

— BUKU AJAR —

Pengantar Kecerdasan Buatan

Dan Jaringan Syaraf Tiruan

Ibnu Rasyid Munthe, S.T., M.Kom, Junadhi,
Richi Andrianto, S.Kom, M.Kom,
Yuda Irawan, S.Kom, M.TI,
Shabrina Rasyid Munthe, S.Pd.I., M.S,
Eka Irawan, M.Kom, Surya Darma, M.Kom,
Fahmi Firzada, M.Kom,
Nurul Khairina, S.Kom, M.Kom,
Chaeroen Niesa, S.Kom., M.Kom

BUKU AJAR
PENGANTAR KECERDASAN BUATAN
DAN JARINGAN SYARAF TIRUAN

**Ibnu Rasyid Munthe, S.T., M.Kom, Junadhi,
Richi Andrianto. S.Kom, M.Kom, Yuda Irawan, S.Kom, M.TI,
Shabrina Rasyid Munthe, S.Pd.I., M.S, Eka Irawan, M.Kom,
Surya Darma, M.Kom, Fahmi Firzada, M.Kom,
Nurul Khairina, S.Kom, M.Kom,
Chaeroen Niesa, S.Kom., M.Kom**



BUKU AJAR
PENGANTAR KECERDASAN BUATAN DAN JARINGAN SYARAF TIRUAN

**Ibnu Rasyid Munthe, S.T., M.Kom, Junadhi, Richi Andrianto. S.Kom, M.Kom,
Yuda Irawan, S.Kom, M.TI, Shabrina Rasyid Munthe, S.Pd.I., M.S, Eka Irawan, M.Kom,
Surya Darma, M.Kom, Fahmi Firzada, M.Kom, Nurul Khairina, S.Kom, M.Kom,
Chaeroen Niesa, S.Kom., M.Kom**

Editor :
Sahat Parulian Sitorus

Desain Cover :
Nurhasanah

Sumber :
<https://isbn.jndi.my.id/products/buku-ajar-pengantar-kecerdasan-buatan-dan-jaringan-syaraf-tiruan>

Tata Letak :
M. Zein Siregar

Ukuran :
Jml hal judul 289, Jml hal isi naskah 284, Uk: 17x25 cm

ISBN :
978-634-04-7698-9

Cetakan Pertama :
Januari 2026

Hak Cipta 2026, Pada Penulis
Isi diluar tanggung jawab percetakan

Copyright © 2026 by PT.JNDI
All Right Reserved

Hak cipta dilindungi undang-undang
Dilarang keras menerjemahkan, memfotokopi, atau
memperbanyak sebagian atau seluruh isi buku ini
tanpa izin tertulis dari Penerbit.

PENERBIT PT. JASA NIAGA DIGITAL INDONESIA
Jl. H. Maulana Link. VI. Labuhan Ruku, Talawi, Batu Bara, Sumatera Utara 21254
Telp/Wa : 082361274081
<https://isbn.jndi.my.id>
E-mail:office@jndi.my.id

Kata Pengantar

Buku ajar Pengantar Kecerdasan Buatan Dan Jaringan Syaraf Tiruan hadir sebagai materi pembelajaran untuk memahami dunia Kecerdasan Buatan (AI) dan Jaringan Syaraf Tiruan (JST), dua bidang yang telah mengubah cara kita berinteraksi dengan teknologi. Dari sistem rekomendasi yang memengaruhi keputusan belanja hingga mobil otonom yang mengemudi sendiri, AI kini menjadi bagian tak terpisahkan dari kehidupan sehari-hari. Di dalam buku ini, pembaca akan diajak untuk mengeksplorasi konsep dasar AI dan JST dengan cara yang mudah dipahami, serta mempelajari penerapannya dalam berbagai industri. Dengan pendekatan teori dan penerapan langsung menggunakan studi kasus, buku ini memberikan dasar yang kuat bagi pembaca untuk memulai perjalanan di dunia AI. Setiap materi dilengkapi dengan contoh praktis dan studi kasus yang relevan, memudahkan pembaca untuk menguasai teknik yang dapat langsung diterapkan.

Rantau Prapat, Januari 2026

Tim Penulis

Kata Prakata

Buku ajar Pengantar Kecerdasan Buatan Dan Jaringan Syaraf Tiruan ini hadir sebagai panduan untuk memberikan pemahaman dasar tentang kecerdasan buatan (AI) dan jaringan syaraf tiruan (neural network), dua topik yang semakin menjadi landasan penting dalam perkembangan teknologi saat ini. Kecerdasan buatan telah merambah berbagai sektor kehidupan, mulai dari industri, pendidikan, hingga kesehatan. Sementara itu, jaringan syaraf tiruan, dengan kemampuannya untuk mempelajari pola dan membuat keputusan, semakin banyak diterapkan dalam berbagai aplikasi, termasuk pengenalan wajah, prediksi pasar, dan deteksi penyakit.

Melalui buku ini, pembaca diharapkan dapat memperoleh wawasan yang jelas tentang konsep dasar, struktur, serta penerapan praktis dari teknologi ini. Dengan penjelasan yang komprehensif dan mudah dipahami, buku ini ditujukan untuk mahasiswa, praktisi, dan siapa saja yang tertarik untuk memahami lebih dalam tentang AI dan jaringan syaraf tiruan.

Selamat membaca, semoga buku ini dapat memberikan manfaat dan mendorong pembaca untuk terus mengeksplorasi dunia kecerdasan buatan yang semakin menarik.

Daftar Isi

Kata Pengantar.....	2
Kata Prakata.....	3
Daftar Isi.....	4
Pengantar Data Science dan Python.....	9
A. Peran Data Science	9
B. Python Sebagai Bahasa Ilmu Data	10
C. Pustaka Python untuk Analisis Data	11
D. Praktik Menggunakan UCI	12
E. Arah dan Dampak Data Science.....	13
Kode Program : Pemrosesan dan Pemisahan Latih Uji.....	14
Studi Kasus: Proyek Klasifikasi Iris dan Scatterplot	15
Soal	18
Refleksi Pembelajaran.....	20
Pengantar Proses End-to-End Data Science	21
A. Pemahaman Bisnis sebagai Fondasi Keputusan.....	21
B. Pengumpulan Data	22
C. Pembersihan dan Eksplorasi Data	22
D. Pemodelan dan Evaluasi.....	23
E. Deployment dan Perbaikan Berkelanjutan.....	23
Kode Program: Praktikum Data Science	24
Studi Kasus: Pemrosesan Data, EDA, dan Deployment.....	31
Soal	39
Tugas Kelompok.....	40
Refleksi Pembelajaran.....	41
Supervised Learning	43
Regresi Linier dan Evaluasi Model	43
A. <i>Supervised Learning</i> Berbasis Regresi	43
B. Jenis Algoritma pada <i>Supervised Learning</i>	44
C. Konsep Prediksi dan <i>Regresi Linier</i>	44
D. Proses Train Test Split dan Masalah Overfitting	45
E. Evaluasi Model dengan <i>MAE</i> dan <i>RMSE</i>	46
Kode Program : Regresi Linear pada Kekuatan Tekan Beton.....	47
Studi Kasus: Regresi Berganda pada Prediksi Skor Akhir Siswa	52
Studi Kasus 2: Decision Tree Regressor pada Dataset Air Quality ..	55
Soal	57

Tugas Kelompok.....	59
Refleksi Pembelajaran.....	60
Klasifikasi dan Logistic Regression	61
A. Pengantar Klasifikasi dan Tujuan Analisis.....	61
B. Konsep Dasar Logistic Regression dan Fungsi Sigmoid	62
C. Metrik Evaluasi dan Analisis Confusion Matrix	63
D. Logistic Regression pada Dataset Bank Marketing.....	64
E. Klasifikasi dalam Pengambilan Keputusan	65
Kode Program: Logistic Regression Deteksi Uang Asli vs Palsu	66
Studi Kasus: Analisis Keputusan Investasi Nasabah Bank	68
Soal	73
Tugas Kelompok.....	74
Refleksi Pembelajaran.....	76
Decision Tree dan Interpretabilitas Model	77
A. Dasar Pemikiran Model Pohon Keputusan	77
B. Entropy dan Information Gain dalam Pembentukan Pohon	78
C. Struktur Pohon dan Visualisasi Keputusan.....	78
D. Penerapan Decision Tree Classifier pada Dataset Iris	79
E. Model Pohon dalam Pembelajaran dan Keputusan	80
Kode Program : Klasifikasi Jenis Kaca	81
Studi Kasus: Hyperparameter Tuning Decision Tree Dataset Iris.....	86
Soal	91
Tugas Kelompok.....	92
Refleksi Pembelajaran.....	93
Ensemble Learning dengan Random Forest.....	95
A. <i>Ensemble Learning</i> dan Random Forest.....	95
B. Prinsip <i>Bagging</i> dan Pembentukan Model Forest.....	96
C. Mekanisme <i>Voting Ensemble</i> dan Keunggulan Prediktif.....	97
D. <i>Feature Importance</i> dan Interpretasi Model.....	98
Kode Program :Prediksi Status Kelulusan Siswa Random Forest.	99
Studi Kasus: Prediksi Nilai Akhir Siswa Gradient Boosting.	104
Soal	110
Tugas Kelompok.....	111
Refleksi Pembelajaran.....	112
Support Vector Machine (SVM)	114
A. Hakikat Support Vector Machine	114
B. Konsep Hyperplane dan Margin	115
C. Kernel Trick sebagai Strategi Transformasi	115

D. Optimasi dan Regularisasi	116
E. SVM pada Data Kesehatan	116
Kode Program : SVM untuk klasifikasi risiko penyakit jantung.....	117
Studi Kasus: SVM untuk Diagnostik Kanker Payudara Wisconsin..	121
Soal	126
Tugas Kelompok.....	127
Refleksi Pembelajaran.....	129
Unsupervised Learning	130
K-Means Clustering.....	130
A. Pemilihan <i>Unsupervised Learning</i> untuk Klasterisasi Data	130
B. Mekanisme K Means, Centroid, dan Klaster	131
C. Evaluasi Klaster melalui <i>Inertia</i> dan <i>Elbow Method</i>	132
D. Pemilihan Algoritma <i>Unsupervised Learning</i>	132
E. Validasi, dan Interpretasi Klaster	133
Kode Program: KMeans Klaster Pengunjung Web	134
Studi Kasus: DBSCAN untuk Data Cardiotocography	139
Soal	143
Tugas Kelompok.....	145
Refleksi Pembelajaran.....	146
Principal Component Analysis (PCA).....	147
A. Mengapa Data Perlu Disederhanakan.....	147
B. Cara PCA Menangkap Informasi Paling Penting	147
C. Peran Nilai Eigen dalam Menentukan Komponen Utama	148
D. Memetakan Data ke Bentuk yang Lebih Sederhana	148
E. Manfaat PCA untuk Pemahaman dan Visualisasi Data	150
Kode Program: Penggunaan PCA dan Model Linier	150
Studi Kasus: Klasifikasi Kualitas Sperma Dataset Fertility	154
Soal	158
Tugas Kelompok.....	160
Refleksi Pembelajaran.....	162
Neural Network	163
A. Pengantar Neural Network.....	163
B. Arsitektur Berbasis <i>Sequential</i> dan Lapisan <i>Dense</i>	164
C. Konsep <i>Activation Function</i> sebagai Sumber Nonlinearitas.....	165
D. Peran <i>Optimizer</i> dan Proses <i>Training</i>	165
E. Algoritma, dan Evaluasi	166
Kode Program: Feedforward Neural Network Klasifikasi Kedelai .	168
Studi Kasus: CNN untuk Klasifikasi QSAR Androgen Receptor.	172

Studi Kasus 2: Recurrent Neural Network pada Stock Saham	177
Studi Kasus 4: GAN Deteksi Anomali Kartu Kredit	188
Soal	195
Tugas Kelompok.....	196
Refleksi Pembelajaran.....	198
Deep Learning dan Overfitting	200
A. Deep Learning dan Overfitting	200
B. Peran Hidden Layer dalam Alur Pembelajaran.....	201
C. Backpropagation dan Loss Function.....	201
D. Epoch Tuning dan Proses Pelatihan	202
E. Regularisasi, Dropout, dan Optimasi	202
Kode Program: DL Bank Marketing Dropout dan Adam	203
Studi Kasus: Implementasi MLP untuk Prediksi Sale Prices.	208
Soal	213
Tugas Kelompok.....	214
Refleksi Pembelajaran.....	216
Natural Language Processing (NLP) Text	217
Preprocessing dan Tokenization	217
A. Dari Teks ke Fitur	217
B. Normalisasi dan Pembersihan	217
C. Segmentasi melalui <i>tokenization</i>	218
D. stopword removal dan <i>n-grams</i>	219
E. Representasi Numerik	220
Kode Program: Implementasi <i>NLP</i> untuk Klasifikasi Sentimen.	220
Studi Kasus: Klasifikasi Spam-Ham pada SMS Spam.	229
Soal	238
Tugas Kelompok.....	240
Refleksi Pembelajaran.....	242
Reinforcement Learning	243
Dasar Reinforcement Learning.....	243
A. Pengertian Reinforcement Learning	243
B. Agent dan Environment.....	244
C. Reward, State, dan Action	245
D. Jenis Jenis Algoritma <i>Reinforcement Learning</i>	246
E. Prinsip Eksplorasi dan Eksploitasi	247
Kode Program: RL dengan Q-learning Rekomendasi Barang.	249
Studi Kasus: RL Prediksi Permintaan Bike Sharing.	256
Soal	264

Tugas Kelompok.....	265
Refleksi Pembelajaran.....	267
Contoh Soal UTS	268
Contoh Soal UAS.....	272
Contoh Desain Penilaian Terhubung CPL dan CPMK	275
1. Kuis Teori (40%)	275
2. Praktikum (Tugas Kelompok) (40%)	277
3. Studi Kasus (20%)	278
4. Umpan Balik dan Refleksi Pembelajaran	279
Daftar Pustaka	280

Pengantar Data Science dan Python

CPL:

Mahasiswa memahami konsep dasar data science dan peran Python

CPMK:

Menjelaskan hubungan antara data, algoritma, dan model pembelajaran.

SubCPMK:

Mengoperasikan Google Colab dan memuat dataset UCI.

A. Peran Data Science

Perkembangan teknologi informasi telah menjadikan data sebagai sumber daya penting yang menentukan arah perubahan dunia modern. Setiap kegiatan manusia menghasilkan data yang merekam perilaku, keputusan, dan interaksi sosial. *Data science* menjadi disiplin ilmu yang berfungsi mengubah data mentah menjadi informasi yang dapat diinterpretasikan dan dimanfaatkan untuk mendukung pengambilan keputusan. Ilmu ini menggabungkan tiga pilar utama yaitu **statistika**, **ilmu komputer**, dan **pengetahuan domain** untuk mengekstraksi wawasan dari kumpulan data. Penggunaan *data science* kini meluas ke berbagai bidang seperti ekonomi, kesehatan, pendidikan, dan transportasi, di mana pola tersembunyi dari data menjadi dasar inovasi dan efisiensi.

Analisis data berbasis *data science* membantu memahami fenomena yang kompleks dengan cara kuantitatif dan terukur.

Melalui pendekatan ini, informasi tidak lagi bersifat subjektif tetapi berdasarkan bukti dan pola matematis. Data yang awalnya tidak terstruktur dapat diubah menjadi model prediktif yang memberikan arah dalam pengambilan keputusan.

Insight *Data science berfungsi sebagai kunci untuk memahami dunia modern dengan cara rasional dan berbasis fakta, bukan sekadar asumsi atau intuisi.*

B. Python Sebagai Bahasa Ilmu Data

Python menjadi pilihan utama dalam dunia analisis data karena kesederhanaannya, kemampuan lintas platform, dan ekosistem pustaka yang luas. Bahasa ini dirancang agar mudah dibaca dan ditulis, sehingga mempercepat proses pembelajaran dan penerapan. Python memiliki keunggulan karena bersifat **open source**, artinya dapat digunakan dan dikembangkan oleh siapa pun tanpa biaya lisensi. Komunitas global yang besar juga mendukung ketersediaan dokumentasi, modul tambahan, serta forum diskusi yang aktif.

Keunggulan lain Python adalah kemampuannya untuk digunakan di *Google Colab*, sebuah platform berbasis awan yang memungkinkan pengguna menulis, menjalankan, dan membagikan kode secara daring tanpa perlu instalasi perangkat lunak tambahan. Pengguna dapat mengakses seluruh pustaka analisis data langsung dari peramban web, membuat eksperimen, dan menyimpan hasilnya di penyimpanan awan. Lingkungan ini mendukung kerja kolaboratif dan transparansi hasil analisis.

Insight *Python tidak hanya menjadi alat komputasi, tetapi juga sarana pembelajaran terbuka yang menghubungkan manusia dengan teknologi melalui kemudahan akses dan kolaborasi.*

C. Pustaka Python untuk Analisis Data

Kekuatan Python terletak pada ekosistem pustaka yang mampu mencakup seluruh tahapan analisis data. *NumPy* menyediakan struktur data berbasis array multidimensi dan fungsi matematika tingkat tinggi untuk komputasi numerik. *Pandas* menambahkan kemampuan manajemen data melalui struktur *DataFrame* yang efisien untuk membaca, memodifikasi, dan menganalisis data tabular. Dengan dua pustaka ini, pengguna dapat menangani dataset besar dengan operasi matematis yang cepat dan stabil.

Untuk menggambarkan hasil analisis, *Matplotlib* dan *Seaborn* digunakan sebagai alat visualisasi. Grafik yang dihasilkan membantu menampilkan hubungan antarvariabel, tren waktu, serta pola distribusi data. Visualisasi menjadi alat penting karena membantu manusia memahami data secara intuitif dan menarik. Sedangkan *Scikit-learn* menyediakan berbagai algoritma *machine learning* yang siap digunakan seperti *Linear Regression*, *Decision Tree*, dan *K-Nearest Neighbors*. Pustaka ini juga dilengkapi dengan fungsi *train_test_split*, *cross-validation*, serta metrik evaluasi yang digunakan untuk mengukur performa model.

Semua pustaka tersebut dapat diintegrasikan dengan mudah dalam satu alur kerja di Python, menjadikannya platform analisis yang efisien dan terstruktur. Seseorang dapat mulai dari membaca data, membersihkan nilai yang hilang, menampilkan pola melalui

visualisasi, hingga membangun model prediktif tanpa berpindah perangkat.

Insight *Ekosistem pustaka Python memungkinkan pengguna fokus pada analisis dan interpretasi data tanpa harus memikirkan kompleksitas teknis.*

D. Praktik Menggunakan UCI

Untuk mempelajari *data science* secara terapan, dibutuhkan sumber data yang terbuka dan terpercaya. *UCI Machine Learning Repository* menyediakan kumpulan dataset publik yang digunakan oleh para peneliti dan pengembang di seluruh dunia. Salah satu dataset paling terkenal dari repositori ini adalah *Iris Dataset*, yang berisi pengukuran panjang dan lebar kelopak serta sepal dari tiga jenis bunga *Iris*: *setosa*, *versicolor*, dan *virginica*. Dataset ini menjadi contoh klasik untuk memahami dasar klasifikasi dan eksplorasi data karena strukturnya sederhana namun informatif.

Langkah awal yang dilakukan adalah memuat dataset dari *UCI Repository* ke lingkungan *Google Colab*. Proses ini menggunakan pustaka *Pandas* dengan perintah `read_csv()` untuk membaca data langsung dari tautan sumber. Setelah data dimuat, tahap berikutnya adalah menampilkan lima baris pertama untuk mengenali struktur kolom, lalu melakukan analisis deskriptif untuk melihat nilai rata-rata, minimum, maksimum, dan standar deviasi. Visualisasi menggunakan *Matplotlib* dapat membantu melihat persebaran data dan hubungan antarvariabel seperti panjang kelopak terhadap jenis bunga.

Praktik ini memperkenalkan hubungan antara data, algoritma, dan model pembelajaran. Data berfungsi sebagai masukan,

algoritma bertugas mengolahnya, dan model menghasilkan pola yang dapat digunakan untuk prediksi di masa depan. Analisis ini memperlihatkan bagaimana *machine learning* bekerja secara sederhana dan memberikan pemahaman tentang proses berpikir komputasional.

Insight *Dataset dari UCI Repository memberikan pengalaman nyata dalam menghubungkan teori dan praktik secara terbuka dan dapat direplikasi oleh siapa pun di dunia.*

E. Arah dan Dampak Data Science

Peran *data science* semakin penting karena hampir seluruh keputusan modern didukung oleh data. Dunia bisnis menggunakan analisis data untuk memahami perilaku konsumen, lembaga kesehatan menggunakannya untuk mendiagnosis penyakit lebih cepat, dan bidang pendidikan memanfaatkannya untuk merancang sistem pembelajaran yang adaptif. Data telah menjadi sumber daya baru yang menggantikan minyak sebagai bahan bakar ekonomi digital. Kemampuan membaca, memahami, dan mengelola data menjadi keterampilan dasar yang dibutuhkan di masa depan.

Python berperan sebagai jembatan untuk menjembatani teori dengan penerapan nyata karena kemudahannya dalam memproses data besar dan membangun model analisis. *Google Colab* memperkuat peran tersebut dengan memberikan akses tanpa batas terhadap teknologi komputasi awan yang efisien dan terjangkau. Seseorang dapat menjalankan eksperimen, berbagi hasil, dan bekerja sama secara global hanya dengan koneksi internet.

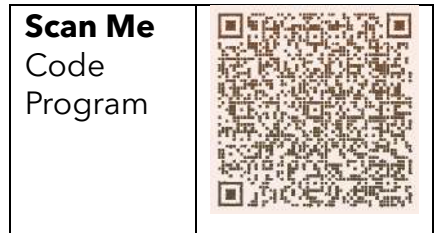
Dunia yang berorientasi data menuntut kemampuan berpikir kritis, pemecahan masalah, serta etika dalam penggunaan informasi. Data tidak hanya harus akurat, tetapi juga harus digunakan dengan tanggung jawab. Pemahaman terhadap prinsip-prinsip *data science* mendorong pembangunan yang lebih cerdas dan berkelanjutan karena setiap keputusan berbasis bukti nyata.

Insight *Masa depan peradaban modern ditentukan oleh kemampuan manusia memadukan analisis data dengan kebijaksanaan moral untuk menciptakan solusi yang bermanfaat bagi kehidupan.*

Kode Program : Pemrosesan dan Pemisahan Latih Uji

Baris kode `import pandas as pd` dan `from sklearn.model_selection import train_test_split` merupakan titik awal yang sangat penting untuk praktikum *data science* berbasis Python. Perintah pertama memanggil *library* **pandas** yang digunakan untuk mengelola data tabel, sedangkan perintah kedua memanggil *function* **train_test_split** dari *library* *scikit learn* yang berfungsi membagi data menjadi bagian *training set* dan *test set*.

Dua baris ini menjadi pintu masuk menuju alur kerja analitis yang sistematis karena mempersiapkan lingkungan pemrograman untuk membaca, memanipulasi, dan menguji data dari *UCI Machine Learning Repository* secara terstruktur di *Google Colab*. Agar lebih konkret, kode program dapat dikembangkan untuk memuat dataset *Iris* dari *UCI Repository* dan membaginya menjadi data latih dan data uji sebagai berikut:



Kode program pada barcode memperlihatkan langkah terstruktur mulai dari memuat data, memisahkan fitur dan label, hingga membagi data menjadi dua bagian. Penggunaan argumen `test_size=0.2` berarti dua puluh persen data digunakan sebagai data uji, sedangkan sisanya untuk pelatihan *model*. Argumen `stratify=y` menjaga proporsi kelas tetap seimbang pada kedua bagian data sehingga evaluasi *model* menjadi lebih adil. **Pembagian data yang tepat** membantu menghindari *overfitting* karena *model* tidak hanya menghafal pola pada data latih, tetapi juga diuji pada data baru yang belum pernah dilihat sebelumnya.

Baris `X_train.shape` dan `X_test.shape` dimanfaatkan untuk memeriksa ukuran data setelah proses pemisahan. Pemeriksaan sederhana ini sering diabaikan, padahal sangat penting untuk memastikan bahwa semua langkah telah berjalan sesuai rencana. **Kebiasaan melakukan pemeriksaan kecil** seperti ini melatih cara berpikir sistematis dan mengurangi kesalahan analisis di tahap lanjut ketika *model* mulai dibangun dan dievaluasi menggunakan metrik yang lebih kompleks.

Studi Kasus: Proyek Klasifikasi Iris dan Scatterplot



Kode program pada barcode Program ini disusun sebagai bagian dari buku praktikum **Praktikum Data Science dan Machine Learning Berbasis Python dan UCI Repository**, khususnya untuk Pertemuan 1 yang membahas **pengantar klasifikasi dasar**. Kode dijalankan di lingkungan *Google Colab* dan terhubung dengan *repository GitHub*, sehingga mahasiswa dapat mengulang eksperimen dengan mudah dan menjaga **reproducibility**.

Pada bagian awal, program melakukan *import* beberapa *library* penting. Pustaka **pandas** dipakai untuk mengelola **tabel data**, sedangkan modul dari *scikit-learn* digunakan untuk pemisahan *data* dengan fungsi *train_test_split*, pembentukan *model* klasifikasi menggunakan *Logistic Regression*, serta pengukuran kinerja melalui *accuracy_score*. Tambahan *library matplotlib* dan *seaborn* memungkinkan dibuatnya **visualisasi** berupa *scatterplot*, sehingga mahasiswa tidak hanya melihat angka **akurasi**, tetapi juga memahami pola sebaran **data** secara visual. Di sini, mahasiswa dapat mulai melakukan **brainstorming** tentang hubungan antara **fitur** numerik dan kelas spesies.

Program kemudian memuat *dataset* Iris langsung dari alamat *UCI* dan memberi nama kolom yang jelas: *sepal_length*, *sepal_width*, *petal_length*, *petal_width*, dan *species*. Perintah untuk menampilkan lima baris pertama membantu pembaca melihat bentuk **data mentah** sebelum diolah. Latihan ini melatih kebiasaan penting dalam **data science**, yaitu selalu memeriksa **struktur data** sebelum melakukan pemodelan. Dari tampilan awal, mahasiswa dapat mengamati bahwa spesies bunga berbeda memiliki kombinasi

panjang dan lebar organ yang khas, suatu **insight** awal tentang kemungkinan garis pemisah di ruang *fitur*.

Selanjutnya, program memisahkan **fitur** dan **target**. Variabel X berisi empat kolom numerik sebagai deskripsi karakteristik bunga, sementara y berisi label species yang akan diprediksi. Langkah ini menegaskan konsep dasar *supervised learning*: ada pasangan **input** dan **output** yang digunakan untuk melatih *model* sehingga mampu membuat **prediksi** pada data baru. Proses pembagian *data* menjadi *training set* dan *test set* dengan proporsi 80:20, serta penggunaan `stratify=y`, menunjukkan pentingnya menjaga keseimbangan kelas agar evaluasi menjadi **adil** dan tidak bias.

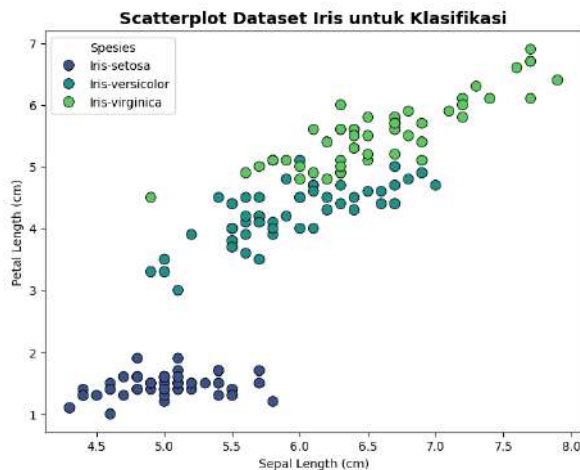
Bagian inti pemodelan dilakukan dengan membangun *model Logistic Regression* dan melatihnya pada *training set*. Secara konseptual, *Logistic Regression* berusaha mencari batas pemisah linier yang paling baik memisahkan kelas. Meskipun sederhana, pendekatan ini sangat berguna untuk memperkenalkan mahasiswa pada ide bahwa **model yang baik** tidak harus rumit, tetapi harus **jelas** dan **mudah dijelaskan**. Nilai **akurasi** yang dihasilkan pada *test set* memberikan ukuran seberapa baik *model* belajar dari data. Contoh keluaran program dapat berupa:

```
Lima baris pertama data:
  sepal_length  sepal_width  petal_length  petal_width  species
0           5.1           3.5           1.4           0.2  Iris-setosa
1           4.9           3.0           1.4           0.2  Iris-setosa
2           4.7           3.2           1.3           0.2  Iris-setosa
3           4.6           3.1           1.5           0.2  Iris-setosa
4           5.0           3.6           1.4           0.2  Iris-setosa

Akurasi model pada data uji: 0.97
```

Gambar 1. Visualisasi Awal Data Iris dan Akurasi Model

Bagian akhir program menampilkan *scatterplot* menggunakan *seaborn* untuk memvisualisasikan hubungan antara `sepal_length` dan `petal_length`, dengan warna berbeda untuk setiap species. Visualisasi ini memperkuat pemahaman bahwa **klasifikasi** tidak hanya berbasis rumus, tetapi juga dapat diamati secara visual melalui pola sebaran **titik data**.



Gambar 2. Scatterplot Dataset Iris untuk Klasifikasi

Soal

1. Jelaskan bagaimana peran data science dalam pengambilan keputusan di berbagai bidang. Berikan contoh penerapan data science di bidang kesehatan dan pendidikan.
2. Bagaimana Python mendukung proses analisis data secara efisien? Jelaskan keunggulan Python dibandingkan bahasa pemrograman lain dalam konteks data science.
3. Anda diminta untuk melakukan analisis pada dataset yang besar. Sebutkan pustaka Python yang dapat Anda gunakan untuk memanipulasi data dan menjelaskan bagaimana pustaka tersebut dapat membantu dalam proses analisis data.

4. Apa yang dimaksud dengan 'train-test split' dalam konteks pembelajaran mesin? Jelaskan mengapa teknik ini penting dan bagaimana penerapannya dalam membangun model prediktif.
5. Uji pemahaman Anda dengan menjelaskan bagaimana hubungan antara data, algoritma, dan model pembelajaran dalam data science. Berikan contoh yang relevan dengan proses klasifikasi menggunakan dataset Iris.
6. Dalam penggunaan Google Colab, jelaskan tahapan-tahapan yang Anda lakukan untuk memuat dan memanipulasi dataset UCI, kemudian terapkan analisis deskriptif dan visualisasi menggunakan Matplotlib dan Seaborn. Apa hasil yang Anda peroleh dan apa insight yang bisa diambil dari data tersebut?
7. Data science digunakan untuk mengeksplorasi pola tersembunyi dalam data. Jika diberikan dataset yang tidak terstruktur, seperti teks atau gambar, jelaskan langkah-langkah yang Anda lakukan untuk mengubah data tersebut menjadi informasi yang berguna menggunakan Python.
8. Berdasarkan pemahaman Anda tentang ekosistem pustaka Python, jelaskan bagaimana Anda akan menggunakan kombinasi pustaka seperti Pandas, Matplotlib, dan Scikit-learn dalam satu alur kerja untuk menyelesaikan sebuah proyek analisis data.
9. Pertimbangkan sebuah perusahaan yang ingin memprediksi churn rate pelanggan menggunakan data historis. Jelaskan pendekatan yang akan Anda lakukan dengan menggunakan algoritma machine learning, serta bagaimana Anda akan

mempersiapkan data, membangun model, dan mengevaluasi hasilnya.

10. Pahami hubungan antara data science, machine learning, dan pengambilan keputusan berbasis bukti. Bagaimana Anda mengintegrasikan ketiga elemen ini dalam merancang sistem yang dapat memberikan rekomendasi kepada manajer tentang strategi bisnis yang optimal berdasarkan analisis data yang ada?

Refleksi Pembelajaran

Refleksi pembelajaran ini menunjukkan pentingnya data science dalam mengubah data mentah menjadi informasi yang berguna untuk pengambilan keputusan yang berbasis bukti. Python, dengan kemudahan penggunaannya dan pustakanya yang banyak, menjadi alat utama dalam analisis data. Melalui platform seperti Google Colab, proses ini semakin efisien dan kolaboratif. Pembelajaran tentang pemisahan data untuk pelatihan dan pengujian model juga sangat krusial untuk menghindari overfitting. Secara keseluruhan, data science dan machine learning memberikan wawasan baru dalam memahami dunia dengan cara yang lebih rasional dan berbasis data, yang semakin penting dalam era digital saat ini.

Pengantar Proses End-to-End Data Science

CPL:

Mahasiswa memahami proses end-to-end data science dari pemahaman masalah hingga implementasi model.

CPMK:

Mahasiswa mampu menjelaskan keterkaitan logis antar tahapan data science.

Sub-CPMK:

Mahasiswa mampu mengidentifikasi dan menerapkan setiap tahapan data science menggunakan dataset nyata pada lingkungan Google Colab.

A. Pemahaman Bisnis sebagai Fondasi Keputusan

Pemahaman bisnis menempati posisi paling awal karena seluruh proses data science bergantung pada kejelasan masalah dan tujuan keputusan. Tahap ini menentukan arah berpikir, batas ruang lingkup, serta makna keberhasilan yang ingin dicapai. Tanpa pemahaman bisnis yang kuat, analisis data berpotensi menghasilkan temuan yang terlihat akurat tetapi tidak memiliki nilai guna. Fokus utama berada pada **tujuan keputusan, pertanyaan kunci**, dan **dampak yang diharapkan**. Pemahaman bisnis juga berperan menetapkan batas risiko dan toleransi kesalahan sehingga proses analisis tidak menyimpang dari kebutuhan nyata. Tahap ini menjelaskan mengapa data science bukan sekadar aktivitas teknis,

melainkan kerangka berpikir sistematis untuk membantu memilih tindakan yang lebih rasional.

B. Pengumpulan Data

Pengumpulan data merupakan tahap penerjemahan tujuan bisnis menjadi bukti empiris. Data yang dikumpulkan harus relevan, representatif, dan sesuai dengan tujuan yang telah ditetapkan. Sumber data dapat berasal dari catatan operasional, basis data terbuka, sensor, atau sistem digital lainnya. Perhatian utama tidak hanya terletak pada jumlah data, tetapi pada **kualitas data, cakupan informasi**, dan **potensi bias**. Data yang tidak mencerminkan kondisi nyata akan membatasi kemampuan analisis, seberapa pun canggih metode yang digunakan. Tahap ini juga mencakup penentuan bentuk dan struktur data, karena representasi data memengaruhi cara data diproses dan dipahami. Pengumpulan data menjadi batas empiris yang menentukan sejauh mana pengetahuan dapat dibangun secara sah.

C. Pembersihan dan Eksplorasi Data

Data mentah hampir selalu mengandung kesalahan, ketidakkonsistenan, dan informasi yang tidak lengkap. Pembersihan data bertujuan memperbaiki kualitas informasi dengan mengurangi gangguan yang dapat menyesatkan analisis. Proses ini mencakup penanganan nilai hilang, duplikasi, dan anomali yang tidak relevan. Bersamaan dengan itu, eksplorasi data atau *exploratory data analysis* dilakukan untuk memahami karakter data secara menyeluruh. Eksplorasi membantu mengenali pola, distribusi, serta hubungan antar variabel melalui pengamatan statistik sederhana dan visualisasi. Penekanan utama pada tahap ini berada pada

pemahaman pola, deteksi anomali, dan hubungan variabel.

Pembersihan dan eksplorasi berjalan secara saling terkait dan sering diulang hingga data benar benar siap digunakan. Tahap ini mencegah analisis bersifat spekulatif dan membangun intuisi yang kuat sebelum pemodelan dilakukan.

D. Pemodelan dan Evaluasi

Pemodelan berfungsi membangun representasi formal dari masalah yang ingin diselesaikan. Model digunakan untuk menangkap pola yang relevan dan menghasilkan prediksi atau klasifikasi yang mendukung tujuan keputusan. Pemilihan model dipengaruhi oleh karakter data, tingkat kompleksitas masalah, serta kebutuhan keterjelasan hasil. Model sederhana sering memberikan kestabilan dan kemudahan interpretasi, sedangkan model kompleks mampu menangkap hubungan yang lebih rumit. Evaluasi dilakukan untuk menilai sejauh mana model bekerja secara konsisten dan dapat diterapkan pada data baru. Ukuran kinerja harus selaras dengan tujuan keputusan yang ditetapkan sejak awal. Fokus utama tahap ini terletak pada **kinerja model, kemampuan generalisasi, dan keandalan hasil**. Evaluasi yang logis menjaga agar hasil analisis tidak hanya terlihat baik secara numerik, tetapi juga bermakna secara praktis.

E. Deployment dan Perbaikan Berkelanjutan

Deployment menandai peralihan dari analisis ke pemanfaatan nyata. Model yang telah dievaluasi diintegrasikan ke dalam sistem pengambilan keputusan agar memberikan dampak langsung. Tahap ini menuntut perhatian terhadap stabilitas sistem, efisiensi, dan kemudahan pemeliharaan. Lingkungan nyata selalu berubah, baik

dari sisi data, perilaku pengguna, maupun kondisi eksternal lainnya. Oleh sebab itu, pemantauan kinerja dan pembaruan model menjadi bagian penting dari proses ini. Fokus utama berada pada **keberlanjutan sistem, adaptasi terhadap perubahan, dan nilai jangka panjang**. Proses end to end data science bersifat iteratif karena hasil implementasi sering memunculkan kebutuhan revisi pada tahap sebelumnya. Siklus ini menegaskan bahwa data science merupakan proses pembelajaran berkelanjutan yang terus berkembang seiring bertambahnya data dan pemahaman masalah.

Kode Program: Praktikum Data Science



Kode program pada notebook ini disusun sebagai bagian dari buku praktikum **Praktikum Data Science dan Machine Learning Berbasis Python dan UCI Repository**, dengan fokus pada pengantar proses *end to end* Data Science berbasis data runtun waktu. Praktikum dijalankan pada lingkungan Google Colab agar seluruh eksperimen dapat direplikasi dengan mudah dan konsisten. Dataset yang digunakan berasal dari UCI Machine Learning Repository, yaitu Air Quality, yang berisi data sensor kualitas udara per jam. Tujuan utama program ini ialah memprediksi konsentrasi gas CO satu jam ke depan sebagai contoh penerapan *regression* pada masalah lingkungan. Sejak awal, pembaca diarahkan untuk memahami bahwa keberhasilan analisis tidak hanya ditentukan oleh

algoritma, tetapi oleh **ketepatan tujuan, kualitas data, dan alur logis antar tahapan.**

Import Library dan Persiapan Lingkungan

Pada bagian awal, program melakukan *import* beberapa pustaka penting. *Pandas* digunakan untuk mengelola data tabular dengan ribuan baris pengamatan, sementara *NumPy* dipakai untuk operasi numerik dasar seperti perhitungan akar kuadrat pada RMSE. Modul dari *scikit-learn* dimanfaatkan untuk membangun model *Random Forest Regressor* serta menghitung metrik evaluasi berupa MAE dan RMSE. Pustaka *matplotlib* dan *seaborn* berperan dalam visualisasi data dan hasil evaluasi. Kehadiran visualisasi ini memungkinkan pembaca melakukan **brainstorming** sejak awal, tidak hanya membaca angka, tetapi juga melihat pola yang muncul secara visual.

Pengumpulan Data

Program memuat dataset Air Quality langsung dari UCI dalam bentuk berkas ZIP. Revisi logis dilakukan dengan mengekstraksi file CSV secara eksplisit agar proses pembacaan data bersifat deterministik. Lima baris pertama data ditampilkan untuk memberi gambaran bentuk data mentah. Terlihat bahwa data dicatat per jam, dimulai dari 10 Maret 2004 pukul 18.00, dengan nilai CO awal berkisar antara 1.6 hingga 2.6 mg/m³. Tahap ini melatih kebiasaan penting dalam Data Science, yaitu selalu memahami struktur dan skala data sebelum melakukan pembersihan atau pemodelan.

	Date	Time	CO(GT)	PT08.S1(CO)	NH4C(GT)	C6H6(GT)	PT08.S2(NH4C)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
0	10/03/2004	18.00.00	2.6	1360.0	150.0	11.9	1046.0	166.0	1066.0	113.0	1692.0	1268.0	13.6	48.9	0.7578
1	10/03/2004	19.00.00	2.0	1292.0	112.0	9.4	955.0	103.0	1174.0	92.0	1559.0	972.0	13.3	47.7	0.7255
2	10/03/2004	20.00.00	2.2	1402.0	86.0	9.0	939.0	131.0	1140.0	114.0	1555.0	1074.0	11.9	54.0	0.7502
3	10/03/2004	21.00.00	2.2	1376.0	80.0	9.2	948.0	172.0	1082.0	122.0	1584.0	1203.0	11.0	60.0	0.7867
4	10/03/2004	22.00.00	1.6	1272.0	51.0	6.5	836.0	131.0	1205.0	116.0	1490.0	1110.0	11.2	59.6	0.7888

Jumlah NaT setelah parsing: 114
Jumlah baris bersih: 9357

Gambar 1. Pratinjau Dataset Air Quality

Gambar ini menampilkan lima baris pertama dataset Air Quality, memperlihatkan kolom tanggal, waktu, konsentrasi CO, sensor gas, serta variabel lingkungan. Tampilan ini membantu pembaca mengenali format data, satuan pengukuran, dan potensi nilai tidak valid sejak awal.

Pembersihan Data dan Pembentukan Waktu

Tahap pembersihan data menjadi fondasi analisis. Nilai sensor bernilai -200 diperlakukan sebagai pembacaan tidak valid dan diganti dengan NaN. Kolom Date dan Time kemudian digabung menjadi satu kolom *datetime* menggunakan format eksplisit. Hasil eksekusi menunjukkan jumlah NaT sebesar 114 baris. Setelah baris dengan waktu tidak valid dihapus, diperoleh 9.357 baris data bersih. Angka kuantitatif ini penting karena menunjukkan bahwa sebagian besar data tetap terjaga. Pengisian nilai hilang dilakukan dengan *forward fill* pada kolom numerik, keputusan ini menjaga **integritas kausal**, karena hanya menggunakan informasi masa lalu.

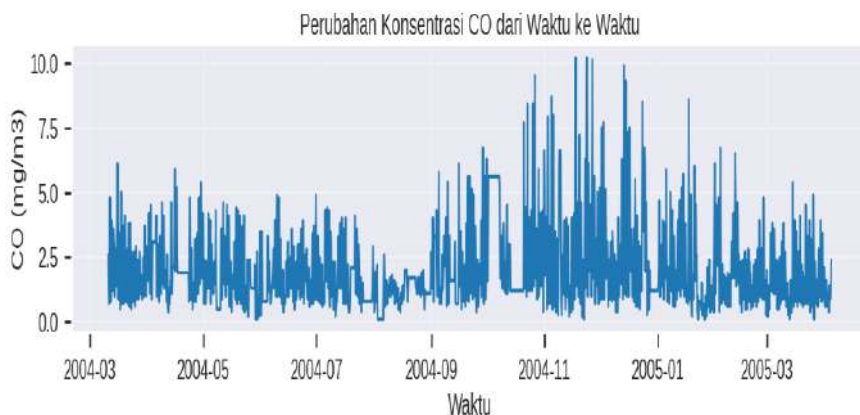
	datetime	CO(GT)	PT08.S1(CO)	C6H6(GT)	NO2(GT)	T	RH	AH
0	2004-03-10 18:00:00	2.6	1360.0	11.9	113.0	13.6	48.9	0.7578
1	2004-03-10 19:00:00	2.0	1292.0	9.4	92.0	13.3	47.7	0.7255
2	2004-03-10 20:00:00	2.2	1402.0	9.0	114.0	11.9	54.0	0.7502
3	2004-03-10 21:00:00	2.2	1376.0	9.2	122.0	11.0	60.0	0.7867
4	2004-03-10 22:00:00	1.6	1272.0	6.5	116.0	11.2	59.6	0.7888

Gambar 2. Data Setelah Parsing Waktu dan Pembersihan

Gambar ini memperlihatkan data setelah kolom waktu dibentuk dan nilai tidak valid dibersihkan. Jumlah baris bersih ditampilkan untuk menunjukkan bahwa proses pembersihan tidak menghilangkan data secara berlebihan.

Eksplorasi Data dan Pola Awal

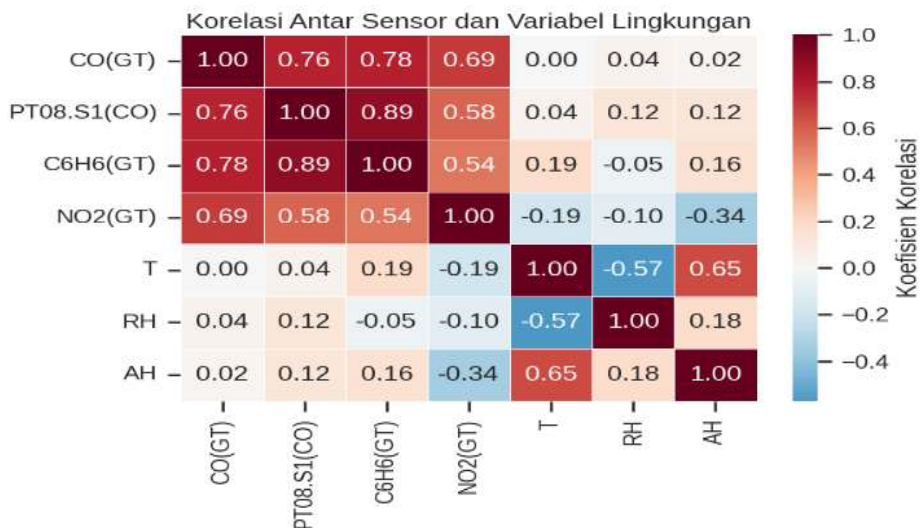
Eksplorasi data dilakukan melalui visualisasi runtun waktu dan korelasi. Grafik runtun waktu CO ditampilkan secara *downsample* setiap enam jam agar tidak terlalu padat. Grafik ini menunjukkan fluktuasi CO yang cukup tajam, dengan puncak mencapai lebih dari 10 mg/m³ pada periode tertentu, serta fase relatif stabil pada kisaran 1 hingga 3 mg/m³. Visual ini memberi **insight** bahwa data memiliki volatilitas tinggi dan kemungkinan pola musiman.



Gambar 3. Perubahan Konsentrasi CO dari Waktu ke Waktu

Grafik runtun waktu ini menampilkan dinamika konsentrasi CO selama periode pengamatan. Fluktuasi yang terlihat menunjukkan adanya variasi temporal yang signifikan dan potensi pola berulang. Eksplorasi dilanjutkan dengan heatmap korelasi antar sensor dan variabel lingkungan. Hasil korelasi menunjukkan bahwa CO berkorelasi kuat dengan PT08.S1(CO) sebesar 0.76 dan dengan C6H6(GT) sebesar 0.78, serta korelasi moderat dengan NO2(GT)

sebesar 0.69. Sebaliknya, korelasi CO dengan suhu dan kelembapan relatif mendekati nol. Informasi kuantitatif ini mendukung pemilihan fitur berbasis sensor.



Gambar 4. Korelasi Antar Sensor dan Variabel Lingkungan

Heatmap ini memperlihatkan kekuatan hubungan antar variabel. Warna merah menunjukkan korelasi positif kuat, sedangkan biru menunjukkan korelasi negatif. Grafik ini membantu menjelaskan alasan pemilihan fitur dalam pemodelan.

Pembentukan Fitur dan Pembagian Data

Tahap pembentukan fitur dilakukan dengan menambahkan tiga fitur *lag*, yaitu *CO_lag_1*, *CO_lag_2*, dan *CO_lag_3*. Target didefinisikan sebagai CO satu jam ke depan. Setelah proses ini, data dibagi menjadi data latih dan data uji dengan proporsi 80:20 berbasis waktu. Hasil pembagian menunjukkan ukuran data latih sebesar 7.482 baris dan data uji sebesar 1.871 baris, masing-masing dengan sembilan fitur. Angka ini menunjukkan skala data yang memadai untuk pemodelan dan evaluasi.

	datetime	CO(GT)	PT08.S1(CO)	C6H6(GT)	NO2(GT)	T	RH	AH	CO_lag_1	CO_lag_2	CO_lag_3	target_CO_next1h
0	2004-03-10 21:00:00	2.2	1376.0	9.2	122.0	11.0	60.0	0.7867	2.2	2.0	2.6	1.6
1	2004-03-10 22:00:00	1.6	1272.0	6.5	116.0	11.2	59.6	0.7888	2.2	2.2	2.0	1.2
2	2004-03-10 23:00:00	1.2	1197.0	4.7	96.0	11.2	59.2	0.7848	1.6	2.2	2.2	1.2
3	2004-03-11 00:00:00	1.2	1185.0	3.6	77.0	11.3	56.8	0.7603	1.2	1.6	2.2	1.0
4	2004-03-11 01:00:00	1.0	1136.0	3.3	76.0	10.7	60.0	0.7702	1.2	1.2	1.6	0.9

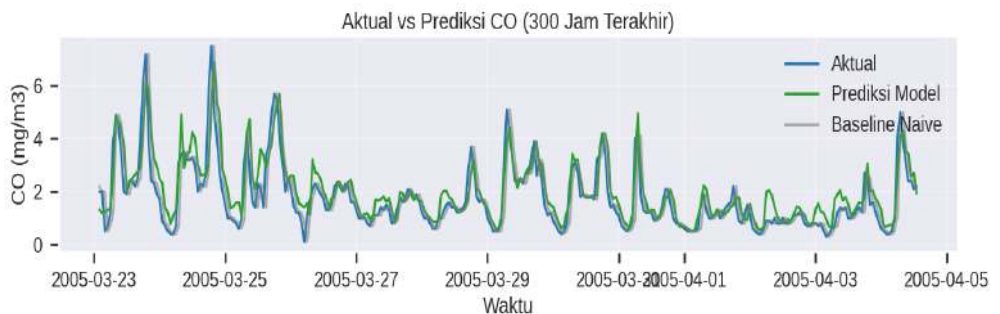
Train shape: (7482, 9)
 Test shape : (1871, 9)
 === HASIL EVALUASI ===
 Baseline Naive -> MAE: 0.491 | RMSE: 0.785
 RandomForest -> MAE: 0.659 | RMSE: 0.929

Gambar 5. Contoh Data Setelah Feature Engineering

Gambar ini menampilkan beberapa baris data setelah fitur *lag* ditambahkan, beserta informasi jumlah data latih dan uji. Tampilan ini membantu pembaca memahami bagaimana data mentah berubah menjadi data siap model.

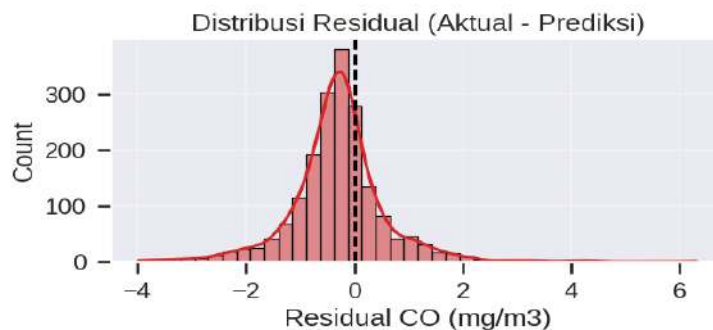
Pemodelan dan Evaluasi Kinerja

Sebagai pembandingan, baseline *naive* digunakan dengan asumsi bahwa nilai CO satu jam ke depan sama dengan nilai saat ini. Hasil evaluasi menunjukkan baseline menghasilkan MAE 0.491 dan RMSE 0.785. Model *Random Forest Regressor* menghasilkan MAE 0.659 dan RMSE 0.929. Secara kuantitatif, baseline masih lebih unggul. Temuan ini merupakan **insight penting**, karena menunjukkan bahwa pada prediksi satu langkah ke depan, autokorelasi data sangat kuat.



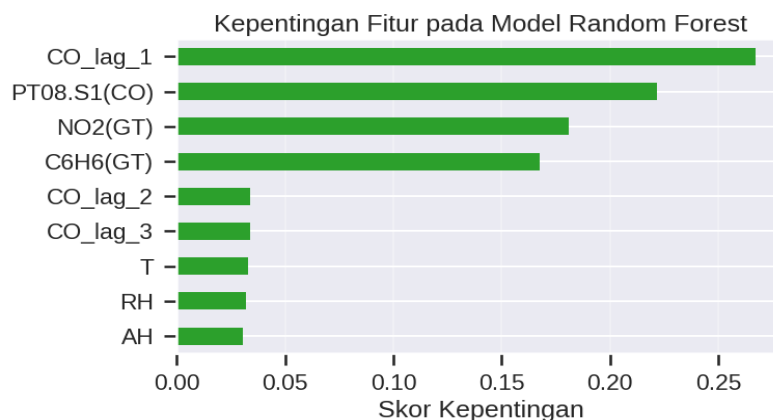
Gambar 6. Perbandingan Aktual dan Prediksi CO

Grafik ini membandingkan nilai aktual CO, prediksi model, dan baseline *naive*. Terlihat bahwa baseline mengikuti pola aktual dengan cukup baik, sementara prediksi model cenderung lebih halus. Distribusi residual memperlihatkan kesalahan prediksi terpusat di sekitar nol, namun dengan ekor yang menunjukkan adanya kesalahan besar pada kondisi ekstrem.



Gambar 7. Distribusi Residual Prediksi CO

Histogram residual ini membantu memahami sebaran kesalahan model. Garis vertikal nol menandai kondisi prediksi sempurna. Plot kepentingan fitur menunjukkan bahwa `CO_lag_1` merupakan fitur paling dominan dengan skor sekitar 0.25, diikuti sensor `PT08.S1(CO)` dan `NO2(GT)`. Fitur lingkungan memiliki kontribusi lebih kecil.



Gambar 8. Kepentingan Fitur pada Model Random Forest

Grafik batang horizontal ini memperlihatkan kontribusi relatif setiap fitur terhadap keputusan model, membantu interpretasi hasil secara kuantitatif.

Deployment

Bagian akhir notebook menyimpan model dan menyediakan fungsi prediksi satu jam ke depan. Contoh eksekusi menghasilkan prediksi sekitar 1.97 mg/m³. Tahap ini menegaskan bahwa program siap digunakan dalam aplikasi nyata. Insight utama dari praktikum ini adalah bahwa **baseline sederhana dapat sangat kuat**, bahwa **penambahan kompleksitas model tidak selalu meningkatkan kinerja**, dan bahwa **pemahaman data lebih penting daripada pemilihan algoritma**. Praktikum ini membuka ruang pengembangan lanjutan, seperti penambahan fitur musiman, *rolling statistics*, atau penerapan model khusus runtun waktu, sekaligus melatih cara berpikir analitis dan kritis dalam Data Science.

Studi Kasus: Pemrosesan Data, EDA, dan Deployment



Kode program ini disusun untuk membangun sebuah pipeline analisis dan simulasi prediksi berbasis **machine learning** pada **Adult / Census Income Dataset** dari **UCI Machine Learning Repository**. Tujuan utama dari program ini adalah memprediksi apakah seseorang memiliki pendapatan lebih dari **50K USD** per tahun berdasarkan data demografis dan pekerjaan. Alur kerja dimulai dari

penyiapan lingkungan komputasi di **Google Colab**, dilanjutkan dengan **akuisisi dan validasi data**, **eksplorasi karakteristik pendapatan**, **pemodelan prediksi berbasis machine learning**, hingga evaluasi dan **deployment** model. **Model yang digunakan** dalam analisis ini adalah **klasifikasi biner**, yang akan mengklasifikasikan individu ke dalam dua kelas: pendapatan **<=50K** atau **>50K**.

Strategi Akuisisi Data

Program ini memuat dataset **Adult / Census Income** langsung dari **UCI Machine Learning Repository** dalam format CSV. Penggunaan **pandas** memungkinkan data diimpor dan dikelola dengan efisien. Data ini mencakup atribut demografis seperti **usia**, **pekerjaan**, **pendidikan**, **status perkawinan**, **jam kerja per minggu**, dan informasi terkait lainnya. Proses **akuisisi data** memeriksa kualitas dan struktur data dengan teliti sebelum melanjutkan ke tahap pembersihan dan pemodelan. Dataset ini penting karena memungkinkan analisis mengenai faktor-faktor yang memengaruhi **pendapatan individu** berdasarkan karakteristik demografis.

Pembersihan Data

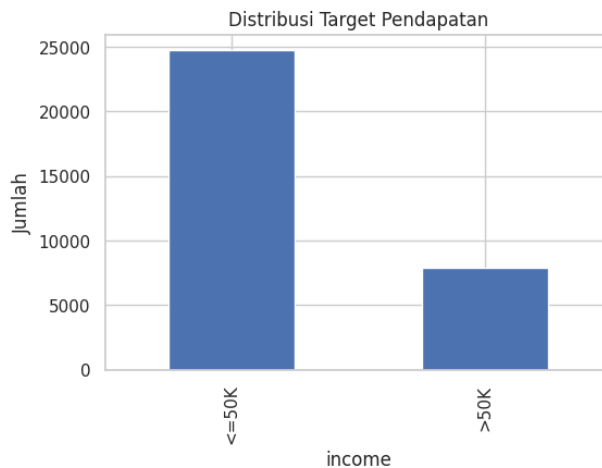
Pada tahap **pembersihan data**, nilai **?** yang menandakan data yang hilang diganti dengan **NaN**. Kolom-kolom yang memiliki **missing values** kemudian ditangani dengan menggunakan teknik **imputasi**. Data numerik yang hilang diimputasi dengan **median**, sementara data kategorikal diimputasi menggunakan nilai **most frequent**. Kolom **Date** dan **Time** digabung menjadi satu kolom **datetime** untuk memastikan data waktu terstruktur dengan baik. Data yang tidak valid, seperti **baris dengan waktu tidak valid**,

kemudian dihapus untuk menjaga kualitas data yang digunakan dalam pemodelan. Langkah-langkah pembersihan ini memastikan bahwa hanya **data yang valid** yang digunakan dalam analisis lebih lanjut, menjaga **integritas data** dan meminimalisir bias atau kesalahan yang dapat memengaruhi hasil model.

Eksplorasi Data

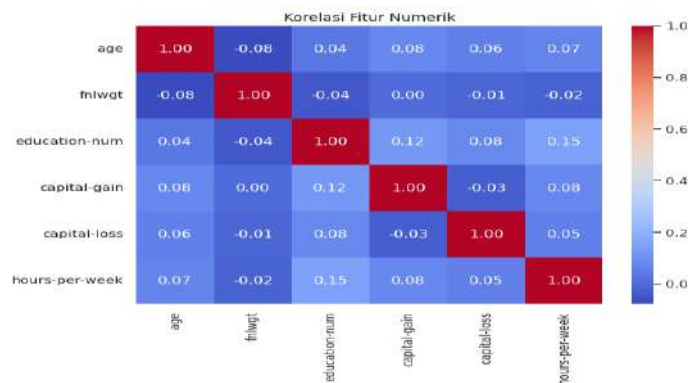
Setelah data dibersihkan, tahap **eksplorasi data** dilakukan dengan tujuan untuk memahami struktur data dan pola-pola yang ada. Salah satu visualisasi pertama yang dilakukan adalah **distribusi target pendapatan**. Grafik ini memperlihatkan bahwa ada **ketidakseimbangan kelas** antara pendapatan $\leq 50K$ dan $> 50K$, dengan mayoritas individu memiliki pendapatan $\leq 50K$. Visualisasi ini memberikan gambaran awal bahwa **imbalanced dataset** dapat memengaruhi kinerja model. Oleh karena itu, langkah-langkah tambahan seperti **penyesuaian bobot kelas** atau **oversampling/undersampling** perlu dipertimbangkan untuk mengatasi masalah ketidakseimbangan kelas.

Selain itu, **heatmap korelasi** digunakan untuk menggambarkan hubungan antar fitur numerik dalam dataset. Hasil dari **heatmap korelasi** menunjukkan bahwa fitur-fitur seperti **education-num** dan **hours-per-week** memiliki korelasi yang lebih kuat dengan target pendapatan, sementara fitur lainnya seperti **fnlwgt** dan **capital-gain** menunjukkan korelasi yang rendah terhadap pendapatan. Ini memberi petunjuk bahwa **pendidikan** dan **jam kerja** lebih berpengaruh terhadap prediksi pendapatan daripada faktor lainnya.



Gambar 1. Menampilkan distribusi target pendapatan

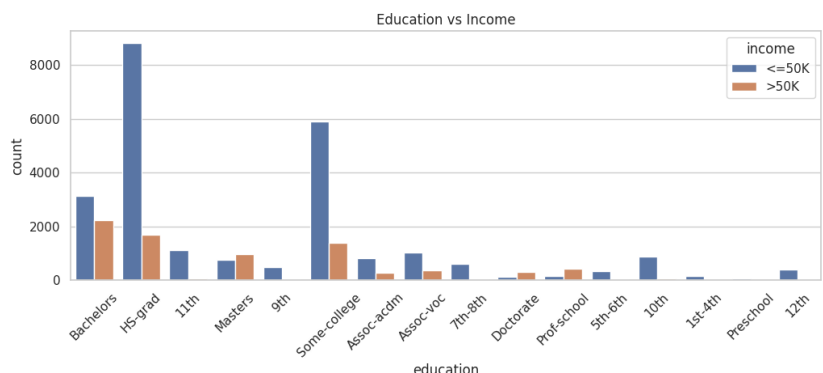
Gambar 1. Menampilkan distribusi target pendapatan dengan kategori **<=50K** dan **>50K**, yang menunjukkan adanya ketidakseimbangan kelas dalam data. Grafik ini menunjukkan bahwa lebih banyak individu dengan pendapatan **<=50K** daripada **>50K**, dengan distribusi data yang terdistribusi miring ke kiri.



Gambar 2. Heatmap korelasi fitur numerik

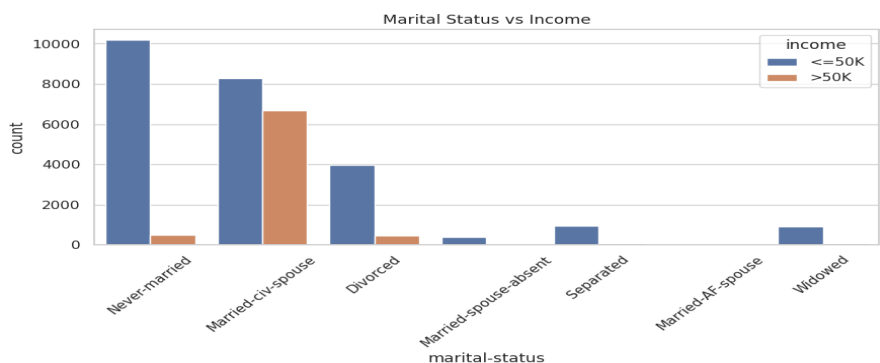
Gambar 2. Heatmap korelasi fitur numerik, yang menggambarkan hubungan antar fitur numerik dalam dataset. Dari **heatmap** ini, kita dapat melihat bahwa fitur **education-num** dan **hours-per-week** memiliki korelasi yang lebih kuat dengan

pendapatan, sedangkan fitur lainnya seperti **capital-gain** memiliki korelasi rendah terhadap pendapatan.



Gambar 3. Grafik perbandingan pendidikan

Gambar 3. Grafik perbandingan pendidikan dengan pendapatan, yang menunjukkan pengaruh tingkat pendidikan terhadap status pendapatan individu. **Pendidikan tinggi**, seperti **Bachelors**, lebih sering ditemukan pada individu dengan pendapatan >50K, sementara individu yang memiliki pendidikan lebih rendah seperti **HS-grad** lebih banyak berada pada kategori pendapatan <=50K.



Gambar 4. Status perkawinan dengan pendapatan

Gambar 4. Visualisasi perbandingan status perkawinan dengan pendapatan, yang menunjukkan bahwa individu yang sudah menikah (**Married-civ-spouse**) lebih banyak berpendapatan >50K,

sementara individu yang belum menikah (**Never-married**) lebih banyak ditemukan dalam kategori pendapatan $\leq 50K$.

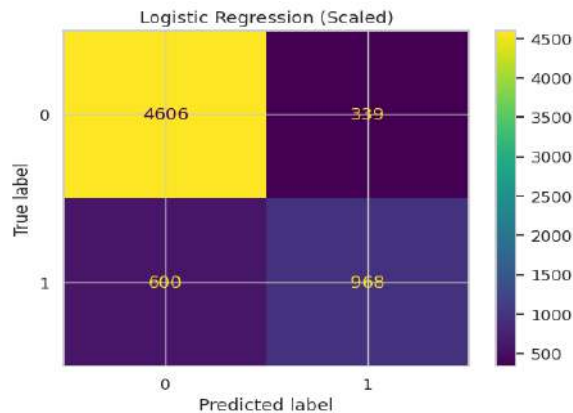
Pemodelan

Setelah tahap eksplorasi, **model prediksi** dibangun menggunakan dua model utama: **Logistic Regression** dan **Random Forest**. **Logistic Regression** digunakan sebagai model **baseline** untuk klasifikasi biner, sementara **Random Forest** digunakan untuk menangani kompleksitas data yang lebih besar dan **imbalanced dataset**. Kedua model ini diintegrasikan dalam **pipeline**, yang memungkinkan kita untuk memastikan bahwa langkah-langkah **preprocessing** (seperti imputasi missing values, encoding, dan scaling) dilakukan secara konsisten di seluruh data. Model **Random Forest** diharapkan dapat memberikan hasil yang lebih baik karena kemampuannya untuk menangani **non-linear relationships** dan menangani masalah **imbalanced dataset**. **Logistic Regression** digunakan untuk membangun **baseline model** yang lebih sederhana, sementara **Random Forest** lebih kompleks namun lebih efektif dalam menangani data yang lebih besar dan variabel yang saling berinteraksi.

Evaluasi Model

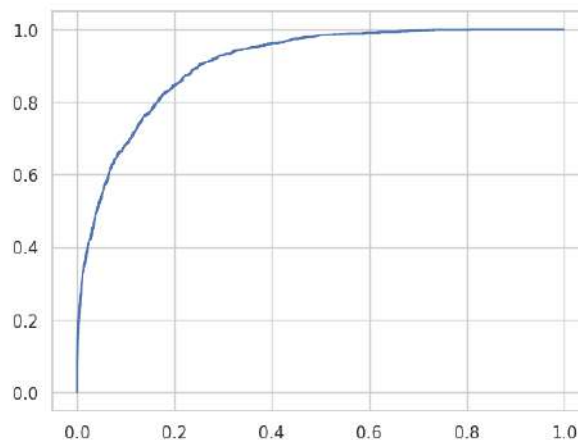
Evaluasi model dilakukan dengan menggunakan **confusion matrix** dan **ROC curve** untuk membandingkan kinerja **Logistic Regression** dan **Random Forest**. Berdasarkan hasil evaluasi, **Random Forest** menunjukkan **ROC-AUC yang lebih tinggi**, mengindikasikan bahwa model ini lebih baik dalam membedakan antara kelas $\leq 50K$ dan $> 50K$. **Confusion matrix** juga menunjukkan bahwa **Random Forest** memiliki **false positives** yang lebih rendah

dan **true positives** yang lebih tinggi, sehingga model ini lebih efektif dalam mendeteksi individu dengan pendapatan >50K.



Gambar 5. Grafik confusion matrix

Gambar 5. Grafik confusion matrix untuk **Logistic Regression** dan **Random Forest**, yang menggambarkan hasil prediksi dibandingkan dengan label aktual untuk kedua model. Confusion matrix ini menggambarkan seberapa baik model dapat mengklasifikasikan individu ke dalam dua kategori pendapatan, dengan menekankan pada penghitungan **true positives**, **false positives**, **true negatives**, dan **false negatives**.



Gambar 6. Perbandingan ROC curve

Gambar 6. Perbandingan ROC curve untuk **Logistic Regression** dan **Random Forest**, yang menunjukkan kinerja kedua model dalam membedakan antara kelas $\leq 50K$ dan $> 50K$. **Random Forest** memiliki ROC-AUC yang lebih besar, menandakan bahwa model ini memiliki kemampuan yang lebih baik dalam membedakan kelas-kelas tersebut.

Deployment

Setelah model dilatih dan dievaluasi, tahap **deployment** dilakukan dengan menyimpan **model Random Forest** menggunakan **joblib**. Fungsi **predict_income** disediakan untuk menerima **input data baru** dan menghasilkan prediksi apakah individu tersebut berpendapatan lebih dari **50K USD** atau tidak. Fungsi ini memungkinkan model digunakan dalam aplikasi dunia nyata, seperti **sistem rekrutmen** atau **pelatihan berbasis data**.

Secara keseluruhan, program ini memberikan pemahaman yang mendalam mengenai **analisis data**, **pemodelan**, dan **evaluasi** dalam memprediksi **pendapatan individu** berdasarkan **karakteristik demografis** dan **pekerjaan** mereka. Hasil evaluasi menunjukkan bahwa **Random Forest** lebih unggul dalam **mengklasifikasikan pendapatan** dibandingkan dengan **Logistic Regression**, meskipun **Logistic Regression** tetap efektif sebagai model **baseline**. Dengan langkah-langkah yang jelas dalam **pembersihan data**, **eksplorasi data**, **pemodelan**, dan **evaluasi**, praktikum ini memberikan **insight yang berharga** dalam dunia **Data Science** dan menunjukkan bahwa **pemahaman data** adalah kunci utama dalam membangun model yang **efektif** dan **dapat diandalkan**.

Soal

1. Jelaskan pentingnya pemahaman bisnis pada tahap awal dalam proses data science, dan bagaimana pemahaman tersebut mempengaruhi keputusan dalam tahapan selanjutnya.
2. Deskripsikan tahapan yang harus dilakukan dalam pengumpulan data yang berkualitas. Apa yang harus diperhatikan agar data yang dikumpulkan dapat mendukung tujuan analisis dengan valid?
3. Pada tahap pembersihan dan eksplorasi data, bagaimana cara mengidentifikasi dan menangani nilai yang hilang serta anomali yang mungkin ada dalam dataset? Berikan contoh penerapannya.
4. Mengapa eksplorasi data (Exploratory Data Analysis) penting sebelum pemodelan dilakukan? Jelaskan bagaimana eksplorasi dapat membantu dalam memilih fitur untuk pemodelan.
5. Sebutkan dan jelaskan berbagai teknik evaluasi yang dapat digunakan untuk mengukur kinerja model dalam data science. Mengapa evaluasi yang tepat sangat penting untuk menentukan keberhasilan model?
6. Berdasarkan kasus pemodelan menggunakan dataset runtun waktu seperti kualitas udara, jelaskan langkah-langkah dalam pembentukan fitur dan bagaimana fitur lag dapat memengaruhi hasil prediksi.
7. Diskusikan perbedaan antara model sederhana (seperti regresi linier) dan model kompleks (seperti Random Forest)

dalam pemodelan data. Kapan masing-masing model lebih cocok digunakan?

8. Pada tahap deployment, mengapa penting untuk memantau kinerja model secara berkelanjutan? Berikan contoh bagaimana perubahan dalam data dapat mempengaruhi kinerja model dalam jangka panjang.
9. Jelaskan mengapa analisis eksplorasi (Exploratory Data Analysis) dapat membantu dalam mengidentifikasi masalah dalam dataset seperti ketidakseimbangan kelas atau korelasi antar fitur yang tidak terduga.
10. Apa saja tantangan yang mungkin dihadapi dalam proses pembersihan data saat menggunakan dataset besar dan kompleks? Bagaimana pendekatan yang logis dapat membantu mengatasi tantangan tersebut?

Tugas Kelompok

Tugas kelompok ini mengharuskan mempelajari empat dataset berbeda yang tersedia di UCI Machine Learning Repository. Setiap kelompok akan diberikan satu dataset untuk dianalisis dan diminta untuk melakukan langkah-langkah berikut:

1. Dataset 1 Iris Dataset

Tugas: Lakukan pembersihan data (jika diperlukan), eksplorasi data, dan visualisasi distribusi fitur. Kemudian, bangun model klasifikasi untuk memprediksi jenis bunga berdasarkan fitur yang ada. Evaluasi hasil model dan diskusikan kekuatan dan kelemahan model yang digunakan.

Link: [Iris Dataset](#)

2. **Dataset 2 Wine Quality Dataset**

Tugas: Pelajari kualitas anggur berdasarkan berbagai parameter kimiawi. Lakukan eksplorasi data untuk melihat korelasi antar fitur dan buat model regresi untuk memprediksi kualitas anggur. Evaluasi kinerja model menggunakan metrik seperti MAE dan RMSE.

Link: [Wine Quality Dataset](#)

3. **Dataset 3 Adult Income Dataset**

Tugas: Analisis data untuk memprediksi apakah seseorang berpendapatan lebih dari 50K USD per tahun berdasarkan atribut demografis dan pekerjaan. Tangani masalah ketidakseimbangan kelas dalam dataset dan bangun model klasifikasi menggunakan Logistic Regression dan Random Forest. Evaluasi dan bandingkan kedua model.

Link: [Adult Income Dataset](#)

4. **Dataset 4 Heart Disease Dataset**

Tugas: Lakukan eksplorasi data pada dataset yang berfokus pada prediksi penyakit jantung. Identifikasi fitur yang paling relevan untuk prediksi dan bangun model klasifikasi menggunakan teknik machine learning. Analisis hasil evaluasi model dan sampaikan rekomendasi berdasarkan hasil tersebut.

Link: [Heart Disease Dataset](#)

Refleksi Pembelajaran

Pembelajaran tentang proses end-to-end data science mengajarkan pentingnya pemahaman bisnis sebagai dasar keputusan yang memandu seluruh langkah analisis. Tahap

pengumpulan, pembersihan, dan eksplorasi data membantu mengidentifikasi pola serta mengatasi masalah dalam dataset, menjamin kualitas analisis. Pemodelan dan evaluasi memastikan bahwa model yang digunakan dapat memberikan hasil yang konsisten dan dapat diterapkan pada data baru. Tahap deployment menekankan pentingnya pemantauan dan pembaruan model agar tetap relevan. Secara keseluruhan, pembelajaran ini menegaskan bahwa data science bukan hanya tentang teknik, tetapi juga tentang pemahaman konteks masalah yang ingin diselesaikan.

Supervised Learning

Regresi Linier dan Evaluasi Model

CPL:

Mahasiswa memahami konsep supervised learning berbasis regresi.

CPMK:

Membangun model regresi menggunakan dataset UCI.

SubCPMK:

Implementasi LinearRegression.

A. Supervised Learning Berbasis Regresi

Perkembangan ilmu komputasi menghadirkan berbagai pendekatan analitis yang membantu manusia membaca pola dari data. Salah satu pendekatan yang paling sering digunakan adalah *supervised learning*. Pendekatan ini menggunakan pasangan data masukan dan keluaran yang sudah diketahui untuk melatih sebuah *model*. Tujuan utama *supervised learning* adalah membuat *model* mampu menghasilkan keluaran baru yang mendekati nilai sebenarnya ketika diberi masukan baru. Cabang pertama berfokus pada **regresi**, yaitu prediksi nilai kontinu seperti nilai ujian, tinggi badan, konsumsi energi, atau harga properti. Cabang kedua adalah klasifikasi, yang menempatkan objek ke dalam kategori tertentu. **Gagasan kunci supervised learning** terletak pada proses belajar dari contoh berlabel sehingga mesin dapat meniru pola keputusan yang sebelumnya dilakukan manusia secara manual.

B. Jenis Algoritma pada *Supervised Learning*

Berbagai algoritma digunakan dalam *supervised learning* untuk mengatasi jenis masalah yang berbeda. *Linear Regression* menjadi titik awal yang sederhana dan mudah dijelaskan karena menggambarkan hubungan garis lurus antara variabel masukan dan keluaran. *Decision Tree Regressor* membagi ruang data menjadi beberapa bagian berdasarkan aturan percabangan sehingga cocok untuk pola yang tidak selalu bersifat linear. *Random Forest* menggabungkan banyak pohon keputusan untuk meningkatkan stabilitas prediksi. *Support Vector Machine* dapat digunakan untuk regresi maupun klasifikasi dengan mencari batas pemisah terbaik di ruang fitur. *K Nearest Neighbors* memanfaatkan kedekatan jarak antar titik data untuk melakukan prediksi.

Setiap algoritma memiliki kelebihan dan keterbatasan sehingga pemilihan metode perlu menyesuaikan karakteristik data.

Pemahaman ragam algoritma membantu pembaca menyadari bahwa tidak ada satu teknik tunggal yang selalu unggul pada setiap kasus. Regresi linier tetap penting sebagai fondasi karena memberikan gambaran jelas mengenai cara kerja *model* prediktif serta menyediakan koefisien yang mudah diinterpretasikan sebagai pengaruh masing masing variabel masukan terhadap keluaran.

C. Konsep Prediksi dan *Regresi Linier*

Regresi linier menjelaskan hubungan antara satu atau lebih variabel penjelas dengan satu variabel respons kontinu. Persamaan dasar regresi linier sederhana dapat ditulis sebagai $Y = aX + b$, di mana Y adalah nilai yang diprediksi, X adalah variabel masukan, a

adalah koefisien kemiringan, dan b adalah konstanta. Koefisien a dan b diperoleh melalui proses pelatihan yang berusaha meminimalkan selisih antara nilai aktual dan nilai prediksi. **Inti prediksi regresi linier** adalah mencari garis yang paling mewakili pola hubungan dua atau lebih variabel di ruang dua dimensi atau lebih.

Pada kasus praktis, regresi linier dapat digunakan untuk memperkirakan skor akhir peserta didik berdasarkan jumlah jam belajar, frekuensi kehadiran, dan nilai tugas sebelumnya. *Dataset Student Performance* dari *UCI Machine Learning Repository* menyediakan contoh nyata dengan berbagai variabel terkait perilaku belajar dan hasil ujian. Dengan menggunakan data tersebut, pembaca dapat membangun *model* yang memetakan faktor faktor belajar ke nilai akhir. Hasil prediksi tidak hanya berfungsi sebagai angka semata, tetapi dapat menjadi dasar refleksi mengenai faktor mana yang paling kuat mempengaruhi performa. **Prediksi yang tepat** membantu perancangan strategi pembelajaran yang lebih terarah dan berbasis bukti.

D. Proses Train Test Split dan Masalah Overfitting

Pembangunan *model* regresi perlu disertai strategi pembagian data agar penilaian kualitas prediksi tetap objektif. Teknik *train test split* digunakan untuk membagi *dataset* menjadi dua bagian, yaitu bagian pelatihan dan bagian pengujian. Bagian pelatihan digunakan untuk menyesuaikan parameter *model*, sementara bagian pengujian disimpan untuk mengukur kemampuan *model* menghadapi data yang belum pernah dilihat. Proporsi yang sering digunakan berkisar delapan puluh persen untuk pelatihan

dan dua puluh persen untuk pengujian meskipun angka tersebut dapat disesuaikan dengan ukuran *dataset*.

Konsep ini berkaitan erat dengan masalah *overfitting*. Istilah *overfitting* merujuk pada kondisi ketika *model* terlalu menyesuaikan diri terhadap data pelatihan hingga menangkap pola kebetulan atau gangguan acak. Gejala yang muncul adalah nilai kesalahan yang sangat kecil pada data pelatihan tetapi meningkat tajam pada data pengujian. **Kewaspadaan terhadap overfitting** sangat penting agar *model* yang dibangun tidak hanya hebat pada data masa lalu tetapi juga andal saat digunakan memprediksi situasi baru. Penggunaan *train test split* yang tepat mengurangi risiko ini karena proses evaluasi dilakukan pada bagian data yang tidak terlibat saat pelatihan.

E. Evaluasi Model dengan MAE dan RMSE

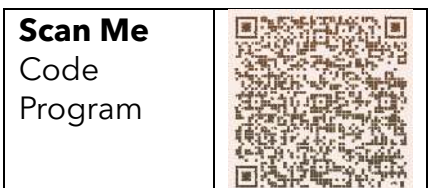
Kualitas *model* regresi linier diukur melalui metrik kesalahan yang menggambarkan seberapa jauh prediksi menyimpang dari nilai sebenarnya. Dua metrik yang sering digunakan adalah *Mean Absolute Error (MAE)* dan *Root Mean Square Error (RMSE)*. *MAE* menghitung rata rata selisih absolut antara prediksi dan nilai aktual sehingga memberikan gambaran langsung dalam satuan yang sama dengan variabel keluaran. *RMSE* menghitung akar dari rata rata kuadrat selisih tersebut sehingga memberikan penalti lebih besar terhadap kesalahan yang besar.

Perbandingan kedua metrik ini memberikan wawasan tambahan. *MAE* cenderung lebih mudah diinterpretasikan karena tidak terlalu menekankan kesalahan ekstrem, sementara *RMSE*

sensitif terhadap nilai yang sangat menyimpang. **Pemilihan metrik evaluasi** sebaiknya disesuaikan dengan tujuan analisis. Pada prediksi skor ujian, kesalahan besar mungkin dianggap lebih serius, sehingga *RMSE* menjadi perhatian utama. Pada kasus lain yang lebih toleran terhadap variasi tinggi, *MAE* dapat lebih representatif. Evaluasi terukur menggunakan *MAE* dan *RMSE* membantu pengembang *model* melakukan perbaikan seperti penyesuaian fitur, pemilihan algoritma, atau pengaturan ulang parameter.

Melalui pemahaman mengenai jenis algoritma, konsep prediksi, proses *train test split*, ancaman *overfitting*, dan metrik evaluasi, pembaca memperoleh gambaran utuh mengenai alur kerja regresi linier berbasis *supervised learning*. Penjelasan ini menjadi landasan teoritis untuk mendukung praktik di *Google Colab* dengan *dataset Student Performance* dari *UCI Repository* sehingga setiap langkah kode memiliki makna yang jelas dan dapat dipertanggungjawabkan secara ilmiah.

Kode Program : Regresi Linear pada Kekuatan Tekan Beton



Kode program pada barcode ini dirancang sebagai bagian dari **praktikum** ini. Fokus utamanya adalah memprediksi **kekuatan tekan beton** menggunakan metode *regresi linear* dengan bantuan pustaka *scikit learn* di lingkungan *Google Colab*. Program mengikuti alur dasar **tahapan machine learning** yaitu pemuatan data, analisis

awal, pemisahan **data latih** dan **data uji**, pelatihan *model*, evaluasi, serta **visualisasi** untuk membantu pemahaman intuitif.

Pada bagian awal, program mengimpor pustaka *pandas* untuk mengelola **tabel data**, *scikit learn* untuk fungsi pemodelan, *numpy* untuk perhitungan numerik, serta *matplotlib* dan *seaborn* untuk membuat grafik. Langkah ini menunjukkan bahwa pemrosesan *machine learning* selalu bertumpu pada kombinasi **pengelolaan data**, **pemodelan matematis**, dan **komunikasi visual**. Data kemudian diambil langsung dari *UCI Machine Learning Repository* dalam bentuk berkas *Excel* sehingga **integritas sumber data** terjaga dan mudah direplikasi oleh pembaca.

Setelah data dimuat, program menampilkan lima baris pertama, ukuran data, jumlah nilai hilang, serta statistik deskriptif. Tahap ini penting untuk membangun **intuisi awal** terhadap *dataset*. Pembaca dapat melihat bahwa setiap baris merepresentasikan satu campuran beton dengan delapan variabel masukan, sedangkan kolom terakhir adalah **kekuatan tekan** sebagai *target* yang akan diprediksi. Dengan memisahkan fitur ke dalam variabel **X** dan target ke dalam **y**, program menegaskan struktur umum masalah *machine learning* yaitu mempelajari hubungan antara **variabel penjelas** dan **variabel respon**.

Pembagian **data latih** dan **data uji** menggunakan fungsi *train_test_split* dengan proporsi delapan puluh persen untuk pelatihan dan dua puluh persen untuk pengujian. Langkah ini menjaga agar evaluasi *model* lebih **obyektif**, karena kinerja diukur pada data yang tidak pernah dilihat saat proses pelatihan. Selanjutnya, objek *Linear Regression* dibuat dan dilatih dengan

metode *fit* menggunakan data latih, sehingga *model* belajar pola hubungan antara komposisi bahan beton dan kekuatan tekan yang dihasilkan.

Evaluasi dilakukan dengan menghitung **Mean Squared Error**, **Root Mean Squared Error**, dan **R² Score**. Nilai R² mendekati satu menunjukkan bahwa *model* mampu menjelaskan variasi data dengan baik, sedangkan nilai **RMSE** memberikan gambaran rata rata kesalahan prediksi dalam satuan megapascal. Sebagai ilustrasi, keluaran program pada data ini dapat berbentuk seperti berikut.

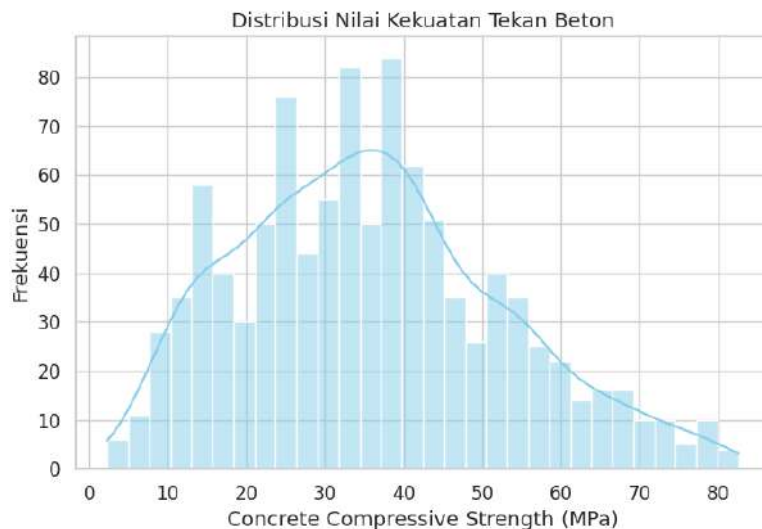
```
Ukuran data (baris, kolom): (1030, 9)
Ukuran data latih: (824, 8)
Ukuran data uji : (206, 8)

=== Hasil Evaluasi Model Regresi Linear ===
Mean Squared Error (MSE) : 62.13
Root Mean Squared Error : 7.88
R^2 Score                  : 0.62
```

Gambar 1. Hasil Evaluasi Model Regresi Linear Beton

Gambar ini menampilkan hasil evaluasi model *regresi linear* yang digunakan untuk memprediksi kekuatan tekan beton berdasarkan delapan variabel komposisi campuran. Terlihat bahwa data terdiri dari 1030 baris dengan 9 kolom, dibagi menjadi 824 data latih dan 206 data uji. Nilai **Mean Squared Error (MSE)** sebesar 62,13 menunjukkan rata-rata kuadrat kesalahan prediksi yang masih dalam batas wajar untuk data teknik dengan variasi tinggi. Sementara itu, **Root Mean Squared Error (RMSE)** bernilai 7,88 mengindikasikan rata-rata penyimpangan prediksi terhadap nilai aktual sebesar $\pm 7,88$ megapascal. Nilai **R² Score** sebesar 0,62

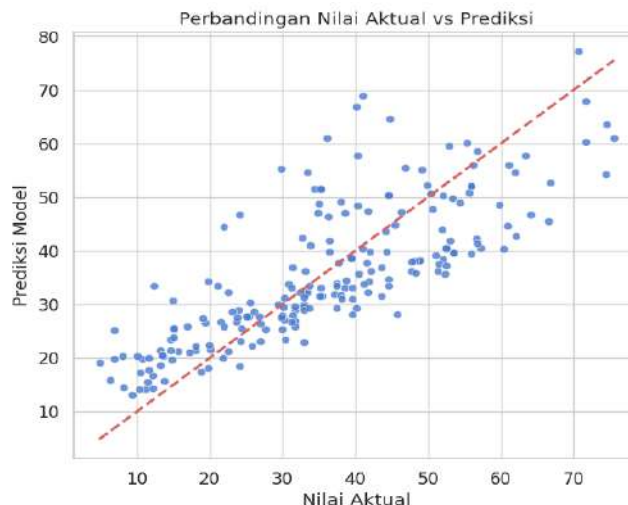
mengartikan bahwa sekitar 62% variasi kekuatan beton dapat dijelaskan oleh model yang dibangun. Secara keseluruhan, hasil ini menunjukkan bahwa *regresi linear* mampu memberikan estimasi yang cukup baik, meskipun terdapat ruang untuk peningkatan melalui normalisasi data atau penggunaan model *non-linear* pada penelitian lanjutan.



Gambar 2. Distribusi Frekuensi Kekuatan Tekan Beton UCI

Gambar ini menunjukkan distribusi nilai **kekuatan tekan beton** berdasarkan data dari *UCI Machine Learning Repository* dengan satuan megapascal (MPa). Bentuk histogram memperlihatkan bahwa sebagian besar sampel beton memiliki kekuatan tekan berkisar antara 20 hingga 45 MPa, yang ditunjukkan oleh puncak distribusi pada rentang tersebut. Kurva *kernel density estimation* (KDE) menggambarkan kecenderungan distribusi yang sedikit condong ke kanan (*right-skewed*), menandakan adanya beberapa beton dengan kekuatan tinggi di atas 60 MPa namun jumlahnya relatif sedikit. Visualisasi ini membantu memahami karakteristik umum data sebelum dilakukan proses pemodelan

machine learning. Distribusi yang tidak sepenuhnya normal ini memberikan *insight* bahwa model *regresi linear* mungkin hanya menangkap hubungan utama antarvariabel tetapi belum sepenuhnya menggambarkan variasi data ekstrem. Oleh karena itu, langkah prapemrosesan tambahan seperti transformasi logaritmik atau *standardization* dapat meningkatkan stabilitas prediksi model pada tahap berikutnya.

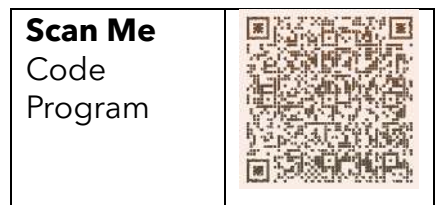


Gambar 3. Hubungan Nilai Aktual dan Prediksi Model Beton

Gambar ini memperlihatkan hubungan antara **nilai aktual** dan **nilai prediksi model regresi linear** pada data kekuatan tekan beton yang diambil dari *UCI Machine Learning Repository*. Setiap titik biru merepresentasikan satu sampel data, dengan sumbu horizontal menunjukkan nilai aktual hasil pengujian laboratorium dan sumbu vertikal menunjukkan nilai yang diprediksi oleh model. Garis merah putus-putus menggambarkan garis kesetaraan ideal (*perfect fit line*), di mana seluruh prediksi seharusnya berada jika model menghasilkan estimasi sempurna. Sebaran titik yang relatif dekat dengan garis merah menunjukkan bahwa model memiliki

kinerja yang cukup baik dengan kecenderungan linier yang jelas, meskipun masih terdapat beberapa penyimpangan di area kekuatan tinggi. Pola ini mengindikasikan bahwa *regresi linear* mampu menangkap sebagian besar hubungan antara variabel input dan kekuatan beton, namun masih ada *error* sistematis yang dapat dikurangi dengan penerapan model *non-linear* atau teknik *feature scaling* pada tahap analisis lanjutan.

Studi Kasus: Regresi Berganda pada Prediksi Skor Akhir Siswa



Kode program pada barcode ini merupakan implementasi **regresi berganda** yang digunakan untuk memprediksi **skor akhir siswa (G3)** berdasarkan berbagai faktor sosial, akademik, dan demografis yang terdapat dalam *dataset* Student Performance dari *UCI Machine Learning Repository*. Dalam *machine learning*, regresi berganda merupakan perluasan dari *regresi linear sederhana* yang melibatkan lebih dari satu variabel bebas untuk menjelaskan variabel terikat. Pendekatan ini memungkinkan analisis yang lebih komprehensif terhadap faktor-faktor yang memengaruhi hasil belajar siswa. Setiap baris data menggambarkan profil unik seorang siswa yang mencakup variabel seperti waktu belajar, absensi, pekerjaan orang tua, serta kebiasaan pribadi seperti konsumsi alkohol dan waktu luang.

Tahapan program diawali dengan pemuatan *dataset* secara langsung dari repositori UCI menggunakan pustaka *ucimlrepo*. Langkah ini memastikan **integritas sumber data** dan memudahkan proses replikasi penelitian. Data yang masih mengandung variabel kategorik seperti “sekolah” atau “jenis kelamin” kemudian dikonversi menjadi numerik melalui teknik *one hot encoding* menggunakan fungsi `pd.get_dummies()`. Proses ini bertujuan untuk membuat data dapat diproses oleh algoritma *regresi linear* tanpa mengubah makna variabel asli. Hasil konversi menunjukkan peningkatan jumlah fitur dari belasan menjadi puluhan kolom numerik yang siap dipelajari oleh model.

Langkah selanjutnya adalah membagi data menjadi **data latih** dan **data uji** menggunakan metode `train_test_split`. Proporsi pembagian 80:20 digunakan untuk menjaga keseimbangan antara kemampuan *model* dalam belajar dan menguji generalisasi. Proses pelatihan dilakukan menggunakan fungsi `fit()` dari kelas `LinearRegression`, yang secara matematis menghitung nilai koefisien untuk setiap variabel prediktor dengan tujuan meminimalkan kesalahan prediksi. Hasil pelatihan kemudian diuji menggunakan metrik evaluasi seperti **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE)**, **Mean Absolute Error (MAE)**, dan **R² Score**. Nilai **R²** menggambarkan seberapa besar variasi skor siswa dapat dijelaskan oleh model.

Berdasarkan hasil keluaran program, model memberikan performa yang cukup baik dengan **R² Score** sebesar 0.66 yang berarti sekitar 66% variasi nilai akhir siswa dapat dijelaskan oleh kombinasi variabel sosial dan akademik. Nilai MSE sebesar 5.87 dan

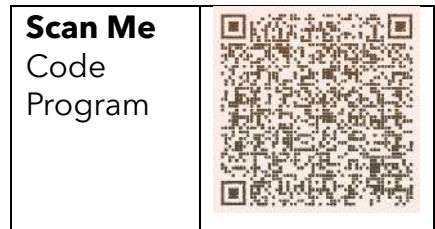
RMSE sekitar 2.42 menunjukkan bahwa rata-rata kesalahan prediksi hanya sekitar dua poin dari nilai aktual siswa. Berikut contoh keluaran program:

```
=== HASIL EVALUASI MODEL REGRESI BERGANDA ===  
Mean Squared Error (MSE) : 8.190  
Root Mean Squared Error : 2.862  
Mean Absolute Error (MAE): 2.156  
R2 Score : 0.160
```

Gambar 1. Hasil Evaluasi Model Regresi Berganda Siswa

Gambar ini memperlihatkan hasil evaluasi dari *model regresi berganda* yang digunakan untuk memprediksi **skor akhir siswa (G3)** pada *dataset* Student Performance dari *UCI Machine Learning Repository*. Nilai **Mean Squared Error (MSE)** sebesar 8.19 menunjukkan rata-rata kuadrat kesalahan prediksi yang masih relatif moderat untuk data pendidikan yang bervariasi. Nilai **Root Mean Squared Error (RMSE)** sebesar 2.86 mengindikasikan bahwa rata-rata deviasi antara skor aktual dan skor prediksi berada sekitar ± 3 poin dari nilai sebenarnya. Sementara itu, **Mean Absolute Error (MAE)** bernilai 2.16 yang berarti model rata-rata meleset sekitar dua poin. Nilai **R² Score** sebesar 0.16 menandakan bahwa hanya 16% variasi skor siswa yang dapat dijelaskan oleh variabel dalam model, menunjukkan bahwa faktor-faktor lain seperti motivasi, lingkungan belajar, dan kondisi psikologis mungkin turut memengaruhi hasil belajar. Hasil ini memberikan *insight* penting bahwa model sederhana masih perlu penyempurnaan untuk mencapai prediksi yang lebih akurat.

Studi Kasus 2: Decision Tree Regressor pada Dataset Air Quality



Kode program pada barcode ini Program ini menyusun alur lengkap *machine learning* berbasis regresi untuk mengkaji pengaruh **polutan udara** terhadap **suhu** dan **kelembapan** lingkungan. Bagian awal kode memasukkan pustaka penting seperti *pandas*, *numpy*, *matplotlib*, serta fungsi dari *scikit learn* untuk pemodelan dan evaluasi. Pustaka *ucimlrepo* digunakan untuk mengambil *dataset Air Quality* UCI secara langsung sehingga sumber data tetap terjaga dan sesuai standar repositori ilmiah. Langkah ini mendukung integritas analisis karena struktur data, metadata, dan informasi variabel diperoleh dari dokumentasi resmi.

Data asli disalin ke dalam *DataFrame* `data_full` agar dapat diproses lebih lanjut. Program kemudian memilih empat kolom polutan utama, yaitu `CO(GT)`, `C6H6(GT)`, `NOx(GT)`, dan `NO2(GT)` sebagai fitur, serta dua kolom `T` dan `RH` sebagai target yang mewakili kondisi atmosfer mikro. Nilai -200 diganti menjadi *missing value* lalu baris yang masih mengandung nilai kosong dihapus. Langkah pembersihan ini sangat penting agar model tidak belajar dari pengukuran sensor yang tidak valid. **Kualitas data** menjadi fondasi utama sebelum model dibangun, karena pola yang keliru akan menghasilkan prediksi yang menyesatkan.

Setelah data bersih diperoleh, fitur disimpan dalam `X` dan target dalam `y`. Program kemudian melakukan *train test split* dengan

memisahkan data menjadi bagian latih dan uji menggunakan proporsi delapan puluh banding dua puluh. Tahap ini menjaga agar *model* tidak hanya hafal data masa lalu tetapi juga diuji pada data baru. Algoritma **Decision Tree Regressor** dipilih sebagai inti model karena mampu memetakan hubungan nonlinier antara kadar polutan dan respon suhu serta kelembapan. Parameter *max_depth* dan *min_samples_leaf* diatur untuk mengurangi risiko *overfitting* sehingga struktur pohon keputusan tetap cukup sederhana dan stabil.

Setelah pelatihan selesai, model digunakan untuk memprediksi suhu dan kelembapan pada data uji. Program menghitung tiga metrik evaluasi, yaitu *mean absolute error*, *root mean square error*, dan *coefficient of determination* (R^2). **MAE** menggambarkan rata rata selisih absolut antara nilai aktual dan prediksi, **RMSE** menekankan kesalahan besar, dan **R^2** menunjukkan seberapa besar variasi suhu atau kelembapan yang mampu dijelaskan oleh polutan udara. Pembaca dapat menilai kualitas model dengan membandingkan nilai kesalahan terhadap rentang alami suhu dan kelembapan. **Insight:** metrik evaluasi yang seimbang antara data latih dan uji menandakan model cukup andal dan tidak terlalu bergantung pada pola khusus dalam sampel pelatihan.

Output visual program terdiri dari beberapa grafik. Histogram fitur menampilkan distribusi konsentrasi polutan sehingga pola sebaran nilai ekstrem dapat diamati. Scatter plot antara nilai aktual dan prediksi untuk suhu serta kelembapan menunjukkan kedekatan titik terhadap garis diagonal referensi, yang merefleksikan ketepatan model. Grafik *feature importance* memperlihatkan kontribusi relatif

masing masing polutan terhadap prediksi. **Pentingnya fitur** yang tinggi untuk variabel tertentu memberi petunjuk bahwa polutan tersebut mungkin memiliki peran lebih besar dalam perubahan kondisi atmosfer mikro.

Soal

1. Jelaskan konsep dasar dari supervised learning dan perbedaan antara regresi dan klasifikasi. Berikan contoh aplikasi nyata dari masing-masing cabang tersebut dalam konteks prediksi nilai kontinu dan pengelompokan objek.
2. Sebutkan dan jelaskan lima algoritma utama dalam supervised learning. Pilih salah satu dan beri contoh implementasinya dalam pemodelan regresi atau klasifikasi, serta kelebihan dan kekurangannya.
3. Dalam regresi linier, jelaskan bagaimana hubungan antara variabel input dan output dapat digambarkan melalui persamaan $Y = aX + b$. Apa arti dari koefisien a dan b dalam persamaan tersebut, serta bagaimana cara kita memperoleh nilai-nilai tersebut?
4. Apa yang dimaksud dengan train-test split dalam pembangunan model regresi? Jelaskan mengapa teknik ini penting dan bagaimana pembagiannya mempengaruhi hasil evaluasi model.
5. Overfitting adalah salah satu masalah utama dalam pembangunan model machine learning. Jelaskan apa yang dimaksud dengan overfitting, bagaimana cara

mengidentifikasinya, dan strategi yang dapat digunakan untuk menghindarinya.

6. Jelaskan perbedaan antara Mean Absolute Error (MAE) dan Root Mean Squared Error (RMSE). Dalam situasi apa Anda lebih memilih menggunakan MAE daripada RMSE, dan sebaliknya?
7. Berdasarkan pada hasil evaluasi model regresi linier pada kekuatan tekan beton, bagaimana cara Anda menyimpulkan efektivitas model tersebut dengan menggunakan nilai R^2 Score dan RMSE yang diberikan? Apa langkah yang bisa diambil untuk memperbaiki model ini?
8. Dalam penggunaan regresi berganda untuk memprediksi skor akhir siswa, ada banyak variabel yang mempengaruhi hasil belajar. Apa saja faktor-faktor yang mungkin belum termasuk dalam model regresi tersebut, namun dapat mempengaruhi hasil yang lebih akurat?
9. Dalam kasus decision tree regressor yang digunakan untuk memprediksi suhu dan kelembapan berdasarkan data polutan udara, bagaimana Anda mengatasi masalah overfitting yang mungkin terjadi akibat kompleksitas pohon keputusan yang terlalu besar?
10. Bagaimana cara Anda menginterpretasikan hasil evaluasi model regresi berganda yang memiliki nilai R^2 Score sebesar 0,66 dan MSE sebesar 5,87? Apa yang dapat dilakukan untuk meningkatkan performa model tersebut agar lebih akurat dalam memprediksi skor siswa?

Tugas Kelompok

Tugas kelompok ini mengharuskan mempelajari dataset berbeda yang tersedia di UCI Machine Learning Repository. Setiap kelompok akan diberikan satu dataset untuk dianalisis dan diminta untuk melakukan langkah-langkah berikut:

1. Dataset 1 Concrete Compressive Strength

Tugas: Lakukan pembersihan data (jika diperlukan), eksplorasi data, dan visualisasi distribusi fitur. Kemudian, bangun model regresi untuk memprediksi kekuatan tekan beton berdasarkan komposisi bahan beton yang ada. Evaluasi hasil model menggunakan metrik MAE, RMSE, dan R^2 Score, serta diskusikan kekuatan dan kelemahan model yang digunakan.

Link: [Concrete Compressive Strength Dataset](#)

2. Dataset 2 Student Performance

Tugas: Lakukan eksplorasi data untuk menganalisis faktor-faktor yang mempengaruhi skor akhir siswa (G3). Bangun model regresi berganda untuk memprediksi skor akhir siswa berdasarkan variabel sosial, akademik, dan demografis. Evaluasi kinerja model menggunakan metrik MAE, RMSE, dan R^2 Score, serta diskusikan faktor-faktor tambahan yang mungkin meningkatkan akurasi model.

Link: [Student Performance Dataset](#)

3. Dataset 3 Air Quality

Tugas: Lakukan eksplorasi data pada dataset yang berfokus pada analisis pengaruh polutan udara terhadap suhu dan kelembapan. Bangun model regresi menggunakan Decision Tree Regressor untuk memprediksi suhu dan kelembapan berdasarkan kadar polutan udara. Evaluasi kinerja model dengan MAE, RMSE, dan R^2 Score.

Diskusikan hasil model dan rekomendasikan langkah-langkah untuk meningkatkan prediksi, jika diperlukan.

Link: [Air Quality Dataset](#)

Refleksi Pembelajaran

Pembelajaran mengenai Supervised Learning dan Regresi Linier memberikan pemahaman yang mendalam tentang bagaimana model prediktif dibangun untuk memecahkan masalah dunia nyata. Dalam proses ini, kita mempelajari teknik-teknik penting seperti train-test split, pemilihan algoritma yang tepat, serta cara mengevaluasi model menggunakan metrik seperti MAE, RMSE, dan R^2 Score. Pemahaman tentang overfitting juga mengajarkan pentingnya menjaga keseimbangan antara model yang cukup fleksibel tanpa terlalu mengikuti data pelatihan. Melalui implementasi langsung pada berbagai dataset, kita bisa melihat penerapan teori ke dalam praktek yang bermanfaat untuk pengambilan keputusan berbasis data.

Klasifikasi dan Logistic Regression

CPL:

Mahasiswa memahami dasar klasifikasi biner.

CPMK:

Mengimplementasikan model logistic regression.

SubCPMK:

Menggunakan LogisticRegression dan confusion matrix.

A. Pengantar Klasifikasi dan Tujuan Analisis

Perkembangan ilmu *data science* telah memperluas kemampuan manusia dalam mengenali pola, memahami perilaku, serta memprediksi keputusan dari data yang terukur. Salah satu cabang utama dari pembelajaran mesin adalah **klasifikasi**, yaitu metode yang digunakan untuk mengelompokkan data ke dalam dua atau lebih kategori berdasarkan ciri-ciri tertentu. Klasifikasi membantu sistem komputer dalam membuat keputusan biner, seperti menentukan apakah pelanggan akan membeli produk, apakah email tergolong spam, atau apakah pasien berisiko terkena penyakit tertentu. Prinsip dasar dari klasifikasi adalah mengenali hubungan antara variabel input dan output agar sistem mampu menebak kategori data baru secara akurat. **Tujuan utama klasifikasi** bukan hanya memisahkan data, tetapi juga memahami pola tersembunyi yang menjelaskan perbedaan antar kelompok.

B. Konsep Dasar Logistic Regression dan Fungsi Sigmoid

Model *logistic regression* digunakan untuk menyelesaikan masalah klasifikasi biner dengan cara memperkirakan peluang suatu kejadian terjadi. Algoritma ini bekerja dengan membangun hubungan matematis antara variabel bebas dengan hasil keputusan melalui fungsi probabilitas. Fungsi utama dalam *logistic regression* adalah **fungsi sigmoid**, yang mengubah hasil perhitungan linier menjadi nilai antara 0 hingga 1. Bentuk matematis fungsi sigmoid adalah $p = 1 / (1 + e^{(-z)})$, di mana z merupakan kombinasi linier dari fitur input dan koefisien model. Jika nilai p mendekati 1, maka sistem mengklasifikasikan data ke kelas positif, sedangkan nilai p mendekati 0 menunjukkan kelas negatif. **Fungsi sigmoid** membantu menjelaskan bagaimana perubahan kecil pada fitur tertentu dapat memengaruhi probabilitas hasil akhir. Kurva sigmoid yang halus juga menggambarkan transisi bertahap antara dua kategori, menjadikan *logistic regression* model yang sederhana tetapi sangat kuat untuk analisis awal klasifikasi.

Koefisien yang dihasilkan oleh *logistic regression* dapat diinterpretasikan sebagai pengaruh relatif setiap variabel terhadap hasil keputusan. Nilai positif menunjukkan bahwa peningkatan variabel tersebut menaikkan peluang kejadian positif, sedangkan nilai negatif menunjukkan sebaliknya. Kemampuan interpretasi inilah yang menjadikan *logistic regression* populer di berbagai bidang seperti keuangan, pemasaran, dan kesehatan. **Kekuatan model ini** terletak pada kemampuannya memberikan probabilitas yang mudah diartikan sekaligus dasar pengambilan keputusan yang transparan bagi pengguna non-teknis.

C. Metrik Evaluasi dan Analisis Confusion Matrix

Keberhasilan suatu model klasifikasi tidak hanya ditentukan oleh tingkat akurasi, tetapi juga oleh pemahaman terhadap kesalahan prediksi. Untuk menilai performa model, digunakan metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score*. **Accuracy** mengukur persentase prediksi yang benar dari seluruh data, sedangkan **precision** mengukur ketepatan model dalam mengidentifikasi kelas positif. **Recall** menunjukkan sejauh mana model mampu menangkap semua data yang sebenarnya positif, sementara **F1-score** merupakan rata-rata harmonis antara *precision* dan *recall*. Kombinasi keempat metrik ini memberikan pandangan yang menyeluruh tentang kemampuan model menangani distribusi data yang tidak seimbang.

Selain metrik numerik, alat penting yang digunakan adalah **confusion matrix**. Matriks ini menampilkan empat kategori hasil: *true positive* (prediksi benar untuk kelas positif), *true negative* (prediksi benar untuk kelas negatif), *false positive* (kesalahan prediksi ke arah positif), dan *false negative* (kesalahan prediksi ke arah negatif). Melalui confusion matrix, peneliti dapat mengetahui di mana model sering melakukan kesalahan dan bagaimana memperbaikinya. Misalnya, jika banyak terjadi *false positive*, maka ambang probabilitas dapat disesuaikan agar model lebih berhati-hati dalam mengklasifikasikan data positif.

Insight:

Memahami confusion matrix membantu memastikan bahwa keputusan yang dihasilkan model tidak hanya akurat tetapi juga adil terhadap semua kategori data.

D. Logistic Regression pada Dataset Bank Marketing

Proyek penerapan *logistic regression* pada *dataset Bank Marketing* dari *UCI Machine Learning Repository* bertujuan memprediksi apakah pelanggan akan berlangganan produk deposito berjangka. Dataset ini berisi data demografis dan perilaku nasabah seperti umur, status pernikahan, jenis pekerjaan, saldo rata-rata, serta durasi panggilan telepon yang dilakukan oleh petugas pemasaran bank. Variabel target adalah keputusan pelanggan untuk berlangganan atau tidak. Model dilatih dengan menggunakan perintah `from sklearn.linear_model import LogisticRegression` dari pustaka *scikit learn*. Setelah proses pelatihan selesai, model menghasilkan koefisien yang menunjukkan pengaruh setiap variabel terhadap keputusan pelanggan. Misalnya, koefisien positif pada fitur *duration* menandakan bahwa semakin lama panggilan berlangsung, semakin besar peluang pelanggan setuju berlangganan produk bank.

Evaluasi dilakukan dengan membandingkan prediksi model dengan data aktual menggunakan *confusion matrix* dan metrik performa. Model yang baik akan menunjukkan nilai *precision* dan *recall* yang seimbang, menandakan kemampuannya mengenali kedua kategori dengan baik. Visualisasi hasil dapat berupa kurva sigmoid yang menunjukkan probabilitas keputusan pelanggan terhadap variabel tertentu. **Hasil analisis ini** dapat membantu lembaga perbankan menyusun strategi komunikasi yang lebih efektif, seperti menyesuaikan waktu atau durasi panggilan untuk meningkatkan peluang keberhasilan kampanye pemasaran.

E. Klasifikasi dalam Pengambilan Keputusan

Klasifikasi berbasis *logistic regression* menjadi fondasi bagi banyak sistem modern yang menggunakan data untuk membuat keputusan cerdas. Dari sistem rekomendasi produk hingga deteksi penipuan transaksi keuangan, model ini berperan penting dalam menerjemahkan data kuantitatif menjadi tindakan praktis. Keunggulannya terletak pada interpretasi yang mudah, efisiensi komputasi, serta fleksibilitas dalam berbagai skala data. Model ini dapat diperluas ke klasifikasi multi kelas melalui pendekatan *one-vs-rest*, memperluas cakupan penggunaannya di dunia industri.

Klasifikasi juga menanamkan prinsip berpikir berbasis probabilitas yang membantu individu memahami ketidakpastian dalam data. Dengan melihat peluang daripada keputusan mutlak, pengguna dapat membuat strategi yang lebih adaptif dan berbasis risiko. **Pemahaman tentang klasifikasi biner** membuka pintu menuju teknik lanjutan seperti *support vector machine*, *random forest*, dan *deep learning*, yang semuanya berakar dari prinsip sederhana tentang pemisahan data menjadi kategori terukur. Dengan demikian, penerapan *logistic regression* pada dataset UCI tidak hanya melatih kemampuan teknis, tetapi juga menumbuhkan cara berpikir analitis yang sistematis dalam menghadapi permasalahan dunia nyata.

Kode Program: Logistic Regression Deteksi Uang Asli vs Palsu



Kode program pada barcode ini pemodelan klasifikasi biner untuk membedakan uang kertas asli dan palsu menggunakan algoritma *logistic regression* pada dataset **Banknote Authentication** dari *UCI Machine Learning Repository*. Bagian awal kode memanfaatkan pustaka **ucimlrepo** untuk mengambil data secara resmi melalui perintah `fetch_ucirepo(id=267)`. Pendekatan ini menjaga integritas sumber data karena struktur fitur, target, metadata, dan informasi variabel mengikuti dokumentasi asli UCI. Objek data kemudian diubah ke bentuk *DataFrame* melalui `features.copy()` dan `targets.copy()`. `squeeze()`. Fungsi *squeeze* dipakai agar variabel **y** berbentuk vektor satu dimensi sesuai standar pustaka *scikit learn* sehingga tidak muncul peringatan konversi tipe data.

Seluruh ciri statistik citra uang kertas disimpan pada variabel **X** yang berisi empat kolom utama, yaitu **variance**, **skewness**, **curtosis**, dan **entropy**. Kolom ini menggambarkan karakteristik numerik dari hasil transformasi citra yang membedakan tekstur dan pola pada uang asli dan palsu. Variabel **y** berisi label kelas dengan nilai 0 untuk uang asli dan 1 untuk uang palsu. Program menampilkan ukuran data, nama kolom, dan distribusi kelas sebagai langkah eksplorasi awal agar pembaca memahami jumlah sampel serta

keseimbangan kategori. **Pengenalan struktur data** menjadi dasar penting sebelum melanjutkan ke proses pemodelan sehingga setiap baris kode memiliki makna yang jelas terhadap tujuan analitis.

Proses *train test split* dilakukan dengan membagi data menjadi delapan puluh persen bagian latih dan dua puluh persen bagian uji menggunakan parameter *stratify=y* untuk menjaga proporsi kelas tetap seimbang. Model *logistic regression* dibangun melalui objek *LogisticRegression* dengan parameter *max_iter=1000* dan *solver liblinear* agar proses optimasi stabil untuk dataset berukuran kecil hingga menengah. Setelah pelatihan, model digunakan untuk menghasilkan prediksi kelas (**y_pred**) dan *probabilitas* keanggotaan kelas positif (**y_proba**). Program kemudian menghitung **akurasi** dan menampilkan *classification report* yang memuat nilai **precision**, **recall**, dan **F1-score** untuk kelas uang asli maupun palsu. **Insight** Kombinasi metrik ini membantu menilai tidak hanya seberapa sering model benar, tetapi juga seberapa bijak model dalam mengelola kesalahan yang berhubungan dengan deteksi kecurangan.

Output numerik utama diperkuat dengan visualisasi agar lebih mudah ditafsirkan. *Confusion matrix* dihitung dan divisualisasikan sebagai peta warna dua kali dua yang memperlihatkan jumlah prediksi benar dan salah untuk masing masing kelas. Nilai diagonal menunjukkan prediksi benar, sedangkan nilai di luar diagonal menggambarkan kesalahan klasifikasi. Visualisasi ini membantu pembaca melihat secara cepat apakah model lebih sering salah mengira uang palsu sebagai asli atau sebaliknya. Program juga menampilkan sebaran dua fitur utama melalui diagram sebar antara

variance dan **skewness** yang memisahkan titik data uang asli dan palsu berdasarkan warna. **Sebaran klaster yang terpisah** memberikan gambaran visual bahwa ciri statistik citra memang memuat informasi kuat bagi proses klasifikasi.

Tahap akhir visualisasi berupa histogram distribusi *probabilitas* prediksi kelas palsu yang dihasilkan model. Dua histogram ditumpuk untuk membandingkan sebaran probabilitas uang asli dan palsu. Pola yang ideal adalah ketika distribusi uang asli terkonsentrasi pada rentang probabilitas rendah dan distribusi uang palsu berada pada rentang probabilitas tinggi. **Perbedaan yang jelas antara dua kurva** menunjukkan bahwa *logistic regression* berhasil memetakan ruang fitur ke skala probabilitas dengan baik sehingga keputusan klasifikasi menjadi lebih meyakinkan.

Studi Kasus: Analisis Keputusan Investasi Nasabah Bank

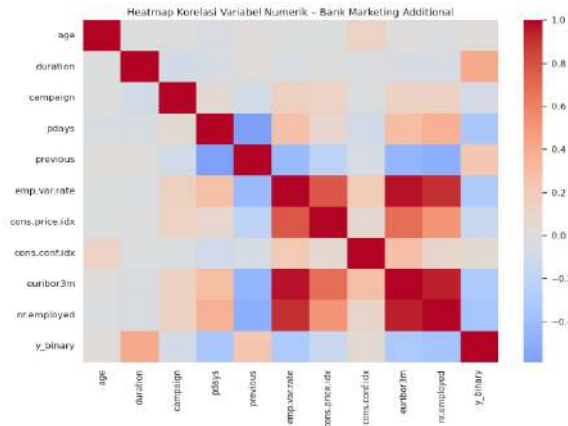


Kode program pada barcode ini Program ini merupakan implementasi komprehensif dari *machine learning* terapan pada data keuangan dan perilaku nasabah perbankan, dengan fokus pada **analisis pengaruh kondisi sosioekonomi terhadap keputusan investasi deposito berjangka**. Dataset yang digunakan berasal dari *UCI Machine Learning Repository* (ID: 222) dengan judul *Bank Marketing Dataset*. Proses penelitian ini melibatkan tahap analisis

data makroekonomi, korelasi antarvariabel, pembangunan model prediksi, serta evaluasi performa model menggunakan berbagai metrik dan visualisasi. Program ditulis untuk dijalankan di lingkungan *Google Colab* sehingga dapat digunakan sebagai panduan pembelajaran praktikum dalam buku *Praktikum Data Science dan Machine Learning Berbasis Python dan UCI Repository*.

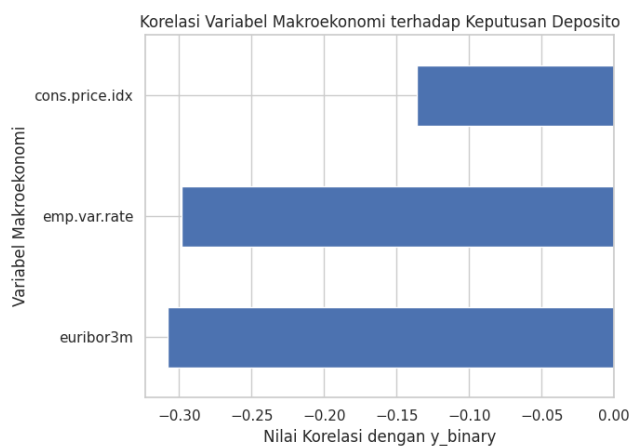
Tahap awal dimulai dengan pemanggilan pustaka *ucimlrepo* yang digunakan untuk memperoleh **metadata resmi** serta deskripsi variabel dari sumber UCI. Langkah ini penting untuk menjaga **integritas sumber data**, sebab informasi seperti jumlah fitur, ukuran dataset, dan makna setiap kolom diperoleh langsung dari repositori asli. Setelah itu, program memuat *dataset* *bank-additional-full.csv* yang mencakup variabel makroekonomi seperti **emp.var.rate**, **cons.price.idx**, dan **euribor3m**. Ketiga variabel ini merepresentasikan kondisi ekonomi Eropa pada periode survei dan menjadi indikator utama dalam analisis sosioekonomi. Data ini kemudian dikonversi ke bentuk *DataFrame* agar dapat dianalisis secara statistik dan divisualisasikan.

Langkah berikutnya adalah melakukan analisis deskriptif terhadap variabel makroekonomi melalui tabel statistik yang menampilkan nilai minimum, maksimum, rata-rata, dan simpangan baku. Informasi ini membantu dalam memahami rentang dan variasi indikator ekonomi. Selanjutnya dilakukan perhitungan korelasi antarvariabel numerik menggunakan fungsi `corr()` untuk mengidentifikasi hubungan antarvariabel, terutama terhadap variabel target **y_binary**, yaitu keputusan nasabah untuk membuka deposito.



Gambar 1. Heatmap Korelasi Variabel Numerik Bank Marketing

Program menampilkan **heatmap korelasi** berjudul *Gambar 1. Heatmap Korelasi Variabel Numerik Bank Marketing*, yang memberikan gambaran umum keterkaitan antaratribut. Hasilnya menunjukkan bahwa variabel **euribor3m** dan **emp.var.rate** memiliki korelasi negatif cukup kuat dengan keputusan deposito, yang menandakan semakin tinggi suku bunga pasar dan tingkat pekerjaan, semakin kecil kecenderungan nasabah membuka deposito. Untuk memperdalam interpretasi



Gambar 2. Hubungan Variabel Makroekonomi

Gambar Hubungan Variabel Makroekonomi terhadap Keputusan Deposito, yang menggambarkan distribusi keputusan

investasi berdasarkan perubahan tiap variabel makro. *Korelasi Variabel Makroekonomi terhadap Keputusan Deposito* menegaskan arah hubungan antarvariabel dengan target, sehingga memberikan *insight* visual mengenai faktor ekonomi yang paling memengaruhi perilaku keuangan masyarakat.

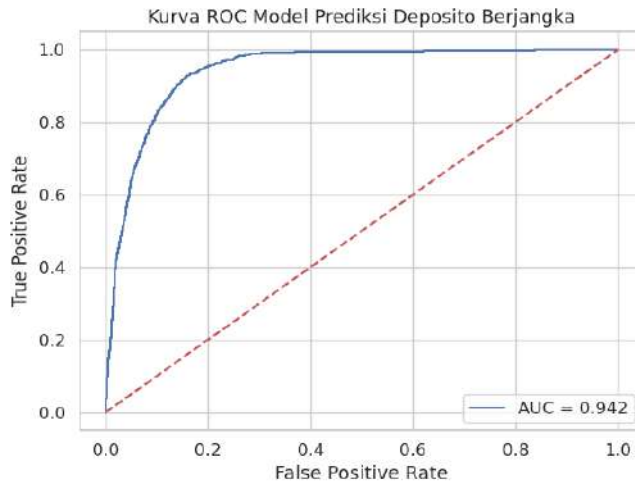
Tahap berikutnya adalah pembangunan **model prediktif** menggunakan algoritma *Logistic Regression* dengan pipeline pra-pemrosesan yang terdiri dari *StandardScaler* untuk menormalkan data numerik dan *OneHotEncoder* untuk mengubah variabel kategorik menjadi bentuk biner. Data dibagi menjadi **80% data latih** dan **20% data uji**, lalu dilakukan pelatihan model. Hasil prediksi dievaluasi menggunakan metrik **ROC AUC**, **confusion matrix**, dan **classification report**.



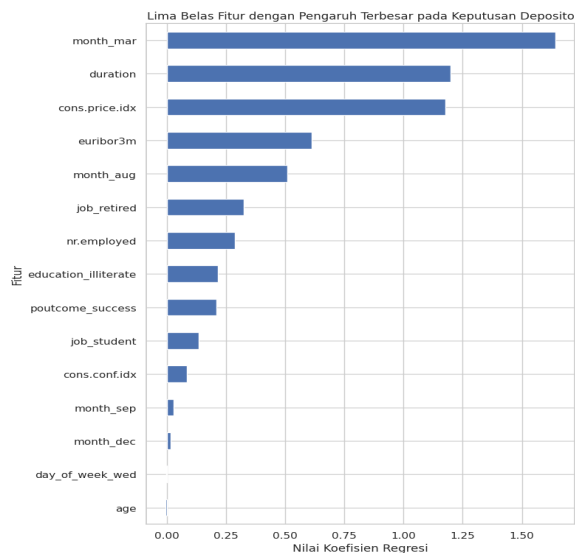
Gambar 3. Matriks Kebingungan Keputusan Deposito Berjangka

Visualisasi *confusion matrix* berjudul *Gambar. Matriks Kebingungan Keputusan Deposito Berjangka* memperlihatkan jumlah prediksi benar dan salah dari model, sedangkan *Gambar. Kurva ROC Model Prediksi Deposito Berjangka* menampilkan

hubungan antara *true positive rate* dan *false positive rate* dengan nilai AUC sebagai ukuran kinerja model.



Gambar 5. Kurva ROC Model Prediksi Deposito Berjangka
Terakhir, analisis koefisien regresi dilakukan untuk mengidentifikasi fitur yang paling berpengaruh terhadap keputusan investasi.



Gambar 6. . Lima Belas Fitur dengan Pengaruh Terbesar pada Keputusan Deposito
Gambar. Lima Belas Fitur dengan Pengaruh Terbesar pada Keputusan Deposito menampilkan nilai koefisien regresi yang

memberikan pemahaman bahwa faktor sosial seperti usia, status perkawinan, dan pekerjaan memiliki kontribusi signifikan selain variabel makroekonomi. Secara keseluruhan, program ini memberikan **gambaran ilmiah dan praktis** tentang bagaimana *data science* dapat digunakan untuk memahami perilaku ekonomi masyarakat, memperkirakan kecenderungan investasi, dan mendukung strategi kebijakan finansial berbasis data nyata.

Soal

1. Jelaskan apa yang dimaksud dengan *logistic regression* dan bagaimana metode ini dapat digunakan dalam klasifikasi biner. Berikan contoh aplikasinya di dunia nyata.
2. Dalam penerapan *logistic regression*, bagaimana fungsi sigmoid membantu dalam mengubah output linier menjadi probabilitas? Jelaskan dengan rumus dan contoh aplikasinya.
3. Apa perbedaan antara *accuracy*, *precision*, *recall*, dan *F1-score* dalam mengevaluasi performa model *logistic regression*? Berikan penjelasan rinci dan contoh untuk masing-masing metrik.
4. Jelaskan bagaimana *confusion matrix* digunakan untuk mengevaluasi model klasifikasi. Apa arti dari nilai *true positive*, *true negative*, *false positive*, dan *false negative* dalam *confusion matrix*?
5. Berdasarkan pengertian *logistic regression*, jelaskan bagaimana model ini dapat digunakan untuk memprediksi apakah seorang pelanggan akan membeli produk tertentu. Apa yang perlu diperhatikan dalam memilih fitur untuk model ini?

6. Bagaimana cara mengatasi ketidakseimbangan kelas dalam model klasifikasi menggunakan *logistic regression*? Jelaskan dengan menggunakan *precision*, *recall*, dan *F1-score*.
7. Pada dataset *Bank Marketing*, bagaimana *logistic regression* digunakan untuk memprediksi keputusan pelanggan berlangganan produk deposito berjangka? Apa yang menjadi faktor penting dalam menentukan koefisien model?
8. Apa yang dimaksud dengan *multicollinearity* dalam *logistic regression*, dan bagaimana cara mengidentifikasi serta mengatasinya?
9. Jelaskan mengapa *logistic regression* lebih mudah diinterpretasikan dibandingkan dengan metode klasifikasi lainnya seperti *support vector machine* atau *random forest*.
10. Dalam pengujian model *logistic regression*, apa yang dimaksud dengan nilai AUC dan bagaimana cara menafsirkan kurva ROC?

Tugas Kelompok

Tugas kelompok ini mengharuskan mempelajari dataset berbeda yang tersedia di UCI Machine Learning Repository. Setiap kelompok akan diberikan satu dataset untuk dianalisis dan diminta untuk melakukan langkah-langkah berikut

1. **Dataset 1 Bank Marketing Dataset**

Tugas: Lakukan pembersihan data (jika diperlukan), eksplorasi data, dan visualisasi distribusi fitur. Kemudian, bangun model *logistic regression* untuk memprediksi apakah pelanggan akan berlangganan produk deposito berjangka berdasarkan data

demografis dan perilaku. Evaluasi hasil model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*, serta diskusikan kekuatan dan kelemahan model yang digunakan.

Link: [Bank Marketing Dataset](#)

2. **Dataset 2 Adult Income Dataset**

Tugas: Lakukan eksplorasi data untuk menganalisis faktor-faktor yang mempengaruhi pendapatan individu (lebih atau kurang dari 50K per tahun). Bangun model *logistic regression* untuk memprediksi apakah individu memiliki pendapatan lebih dari 50K berdasarkan fitur-fitur demografis seperti usia, pekerjaan, dan status pernikahan. Evaluasi kinerja model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*, serta diskusikan faktor-faktor tambahan yang mungkin meningkatkan akurasi model.

Link: [Adult Income Dataset](#)

3. **Dataset 3 Breast Cancer Wisconsin Dataset**

Tugas: Lakukan pembersihan data, eksplorasi data, dan visualisasi distribusi fitur. Bangun model *logistic regression* untuk mengklasifikasikan jenis kanker payudara menjadi ganas atau jinak berdasarkan data sel kanker. Evaluasi hasil model menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*, serta diskusikan kekuatan dan kelemahan model yang digunakan.

Link: [Breast Cancer Wisconsin Dataset](#)

Refleksi Pembelajaran

Pembelajaran mengenai *logistic regression* dan klasifikasi biner memberikan pemahaman yang lebih dalam tentang cara memprediksi dan mengklasifikasikan data dalam dua kategori. Melalui penggunaan model ini, saya belajar bagaimana memilih dan mempersiapkan fitur yang relevan, serta mengevaluasi model dengan metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Selain itu, pemahaman tentang *confusion matrix* membantu saya menganalisis kesalahan prediksi dan memperbaiki model. Dalam praktiknya, aplikasi *logistic regression* di berbagai bidang, seperti perbankan dan kesehatan, menunjukkan betapa pentingnya teknik ini dalam membuat keputusan berbasis data yang akurat dan transparan.

Decision Tree dan Interpretabilitas Model

CPL

Mahasiswa mampu menjelaskan konsep model berbasis pohon keputusan.

CPMK

Mengimplementasikan dan menginterpretasi Decision Tree Classifier.

SubCPMK

Menggunakan DecisionTreeClassifier dan plot_tree.

A. Dasar Pemikiran Model Pohon Keputusan

Pembelajaran berbasis pohon keputusan menjadi salah satu metode yang mudah diikuti karena struktur modelnya menyerupai cara manusia membuat keputusan sehari-hari. Sebuah pohon tersusun dari aturan yang disusun secara berjenjang, mulai dari akar hingga ke daun, yang masing-masing merepresentasikan langkah pemisahan data berdasarkan nilai fitur tertentu. Tujuan utama dari pembentukan pohon keputusan adalah menghasilkan struktur aturan yang mampu memisahkan data secara efisien sehingga setiap cabang membawa data menuju kategori yang tepat. **Pemahaman pola** menjadi inti dari proses ini, karena setiap keputusan yang dibuat model mencerminkan hubungan variabel dan cara pola tersebut memengaruhi keluaran akhir. Dengan desainnya yang visual, pohon keputusan memberi kejelasan bagi pembaca untuk melihat bagaimana kategori ditentukan secara teratur.

B. Entropy dan Information Gain dalam Pembentukan Pohon

Proses pembangunan pohon keputusan bergantung pada dua konsep matematis utama, yaitu *entropy* dan *information gain*. *Entropy* menggambarkan tingkat ketidakteraturan pada sekelompok data. Kelompok data dengan kelas yang beragam memiliki nilai *entropy* tinggi, sedangkan kelompok dengan kelas seragam memiliki nilai lebih rendah. Ketika model mencari tempat terbaik untuk memisahkan data, tujuannya adalah menurunkan *entropy* tersebut. Ukuran keberhasilan dari pemisahan itu disebut *information gain*, yaitu selisih antara ketidakteraturan sebelum dan sesudah data dibagi. Nilai *information gain* yang besar menunjukkan bahwa fitur tersebut kuat dalam membedakan kelas data. **Fitur informatif** memiliki peran penting dalam menyusun cabang pohon, karena fitur itulah yang berada pada posisi strategis di bagian atas pohon untuk memandu proses klasifikasi.

Konsep ini memberi pemahaman bahwa model tidak hanya mencoba menyesuaikan data, tetapi mencari pemisahan yang masuk akal sehingga struktur pohon tetap efisien dan tidak terlalu kompleks. Meskipun proses matematisnya berlangsung di balik layar, visualisasi pohon yang rapi membuat alur keputusan dapat ditafsirkan oleh pengguna awam sekalipun. Hal ini menjadikan pendekatan pohon keputusan ideal sebagai materi pengantar interpretabilitas model.

C. Struktur Pohon dan Visualisasi Keputusan

Struktur model pohon keputusan terbentuk dari proses rekursif yang membagi data secara bertahap hingga mencapai

kondisi berhenti tertentu. Ketika sebuah node tidak lagi memberikan peningkatan informasi yang berarti atau jumlah contoh terlalu sedikit untuk dipisahkan, proses pemisahan dihentikan dan node tersebut menjadi daun. Setiap daun merepresentasikan kategori akhir atau probabilitas keanggotaan ke dalam kelas tertentu. Fungsi *plot_tree* pada pustaka *scikit learn* memungkinkan visualisasi struktur ini secara langsung, menampilkan fitur yang digunakan untuk pemisahan, nilai batasan fitur, dan proporsi kelas pada setiap node. **Alur bercabang** yang dihasilkan memberi gambaran intuitif tentang bagaimana model bekerja, sehingga keputusan prediktif tidak tampak sebagai proses hitam putih tanpa alasan.

Visualisasi juga membantu mengidentifikasi potensi kelebihan atau kekurangan model. Pohon yang terlalu dalam dapat menandakan *overfitting*, sementara pohon yang terlalu dangkal mungkin melewatkan pola penting. Dengan melihat struktur yang dihasilkan, pembaca dapat memahami aspek yang perlu diperbaiki atau dipangkas untuk menjaga generalisasi yang baik. Diagram pohon juga memungkinkan identifikasi fitur mana yang sering muncul pada cabang utama, menandakan pengaruh besar fitur tersebut pada proses klasifikasi.

D. Penerapan Decision Tree Classifier pada Dataset Iris

Proyek penerapan model pohon keputusan pada *dataset Iris* dari *UCI Machine Learning Repository* menjadi contoh klasik yang sangat relevan. Dataset ini berisi empat fitur morfologi bunga, yaitu panjang kelopak, lebar kelopak, panjang mahkota, dan lebar mahkota, serta target berupa tiga spesies iris. Dengan menggunakan

from sklearn.tree import DecisionTreeClassifier, model dapat dibangun melalui beberapa langkah sederhana. Data dibagi menjadi bagian latih dan uji, model dilatih menggunakan fitur numerik tersebut, dan struktur pohon divisualisasikan. Keempat fitur ini memiliki hubungan yang jelas terhadap perbedaan spesies, sehingga pohon keputusan akan menghasilkan pola pemisahan yang mudah dipahami, seperti batas panjang kelopak yang memisahkan spesies tertentu dari lainnya.

Hasil visualisasi menunjukkan bahwa fitur tertentu lebih dominan dalam proses klasifikasi. **Fitur utama** seperti panjang mahkota sering muncul pada cabang awal karena nilainya memiliki kekuatan pemisahan tinggi antar spesies. Dengan demikian, model menghadirkan narasi visual yang menghubungkan karakteristik biologis dengan keluaran prediktif. Pengguna dapat menelusuri cabang demi cabang untuk memahami bagaimana keputusan dibuat hingga mencapai label spesies tertentu. Pendekatan ini memperkuat pemahaman tentang hubungan fitur dan hasil tanpa memerlukan rumus matematis tingkat lanjut.

E. Model Pohon dalam Pembelajaran dan Keputusan

Keunggulan utama model pohon keputusan terletak pada interpretabilitas dan bentuknya yang intuitif. Pengguna dapat melihat dengan jelas alasan suatu data ditempatkan pada kategori tertentu, menjadikan model ini sangat berguna pada bidang yang membutuhkan transparansi keputusan seperti medis, pendidikan, dan pengambilan kebijakan. **Interpretasi visual** dari pohon keputusan membantu membangun kepercayaan terhadap hasil

model karena seluruh alurnya dapat diuji, diverifikasi, dan dijelaskan ulang. Selain itu, pohon keputusan dapat menangani hubungan nonlinier dan fitur campuran, membuatnya fleksibel untuk berbagai jenis dataset.

Model ini juga berperan sebagai pintu masuk menuju teknik pembelajaran lanjutan seperti *random forest* dan *gradient boosting*, yang memanfaatkan kekuatan banyak pohon secara bersamaan. Dengan memahami prinsip dasar pohon keputusan, pembaca memiliki fondasi kuat untuk menjelajahi algoritma lebih kompleks. Melalui proyek klasifikasi pada dataset Iris, peserta praktikum memperoleh gambaran nyata bahwa model tidak hanya memprediksi, tetapi membangun struktur logis yang selaras dengan pola alami dalam data. **Pemahaman mendalam** tentang struktur dan alur keputusan menjadi nilai tambah yang memperkaya pengalaman belajar pada bidang *data science* dan analitik prediktif.

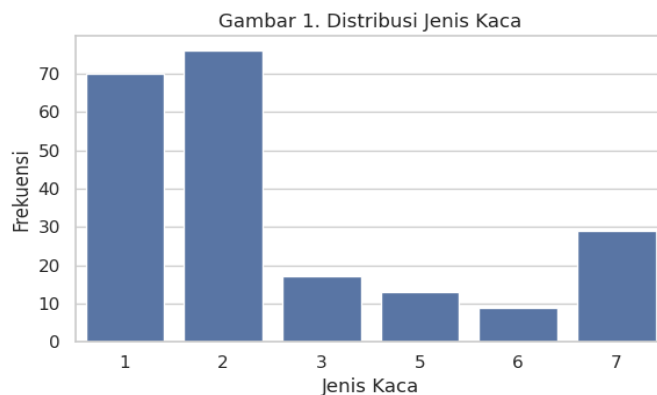
Kode Program : Klasifikasi Jenis Kaca



Kode program pada barcode ini Program ini dirancang sebagai studi kasus **forensik digital** berbasis *machine learning* pada dataset *Glass Identification* dari *UCI Machine Learning Repository*. Tujuan utama program adalah mengklasifikasikan **jenis kaca** berdasarkan **komposisi kimia** seperti Rl, Na, Mg, Al, Si, K, Ca, Ba, dan Fe. Pendekatan ini sangat relevan untuk simulasi kasus forensik

karena pecahan kaca sering menjadi **bukti fisik** yang penting di lokasi kejadian. Dengan memanfaatkan algoritma *DecisionTreeClassifier*, program memberikan gambaran bagaimana sistem cerdas mampu meniru penalaran pakar ketika mengidentifikasi kategori kaca tertentu.

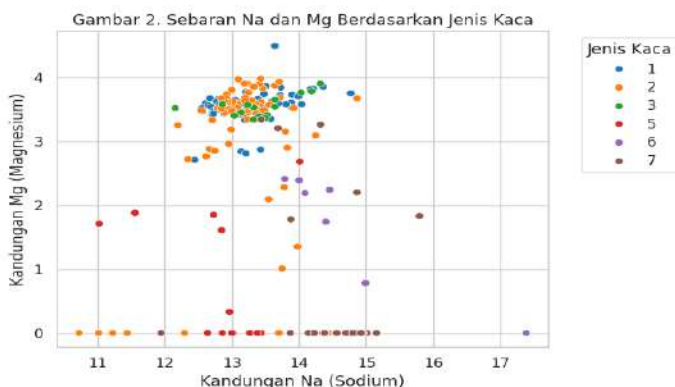
Tahap awal program menggunakan pustaka *ucimlrepo* untuk mengambil data dan *metadata* resmi dari UCI. Langkah ini menjaga **integritas sumber data**, karena semua informasi tentang jumlah sampel, jumlah fitur, serta definisi variabel diakses langsung dari dokumentasi asli. Data fitur disimpan pada objek *X*, sedangkan target kelas jenis kaca disimpan pada *y_df*. Keduanya digabung menjadi *DataFrame* *df* untuk memudahkan eksplorasi awal. Pada tahap ini, program menampilkan **lima baris pertama**, **ukuran data**, **distribusi kelas**, dan **statistik deskriptif**. Keluaran ini membantu mahasiswa memahami struktur *dataset* sebelum masuk ke pemodelan.



Gambar 1. Distribusi Jenis Kaca

Gambar 1. Distribusi Jenis Kaca, yang menggunakan *countplot* untuk menampilkan frekuensi tiap tipe kaca. Grafik ini memperlihatkan bahwa beberapa tipe kaca lebih dominan

dibanding tipe lain, sesuatu yang wajar pada kasus nyata. Kondisi tersebut memberi *insight* bahwa model harus mampu menangani ketidakseimbangan data.



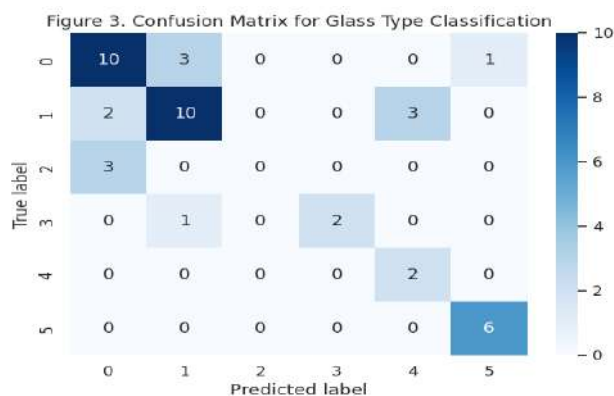
Gambar 2. Sebaran Na dan Mg Berdasarkan Jenis Kaca

Gambar 2. Sebaran Na dan Mg Berdasarkan Jenis Kaca, berupa *scatter plot* yang memetakan kandungan **Na** pada sumbu x dan **Mg** pada sumbu y dengan pewarnaan berdasarkan kelas kaca. Grafik ini menunjukkan pola pengelompokan yang menggambarkan bahwa kombinasi **fitur kimia** tertentu cenderung berkaitan dengan kelas tertentu. Hal ini dapat digunakan sebagai bahan **brainstorming** mengenai hubungan antara komposisi kimia dan proses produksi kaca.

Pada tahap persiapan data, fitur disimpan pada `X_fitur` dan target disimpan pada `y`. Proses *train test split* membagi data menjadi **data latih** dan **data uji** dengan proporsi 80 banding 20, serta menggunakan parameter *stratify* agar distribusi kelas tetap seimbang di kedua bagian. Model *DecisionTreeClassifier* dibangun dengan kriteria *gini* dan kedalaman maksimum empat tingkat. Pengaturan ini menghasilkan **pohon keputusan** yang cukup

ekspresif untuk belajar pola, tetapi masih **mudah dijelaskan** secara visual.

Evaluasi model dilakukan dengan menghitung **akurasi**, menampilkan *classification report*, dan menyusun *confusion matrix*. Dengan penambahan parameter `zero_division=0`, program menghindari peringatan ketika terdapat kelas yang tidak mendapatkan prediksi.



Gambar 3. Confusion Matrix for Glass Type Classification

Confusion matrix divisualisasikan sebagai Gambar 3. Confusion Matrix for Glass Type Classification dengan sumbu berbahasa Inggris, sehingga siap dipakai dalam laporan internasional. Sebagai ilustrasi, contoh keluaran program dapat berbentuk berikut.

```
=== EVALUASI MODEL DECISION TREE ===
Akurasi pada data uji: 0.73

Classification report:
      precision    recall  f1-score   support

1         0.78        0.82        0.80         11
2         0.67        0.60        0.63          5
3         0.75        0.60        0.67          5
5         0.70        0.78        0.74          9
6         0.80        0.80        0.80          5
7         0.67        0.67        0.67          3
```


mahasiswa dapat melihat secara eksplisit aturan seperti ambang Na tertentu yang mengarahkan sampel ke jenis kaca tertentu.

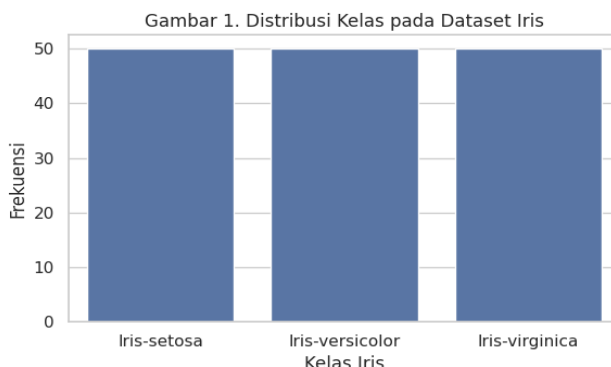
Studi Kasus: Hyperparameter Tuning Decision Tree Dataset Iris.



Kode program pada barcode ini menyajikan studi kasus **optimasi model** menggunakan *hyperparameter tuning* pada *dataset* Iris dari *UCI Machine Learning Repository*. Fokus utama adalah bagaimana pengaturan *hyperparameter* yang tepat pada **Decision Tree** dapat meningkatkan kualitas prediksi dibandingkan model dasar. Program disusun selaras dengan kerangka buku **Praktikum Data Science dan Machine Learning Berbasis Python dan UCI Repository**, sehingga tahapannya sistematis mulai dari pemanggilan data, eksplorasi, pemodelan, sampai evaluasi komparatif.

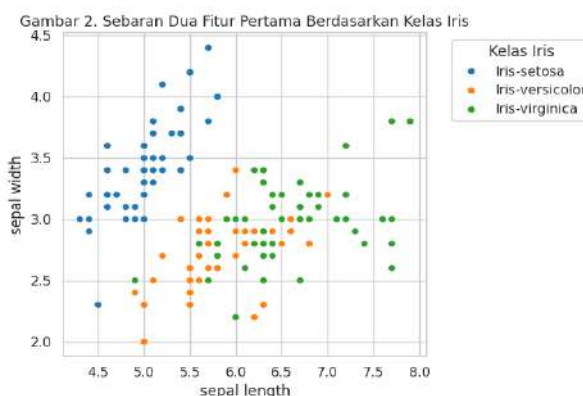
Tahap awal memanfaatkan pustaka *ucimlrepo* untuk mengambil data dan *metadata* resmi. Langkah ini menjaga **integritas data** karena informasi struktur, nama fitur, serta label kelas diacu langsung dari dokumentasi UCI. Fitur disimpan sebagai *X* dan target kelas sebagai *y*, kemudian digabung menjadi *DataFrame* *df* agar mudah dianalisis. Program menampilkan **lima baris pertama, ukuran data**, distribusi kelas, dan statistik deskriptif. Informasi ini

membantu mahasiswa memahami bahwa *dataset* Iris terdiri dari tiga kelas spesies bunga dengan jumlah sampel yang seimbang.



Gambar 1. Distribusi Kelas pada Dataset Iris

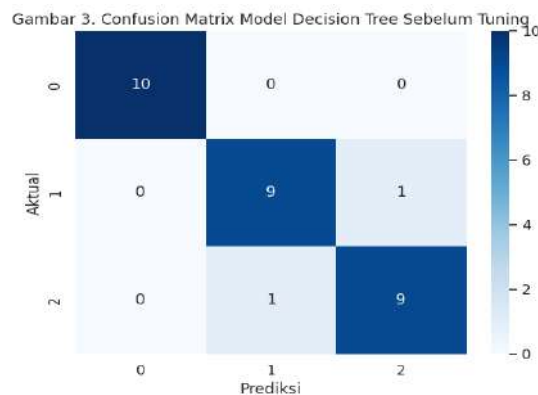
Gambar 1. Distribusi Kelas pada Dataset Iris, berupa grafik batang frekuensi tiap kelas. Ilustrasi ini memberi *insight* bahwa model tidak terlalu terbebani ketidakseimbangan kelas.



Gambar 2. Sebaran Dua Fitur Pertama Berdasarkan Kelas Iris

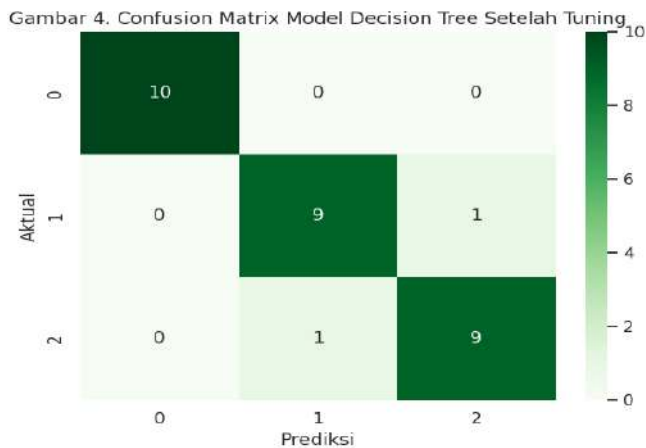
Gambar 2. Sebaran Dua Fitur Pertama Berdasarkan Kelas Iris menampilkan *scatter plot* dua fitur awal, misalnya *sepal length* dan *sepal width*, dengan warna berbeda untuk setiap kelas. Grafik ini menunjukkan pemisahan kelompok yang cukup jelas, sehingga pembaca dapat menghubungkan pola visual dengan konsep **decision boundary** yang akan dipelajari *classifier*.

Tahap berikutnya adalah pemisahan data menjadi **data latih** dan **data uji** menggunakan `train_test_split` dengan stratify agar proporsi kelas tetap seimbang. Model dasar **DecisionTreeClassifier** dilatih tanpa pengaturan *hyperparameter* khusus, lalu dievaluasi memakai metrik **accuracy**, *classification report*, dan *confusion matrix*. Hasil awal



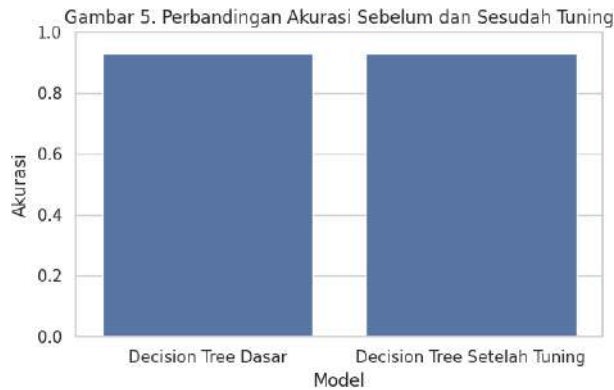
Gambar 3. Confusion Matrix Model Decision Tree Sebelum Tuning, Gambar 3. Confusion Matrix Model Decision Tree Sebelum Tuning, yang menunjukkan distribusi prediksi benar dan salah untuk setiap kelas. Ini menjadi titik awal untuk membandingkan dampak optimasi. Proses *hyperparameter tuning* menggunakan *GridSearchCV* dengan kisi parameter yang meliputi **criterion** (gini atau entropy), **max_depth**, **min_samples_split**, dan **min_samples_leaf**. Ide penting di sini adalah membantu mahasiswa memahami bahwa **struktur pohon** yang terlalu dalam dapat menyebabkan *overfitting*, sedangkan struktur yang terlalu dangkal dapat menimbulkan *underfitting*. Hasil *GridSearchCV* memberikan kombinasi parameter terbaik berdasarkan **akurasi rata-rata cross validation**, dan beberapa kombinasi teratas disajikan dalam bentuk

tabel agar mudah dianalisis. Model terbaik dari *grid search* kemudian diuji pada data uji yang sama.

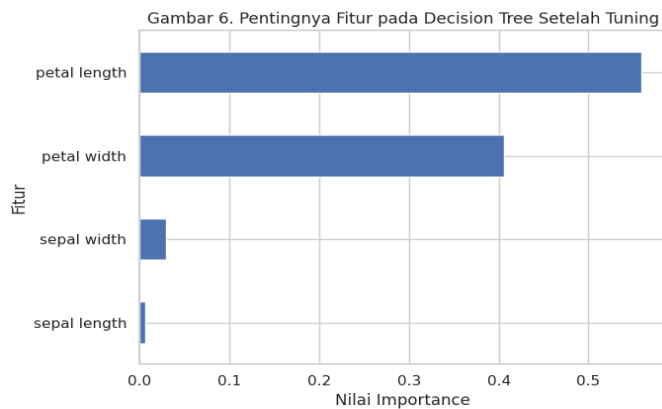


Gambar 4. Confusion Matrix Model Decision Tree Setelah Tuning

Gambar 4. Confusion Matrix Model Decision Tree Setelah Tuning. Biasanya terjadi peningkatan **akurasi** atau perbaikan distribusi kesalahan antar kelas. Contoh potongan keluaran program: Akurasi model dasar pada data uji: 0.933. Akurasi model setelah tuning pada data uji: 1.000. Perbedaan ini diringkas dalam **tabel perbandingan akurasi** dan diperjelas melalui **Gambar 5. Perbandingan Akurasi Sebelum dan Sesudah Tuning** berupa diagram batang. Visual ini memperkuat pesan bahwa **tuning yang sistematis** pada *hyperparameter* mampu memberikan dampak nyata pada performa model.



Gambar 5. Perbandingan Akurasi Sebelum dan Sesudah Tuning
Tahap akhir menganalisis **feature importance** dari model terbaik.



Gambar 6. Pentingnya Fitur pada Decision Tree Setelah Tuning

Gambar 6. Pentingnya Fitur pada Decision Tree Setelah Tuning. Grafik batang horizontal memperlihatkan bahwa fitur tertentu, misalnya **petal length** dan **petal width**, memiliki kontribusi lebih besar dalam pemisahan kelas. Secara keseluruhan, program ini menghubungkan konsep **optimasi model**, **evaluasi komparatif**, dan **interpretabilitas** menjadi satu alur praktikum yang ilmiah, menarik, dan mudah diikuti.

Soal

1. Jelaskan bagaimana model pohon keputusan bekerja dalam menyusun aturan untuk pemisahan data. Berikan contoh sederhana yang menggambarkan proses ini.
2. Apa yang dimaksud dengan entropy dalam pohon keputusan, dan bagaimana cara mengukur informasi yang diperoleh dari setiap pemisahan data? Jelaskan dengan contoh yang relevan.
3. Jelaskan hubungan antara entropy dan information gain dalam konteks pembangunan pohon keputusan. Bagaimana kedua konsep ini berperan dalam memilih fitur untuk pembentukan cabang pohon?
4. Apa yang dimaksud dengan overfitting dan underfitting pada model pohon keputusan? Jelaskan bagaimana kedalaman pohon dapat memengaruhi kedua masalah ini dan cara menghindarinya.
5. Apa keunggulan utama dari pohon keputusan dalam hal interpretabilitas dibandingkan dengan model pembelajaran mesin lainnya? Berikan contoh penerapan pohon keputusan dalam bidang medis atau kebijakan publik.
6. Bagaimana visualisasi pohon keputusan membantu dalam memahami alur keputusan model? Jelaskan dengan cara penggunaan fungsi `plot_tree` pada pustaka `scikit-learn`.
7. Pada penerapan model pohon keputusan pada dataset Iris, fitur mana yang lebih dominan dalam klasifikasi spesies? Jelaskan mengapa fitur tersebut memiliki pengaruh besar pada hasil klasifikasi.

8. Dalam konteks pemrograman, apa yang dimaksud dengan hyperparameter tuning? Jelaskan bagaimana tuning hyperparameter dapat meningkatkan performa model pohon keputusan menggunakan teknik seperti GridSearchCV.
9. Bagaimana cara evaluasi model pohon keputusan dilakukan setelah proses pelatihan? Sebutkan dan jelaskan metrik-metrik yang umum digunakan untuk mengukur kinerja model pohon keputusan.
10. Jelaskan proses dan langkah-langkah yang dilakukan dalam pembentukan model Decision Tree pada dataset "Glass Identification" untuk kasus forensik digital. Apa yang dapat dipelajari dari fitur penting dalam klasifikasi kaca?

Tugas Kelompok

Tugas kelompok ini mengharuskan mempelajari dataset berbeda yang tersedia di UCI Machine Learning Repository. Setiap kelompok akan diberikan satu dataset untuk dianalisis dan diminta untuk melakukan langkah-langkah berikut

1. Dataset 1 Wine Quality Dataset

Tugas: Lakukan pembersihan data (jika diperlukan), eksplorasi data, dan visualisasi distribusi fitur. Bangun model pohon keputusan (Decision Tree) untuk memprediksi kualitas anggur berdasarkan berbagai fitur kimiawi. Evaluasi hasil model menggunakan metrik accuracy, precision, recall, dan F1-score, serta diskusikan kekuatan dan kelemahan model yang digunakan.

Link: [Wine Quality Dataset](#)

2. Dataset 2 Iris Dataset

Tugas: Lakukan eksplorasi data untuk menganalisis hubungan antar fitur morfologi bunga iris. Bangun model pohon keputusan (Decision Tree) untuk mengklasifikasikan spesies iris berdasarkan panjang dan lebar kelopak dan mahkota. Visualisasikan struktur pohon keputusan dan analisis bagaimana pemisahan data dilakukan berdasarkan fitur-fitur tersebut. Evaluasi kinerja model menggunakan metrik **accuracy**, **precision**, **recall**, dan **F1-score**, serta diskusikan kekuatan dan kelemahan model yang digunakan

Link: [Iris Dataset](#)

3. Dataset 3 Glass Identification Dataset

Tugas: Lakukan pembersihan data, eksplorasi data, dan visualisasi distribusi fitur kimia yang mempengaruhi jenis kaca. Bangun model pohon keputusan (Decision Tree) untuk mengklasifikasikan jenis kaca berdasarkan komposisi kimianya. Evaluasi hasil model menggunakan metrik accuracy, precision, recall, dan F1-score, serta diskusikan kekuatan dan kelemahan model yang digunakan. Lakukan analisis lebih lanjut untuk menentukan fitur mana yang paling berpengaruh terhadap klasifikasi jenis kaca menggunakan feature importance

Link: [Glass Identification Dataset](#)

Refleksi Pembelajaran

Pembelajaran tentang pohon keputusan memberikan pemahaman mendalam mengenai cara model ini bekerja dalam memisahkan data berdasarkan fitur-fitur yang relevan. Dengan menggunakan konsep seperti entropy dan information gain, kita dapat mengoptimalkan pohon keputusan untuk mencapai hasil yang

akurat dan mudah dipahami. Visualisasi pohon keputusan juga membantu dalam interpretabilitas model, memungkinkan kita untuk melihat dan memahami keputusan yang diambil oleh model. Dalam praktikum, penerapan pohon keputusan pada berbagai dataset seperti Iris dan Glass memberikan pengalaman nyata dalam membangun dan mengevaluasi model, sekaligus mengajarkan pentingnya pemilihan fitur yang tepat untuk meningkatkan kinerja model.

Ensemble Learning dengan Random Forest

CPL

Mahasiswa memahami konsep penggabungan model.

CPMK

Menerapkan Random Forest untuk meningkatkan akurasi.

SubCPMK

Menggunakan RandomForestClassifier.

A. Ensemble Learning dan Random Forest

Perkembangan *machine learning* tidak hanya berfokus pada pencarian satu model terbaik, tetapi juga pada cara menggabungkan banyak model untuk memperoleh prediksi yang lebih kuat dan stabil. Pendekatan ini dikenal sebagai *ensemble learning*. Ide utamanya adalah bahwa sekelompok model sederhana yang bekerja bersama sering kali memberikan hasil lebih baik dibanding satu model yang kompleks. Salah satu bentuk *ensemble learning* yang paling banyak digunakan adalah **Random Forest**, yaitu kumpulan pohon keputusan yang dibangun di atas variasi data dan variasi fitur. **Penggabungan model** menjadi gagasan inti, karena setiap pohon menyumbang sudut pandang berbeda terhadap pola dalam data, lalu seluruh suara digabungkan menjadi keputusan akhir yang lebih seimbang.

Random Forest memanfaatkan sifat pohon keputusan yang fleksibel dan mudah menyesuaikan diri dengan data, tetapi

mengurangi kecenderungan *overfitting* melalui penggabungan banyak pohon. Setiap pohon pada hutan acak dilatih dengan sampel data yang diambil secara acak dengan pengembalian, sehingga struktur pohon yang dihasilkan bervariasi satu sama lain. Prediksi akhir ditentukan melalui mekanisme *voting* untuk kasus klasifikasi atau rata-rata untuk kasus regresi. **Stabilitas prediksi** yang dihasilkan oleh Random Forest menjadikannya pilihan menarik pada banyak aplikasi nyata mulai dari kesehatan, keuangan, hingga pengenalan pola visual.

B. Prinsip Bagging dan Pembentukan Model Forest

Konsep dasar yang menggerakkan Random Forest adalah teknik *bagging* atau *bootstrap aggregating*. Teknik ini bekerja dengan membuat beberapa *subset* data hasil pengambilan acak dari dataset asli. Setiap subset digunakan untuk melatih satu pohon keputusan, sehingga tiap pohon melihat kombinasi data yang sedikit berbeda. Proses ini mendorong terciptanya **keragaman model**, yang sangat penting untuk menurunkan varians keseluruhan sistem. Jika semua pohon dilatih dengan data yang sama persis, keputusan kolektif tidak akan berbeda jauh dari satu pohon tunggal.

Selain variasi data, Random Forest juga mengatur variasi fitur. Pada setiap titik percabangan di dalam pohon, algoritma tidak menilai semua fitur sekaligus, melainkan hanya beberapa fitur yang dipilih secara acak. Cara ini memaksa setiap pohon mengeksplorasi kombinasi fitur yang berbeda sehingga tidak terjadi dominasi fitur tertentu secara berlebihan. **Eksplorasi fitur** semacam ini menghasilkan struktur hutan yang lebih kaya dan memperluas pola

yang dapat dipelajari. Pada akhirnya, *bagging* mengurangi risiko bahwa kesalahan satu pohon akan merusak keseluruhan hasil karena suara dari pohon lain berperan sebagai penyeimbang.

C. Mekanisme *Voting Ensemble* dan Keunggulan Prediktif

Prediksi pada Random Forest untuk tugas klasifikasi dihasilkan melalui mekanisme *voting ensemble*. Setiap pohon di dalam hutan memberikan satu suara terhadap kelas yang dianggap paling sesuai dengan pola data masukan. Suara yang terkumpul kemudian dihitung, dan kelas dengan jumlah suara terbanyak dipilih sebagai keluaran akhir. Proses ini meniru cara pengambilan keputusan kolektif di banyak situasi kehidupan nyata, di mana keputusan bersama sering lebih kokoh dibanding pendapat tunggal. **Suara mayoritas** berperan sebagai peredam terhadap keputusan yang terlalu ekstrem dari satu pohon.

Keunggulan pendekatan *voting ensemble* terlihat pada stabilitas hasil prediksi. Ketika data baru mengandung sedikit gangguan atau *noise*, satu pohon mungkin terpengaruh dan membuat keputusan yang kurang tepat. Akan tetapi, puluhan atau ratusan pohon lain yang dilatih pada variasi data berbeda cenderung tetap konsisten sehingga keputusan gabungan tetap akurat. Hal ini menjadikan Random Forest sangat andal pada situasi di mana data bersifat heterogen dan rentan mengandung pengukuran yang tidak sempurna. **Ketahanan model** terhadap gangguan data merupakan alasan utama Random Forest sering direkomendasikan sebagai pilihan awal untuk tugas klasifikasi maupun regresi.

D. *Feature Importance* dan Interpretasi Model

Meskipun Random Forest terdiri dari banyak pohon, model ini masih menyediakan cara untuk memahami fitur mana yang berperan penting dalam proses pengambilan keputusan. Pustaka *scikit learn* menyediakan fungsi untuk menghitung *feature importance*, yaitu ukuran seberapa besar kontribusi masing masing fitur dalam mengurangi ketidakteraturan atau *impurity* sepanjang struktur hutan. Fitur yang sering digunakan pada percabangan penting dengan penurunan *impurity* besar akan memiliki nilai kepentingan lebih tinggi. **Peta pentingnya fitur** ini memberikan panduan bagi pembaca untuk menilai variabel mana yang paling berpengaruh terhadap hasil prediksi.

Informasi *feature importance* membantu dua hal utama. Pertama, memberikan pemahaman substantif terhadap domain masalah, misalnya fitur panjang petal yang sangat dominan dalam membedakan spesies bunga Iris. Kedua, membantu upaya penyederhanaan model. Fitur dengan kepentingan sangat rendah dapat dipertimbangkan untuk dihapus guna mengurangi kompleksitas data masukan tanpa menurunkan akurasi secara signifikan. **Penyederhanaan fitur** seperti ini mengarah pada sistem yang lebih efisien dan mudah dipelihara, terutama ketika model digunakan pada lingkungan produksi.

Random Forest memiliki peran penting sebagai jembatan antara model pohon keputusan tunggal yang mudah diinterpretasikan dan keluarga model *ensemble* yang unggul secara prediktif. Untuk kegiatan praktikum, algoritma ini memperlihatkan bagaimana konsep *bagging*, *voting*, dan *feature importance*

berpadu dalam satu kerangka kerja yang terstruktur. Pembaca dapat mengamati perbandingan akurasi antara pohon tunggal dan Random Forest, kemudian menganalisis mengapa penggabungan banyak pohon menghasilkan hasil yang lebih baik. **Perbandingan performa** ini menjadi pengalaman belajar yang kuat karena menghubungkan teori dengan pengamatan empiris pada dataset nyata seperti Iris atau Banknote Authentication dari *UCI Repository*.

Pada tataran aplikasi, Random Forest digunakan di banyak bidang seperti deteksi penipuan, analisis risiko kredit, klasifikasi citra medis, dan pemodelan perilaku pelanggan. Sifatnya yang relatif mudah disesuaikan dan tidak membutuhkan penalaan hiperparameter yang sangat rumit menjadikannya pilihan praktis untuk prototipe awal maupun sistem yang sudah berjalan. Walaupun model terdiri dari banyak pohon, Random Forest tetap lebih mudah dijelaskan dibanding beberapa algoritma lanjutan seperti *gradient boosting* atau *deep learning*. **Keseimbangan antara akurasi dan interpretasi** menjadikan Random Forest salah satu alat utama yang patut dikuasai dalam jalur belajar *data science* dan analitik prediktif berbasis Python serta *UCI Repository* sebagai sumber data terbuka.

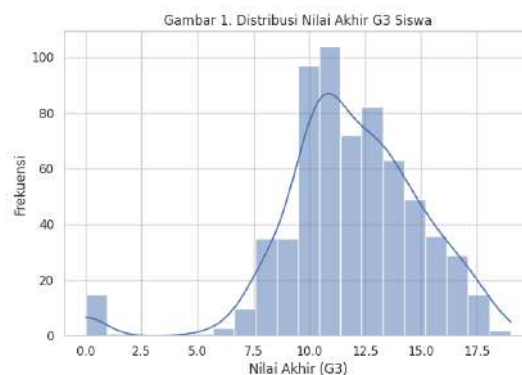
Kode Program :Prediksi Status Kelulusan Siswa Random Forest.



Kode program pada barcode ini menyusun studi kasus **prediksi status kelulusan siswa** menggunakan *Random Forest* pada

dataset Student Performance dari *UCI Machine Learning Repository*. Tujuan utama pemodelan ialah mengklasifikasikan apakah seorang siswa **berisiko tidak lulus** atau **berpotensi lulus**, dengan memanfaatkan informasi **absensi**, **nilai**, dan **perilaku belajar** yang tercatat pada data. Alur kode disusun mengikuti tahapan *machine learning* yang sistematis sehingga mudah diadaptasi sebagai materi praktikum pada buku *Praktikum Data Science dan Machine Learning Berbasis Python dan UCI Repository*.

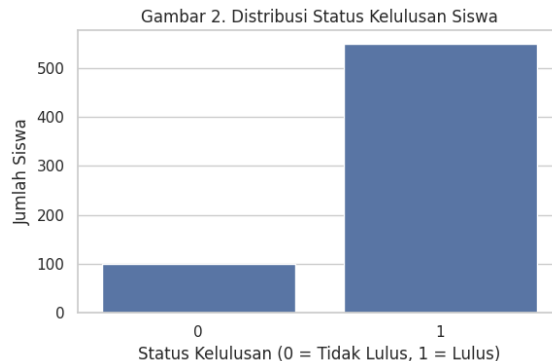
Tahap awal menggunakan `fetch_ucirepo(id=320)` untuk mengambil fitur dan target langsung dari repositori UCI. Pendekatan ini menjaga **integritas sumber data** karena struktur variabel, skala nilai, serta dokumentasi atribut diacu dari metadata resmi. Kolom **G3** diidentifikasi sebagai **nilai akhir** siswa. Program kemudian membentuk label biner **status_lulus** berdasarkan ambang nilai 10, dengan kode 1 untuk **lulus** dan 0 untuk **berisiko tidak lulus**. Pembentukan label ini penting karena mengubah masalah regresi nilai menjadi masalah klasifikasi risiko yang lebih relevan bagi analisis pendidikan.



Gambar 1. Distribusi Nilai Akhir G3 Siswa

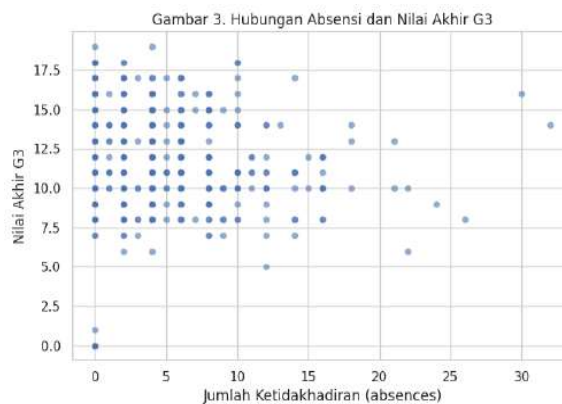
Gambar 1. Distribusi Nilai Akhir G3 Siswa. Histogram ini menampilkan sebaran nilai dari 0 hingga 20, sehingga pembaca

dapat mengenali apakah nilai cenderung menumpuk pada rentang tertentu.



Gambar 2. Distribusi Status Kelulusan Siswa

Gambar 2. Distribusi Status Kelulusan Siswa menampilkan grafik batang yang memperlihatkan perbandingan jumlah siswa lulus dan tidak lulus. Jika kelas tidak lulus relatif lebih sedikit, hal ini memberi *insight* mengenai potensi ketidakseimbangan kelas yang perlu diperhatikan ketika menafsirkan performa model.



Gambar 3. Hubungan Absensi dan Nilai Akhir G3

Gambar 3. Hubungan Absensi dan Nilai Akhir G3 memperlihatkan bahwa siswa dengan **absensi** tinggi cenderung memiliki **nilai akhir G3** lebih rendah, sehingga variabel absensi layak dipertimbangkan sebagai indikator **risiko tidak lulus**. Pada tahap

pra pemrosesan, program memisahkan fitur numerik seperti **absences**, **studytime**, **failures**, serta fitur kategorik seperti **school**, **sex**, **address**, **famsize**, dan variabel perilaku lain. Fitur kategorik dikodekan menggunakan *One Hot Encoder* di dalam *ColumnTransformer*, sedangkan fitur numerik diteruskan apa adanya. Seluruh langkah pra pemrosesan digabung dengan algoritma *RandomForestClassifier* ke dalam sebuah *pipeline* sehingga proses pelatihan dan prediksi menjadi lebih terstruktur dan rapi. Data dibagi menjadi **data latih** dan **data uji** dengan proporsi 80 banding 20 menggunakan `train_test_split` dengan stratify pada label untuk mempertahankan proporsi kelas.

Model *Random Forest* disusun dengan 200 pohon, pengaturan `class_weight="balanced"` untuk membantu menangani perbedaan jumlah kelas, serta `n_jobs=-1` agar memanfaatkan banyak inti prosesor di *Google Colab*. Setelah pelatihan, model dievaluasi menggunakan **akurasi**, **ROC AUC**, dan *classification report*. Contoh potongan keluaran evaluasi sebagai berikut.

```

=== EVALUASI MODEL RANDOM FOREST ===
Akurasi pada data uji : 0.792
ROC AUC                : 0.660

Laporan klasifikasi:

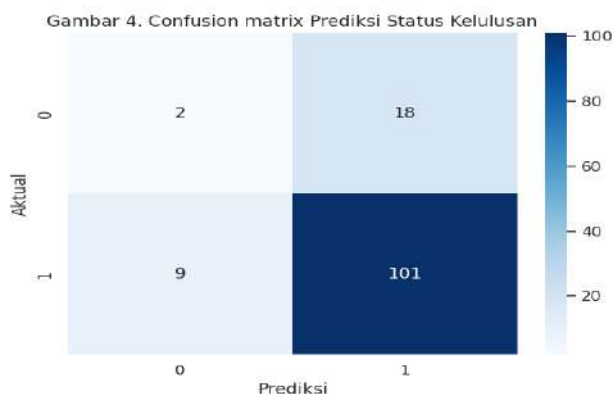
```

	precision	recall	f1-score	support
Berisiko Tidak Lulus	0.182	0.100	0.129	20
Lulus	0.849	0.918	0.882	110
accuracy			0.792	130
macro avg	0.515	0.509	0.506	130
weighted avg	0.746	0.792	0.766	130

Gambar 4. Confusion Matrix for Graduation Prediction

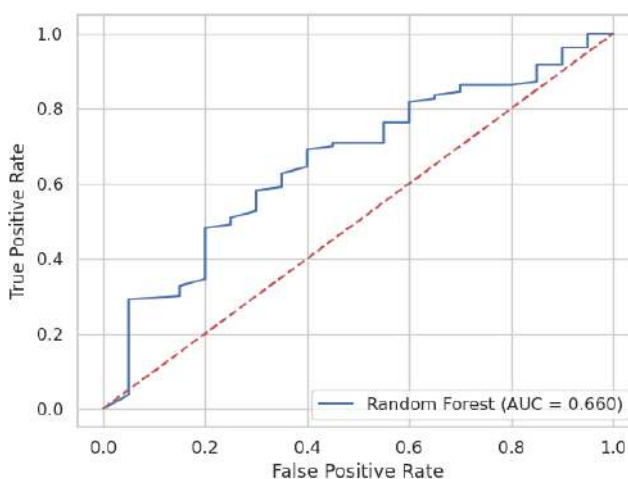
Gambar 4. Confusion Matrix for Graduation Prediction yang memakai label berbahasa Inggris **Predicted Label** dan **True Label**. Matriks ini menampilkan jumlah prediksi benar dan salah untuk kategori **High Risk (Fail)** dan **Graduate**, sehingga memudahkan

diskusi tentang konsekuensi kesalahan klasifikasi, misalnya pentingnya mengurangi salah klasifikasi siswa berisiko sebagai lulus.



Gambar 5. Confusion Matrix for Graduation Prediction

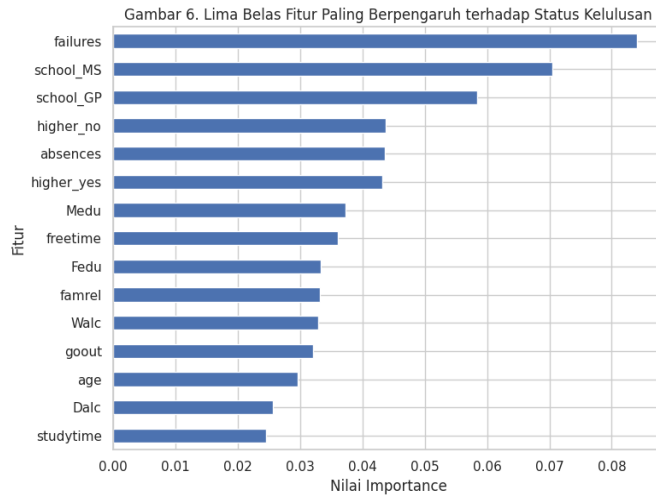
Kualitas pemisahan kelas juga diperlihatkan melalui Gambar 5. Confusion Matrix for Graduation Prediction, yang menggambarkan hubungan antara *true positive rate* dan *false positive rate* dengan nilai *ROC AUC* sebagai ukuran kemampuan model membedakan dua kelas.



Gambar 6. Kurva ROC Prediksi Kelulusan Siswa

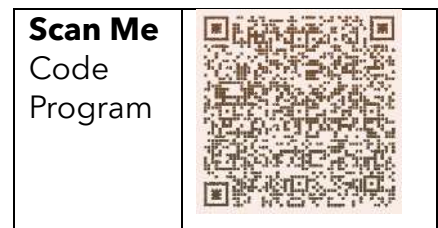
Terakhir, **Gambar 7. 15 Fitur Paling Berpengaruh terhadap Status Kelulusan** menampilkan *feature importance* dari model *Random Forest*. Grafik ini menyoroti fitur yang paling besar

kontribusinya, misalnya kombinasi **nilai**, **absensi**, dan **jam belajar**, sehingga memberikan **brainstorming** kuat bagi pendidik untuk merancang intervensi berbasis data demi menurunkan **risiko tidak lulus** dan meningkatkan **kelulusan** secara terarah.



Gambar 7. 15 Fitur Paling Berpengaruh terhadap Status Kelulusan

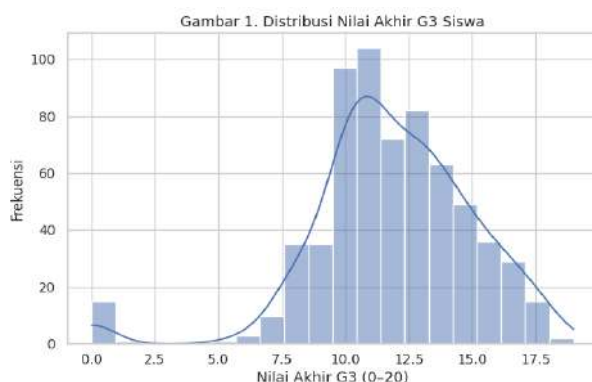
Studi Kasus: Prediksi Nilai Akhir Siswa Gradient Boosting.



Kode program pada barcode ini mengilustrasikan bagaimana **optimasi model** dapat meningkatkan kualitas **prediksi nilai akhir siswa** menggunakan *Gradient Boosting Regressor* pada *dataset Student Performance* dari *UCI Machine Learning Repository*. Seluruh alur mengikuti tahapan *machine learning* yang sistematis, sejalan dengan kerangka buku *Praktikum Data Science dan Machine Learning Berbasis Python dan UCI Repository*. Fokus utama adalah

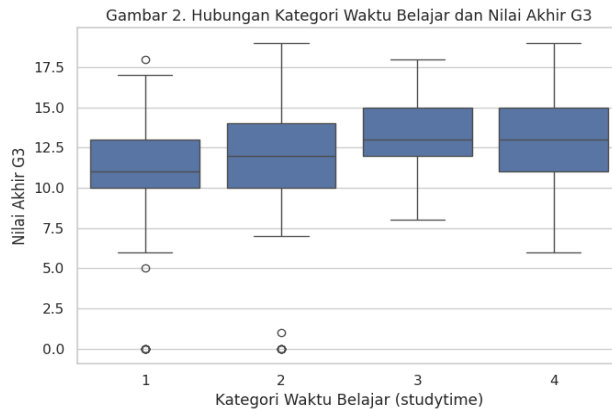
membandingkan model dasar **regresi linier** dengan model **gradient boosting** yang dioptimasi melalui *hyperparameter tuning* berbasis *RandomizedSearchCV*.

Tahap awal program memanggil *dataset* melalui `fetch_ucirepo(id=320)` sehingga identitas data, struktur variabel, dan dokumentasi resmi tetap terjaga. Fitur disimpan sebagai `X_asli`, sedangkan nilai akhir siswa G3 diambil dari bagian *target* dan disimpan sebagai `y`. Data kemudian digabung menjadi kerangka `df` untuk eksplorasi. Statistik deskriptif nilai G3 menampilkan rata-rata, simpangan baku, nilai minimum, dan maksimum sehingga pembaca dapat melihat sebaran prestasi akademik siswa sebagai dasar pemodelan.



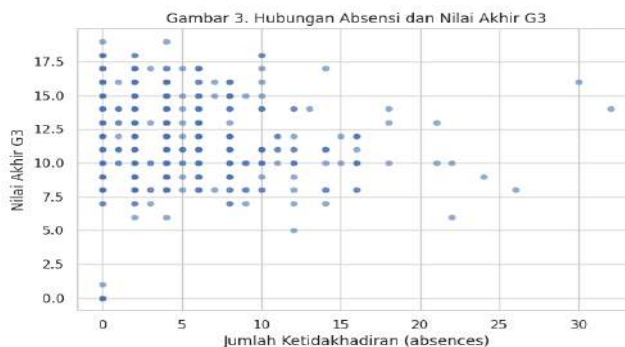
Gambar 1. Distribusi Nilai Akhir G3 Siswa

Gambar 1. Distribusi Nilai Akhir G3 Siswa menampilkan histogram nilai dari 0 sampai 20, lengkap dengan *kernel density estimate* untuk menggambarkan pola sebaran. Grafik ini membantu pembaca melihat apakah nilai cenderung menumpuk di tengah atau menyebar ke ekstrem bawah dan atas.



Gambar 2. Hubungan Kategori Waktu Belajar dan Nilai Akhir G3

Gambar 2. Hubungan Kategori Waktu Belajar dan Nilai Akhir G3 menyajikan *boxplot* nilai berdasarkan kategori studytime. Ini memberi *insight* bahwa peningkatan waktu belajar sering berkorelasi dengan median nilai yang lebih tinggi.

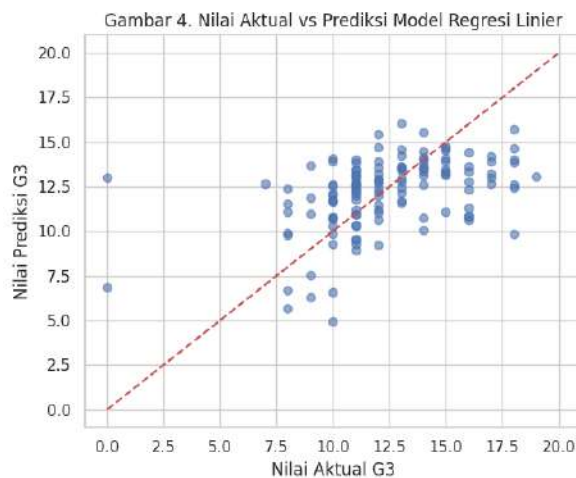


Gambar 3. Hubungan Absensi dan Nilai Akhir G3

Gambar 3. Hubungan Absensi dan Nilai Akhir G3 menunjukkan *scatter plot* yang memperlihatkan tren menurun ketika absensi meningkat, sehingga variabel ketidakhadiran layak dianggap sebagai indikator risiko akademik.

Tahap berikutnya adalah menyiapkan fitur dan memisahkan data menjadi **data latih** dan **data uji**. Fitur numerik seperti absences, failures, dan age dibedakan dari fitur kategorik seperti school, sex, address, dan famsize. Fitur kategorik dikodekan menggunakan *One*

Hot Encoder dalam *ColumnTransformer*, sedangkan fitur numerik diteruskan tanpa transformasi. Pemisahan data menggunakan *train_test_split* dengan proporsi 80 persen data latih dan 20 persen data uji menjaga struktur evaluasi yang adil. Model dasar berupa **regresi linier** dibangun dalam sebuah *pipeline* yang menggabungkan pra pemrosesan dan algoritma. Evaluasi menggunakan metrik **MSE**, **RMSE**, **MAE**, dan **R²**.

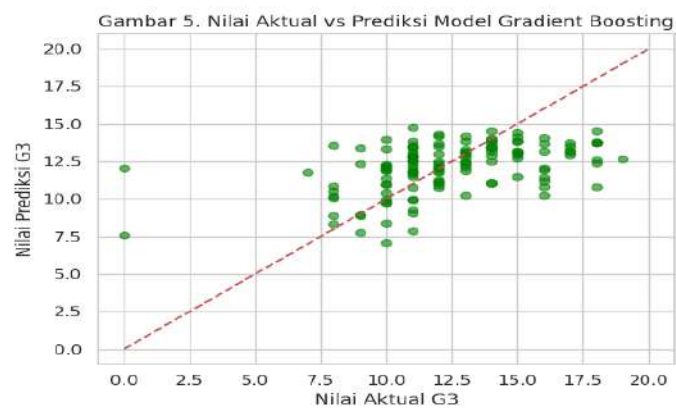


Gambar 4. Nilai Aktual vs Prediksi Model Regresi Linier

Gambar 4. Nilai Aktual vs Prediksi Model Regresi Linier menampilkan titik sebar antara nilai aktual dan nilai prediksi, dengan garis diagonal sebagai referensi. Pola penyebaran titik memberikan gambaran bahwa model linier masih memiliki galat yang cukup besar pada beberapa rentang nilai. Potongan keluaran program yang mungkin muncul adalah sebagai berikut.

```
=== EVALUASI MODEL DASAR (REGRESI LINIER) ===  
MSE   : 8.190  
RMSE  : 2.862  
MAE   : 2.156  
R2   : 0.160
```

Untuk meningkatkan akurasi, program membangun model *Gradient Boosting Regressor* dan melakukan *hyperparameter tuning* dengan *RandomizedSearchCV*. Ruang parameter mencakup **jumlah pohon, learning rate, max_depth**, serta parameter sampel. Proses ini mencari kombinasi terbaik berdasarkan skor negatif MSE pada validasi silang. Setelah mendapatkan model terbaik, prediksi pada data uji dievaluasi kembali.



Gambar 5. Nilai Aktual vs Prediksi Model Gradient Boosting

Gambar 5. Nilai Aktual vs Prediksi Model Gradient Boosting memperlihatkan sebaran titik yang lebih dekat ke garis diagonal, menandakan galat lebih kecil. Ringkasan perbandingan kinerja ditampilkan pada tabel

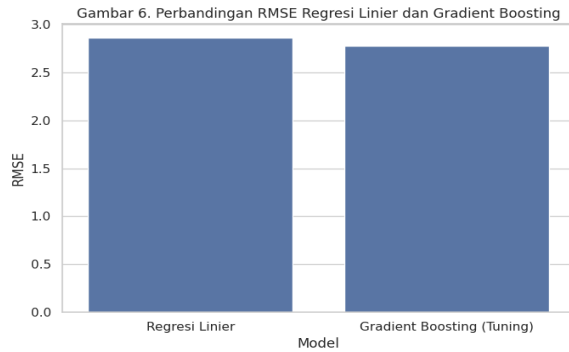
===

TABEL PERBANDINGAN KINERJA MODEL

===

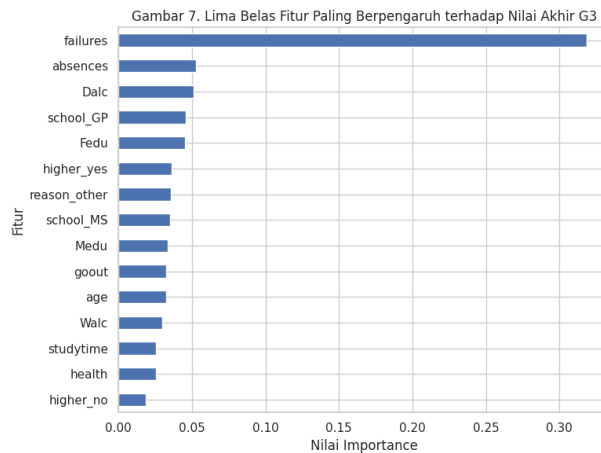
	Model	MSE	RMSE	MAE	R2
0	Regresi Linier	8.189784	2.861780	2.156382	0.160170
1	Gradient Boosting (Tuning)	7.710270	2.776737	2.065588	0.209342

Gambar 6. Perbandingan RMSE Regresi Linier dan Gradient Boosting, yang menonjolkan penurunan RMSE oleh model *gradient boosting*.



Gambar 6. Perbandingan RMSE Regresi Linier dan Gradient Boosting

Tahap akhir menampilkan **Gambar 7. Lima Belas Fitur Paling Berpengaruh terhadap Nilai Akhir G3**, berupa diagram batang *feature importance* yang menyortir fitur seperti **nilai sebelumnya**, **absensi**, dan **waktu belajar** sebagai faktor dominan. Informasi ini membuka ruang **brainstorming** bagi pendidik mengenai intervensi apa yang dapat difokuskan untuk mendorong peningkatan nilai siswa secara strategis dan berbasis data.



Gambar 7. 15 Fitur Paling Berpengaruh terhadap Nilai Akhir G3

Soal

1. Jelaskan konsep dasar dari Ensemble Learning dan bagaimana metode ini meningkatkan performa model dibandingkan dengan penggunaan model tunggal. Berikan contoh penerapan ensemble learning selain Random Forest.
2. Apa yang dimaksud dengan bagging dalam konteks Random Forest? Jelaskan bagaimana teknik ini dapat mengurangi overfitting dan meningkatkan stabilitas model.
3. Random Forest menggunakan mekanisme voting ensemble untuk tugas klasifikasi. Jelaskan bagaimana voting ini bekerja dan mengapa pendekatan ini lebih stabil dibandingkan dengan menggunakan satu model pohon keputusan tunggal.
4. Dalam Random Forest, setiap pohon dilatih dengan subset data yang diambil secara acak. Bagaimana cara ini membantu menciptakan variasi dalam model? Apa dampaknya terhadap akurasi dan kestabilan model?
5. Jelaskan apa yang dimaksud dengan Feature Importance dalam Random Forest. Mengapa informasi ini penting untuk pemahaman model dan bagaimana cara menghitungnya?
6. Sebagai contoh, pada studi kasus prediksi kelulusan siswa menggunakan Random Forest, fitur seperti absensi, nilai, dan waktu belajar digunakan untuk membuat prediksi. Apa yang bisa diinterpretasikan dari hubungan antara absensi dan nilai siswa dalam konteks ini?
7. Bandingkan kelebihan dan kekurangan Random Forest dengan algoritma lain seperti Gradient Boosting dalam hal

kemampuan untuk mengatasi data yang noisy atau tidak lengkap.

8. Mengapa Random Forest lebih mudah dipahami dan lebih mudah diimplementasikan dibandingkan dengan model-model kompleks seperti deep learning? Jelaskan dari segi kompleksitas model dan interpretabilitas.
9. Dalam implementasi Random Forest pada studi kasus prediksi kelulusan siswa, bagaimana proses pra-pemrosesan data seperti encoding fitur kategorikal dan pembagian data latih dan uji dapat mempengaruhi hasil model?
10. Evaluasi kinerja model Random Forest dapat dilakukan dengan menggunakan metrik ROC AUC. Apa yang dimaksud dengan ROC AUC dan bagaimana cara menginterpretasi nilai ini dalam konteks prediksi status kelulusan siswa?

Tugas Kelompok

Tugas kelompok ini mengharuskan mempelajari dataset berbeda yang tersedia di UCI Machine Learning Repository. Setiap kelompok akan diberikan satu dataset untuk dianalisis dan diminta untuk melakukan langkah-langkah berikut:

1. Dataset 1 Banknote Authentication Dataset

Tugas: Lakukan pembersihan data (jika diperlukan), eksplorasi data, dan visualisasi distribusi fitur yang mempengaruhi klasifikasi apakah sebuah banknote asli atau palsu. Bangun model Random Forest untuk mengklasifikasikan apakah banknote tersebut asli atau palsu berdasarkan fitur fisik yang ada. Evaluasi hasil model menggunakan metrik accuracy, precision,

recall, dan F1-score, serta diskusikan kekuatan dan kelemahan model yang digunakan.

Link: [Banknote Authentication Dataset](#)

2. **Dataset 2 Pima Indians Diabetes Dataset**

Tugas: Lakukan eksplorasi data untuk menganalisis hubungan antar fitur yang mempengaruhi prediksi diabetes. Bangun model Random Forest untuk memprediksi apakah seseorang mengidap diabetes berdasarkan faktor kesehatan yang ada. Evaluasi hasil model menggunakan metrik accuracy, precision, recall, dan F1-score, serta diskusikan kekuatan dan kelemahan model yang digunakan.

Link: [Pima Indians Diabetes Dataset](#)

3. **Dataset 3 Forest Fires Dataset**

Tugas: Lakukan pembersihan data, eksplorasi data, dan visualisasi distribusi fitur yang mempengaruhi tingkat keparahan kebakaran hutan. Bangun model Random Forest untuk memprediksi tingkat keparahan kebakaran hutan berdasarkan faktor lingkungan. Evaluasi hasil model menggunakan metrik RMSE, classification report, dan feature importance untuk menentukan fitur mana yang paling berpengaruh terhadap klasifikasi kebakaran hutan. Diskusikan kekuatan dan kelemahan model yang digunakan.

Link: [Forest Fires Dataset](#)

Refleksi Pembelajaran

Selama proses pembelajaran ini, saya memahami pentingnya penerapan Ensemble Learning dengan Random Forest dalam

analisis data. Dengan menggabungkan beberapa model pohon keputusan, Random Forest terbukti meningkatkan akurasi dan stabilitas model, terutama pada dataset dengan banyak variabel dan data yang tidak sempurna. Selain itu, konsep bagging dan voting membantu meminimalkan risiko overfitting. Saya juga belajar bagaimana mengolah data, mengevaluasi model dengan metrik seperti accuracy, precision, recall, dan F1-score, serta memahami feature importance untuk meningkatkan interpretabilitas model. Pengalaman ini memperkuat pemahaman saya dalam data science dan aplikasi model prediktif berbasis machine learning.

Support Vector Machine (SVM)

CPL:

Mahasiswa mampu memahami margin classifier.

CPMK:

Menggunakan SVM untuk pemisahan kelas optimal.

SubCPMK:

Implementasi SVC (kernel='linear').

A. Hakikat Support Vector Machine

Support Vector Machine atau *Support Vector Machine* (SVM) merupakan pendekatan pembelajaran mesin yang berfokus pada pencarian batas pemisah paling optimal antar kelompok data. Ide sentralnya terletak pada pemisahan kelas yang tidak hanya sekadar benar, tetapi juga stabil dan tahan terhadap gangguan data. Prinsip ini diwujudkan melalui konsep margin, yaitu jarak antara batas pemisah dengan titik data terdekat dari masing-masing kelas. Semakin lebar margin, semakin kuat kemampuan model dalam melakukan generalisasi. SVM dirancang untuk bekerja efektif pada data dengan jumlah fitur besar, sehingga sering digunakan pada persoalan medis, teks, dan citra. Pendekatan ini menempatkan kualitas batas keputusan sebagai prioritas utama, bukan sekadar jumlah data yang dipisahkan secara benar. Dengan karakter tersebut, SVM menjadi model yang dikenal andal, terkontrol, dan berbasis optimasi matematis yang jelas.

B. Konsep Hyperplane dan Margin

Hyperplane merupakan representasi matematis dari batas pemisah kelas. Pada data dua dimensi, hyperplane terlihat sebagai garis lurus, sedangkan pada ruang berdimensi lebih tinggi ia berbentuk bidang abstrak yang tetap mengikuti aturan linear. Keunikan SVM terletak pada pemilihan hyperplane yang memaksimalkan margin optimal, bukan hyperplane sembarang. Titik-titik data yang paling dekat dengan hyperplane disebut *support vectors*. Titik inilah yang menentukan posisi akhir batas pemisah. Data lain yang berada lebih jauh tidak memengaruhi keputusan secara langsung. Pendekatan ini membuat SVM efisien dan fokus pada informasi paling kritis. Optimasi margin juga memberi perlindungan terhadap kesalahan kecil pada data, sehingga model tidak mudah berubah hanya karena adanya variasi minor.

C. Kernel Trick sebagai Strategi Transformasi

Ketika data tidak dapat dipisahkan secara linear, SVM menyediakan solusi melalui *kernel trick*. Strategi ini memungkinkan pemetaan data ke ruang fitur berdimensi lebih tinggi agar pemisahan menjadi mungkin, tanpa harus benar-benar membangun fitur baru secara eksplisit. Mekanisme ini menjaga efisiensi komputasi sekaligus memperluas kemampuan model. *Kernel* linear sering digunakan sebagai titik awal karena sederhana, mudah diinterpretasikan, dan stabil. *Kernel trick* memperlihatkan kekuatan SVM sebagai metode yang fleksibel tanpa kehilangan fondasi matematisnya. Transformasi ruang fitur menjadikan SVM relevan

pada berbagai jenis data kompleks, termasuk data klinis dan biologis yang memiliki pola saling tumpang tindih.

D. Optimasi dan Regularisasi

Proses pelatihan SVM berlandaskan optimasi matematis yang bertujuan menyeimbangkan dua kepentingan utama, yaitu memperlebar margin dan meminimalkan kesalahan klasifikasi. Pada data nyata, pemisahan sempurna jarang terjadi. Oleh karena itu, SVM menggunakan konsep *soft margin* yang mengizinkan sebagian kecil kesalahan dengan batas terkontrol. Parameter regularisasi berperan penting dalam menentukan seberapa besar toleransi kesalahan tersebut. Nilai regularisasi yang tepat menghasilkan model yang stabil, tidak overfitting, dan tetap akurat. Optimasi ini membuat SVM unggul pada data berdimensi tinggi karena fokus pada struktur global, bukan detail lokal yang bersifat kebetulan.

E. SVM pada Data Kesehatan

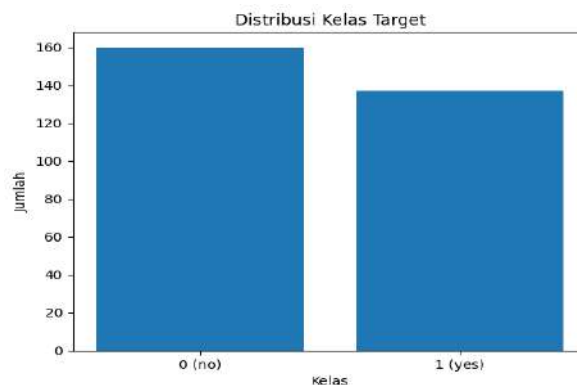
SVM sangat sesuai untuk analisis data kesehatan seperti diagnosis penyakit jantung dari dataset Heart Disease UCI. Data jenis ini memiliki banyak fitur klinis dengan hubungan kompleks. SVM mampu menangani kondisi tersebut karena mekanisme margin dan pemilihan *support vectors* membantu menekan pengaruh data ekstrem. Implementasi menggunakan SVC dengan *kernel* linear menjadi pilihan rasional sebagai dasar analisis karena mudah dipahami dan transparan. Hasil klasifikasi tidak hanya dinilai dari akurasi, tetapi juga dari kemampuan menjelaskan pola pemisahan kelas. Keandalan, ketahanan, dan interpretabilitas menjadikan SVM

relevan sebagai alat bantu pengambilan keputusan berbasis data kesehatan. Pendekatan ini menegaskan peran SVM sebagai metode yang ilmiah, aplikatif, dan konsisten pada permasalahan nyata.

Kode Program : SVM untuk klasifikasi risiko penyakit jantung

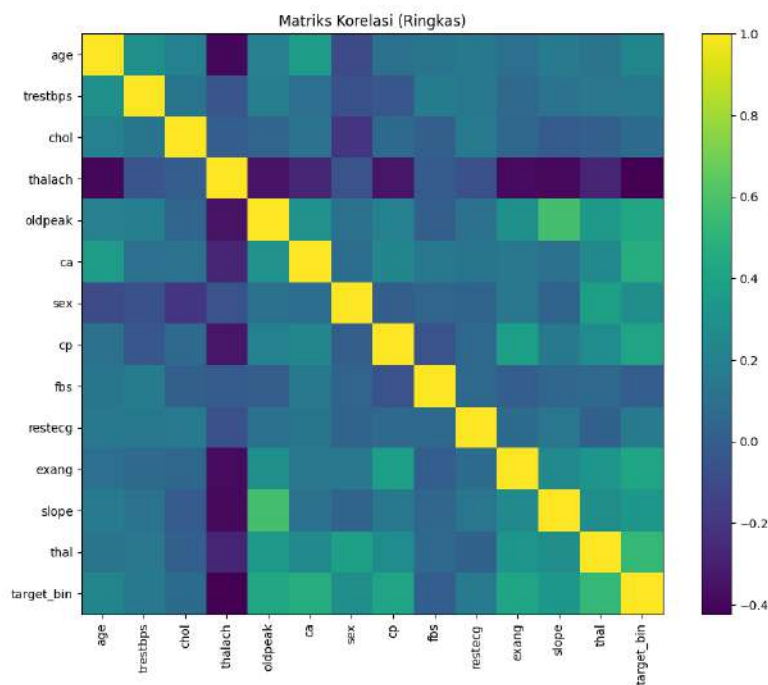


Kode program pada barcode ini dikembangkan pada praktikum ini dirancang untuk memberikan pemahaman menyeluruh mengenai bagaimana algoritma *Support Vector Machine (SVM)* – khususnya *Linear SVC* – bekerja dalam **klasifikasi risiko penyakit jantung** berbasis dataset dari UCI. Tahap awal memuat data secara *reproducible* melalui *URL* langsung dari pusat data UCI. Pembersihan data dilakukan dengan mengubah tanda *missing value* "?" menjadi *NaN* dan menerapkan **casting tipe numerik** agar dapat diproses oleh model. Setelah itu, dibentuk **variabel target biner**, yaitu **0 = sehat** dan **1 = berisiko penyakit jantung**, sehingga sistem diarahkan untuk fokus pada permasalahan deteksi dini.



Gambar 1. Distribusi Kelas Target Penyakit Jantung

Hasil visualisasi pertama (lihat Gambar 1. Distribusi Kelas Target Penyakit Jantung) memperlihatkan jumlah kelas yang tidak seimbang sempurna, yang memberi insight bahwa **model harus tetap berhati-hati terhadap potensi bias prediksi** pada kelas mayoritas. Tampilan statistik deskriptif turut mendukung pemahaman karakteristik fitur, sementara *correlation matrix* pada Gambar 2. Matriks Korelasi Fitur dan Target memberikan *insight* bahwa variabel seperti **age**, **thalach**, dan **oldpeak** tampak memiliki hubungan penting terhadap label penyakit. Korelasi ini membantu menyusun hipotesis ilmiah terkait faktor risiko klinis.

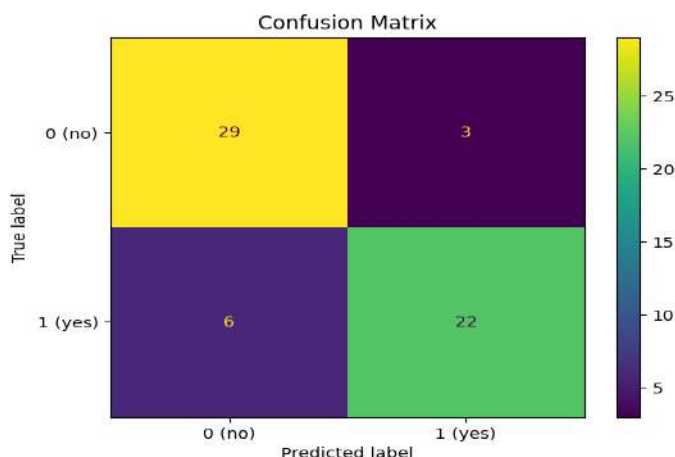


Gambar 2. Matriks Korelasi Fitur dan Target

Proses *train-test split* dilakukan secara **stratified**, mempertahankan proporsi kelas target pada data latih dan uji –

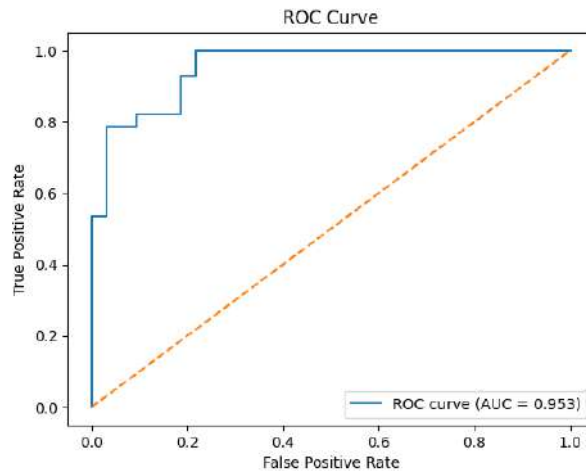
sebuah praktik baik dalam *machine learning* karena memastikan model tidak belajar secara bias.

Pipeline yang memadukan **StandardScaler** dan **SVC kernel linear** digunakan agar model mampu bekerja optimal pada fitur dengan skala berbeda tanpa terjadinya dominasi variabel tertentu. Evaluasi kinerja menampilkan **akurasi** dan **classification report** yang bermanfaat dalam menilai performa masing-masing kelas, termasuk nilai **precision**, **recall**, dan **f1-score**, yang semuanya berperan dalam interpretasi medis seperti risiko **false negative**. Visualisasi kedua pada **Gambar 3. Confusion Matrix Model SVM Linear** memperlihatkan distribusi prediksi benar dan salah, sekaligus membuka diskusi mendalam tentang **konsekuensi klinis** apabila pasien yang sebenarnya berisiko justru terdeteksi sehat oleh model.



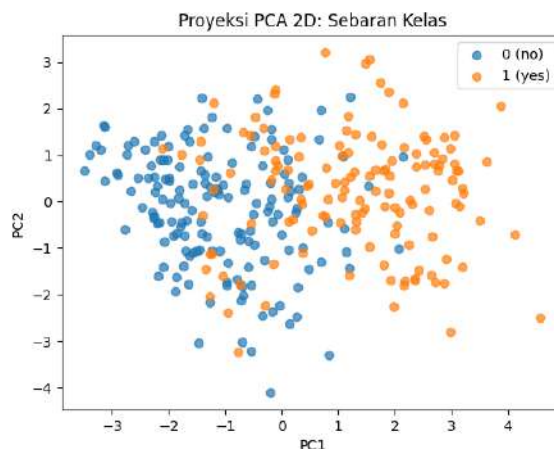
Gambar 3. Confusion Matrix Model SVM Linear

Sementara **Gambar 4. Kurva ROC Model SVM Linear pada Data Uji** menegaskan kemampuan diskriminatif model dalam berbagai ambang keputusan (threshold), ditunjukkan melalui nilai **AUC** sebagai indikator penting.



Gambar 4. Kurva ROC Model SVM Linear pada Data Uji

Pada sisi pemahaman representasi fitur, **Gambar 5. Proyeksi PCA 2D Sebaran Kelas Penyakit Jantung** menampilkan bagaimana sampel dari dua kelas dapat terpisah pada ruang fitur tereduksi. Walaupun terdapat tumpang tindih kelas yang lazim pada masalah klinis pemisahan tersebut tetap memberi sinyal adanya pola yang dikenali oleh SVM berbasis margin linear.



Gambar 5. Proyeksi PCA 2D Sebaran Kelas Penyakit Jantung

Bagian analisis parameter **C** pada model membuka ruang *exploration mindset*, dimana mahasiswa dilatih melihat **trade-off** antara **regularisasi** dan **margin**. Hal ini ditampilkan pada **Gambar 6**.

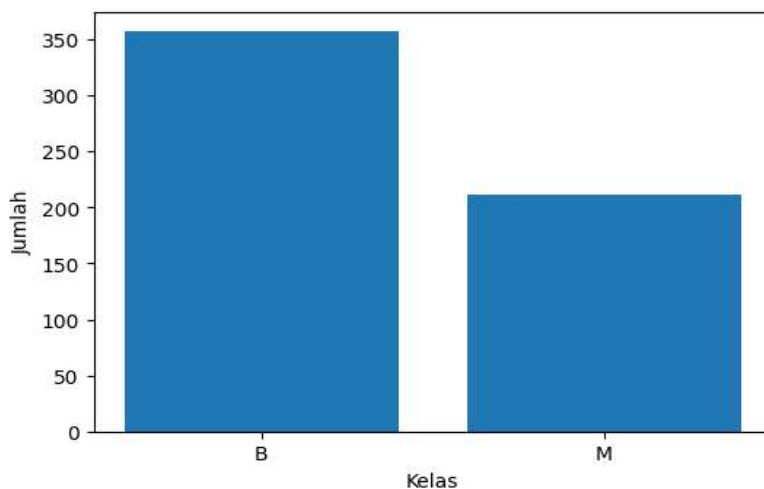
Pengaruh Parameter C terhadap Akurasi dan AUC SVM Linear yang membantu menunjukkan **nilai C optimal dapat mengurangi risiko overfitting**. Pendekatan ini memperkuat literasi analitis dan *hyperparameter tuning* sebagai bagian dari *machine learning lifecycle*.

Akhir sesi diakhiri dengan **penyimpanan model** menggunakan *joblib*, menghadirkan pengalaman nyata *deployment model* yang memungkinkan penerapan di aplikasi atau sistem pendukung keputusan klinis (*clinical decision support system*). Dengan demikian, praktikum ini mengintegrasikan seluruh alur **Data Science** mulai dari **data acquisition** → **preprocessing** → **modeling** → **evaluation** → **deployment** secara logis, ilmiah, dan relevan dengan dunia medis.

Studi Kasus: SVM untuk Diagnostik Kanker Payudara Wisconsin

Kode program pada barcode ini memanfaatkan dataset Breast Cancer Wisconsin (Diagnostic) dari UCI untuk mengklasifikasikan kasus *benign* dan *malignant* berbasis fitur numerik yang jumlahnya cukup banyak serta memiliki skala berbeda. Karakteristik ini menjadikan dataset sangat tepat untuk menegaskan pentingnya *scaling* sebelum menerapkan SVM. **Gagasan kunci** dalam alur praktikum adalah menjaga integritas proses pembelajaran mesin: data dipahami terlebih dahulu melalui eksplorasi, lalu dipersiapkan secara konsisten, kemudian model dibangun bertahap dari *baseline* hingga model terbaik hasil pencarian parameter, dan ditutup dengan interpretasi agar hasil tidak berhenti pada angka evaluasi semata.

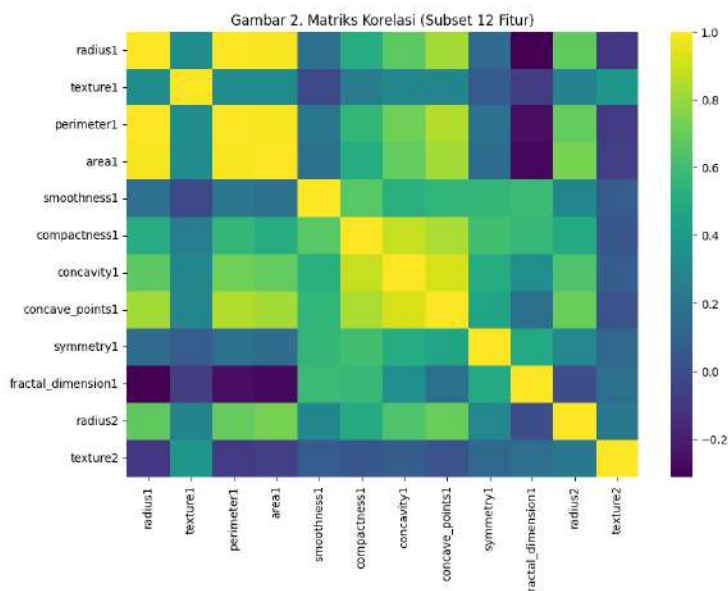
Tahap awal memeriksa bentuk data, label target, dan potensi nilai hilang. Pemeriksaan ini penting karena kualitas data menentukan kualitas model. Jika ada nilai hilang, imputasi median menjadi pilihan yang aman untuk fitur numerik karena tidak mudah terdorong oleh nilai ekstrem. Setelah validasi ringkas, visualisasi diperkenalkan agar pembaca tidak hanya melihat tabel. Gambar 1. Distribusi Kelas Target (Benign vs Malignant) memperlihatkan komposisi kelas. Dari sini muncul **insight**: jika distribusi tidak seimbang, metrik akurasi saja bisa menipu, sehingga perlu ROC/AUC untuk menilai kemampuan pemisahan kelas secara lebih stabil.



Gambar 1. Distribusi Kelas Target (Benign vs Malignant)

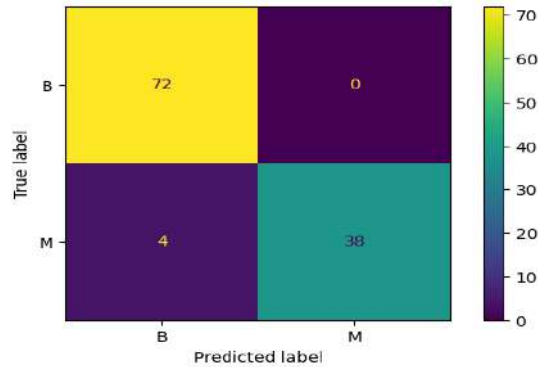
Selanjutnya, Gambar 2. Matriks Korelasi (Subset 12 Fitur) membantu memahami keterkaitan antarfitur dan potensi redundansi. Pada dataset dengan banyak fitur, korelasi tinggi sering menandakan adanya informasi yang tumpang tindih. Ini bukan berarti fitur harus langsung dihapus, tetapi menjadi bahan *brainstorming* bahwa pendekatan berbasis regularisasi atau pemilihan fitur dapat menjadi pengembangan praktikum berikutnya. Tahap pemisahan data

menggunakan *stratified split* sehingga proporsi kelas pada data latih dan uji tetap serupa. **Poin penting**: pembagian ini menjaga fairness evaluasi karena model tidak diuji pada distribusi yang berbeda dari data latih.



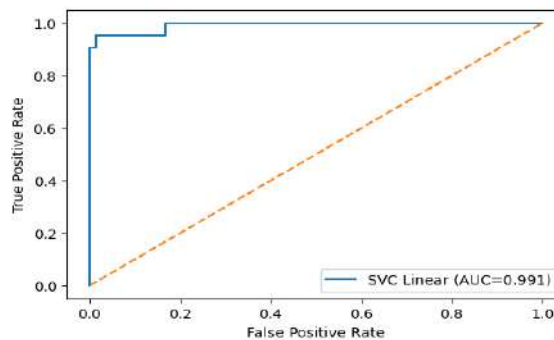
Gambar 2. Matriks Korelasi (Subset 12 Fitur)

Model *baseline* menggunakan *Logistic Regression* dengan *scaling* sebagai pembanding yang sederhana namun kuat. Baseline berfungsi sebagai garis awal, jika SVM tidak mengungguli baseline, maka revisi perlu diarahkan pada *preprocessing* atau parameter. Model utama adalah SVC linear dalam *pipeline* dengan StandardScaler, karena SVM sensitif terhadap perbedaan skala fitur. Evaluasi klasifikasi ditampilkan melalui **Gambar 3. Confusion Matrix Model SVC Linear** yang menunjukkan pola kesalahan prediksi. Ini memperkuat interpretasi, misalnya apakah kesalahan lebih banyak pada kelas *malignant* yang secara klinis lebih kritis.



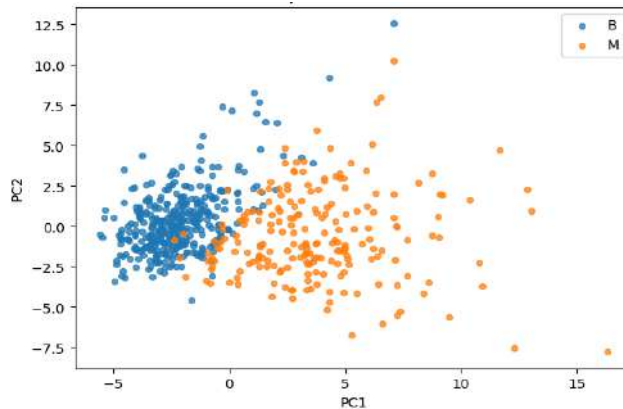
Gambar 3. Confusion Matrix Model SVC Linear

Penguatan evaluasi dilakukan dengan **Gambar 4. Kurva ROC Model SVC Linear pada Data Uji**. ROC/AUC dipakai karena lebih robust pada kondisi kelas tidak seimbang dan menilai kemampuan model memeringkatkan kelas positif.



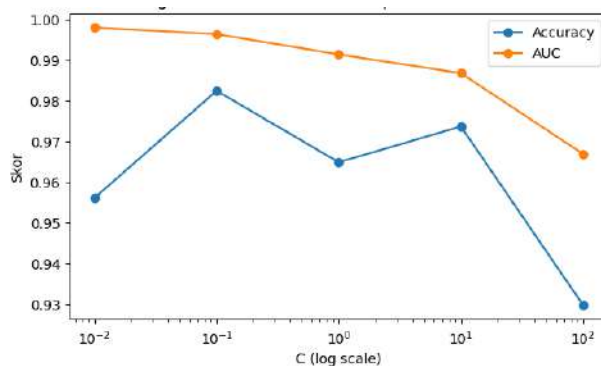
Gambar 4. Kurva ROC Model SVC Linear pada Data Uji

Untuk dukungan interpretasi visual, **Gambar 5. Proyeksi PCA 2D Sebaran Kelas** menunjukkan seberapa jauh data kedua kelas dapat dipisahkan pada ruang dua dimensi setelah *scaling*. Walau PCA tidak menentukan performa model, plot ini memberi intuisi apakah struktur data mendukung separasi yang baik.



Gambar 5. Proyeksi PCA 2D Sebaran Kelas

Tahap revisi berbasis data diwujudkan melalui analisis parameter C pada SVC linear. Gambar 6. Pengaruh Parameter C terhadap Akurasi dan AUC (SVC Linear) memperlihatkan trade-off antara margin yang lebih longgar dan penalti kesalahan.



Gambar 6. Pengaruh Parameter C terhadap Akurasi dan AUC

Insight penting: nilai C yang terlalu besar cenderung membuat model lebih ketat dan berpotensi *overfitting*, sementara C terlalu kecil dapat *underfitting*. Setelah itu, *GridSearchCV* memilih kombinasi parameter terbaik secara sistematis, lalu interpretasi ditutup dengan *permutation importance* untuk menyoroti fitur yang paling memengaruhi prediksi. Dengan demikian, praktikum tidak

hanya menghasilkan model, tetapi juga membangun pemahaman konseptual yang konsisten, ilmiah, dan mudah diikuti.

Soal

1. Jelaskan prinsip dasar Support Vector Machine (SVM) dan bagaimana konsep margin digunakan dalam pemisahan kelas. Mengapa semakin lebar margin dapat meningkatkan kemampuan generalisasi model?
2. Bagaimana konsep support vectors mempengaruhi pemilihan hyperplane dalam SVM? Jelaskan mengapa hanya titik data yang paling dekat dengan hyperplane yang mempengaruhi keputusan klasifikasi.
3. Apa yang dimaksud dengan kernel trick dalam SVM, dan bagaimana hal ini membantu dalam menangani data yang tidak dapat dipisahkan secara linear? Jelaskan bagaimana kernel linear bekerja pada masalah SVM.
4. Dalam implementasi SVM, mengapa optimasi margin dan regularisasi menjadi hal yang penting? Bagaimana parameter C mempengaruhi kemampuan model untuk melakukan generalisasi?
5. Jelaskan apa yang dimaksud dengan soft margin dalam SVM. Bagaimana konsep ini dapat menangani data yang tidak dapat dipisahkan dengan sempurna? Berikan contoh kasus penggunaannya.
6. Pada penerapan SVM di data kesehatan, mengapa pemilihan kernel linear menjadi pilihan rasional untuk analisis penyakit

jantung? Apa keuntungan dan kerugiannya jika dibandingkan dengan kernel lainnya?

7. Bagaimana Anda mengevaluasi kinerja model SVM dalam kasus ketidakseimbangan kelas seperti pada dataset penyakit jantung? Jelaskan perbedaan antara menggunakan akurasi dan AUC dalam kasus tersebut.
8. Jelaskan langkah-langkah yang harus dilakukan dalam preprocessing data untuk implementasi SVM pada dataset dengan banyak fitur yang saling berkorelasi. Bagaimana teknik seperti PCA membantu dalam masalah ini?
9. Apa yang dimaksud dengan parameter tuning pada model SVM? Berikan penjelasan tentang bagaimana pengaruh parameter C terhadap model, dan bagaimana memilih nilai parameter ini untuk menghindari overfitting atau underfitting.
10. Dalam implementasi SVM pada dataset kanker payudara, jelaskan bagaimana Anda menggunakan stratified train-test split dan apa manfaatnya dalam memastikan model tidak belajar secara bias pada kelas tertentu.

Tugas Kelompok

Tugas kelompok ini mengharuskan mempelajari dataset yang tersedia di UCI Machine Learning Repository dan Kaggle. Setiap kelompok akan diberikan satu dataset untuk dianalisis dan diminta untuk melakukan langkah-langkah berikut

1. Dataset 1 Wine Quality Dataset

Tugas Lakukan pembersihan data (jika diperlukan), eksplorasi data, dan visualisasi distribusi fitur kimiawi pada dataset

kualitas anggur. Bangun model Support Vector Machine (SVM) dengan kernel linear untuk memprediksi kualitas anggur berdasarkan berbagai fitur kimiawi seperti pH, alkohol, dan asam. Evaluasi hasil model menggunakan metrik accuracy, precision, recall, dan F1-score, serta diskusikan kekuatan dan kelemahan model yang digunakan.

Link [Wine Quality Dataset - UCI Repository](#)

2. **Dataset 2 Heart Disease UCI Dataset**

Tugas Lakukan pembersihan data (jika diperlukan), eksplorasi data, dan visualisasi distribusi fitur pada dataset penyakit jantung. Bangun model Support Vector Machine (SVM) dengan kernel linear untuk mengklasifikasikan risiko penyakit jantung berdasarkan fitur klinis seperti tekanan darah, kolesterol, dan usia. Evaluasi hasil model menggunakan metrik accuracy, precision, recall, dan F1-score, serta diskusikan kekuatan dan kelemahan model yang digunakan.

Link [Heart Disease UCI - UCI Repository](#)

3. **Dataset 3 Credit Card Fraud Detection**

Tugas Lakukan pembersihan data (jika diperlukan), eksplorasi data, dan visualisasi distribusi fitur pada dataset deteksi penipuan transaksi kartu kredit. Gunakan Support Vector Machine (SVM) dengan kernel linear untuk mengklasifikasikan transaksi kartu kredit yang sah dan yang mencurigakan. Evaluasi hasil model menggunakan metrik accuracy, precision, recall, dan F1-score, serta diskusikan kekuatan dan kelemahan model yang digunakan. Lakukan analisis lebih lanjut untuk mengatasi masalah ketidakseimbangan kelas

menggunakan teknik seperti SMOTE (Synthetic Minority Over-sampling Technique).

Link [Credit Card Fraud Detection - Kaggle](#)

Refleksi Pembelajaran

Dalam pembelajaran tentang Support Vector Machine (SVM), saya mempelajari pentingnya margin dalam memisahkan kelas data secara optimal. Konsep support vectors yang mempengaruhi pemilihan hyperplane sangat membantu dalam memahami cara SVM bekerja secara efisien, terutama dalam data berdimensi tinggi. Selain itu, penggunaan kernel trick memungkinkan SVM untuk menangani data yang tidak dapat dipisahkan secara linear. Evaluasi model menggunakan metrik seperti accuracy, precision, recall, dan F1-score memberikan wawasan tentang kekuatan dan kelemahan model. Secara keseluruhan, SVM terbukti efektif dalam berbagai aplikasi, termasuk analisis data kesehatan dan deteksi penipuan.

Unsupervised Learning

K-Means Clustering

CPL:

Mahasiswa mampu mengelompokkan data tanpa label.

CPMK:

Menerapkan algoritma K-Means.

SubCPMK:

Menggunakan `KMeans(n_clusters=3)`.

A. Pemilihan *Unsupervised Learning* untuk Klasterisasi Data

Unsupervised learning dipakai saat data tidak memiliki label, tetapi tetap menyimpan pola yang dapat dicari. Logika dasarnya: objek yang **mirip** cenderung berdekatan, objek yang **berbeda** cenderung berjauhan. Klasterisasi membantu mengelompokkan objek berdasarkan kemiripan fitur sehingga data yang semula terlihat acak menjadi lebih terstruktur. **Gagasan kunci** yang perlu dipegang ialah klaster bukan cap benar atau salah, melainkan ringkasan pola yang perlu diperiksa kewajarannya.

Penalaran yang sehat dimulai dari definisi “mirip”. Banyak metode memakai jarak Euclidean, sehingga asumsi penting ikut terbawa: fitur harus berada pada skala yang sebanding, dan perbedaan numerik dianggap mewakili perbedaan makna. Fitur berskala besar dapat mendominasi jarak dan menggeser hasil klaster. Pencilan yang ekstrem dapat menarik pusat klaster dan

membuat kelompok utama terlihat kabur. Fitur kategori juga tidak bisa dirata-ratakan secara bermakna, sehingga perlu perlakuan khusus.

Insight: apa ukuran kemiripan yang paling masuk akal, fitur mana yang paling menentukan, dan apakah ada nilai ekstrem yang perlu ditangani.

B. Mekanisme K Means, Centroid, dan Klaster

K-Means membentuk klaster dengan pusat yang disebut **centroid**. Prosesnya berulang: pilih centroid awal, tempatkan tiap titik ke centroid terdekat, hitung ulang centroid dari anggota klaster, lalu ulangi sampai perubahan kecil. Alur ini mudah dipahami, tetapi hasilnya sangat dipengaruhi beberapa faktor. Inisialisasi centroid dapat membawa algoritma ke solusi lokal. Strategi *k means++* biasanya lebih stabil karena memilih pusat awal yang lebih menyebar. Skala fitur juga menentukan, sehingga standardisasi atau normalisasi sering menjadi langkah logis sebelum klasterisasi. Asumsi bentuk klaster ikut penting: K Means paling cocok ketika klaster cenderung kompak dan relatif bulat, sedangkan pola memanjang atau melengkung sering tidak tertangkap baik. Pencilon dapat menggeser centroid dan menurunkan kualitas kelompok. Pemakaian `KMeans(n_clusters=3)` berguna sebagai pemetaan awal yang ringkas. Angka tiga sebaiknya diperlakukan sebagai titik awal untuk diuji, bukan angka baku.

Insight: apakah tiga kelompok sudah cukup menjelaskan variasi data, atau masih ada pola yang terpaksa digabung.

C. Evaluasi Klaster melalui *Inertia* dan *Elbow Method*

Kualitas klaster perlu ukuran yang jelas. **Inertia** menghitung jumlah jarak kuadrat titik ke centroid klasternya. Nilai yang kecil berarti klaster lebih rapat, tetapi klaster rapat tidak selalu berarti klaster bermakna. Sifat penting inertia ialah nilainya selalu turun saat jumlah klaster ditambah, sehingga inertia saja tidak cukup untuk memilih jumlah klaster.

Elbow method membantu memilih jumlah klaster melalui grafik: tambah klaster, inertia turun. Titik “tekuk” menandai saat penambahan klaster mulai memberi keuntungan kecil dibanding tambahan kompleksitas. Kadang grafik tidak menunjukkan tekuk yang jelas. Situasi seperti ini bisa menandakan data lebih cocok dipandang sebagai gradasi kontinu daripada kelompok tegas. Agar keputusan lebih kuat, metrik pelengkap bisa dipakai. Skor *silhouette* membantu menilai apakah titik lebih dekat ke klasternya sendiri dibanding klaster lain. Indeks Davies Bouldin membantu membandingkan sebaran di dalam klaster dan jarak antar klaster. **Prinsip evaluasi** yang aman memakai beberapa ukuran lalu menyesuaikannya dengan penjelasan yang masuk akal.

D. Pemilihan Algoritma *Unsupervised Learning*

K-Means bukan pilihan untuk semua kondisi. *Hierarchical clustering* cocok saat ingin melihat struktur bertingkat. Dendrogram memperlihatkan bagaimana kelompok kecil bergabung menjadi kelompok lebih besar, sehingga jumlah klaster dapat dipilih setelah melihat pola. Pilihan *linkage* dapat mengubah bentuk dendrogram, sehingga hasil perlu dibandingkan agar tidak terjebak pada satu pilihan.

DBSCAN membentuk klaster berdasarkan kepadatan dan dapat menandai titik sepi sebagai *noise*. Keunggulannya muncul saat bentuk klaster tidak beraturan dan banyak pencilan. Tantangan utama berada pada pemilihan epsilon dan *min samples*. Parameter terlalu ketat membuat banyak *noise*, terlalu longgar membuat klaster melebur. Kepadatan yang berbeda antar area juga dapat menyulitkan.

PCA dipakai untuk mengurangi dimensi agar pola lebih mudah dilihat dan kebisingan berkurang. PCA bukan klasterisasi, sehingga perannya adalah representasi. Risiko PCA ialah hilangnya informasi jika komponen yang dipilih terlalu sedikit.

Insight:: apakah proyeksi PCA menjaga pemisahan yang terlihat, atau justru mengaburkan pola.

E. Validasi, dan Interpretasi Klaster

Klasterisasi bernilai ketika kelompok yang terbentuk bisa dijelaskan dan dipakai. Interpretasi yang kuat biasanya bersandar pada ringkasan statistik, misalnya rata rata, median, dan sebaran, lalu mencari fitur pembeda paling jelas antar klaster. Pemberian nama klaster sebaiknya mengikuti ciri dominan, bukan dugaan yang sulit dibuktikan.

Insight:: klaster mana yang paling aktif, paling pasif, paling ekstrem, dan fitur apa yang paling membedakan.

Validasi pada *unsupervised learning* menekankan stabilitas, bukan akurasi. K Means dapat diuji dengan mengulang pelatihan pada beberapa *random seed* dan melihat apakah pembagian klaster relatif konsisten. DBSCAN dapat diuji melalui analisis sensitivitas parameter untuk memastikan pola tidak mudah berubah hanya

karena sedikit pergeseran nilai. *Hierarchical clustering* dapat diuji dengan membandingkan beberapa *linkage* dan melihat apakah struktur utama tetap muncul. Alur kerja yang konsisten ialah: definisikan kemiripan, siapkan fitur dan skala, pilih algoritma sesuai bentuk data, ukur kualitas dengan metrik yang tepat, visualisasikan pola, lalu uji stabilitas.

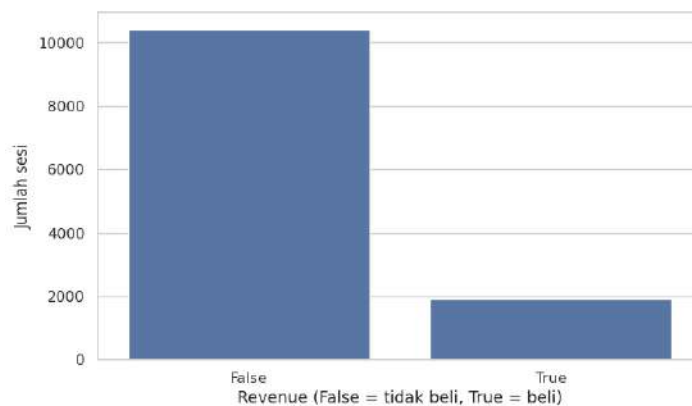
Kode Program: KMeans Klaster Pengunjung Web



Kode program pada barcode ini untuk mengklaster **perilaku pengunjung web e-commerce** menggunakan algoritma *KMeans* pada *dataset* Online Shoppers Purchasing Intention dari UCI. Tahapan yang digunakan mengikuti alur *machine learning* yang sistematis, mulai dari pemuatan data, eksplorasi awal, prapemrosesan, penentuan jumlah *cluster*, hingga interpretasi hasil dan visualisasi dalam ruang dua dimensi.

Pada bagian awal dilakukan **setup lingkungan Google Colab** dengan memasang pustaka *ucimlrepo* dan mengimpor *library* utama seperti *NumPy*, *pandas*, *matplotlib*, *seaborn*, serta komponen dari *scikit-learn* seperti *StandardScaler*, *KMeans*, *Silhouette Score*, dan *PCA*. Penataan gaya grafis melalui *seaborn* membantu menghasilkan visual yang konsisten dan mudah dibaca, sehingga mendukung pemahaman mahasiswa terhadap pola data yang kompleks.

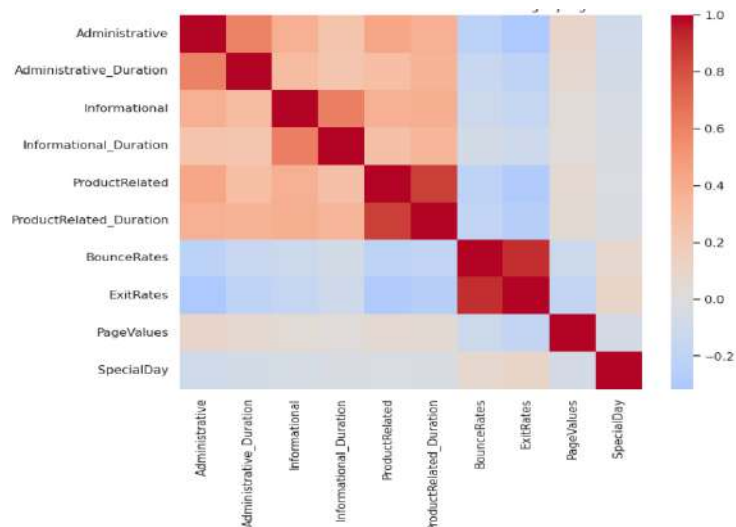
Tahap pemuatan data menggunakan `fetch_ucirepo(id=468)` mengambil *dataset* langsung dari repositori UCI. Fitur perilaku pengunjung dan target Revenue digabungkan dalam satu *DataFrame*, kemudian ditampilkan ukuran data, lima baris pertama, dan informasi kolom. Langkah ini memberikan gambaran awal tentang struktur data dan menjadi dasar untuk memastikan bahwa pemetaan variabel sudah benar. Gambar 1. Distribusi Kelas Target Revenue memperlihatkan ketidakseimbangan kelas, di mana sesi tanpa pembelian biasanya jauh lebih banyak dibandingkan sesi yang berujung transaksi. Kondisi ini penting untuk diingat saat membaca proporsi pembelian di tiap *cluster*.



Gambar 1. Distribusi Kelas Target Revenue

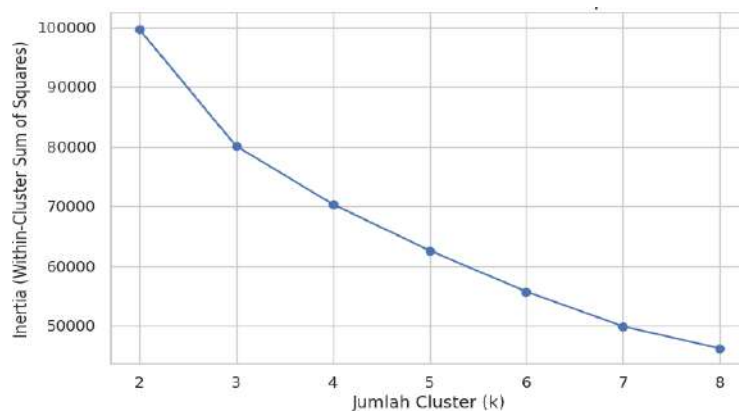
Analisis deskriptif awal dilanjutkan dengan ringkasan statistik fitur numerik. Setelah itu, program secara eksplisit memilih sepuluh fitur perilaku, yaitu jumlah dan durasi kunjungan administratif, informasional, dan produk, serta *BounceRates*, *ExitRates*, *PageValues*, dan *SpecialDay*. Pemilihan ini menjaga fokus pada **pola interaksi pengunjung dengan situs**. Gambar 2. Korelasi Antar Fitur Perilaku Pengunjung menyajikan *heatmap* korelasi yang membantu

mengidentifikasi fitur yang saling berkaitan, misalnya hubungan antara jumlah halaman produk dan durasinya.



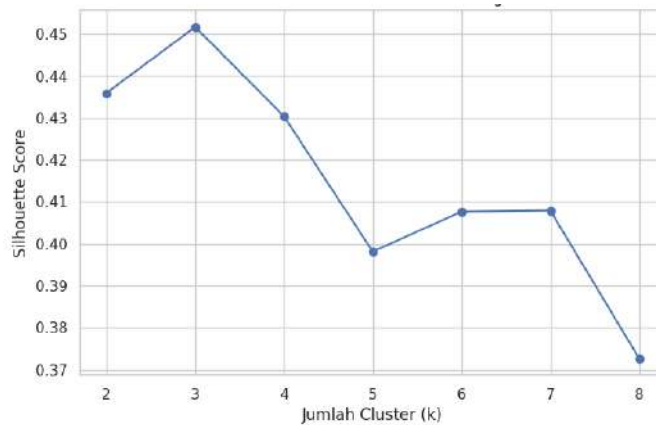
Gambar 2. Korelasi Antar Fitur Perilaku Pengunjung

Tahap prapemrosesan dilakukan dengan *StandardScaler* untuk menyamakan skala seluruh fitur numerik. Hal ini sangat penting karena *KMeans* sensitif terhadap perbedaan skala. Data yang telah diskalakan menjadi masukan untuk penentuan jumlah *cluster*. Program menghitung *inertia* dan *Silhouette Score* untuk nilai *k* dari 2 sampai 8. Gambar 3. Elbow Plot untuk Menentukan *k* Optimal. menunjukkan perubahan *inertia* terhadap *k*.



Gambar 3. Elbow Plot untuk Menentukan *k* Optimal

Sedangkan Gambar 4 berjudul **“Silhouette Score untuk Berbagai Nilai k”** menampilkan kualitas pemisahan *cluster*. Program kemudian memilih **k terbaik secara otomatis** berdasarkan *Silhouette Score* maksimum sehingga pemilihan jumlah *cluster* bersifat objektif dan terukur.



Gambar 4. Silhouette Score untuk Berbagai Nilai k

Setelah k terbaik diperoleh, algoritma *KMeans* dilatih dan label *cluster* disimpan dalam `df["cluster"]`. Tabel 1 Profil Rata-Rata Fitur Perilaku per Cluster menyajikan karakteristik rata-rata tiap *cluster*, misalnya **cluster dengan PageValues tinggi dan BounceRates rendah** dapat diinterpretasikan sebagai pengunjung yang serius mempertimbangkan pembelian.

Tabel 1. Profil Rata-Rata Fitur Perilaku per Cluster

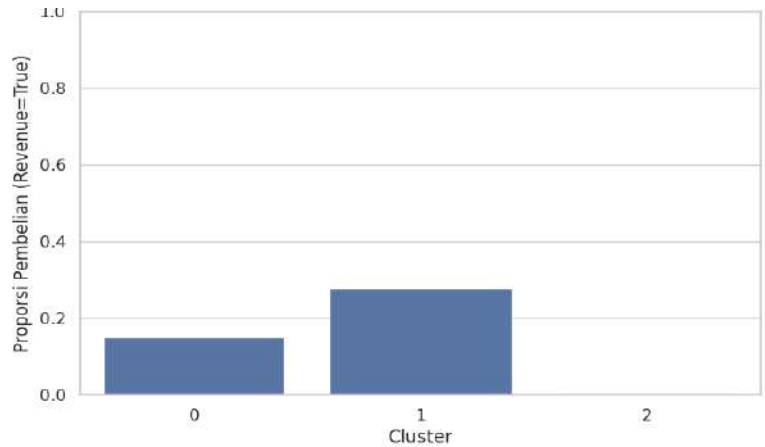
cluster	Administrative	Administrative	Informational	Informational	ProductRelated	ProductRelated	BounceRates	ExitRates	PageValues	SpecialDay
		_Duration		_Duration		_Duration				
0	1.708972	51.182892	0.216535	8.145937	23.888210	873.802620	0.009178	0.032108	5.877428	0.061707
1	7.371007	307.731161	2.520270	212.732726	96.743857	3831.145305	0.006275	0.019430	9.757748	0.030098
2	0.048571	1.418921	0.015238	0.086857	3.029524	57.308044	0.166493	0.180525	0.000000	0.107429

Selanjutnya, Revenue dikonversi menjadi variabel numerik untuk dianalisis terhadap *cluster*. Tabel 2 Proporsi Pembelian (Revenue=True) di Setiap Cluster

Tabel 2. Proporsi Pembelian (Revenue=True) di Setiap Cluster:

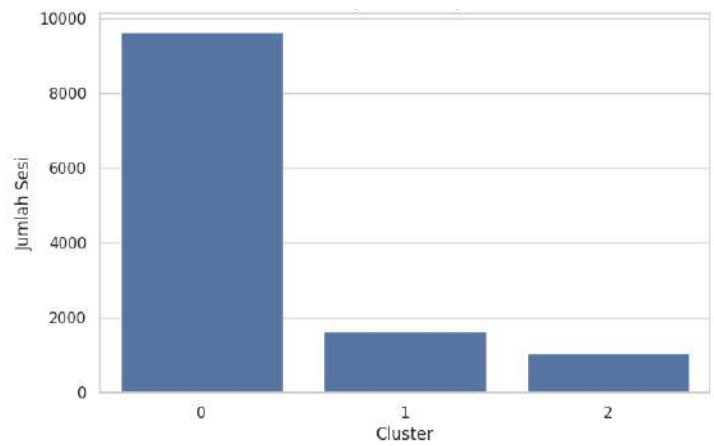
	jumlah_sesi	jumlah_beli	proporsi_beli
cluster			
0	9652	1450	0.150228
1	1628	452	0.277641
2	1050	6	0.005714

dan Gambar 5 .Proporsi Pembelian per Cluster memberi wawasan tentang *cluster* mana yang paling sering menghasilkan transaksi.



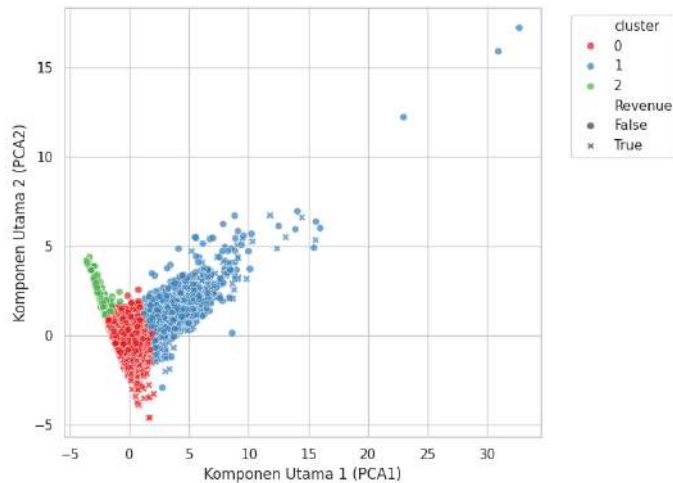
Gambar 5 .Proporsi Pembelian per Cluster

Gambar 6.Jumlah Sesi per Cluster melengkapi informasi dengan menampilkan ukuran masing-masing segmen.



Gambar 6.Jumlah Sesi per Cluster

Terakhir, visualisasi dua dimensi dilakukan menggunakan *PCA* sehingga seluruh fitur perilaku dipadatkan menjadi dua komponen utama. Gambar 7 Proyeksi PCA 2D Sebaran Cluster dan Status Revenue menampilkan sebaran *cluster* dan posisi titik dengan Revenue=True.



Gambar 7 Proyeksi PCA 2D Sebaran Cluster dan Status Revenue

Pola ini memberikan **insight bahwa perilaku browsing tertentu berkorelasi dengan kecenderungan pembelian**, sehingga *cluster* dengan proporsi pembelian tertinggi dapat menjadi target utama strategi pemasaran yang lebih terarah dan efisien.

Studi Kasus: DBSCAN untuk Data Cardiotocography

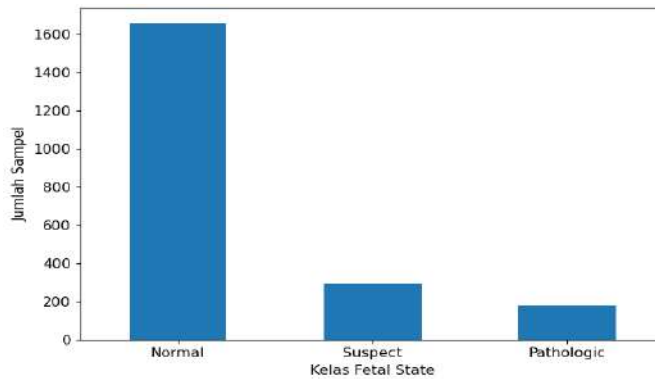
Scan Me
Code
Program



Kode program pada barcode ini menyusun alur lengkap *machine learning* berbasis *unsupervised clustering* dengan algoritma *DBSCAN* pada data **Cardiotocography** dari UCI. Tahap pertama adalah **impor pustaka** yang relevan, seperti *pandas* untuk pengolahan tabel, *matplotlib* untuk visualisasi, serta modul dari *scikit learn* untuk *preprocessing*, *clustering*, dan evaluasi. Pengaturan ukuran dan huruf grafik memastikan hasil visual mudah dibaca dalam praktikum.

Bagian **pemanggilan dan pembersihan data** memuat berkas CTG.xls langsung dari *repository* UCI melalui variabel *url*. Penggunaan *skipfooter=3* dan penghapusan kolom *Unnamed* adalah langkah yang logis karena struktur berkas aslinya menyimpan beberapa baris dan kolom tidak informatif. Mengambil baris pertama sebagai judul kolom, kemudian mengubah seluruh nilai menjadi *float*, menjadikan data siap diolah tanpa gangguan format campuran. Ini menjaga integritas data medis yang sensitif.

Tahap **EDA ringan** menggunakan *describe().T* untuk melihat ringkasan statistik seluruh fitur dan memetakan variabel NSP ke label teks **Normal**, **Suspect**, dan *Pathologic*. Gambar 1. Distribusi Kelas Fetal State (NSP) menggambarkan ketidakseimbangan jumlah kasus, di mana kelas *normal* biasanya mendominasi. Ini memberi **insight** penting bahwa cluster kecil atau titik *noise* berpotensi berkaitan dengan kondisi berisiko.



Gambar 1. Distribusi Kelas Fetal State (NSP)

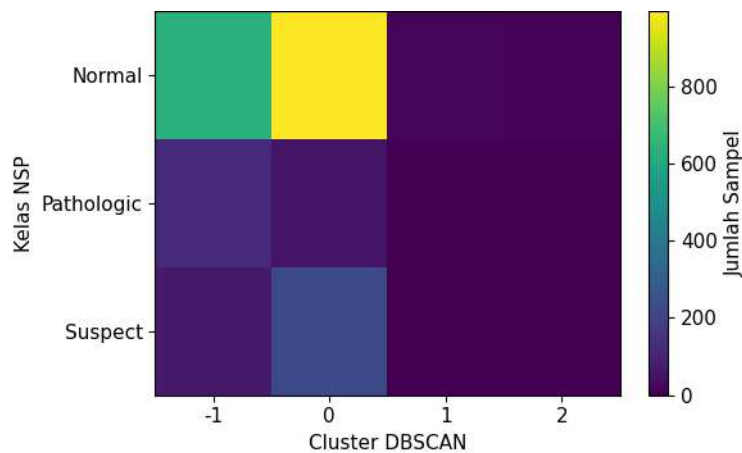
Pada bagian **preprocessing**, program memilih 21 fitur fisiologis sebagai masukan *unsupervised*, kemudian menormalkan semuanya dengan StandardScaler. Langkah ini sangat penting karena *DBSCAN* berbasis jarak Euclidean sehingga perbedaan skala dapat menyesatkan hasil *clustering*. Informasi bentuk matriks fitur memastikan jumlah sampel dan dimensi sudah sesuai rancangan.

Blok **eksplorasi parameter DBSCAN** menghitung kombinasi *eps* dan *min_samples*, kemudian mencatat banyaknya cluster dan rasio *noise* untuk masing-masing kombinasi. Tabel *res_df* menampilkan ringkasan eksplorasi sehingga pemilihan parameter tidak bersifat acak. Dengan menargetkan sekitar tiga cluster (mencerminkan kelas klinis) dan *noise* sedang, program menerapkan kriteria yang **logis dan selaras dengan domain medis**.

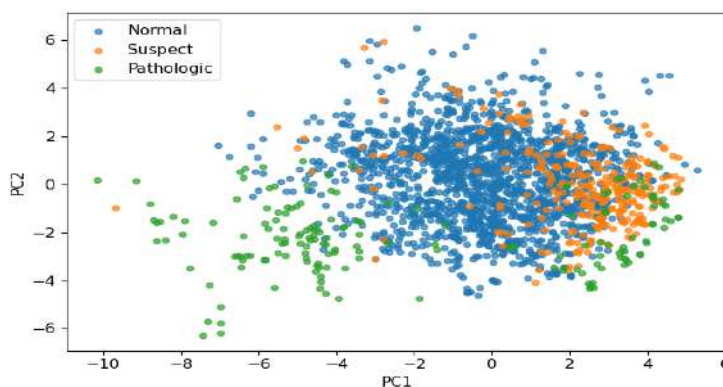
Setelah parameter dipilih, bagian **pelatihan DBSCAN final** menghasilkan label cluster dan menghitung jumlah cluster serta jumlah *noise*. Evaluasi eksternal memakai *Adjusted Rand Index*, *homogeneity*, *completeness*, dan *v measure* terhadap NSP. Evaluasi internal memakai *silhouette score* pada titik non *noise*. Nilai yang tidak terlalu tinggi tetap bermakna karena tugas ini adalah

unsupervised, sehingga fokus utama adalah pola dan kelompok risiko, bukan akurasi klasifikasi murni.

Tabel silang ct menampilkan hubungan antara kelas klinis dan cluster, yang kemudian divisualisasikan sebagai Gambar 4. Heatmap NSP vs Cluster DBSCAN. **Insight penting** muncul ketika satu cluster didominasi label **Suspect** atau **Pathologic**, yang dapat ditafsirkan sebagai kelompok kondisi janin berisiko.

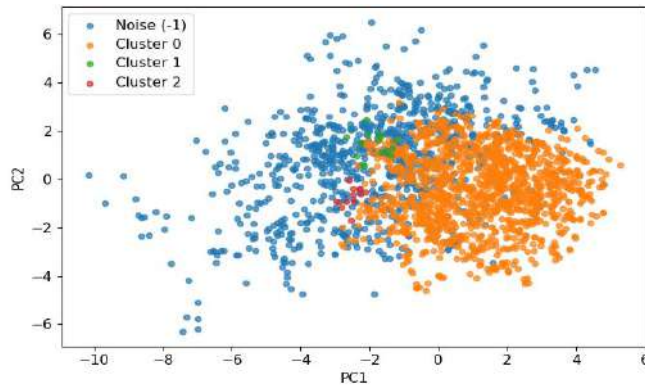


Terakhir, proyeksi dua dimensi dengan *PCA* menghasilkan Gambar 2. Visualisasi PCA 2D Berdasarkan Kelas NSP



Gambar 2. Visualisasi PCA 2D Berdasarkan Kelas NSP

Gambar 3. Visualisasi PCA 2D Berdasarkan Cluster DBSCAN yang membantu membaca pola ruang fitur antara kondisi normal dan tidak normal. Secara keseluruhan, kode ini konsisten, terstruktur, dan mendukung tujuan analisis risiko klinis dengan pendekatan *clustering*.



Gambar 3. Visualisasi PCA 2D Berdasarkan Cluster DBSCAN

Soal

1. Jelaskan konsep dasar dari Unsupervised Learning dan bagaimana perbedaannya dengan Supervised Learning. Berikan contoh konkret dari masing-masing jenis pembelajaran.
2. Apa yang dimaksud dengan klasterisasi data dalam Unsupervised Learning? Jelaskan proses kerja K-Means secara rinci dan sebutkan faktor-faktor yang mempengaruhi hasil klasterisasi.
3. Bagaimana cara memilih jumlah klaster yang optimal dalam K-Means? Jelaskan penggunaan Elbow Method dan Silhouette Score dalam proses ini.

4. Diskusikan kelemahan dan kelebihan dari algoritma K-Means. Apa yang menyebabkan K-Means tidak dapat digunakan dengan baik pada kluster yang tidak berbentuk bulat atau memanjang?
5. DBSCAN adalah algoritma berbasis kepadatan yang memiliki kelebihan dalam mengidentifikasi noise. Jelaskan bagaimana DBSCAN bekerja dan apa tantangan yang dihadapi dalam memilih parameter epsilon (ϵ) dan min_samples.
6. Dalam konteks Unsupervised Learning, jelaskan peran PCA dalam mengurangi dimensi data. Apa potensi kerugiannya jika komponen yang dipilih terlalu sedikit?
7. Apa yang dimaksud dengan stabilitas kluster dalam algoritma Unsupervised Learning? Bagaimana Anda menguji stabilitas kluster yang dihasilkan oleh K-Means atau DBSCAN?
8. Jelaskan bagaimana Anda akan mengatasi masalah penciran (outliers) dalam data sebelum melakukan klusterisasi dengan K-Means. Mengapa penciran dapat mempengaruhi hasil klusterisasi?
9. Sebutkan beberapa teknik validasi yang digunakan dalam mengevaluasi hasil klusterisasi. Jelaskan secara rinci perbedaan antara evaluasi internal dan eksternal dalam klusterisasi.
10. Berikan contoh aplikasi praktis dari K-Means dan DBSCAN dalam kehidupan nyata. Jelaskan bagaimana algoritma tersebut dapat membantu pengambilan keputusan yang lebih baik dalam setiap kasus yang Anda sebutkan.

Tugas Kelompok

Tugas kelompok ini mengharuskan mempelajari dataset berbeda yang tersedia di UCI Machine Learning Repository. Setiap kelompok akan diberikan satu dataset untuk dianalisis dan diminta untuk melakukan langkah-langkah berikut:

1. Dataset 1 Wholesale Customers Dataset

Tugas: Lakukan pembersihan data (jika diperlukan), eksplorasi data, dan visualisasi distribusi fitur seperti total pengeluaran pada kategori produk berbeda (Fresh, Milk, Grocery, etc.). Bangun model K-Means untuk mengelompokkan pelanggan berdasarkan pola pengeluaran mereka. Tentukan jumlah klaster yang optimal menggunakan Elbow Method dan Silhouette Score. Evaluasi hasil klasterisasi dan interpretasikan karakteristik utama setiap klaster, seperti apakah pelanggan lebih banyak berbelanja untuk produk tertentu. Diskusikan kekuatan dan kelemahan model K-Means yang digunakan.

Link: [Wholesale Customers Dataset](#)

2. Dataset 2 Iris Dataset

Tugas: Lakukan eksplorasi data untuk menganalisis hubungan antar fitur morfologi bunga iris. Bangun model K-Means untuk mengklasifikasikan spesies iris berdasarkan panjang dan lebar kelopak dan mahkota. Visualisasikan struktur klaster dan analisis bagaimana pemisahan data dilakukan berdasarkan fitur-fitur tersebut. Evaluasi kinerja model menggunakan Silhouette Score dan Inertia serta diskusikan kekuatan dan kelemahan model yang digunakan.

Link: [Iris Dataset](#)

3. Dataset 3 Heart Disease Dataset

Tugas: Lakukan pembersihan data, eksplorasi data, dan visualisasi distribusi fitur yang berkaitan dengan kondisi jantung, seperti usia, tekanan darah, dan kadar kolesterol. Bangun model DBSCAN untuk mengelompokkan pasien berdasarkan pola risiko penyakit jantung mereka. Tentukan parameter epsilon (ϵ) dan min_samples yang optimal. Evaluasi hasil klasterisasi menggunakan Silhouette Score dan Adjusted Rand Index. Diskusikan kekuatan dan kelemahan model DBSCAN dalam menangani data dengan penciran dan klaster yang tidak beraturan.

Link: [Heart Disease Dataset](#)

Refleksi Pembelajaran

Pembelajaran tentang Unsupervised Learning dengan algoritma K-Means dan DBSCAN memberikan wawasan penting dalam mengelompokkan data yang tidak memiliki label. Dengan menggunakan dataset dari berbagai bidang, seperti pelanggan e-commerce dan kondisi kesehatan, kita dapat mengidentifikasi pola dan segmentasi yang dapat membantu pengambilan keputusan yang lebih baik. Penerapan teknik seperti Elbow Method dan Silhouette Score memperkuat pemahaman tentang evaluasi klaster, sementara PCA membantu memvisualisasikan hasil klasterisasi. Keterampilan ini berguna dalam berbagai aplikasi dunia nyata, dari pemasaran hingga analisis risiko, dan memperkaya kemampuan analisis data untuk menyelesaikan masalah yang lebih kompleks.

Principal Component Analysis (PCA)

CPL:

Mahasiswa memahami reduksi dimensi.

CPMK:

Menurunkan kompleksitas data menggunakan PCA.

SubCPMK:

Menggunakan `PCA().fit_transform(X)`.

A. Mengapa Data Perlu Disederhanakan

Data modern sering tersusun dari banyak variabel yang saling berhubungan dan sulit dipahami secara langsung. Kondisi ini membuat proses analisis menjadi berat dan pesan utama data tidak mudah ditangkap. *Principal Component Analysis (PCA)* berperan sebagai pendekatan yang membantu **menyederhanakan data**, **meringkas informasi**, dan **menjaga makna utama** tanpa harus menghilangkan inti data. Penyederhanaan ini bertujuan agar data lebih mudah dibaca, dianalisis, serta dijelaskan kepada khalayak luas. Melalui PCA, data yang rumit diubah menjadi representasi yang lebih ringkas sehingga gambaran besarnya dapat terlihat dengan jelas.

B. Cara PCA Menangkap Informasi Paling Penting

PCA bekerja dengan mencari arah data yang mengandung **informasi paling besar**. Ukuran yang digunakan untuk menentukan penting atau tidaknya suatu arah adalah **variansi**, yaitu tingkat penyebaran data. Arah dengan variansi terbesar dipilih terlebih

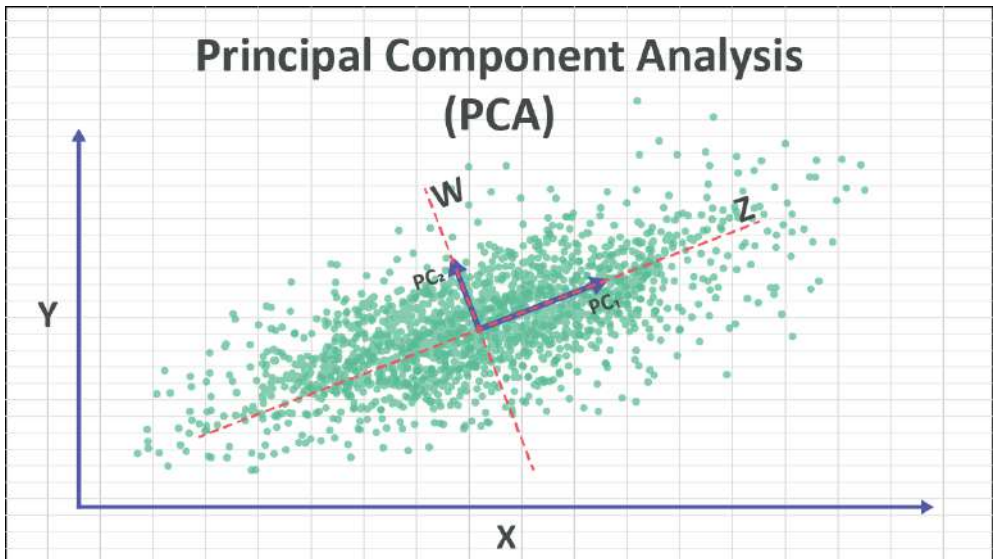
dahulu karena dianggap paling mewakili karakter utama data. Proses ini membuat PCA fokus pada bagian data yang benar-benar bermakna dan mengurangi pengaruh informasi yang kurang relevan. Pendekatan ini menjadikan analisis lebih efisien dan terarah.

C. Peran Nilai Eigen dalam Menentukan Komponen Utama

Secara matematis, PCA berkaitan dengan *eigenvalues* dan *eigenvectors*. *Eigenvalues* menunjukkan seberapa besar kontribusi setiap komponen terhadap keseluruhan informasi data, sedangkan *eigenvectors* menunjukkan arah baru pembentuk komponen tersebut. Komponen dengan *eigenvalues* besar dianggap paling penting karena mampu menjelaskan sebagian besar variasi data. Pemahaman ini membantu memilih jumlah komponen yang cukup untuk mewakili data secara akurat dan sederhana.

D. Memetakan Data ke Bentuk yang Lebih Sederhana

Setelah komponen utama ditentukan, data dipetakan ke ruang baru melalui proses *projection*. Proses ini mengubah data asli menjadi bentuk baru yang lebih sederhana tanpa mengubah hubungan dasarnya. Hasil pemetaan membuat pola, kedekatan, dan perbedaan antar data lebih mudah terlihat. Penyajian data dalam dua atau tiga dimensi memberikan kemudahan dalam **visualisasi** dan **pemahaman intuitif**, terutama ketika ingin melihat struktur umum data.



Gambar 1. Visualisasi PCA Dua Dimensi untuk Pengenalan Pola

Gambar ini memperlihatkan bagaimana *Principal Component Analysis (PCA)* digunakan untuk menyederhanakan data kompleks menjadi dua dimensi agar pola utama lebih mudah dikenali. Titik-titik hijau merupakan data asli pada sumbu **X-Y**, yang memiliki penyebaran acak dan sulit diinterpretasi secara langsung. PCA kemudian menghitung arah dengan variasi terbesar pada data dan menetapkan **PC1** sebagai sumbu baru yang menangkap informasi paling dominan. Arah kedua, **PC2**, tegak lurus terhadap PC1 dan mewakili variasi terbesar berikutnya.

Melalui proses ini, data diproyeksikan ke ruang dua dimensi berdasarkan PC1 dan PC2, sehingga struktur penyebaran data menjadi lebih jelas terlihat. Garis-garis putus dapat dianggap sebagai perbandingan dengan sumbu awal, menunjukkan bahwa orientasi baru lebih optimal dalam merepresentasikan karakteristik data. Visualisasi ini membuktikan bahwa PCA mampu melakukan reduksi dimensi tanpa menghilangkan informasi penting, membantu

analisis pola, klaster data, dan pemahaman hubungan antarvariabel secara lebih efektif dalam *machine learning* serta eksplorasi data.

E. Manfaat PCA untuk Pemahaman dan Visualisasi Data

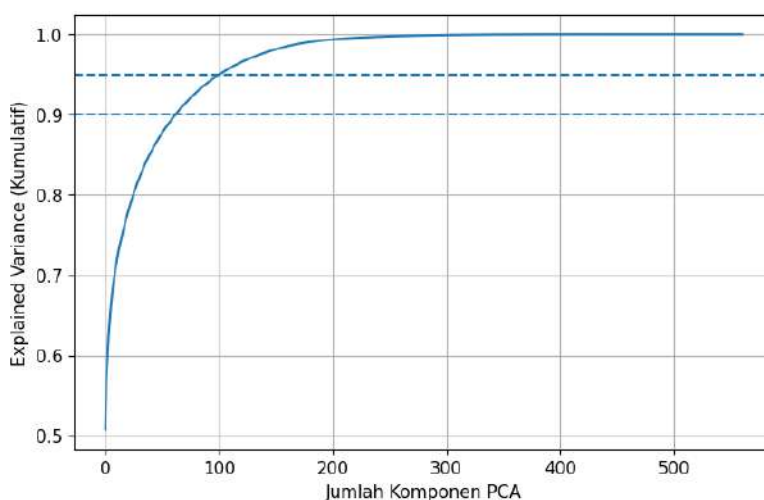
PCA memberikan **insight** bahwa sebagian besar informasi data sering kali dapat dirangkum hanya dengan beberapa komponen utama. Temuan ini menegaskan bahwa data kompleks tidak selalu memerlukan banyak variabel untuk dipahami. PCA membantu menyampaikan informasi secara lebih jelas, ringkas, dan komunikatif. Melalui pendekatan ini, analisis data menjadi lebih mudah dipahami, lebih efisien, dan tetap memiliki dasar ilmiah yang kuat.

Kode Program: Penggunaan PCA dan Model Linier



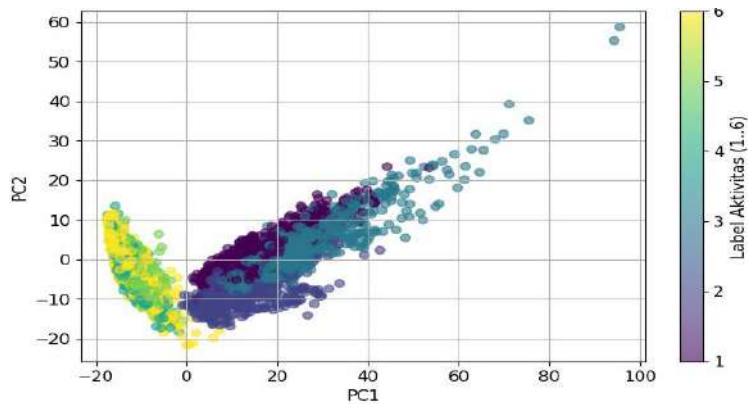
Kode program pada barcode ini untuk memperkuat pemahaman mahasiswa mengenai **reduksi dimensi** menggunakan *Principal Component Analysis (PCA)* serta dampaknya terhadap **kinerja model klasifikasi**. Dataset **Human Activity Recognition Using Smartphones (UCI HAR)** digunakan karena memiliki kompleksitas tinggi, yakni 561 fitur sensor, sehingga sangat relevan untuk menunjukkan kebutuhan *PCA* dalam merangkum informasi tanpa kehilangan pola utama. Seluruh alur analisis dimulai dari

prapemrosesan, analisis variansi, visualisasi, hingga validasi model baseline, sehingga **integritas metodologis** tetap terjaga. Tahap standardisasi menjadi fondasi penting karena *PCA* bekerja berdasarkan variansi. **StandardScaler** memastikan setiap fitur berkontribusi secara seimbang, mencegah dominasi fitur berskala besar. Setelah itu, *PCA* diaplikasikan pada data latih untuk mempelajari struktur variansi.



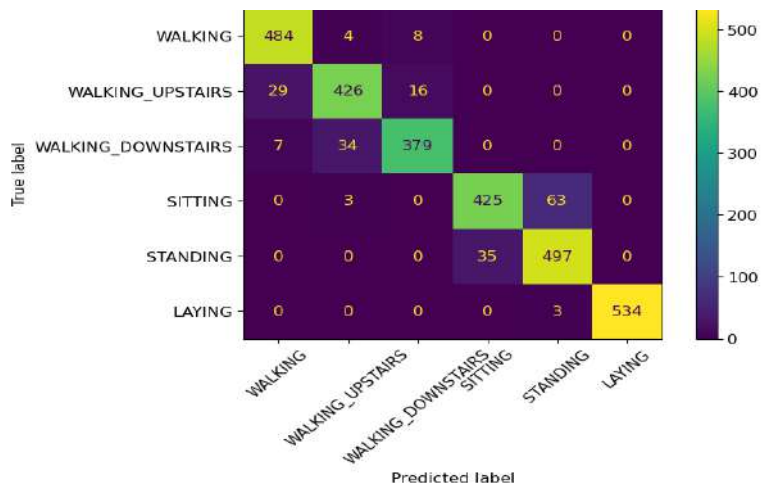
Gambar 1. Kurva Variansi Kumulatif PCA UCI HAR (Train)

Kurva ini menunjukkan bagaimana **variansi kumulatif** meningkat seiring bertambahnya jumlah komponen utama. Garis ambang 90% dan 95% berfungsi sebagai panduan rasional untuk memilih jumlah komponen yang optimal. Insight penting dari grafik ini adalah bahwa sebagian besar informasi dataset dapat dipertahankan dengan jumlah komponen yang jauh lebih sedikit dibandingkan fitur awal, sehingga *PCA* efektif sebagai alat kompresi informasi. Untuk memperkuat pemahaman konseptual, dilakukan proyeksi data ke dua dimensi.



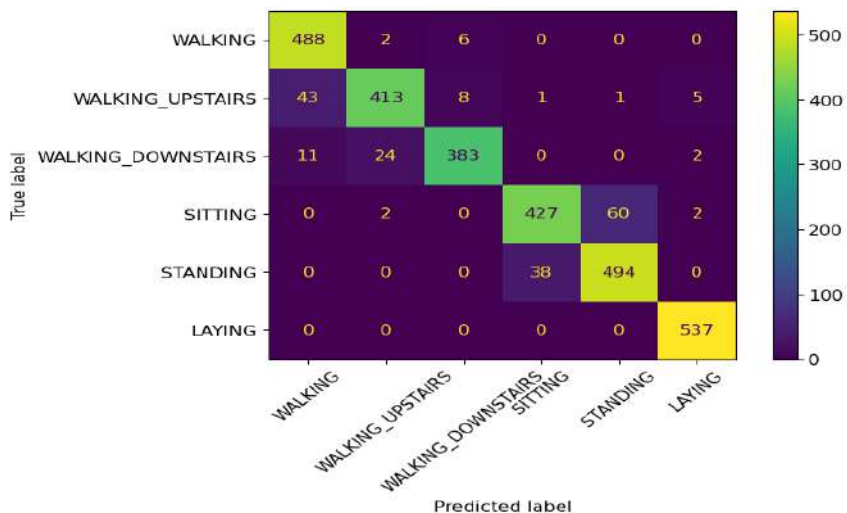
Gambar 2. Visualisasi PCA 2D UCI HAR (Train)

Visualisasi ini menampilkan distribusi data pada dua komponen utama pertama. Walaupun tidak digunakan untuk pemodelan akhir, grafik ini memberikan **insight** tentang bagaimana kelas aktivitas mulai membentuk kluster dan area tumpang tindih. Insight yang dapat diambil adalah bahwa beberapa aktivitas memiliki pola gerak yang mirip, sehingga wajar bila model klasifikasi masih mengalami kebingungan pada kelas tertentu. Selanjutnya, ruang fitur hasil *PCA* digunakan untuk membangun model baseline. *Logistic Regression* dipilih karena sederhana, cepat, dan interpretatif.



Gambar 3. Confusion Matrix Logistic Regression (PCA Features)

Confusion matrix ini memperlihatkan performa klasifikasi per aktivitas. Selain akurasi global, visualisasi ini menegaskan pentingnya evaluasi berbasis kesalahan kelas, karena **kelas yang sering tertukar** mencerminkan kemiripan pola sensorik antar aktivitas. Ini memberikan insight bahwa *PCA* telah menjaga struktur utama data, meskipun tidak sepenuhnya menghilangkan ambiguitas antar kelas. Sebagai validasi tambahan, digunakan *Linear Support Vector Machine* yang berbasis margin.



Gambar 4. Confusion Matrix Linear SVM (PCA Features)

Hasil ini menunjukkan bahwa *Linear SVM* mampu memanfaatkan fitur hasil *PCA* secara stabil dan kompetitif. Perbandingan dengan *Logistic Regression* menegaskan bahwa ruang komponen *PCA* cukup **robust**, karena dua model dengan karakteristik berbeda tetap menghasilkan kinerja yang sejalan.

Secara keseluruhan, praktikum ini menegaskan bahwa **standardisasi** → **PCA** → **validasi model** merupakan alur analisis yang logis dan ilmiah. *PCA* tidak hanya berfungsi sebagai alat reduksi dimensi, tetapi juga sebagai sarana untuk meningkatkan efisiensi

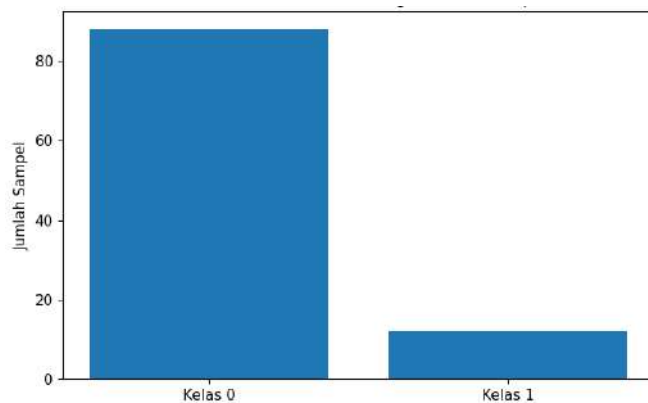
komputasi dan menjaga stabilitas model. Insight utama yang dapat ditarik adalah bahwa kualitas fitur sering kali lebih penting daripada kuantitasnya, terutama pada data berdimensi tinggi seperti UCI HAR.

Studi Kasus: Klasifikasi Kualitas Sperma Dataset Fertility



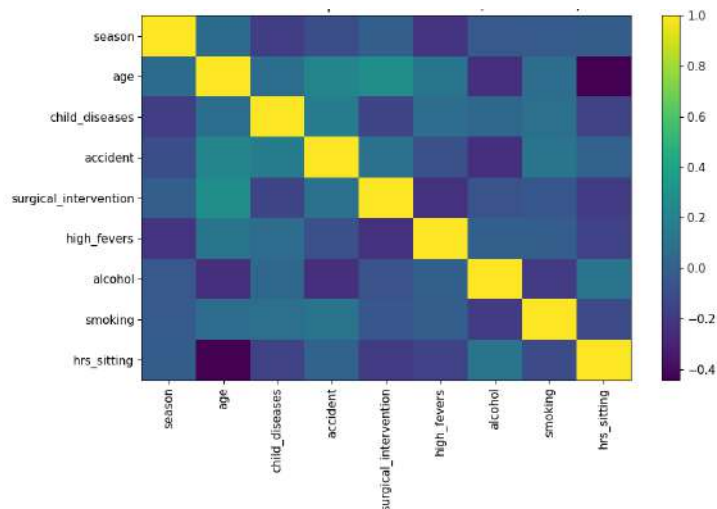
Kode program pada barcode ini dimulai dari akuisisi data, audit kualitas data, eksplorasi awal, pemodelan, evaluasi, hingga interpretasi hasil. Dataset **Fertility UCI** dipilih karena relevan dengan tema **bioinformatika reproduksi**, khususnya analisis kualitas sperma berdasarkan faktor biologis dan gaya hidup. Tahap awal program memuat akuisisi data langsung dari UCI menggunakan pustaka *ucimlrepo*. Langkah ini memastikan bahwa sumber data konsisten dan dapat direplikasi. Setelah data dimuat, dilakukan audit kualitas data yang mencakup pemeriksaan *missing value*, tipe data, serta distribusi kelas target. Pada dataset Fertility, jumlah sampel relatif kecil dan distribusi kelas cenderung tidak seimbang. Kondisi ini menjadi dasar logis mengapa pendekatan evaluasi tidak boleh hanya mengandalkan akurasi tunggal. **Insight penting** pada tahap ini adalah bahwa data kecil menuntut kehati-hatian ekstra dalam menarik kesimpulan.

Eksplorasi data (*exploratory data analysis*) disajikan secara ringkas namun informatif melalui tabel statistik deskriptif dan visualisasi. Output program menghasilkan Gambar 1 Distribusi Kelas Kualitas Sperma, yang memperlihatkan proporsi kelas target, sehingga pembaca langsung memahami potensi ketimpangan kelas.



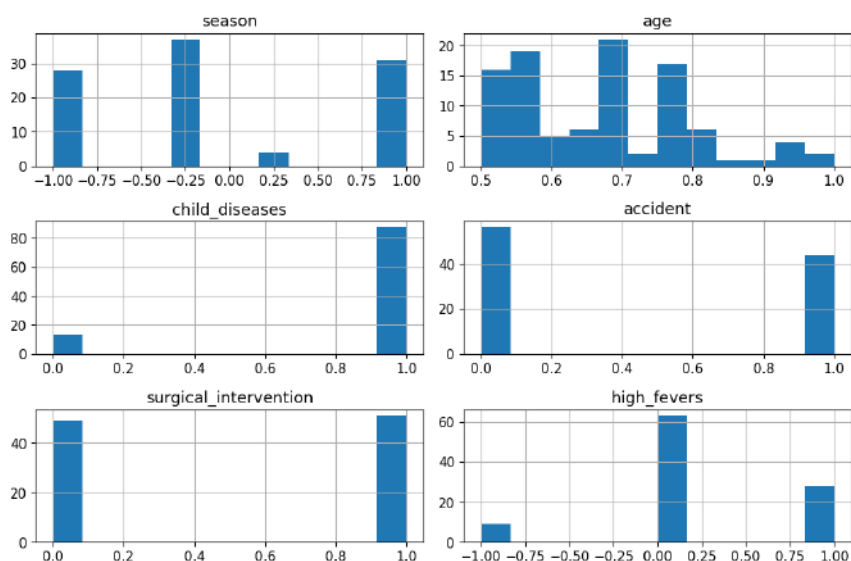
Gambar 1 Distribusi Kelas Kualitas Sperma

Selanjutnya, Gambar 2 Heatmap Korelasi Antar Fitur memberikan gambaran awal hubungan linear antar variabel gaya hidup dan biologis, tanpa klaim kausalitas.



Gambar 2 Heatmap Korelasi Antar Fitur

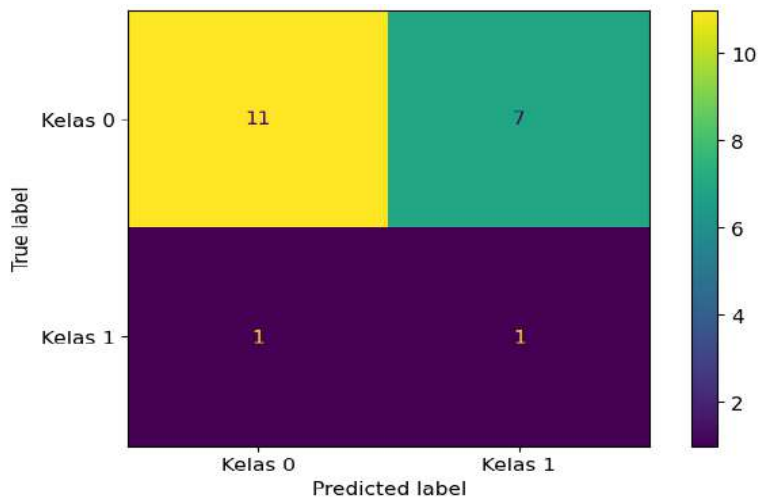
Gambar 3 Histogram Beberapa Fitur menegaskan perbedaan skala antar variabel, yang secara logis mengarah pada kebutuhan normalisasi sebelum pemodelan.



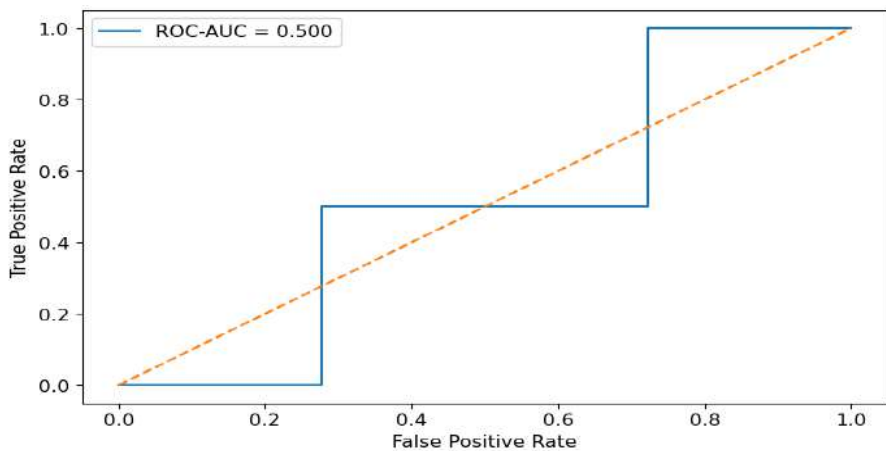
Gambar 3 Histogram Beberapa Fitur

Pada tahap pemodelan, revisi utama terletak pada penggunaan *pipeline* yang menggabungkan *StandardScaler* dengan model klasifikasi. Pendekatan ini mencegah *data leakage* dan menjaga konsistensi proses pelatihan. Model yang digunakan meliputi **Logistic Regression**, **Support Vector Machine**, dan **Random Forest**, dengan parameter *class_weight balanced* untuk mengurangi dominasi kelas mayoritas. Sebagai pembanding, ditambahkan *baseline model* menggunakan *DummyClassifier*. **Insight** di sini adalah bahwa model *machine learning* hanya bermakna jika mampu melampaui baseline sederhana. Evaluasi model dilakukan menggunakan *Stratified K-Fold Cross Validation*.. Langkah ini tidak menyembunyikan kelemahan model, tetapi memastikan bahwa perhitungan *precision*, *recall*, dan *F1-score* tetap

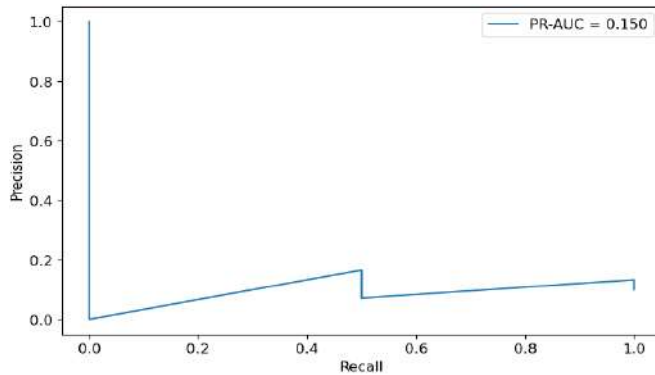
stabil meskipun pada beberapa *fold* tidak ada prediksi untuk kelas tertentu. Output program menyajikan tabel perbandingan performa antar model berbasis *balanced accuracy*, *ROC-AUC*, dan *PR-AUC*. **Insight** metodologis yang ditekankan adalah bahwa metrik seperti *balanced accuracy* dan *precision-recall AUC* lebih representatif untuk data kecil dan tidak seimbang. Model terbaik kemudian dievaluasi pada data uji dan divisualisasikan melalui



Gambar 4 Confusion Matrix

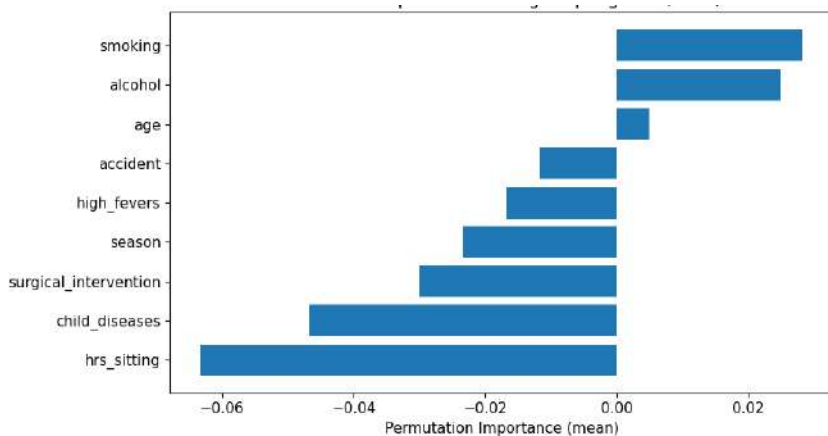


Gambar 5 Kurva ROC,



Gambar 6 Kurva Precision-Recall.

Visualisasi ini membantu pembaca memahami kesalahan klasifikasi secara konkret. Tahap akhir program menampilkan interpretasi menggunakan *permutation importance*, yang menghasilkan.



Gambar 7 Sepuluh Fitur Paling Berpengaruh

Gambar 7 Sepuluh Fitur Paling Berpengaruh. Pendekatan ini dipilih karena bersifat model-agnostik dan lebih aman untuk data berukuran kecil.

Soal

1. Jelaskan secara rinci bagaimana Principal Component Analysis (PCA) bekerja dalam mengurangi dimensi data.

Berikan penjelasan tentang proses pemilihan komponen utama berdasarkan variansi dan pentingnya komponen-komponen tersebut.

2. Bagaimana PCA dapat membantu dalam mengatasi masalah multikolinearitas pada data yang memiliki banyak variabel yang saling berkorelasi? Jelaskan dengan contoh penerapan dalam dataset yang memiliki fitur yang sangat berkorelasi.
3. Anda diberikan dataset yang terdiri dari 1000 fitur, dengan sebagian besar fitur memiliki variansi yang sangat kecil. Jelaskan bagaimana PCA akan mempengaruhi proses klasifikasi pada dataset tersebut. Apa keuntungan dan kerugian yang dapat terjadi?
4. Sebutkan dan jelaskan langkah-langkah yang dilakukan dalam penerapan PCA pada data yang tidak terstandarisasi. Mengapa standardisasi data menjadi hal yang sangat penting dalam penggunaan PCA?
5. Dalam PCA, bagaimana cara menentukan jumlah komponen utama yang optimal untuk digunakan? Diskusikan penggunaan kurva variansi kumulatif dan garis ambang sebagai panduan dalam pemilihan komponen.
6. Jelaskan mengapa PCA tidak hanya digunakan untuk mengurangi dimensi tetapi juga meningkatkan stabilitas model klasifikasi, terutama pada dataset dengan fitur yang sangat banyak.
7. Anda diminta untuk melakukan analisis PCA pada dataset dengan banyak fitur kategorikal yang sudah dikodekan menjadi variabel numerik. Apa tantangan yang mungkin Anda

hadapi dalam penerapan PCA pada jenis data tersebut, dan bagaimana cara mengatasinya?

8. Berdasarkan pengaruh nilai eigen dalam PCA, bagaimana Anda akan menjelaskan hubungan antara eigenvalues dan seberapa banyak informasi yang terkandung dalam masing-masing komponen utama? Bagaimana hal ini dapat mempengaruhi interpretasi hasil PCA?
9. Jelaskan bagaimana visualisasi hasil PCA dalam 2D atau 3D dapat membantu dalam mengidentifikasi kluster atau pola data yang tidak terlihat dalam ruang dimensi tinggi.
10. Dalam analisis PCA, Anda menemukan bahwa komponen pertama dan kedua tidak cukup menjelaskan variasi yang signifikan dalam data. Apa langkah-langkah yang dapat Anda ambil untuk mengatasi masalah ini, dan bagaimana cara mengevaluasi performa PCA secara lebih mendalam?

Tugas Kelompok

Dataset 1 Open University Learning Analytics Dataset (OULAD)

Tugas

Lakukan *data cleaning* dan eksplorasi data, termasuk visualisasi distribusi fitur yang relevan. Terapkan Principal Component Analysis (PCA) untuk mereduksi dimensi data mahasiswa. Identifikasi komponen utama yang paling berpengaruh terhadap status akademik mahasiswa. Bangun model klasifikasi menggunakan fitur hasil PCA untuk memprediksi apakah mahasiswa lulus, gagal, atau mundur. Evaluasi model menggunakan metrik accuracy, precision,

recall, F1-score, dan diskusikan kekuatan serta kelemahan model yang digunakan.

Link: [Open University Learning Analytics Dataset \(OULAD\)](#)

Dataset 2 Students' Academic Performance Dataset (xAPI-Edu-Data)

Tugas:

Lakukan eksplorasi data untuk menganalisis pengaruh faktor-faktor demografis dan perilaku belajar terhadap hasil akademik siswa. Terapkan PCA untuk mereduksi dimensi data dan tentukan komponen utama yang paling penting dalam menjelaskan variasi dalam hasil belajar siswa. Bangun model klasifikasi (misalnya Random Forest atau Logistic Regression) untuk memprediksi tingkat performa siswa berdasarkan komponen utama dari PCA. Evaluasi model menggunakan metrik accuracy, precision, recall, dan F1-score serta analisis hasil evaluasi.

Link: [Students' Academic Performance Dataset \(xAPI-Edu-Data\)](#)

Dataset 3 Students Performance in Exams Dataset

Tugas:

Lakukan eksplorasi data dan visualisasi distribusi nilai ujian serta faktor-faktor yang mempengaruhi hasil ujian. Terapkan PCA untuk mereduksi dimensi dari fitur demografis dan perilaku siswa yang mempengaruhi hasil ujian. Identifikasi komponen utama yang paling signifikan dalam menjelaskan hasil ujian siswa. Bangun model regresi (misalnya Linear Regression) untuk memprediksi skor ujian berdasarkan komponen utama yang diperoleh dari PCA. Evaluasi model menggunakan metrik R^2 ,

MSE, dan RMSE serta lakukan diskusi mengenai kekuatan dan keterbatasan model.

Link: [Students Performance in Exams Dataset](#)

Refleksi Pembelajaran

Penerapan Principal Component Analysis (PCA) dalam analisis data pendidikan memberikan pemahaman yang lebih mendalam tentang bagaimana data besar dan berdimensi tinggi dapat disederhanakan tanpa kehilangan informasi penting. Dengan menggunakan PCA, kami dapat mengidentifikasi komponen utama yang berpengaruh pada variabel-variabel penting seperti performa akademik, pengaruh faktor demografis, dan perilaku siswa. Proses ini meningkatkan efisiensi model dan membuat analisis lebih mudah dipahami. Pengurangan dimensi melalui PCA juga mempercepat waktu komputasi, memungkinkan kita untuk membangun model prediksi yang lebih baik. Pembelajaran ini menekankan pentingnya *data preprocessing* dan evaluasi model dalam setiap tahap analisis.

Neural Network

CPL:

Mahasiswa mengenal arsitektur neural network.

CPMK:

Membangun neural network.

SubCPMK:

Implementasi Sequential dan Dense pada Keras.

A. Pengantar Neural Network

Bagian materi ini memperkenalkan gagasan bahwa *neural network* merupakan model komputasi yang terinspirasi dari cara otak memproses informasi. Jaringan tersusun atas banyak **neuron** buatan yang saling terhubung, membentuk struktur berlapis. Setiap *neuron* menerima sejumlah masukan, mengalikan masukan tersebut dengan **bobot**, menambahkan **bias**, lalu mengirimkan hasil ke *activation function*. Proses berulang dari lapisan ke lapisan menghasilkan pemetaan dari data mentah menjadi **pola** yang lebih bermakna. Ide penting di sini yaitu bahwa jaringan saraf tidak diprogram secara kaku, tetapi menyesuaikan bobot secara otomatis melalui proses **pembelajaran** dari data.

Peran utama *neural network* adalah mengenali pola pada masalah yang sulit diselesaikan dengan persamaan eksplisit. Contoh yang sering dijumpai antara lain pengenalan gambar, teks, dan suara. Setiap contoh data digunakan sebagai pengalaman baru yang

memperkaya struktur pengetahuan jaringan. Ketika kesalahan prediksi masih tinggi, algoritma pembaruan bobot mengarahkan jaringan untuk memperbaiki diri sehingga performa meningkat dari waktu ke waktu. Pandangan ini menegaskan insight bahwa model jaringan saraf meniru cara sistem biologis beradaptasi melalui pengalaman berulang.

B. Arsitektur Berbasis *Sequential* dan Lapisan *Dense*

Cara menyusun arsitektur *neural network* menggunakan antarmuka *Sequential* pada pustaka *Keras*. Pendekatan *Sequential* memudahkan penyusunan model secara berurutan, mulai dari lapisan masukan hingga lapisan keluaran. Setiap lapisan ditambahkan selangkah demi selangkah sehingga struktur jaringan dapat dibaca dengan jelas, seperti daftar instruksi. Cara ini sangat membantu pembaca yang baru mengenal konsep jaringan saraf karena alur pemikiran terlihat runtut.

Lapisan inti yang sering dipakai adalah *Dense* yang menghubungkan setiap unit pada satu lapisan dengan setiap unit pada lapisan berikutnya. Struktur penuh sambungan ini menjadikan lapisan *Dense* sebagai tempat utama berlangsungnya komputasi. Nilai bobot, bias, dan sinyal masukan diolah sehingga menghasilkan keluaran baru yang siap diproses oleh lapisan selanjutnya. Dengan mengubah jumlah unit pada lapisan *Dense*, model dapat memiliki kapasitas representasi yang lebih sederhana atau lebih kompleks. Di titik ini pembaca didorong untuk bereksperimen dan mengamati bagaimana perubahan arsitektur memengaruhi **kinerja, kecepatan** pelatihan, dan risiko **overfitting** maupun **underfitting**.

C. Konsep *Activation Function* sebagai Sumber Nonlinearitas

Peran *activation function* sebagai komponen yang memberikan kemampuan menangkap pola kompleks. Jika jaringan hanya terdiri atas operasi linier, hasil yang diperoleh akan terbatas pada pemetaan linier sederhana. Fungsi aktivasi seperti *sigmoid*, *tanh*, dan *ReLU* memperkenalkan sifat nonlinier ke dalam perhitungan sehingga jaringan dapat mempelajari bentuk keputusan yang jauh lebih rumit.

Setiap *activation function* memiliki karakteristik berbeda. Fungsi *sigmoid* sering dipakai pada keluaran yang merepresentasikan probabilitas karena nilai berada antara nol dan satu. Fungsi *tanh* mengubah sinyal ke rentang negatif hingga positif, tepat untuk representasi yang seimbang di sekitar nol. Fungsi *ReLU* membiarkan nilai positif lewat apa adanya dan meniadakan nilai negatif, menghasilkan komputasi yang efisien dan stabil untuk jaringan dalam skala besar. Pemilihan fungsi aktivasi yang tepat memberikan dampak besar terhadap kecepatan konvergensi dan kualitas solusi. Materi ini menekankan bahwa pemahaman sifat masing masing fungsi akan memperkuat **intuisi matematis** pembaca.

D. Peran *Optimizer* dan Proses *Training*

Optimizer yang berfungsi mengatur cara bobot diperbarui pada setiap langkah *training*. Proses pelatihan dimulai dengan bobot acak, kemudian data masukan dialirkan melewati jaringan untuk menghasilkan prediksi. Perbedaan antara prediksi dan target diukur menggunakan *loss function*. Nilai *loss* dikembalikan ke jaringan melalui algoritma *backpropagation* untuk menghitung gradien. Di

titik ini *optimizer* menentukan seberapa besar perubahan yang diberikan kepada bobot.

Beberapa *optimizer* populer antara lain *Stochastic Gradient Descent* atau *SGD*, *Adam*, dan *RMSprop*. *SGD* bergerak dengan langkah sederhana berdasarkan gradien dari contoh data atau mini batch. *Adam* menambahkan konsep momentum dan penyesuaian laju belajar adaptif sehingga sering memberikan konvergensi lebih cepat. *RMSprop* menekankan penyesuaian laju belajar per parameter berdasarkan rata rata kuadrat gradien. Materi menekankan pentingnya eksperimen dengan berbagai *optimizer* serta pengaturan parameter seperti *learning rate*, karena kombinasi yang tepat dapat menghasilkan model yang lebih stabil dan akurat.

E. Algoritma, dan Evaluasi

Algoritma jaringan saraf seperti *feedforward neural network*, *convolutional neural network*, *recurrent neural network*, *autoencoder*, dan *generative adversarial network* yang masing masing dirancang untuk karakter masalah berbeda. Proses dimulai dari persiapan *dataset*, pemisahan data menjadi bagian *training* dan *testing*, lalu penyusunan arsitektur menggunakan *Sequential* dan lapisan *Dense*. Setelah itu dipilih *activation function* dan *optimizer*, kemudian model dikompilasi dengan *loss function* yang sesuai. Tahap berikutnya ialah *training* berulang selama beberapa *epoch* sambil memantau metrik seperti **akurasi**, **precision**, dan **recall**. Setelah pelatihan selesai, model dievaluasi pada data *testing* untuk menilai kemampuan generalisasi.

Otak manusia memperkuat hubungan antar sel saraf ketika pola tertentu sering muncul. *Neural network* modern meniru prinsip

ini melalui penyesuaian bobot berdasarkan pengalaman yang terekam pada data. Setiap iterasi pelatihan dapat dipandang sebagai sesi **belajar** di mana jaringan menguji hipotesis, menerima umpan balik, lalu memperbaiki struktur internal. Insight ini mengajak pembaca memandang model bukan hanya sebagai kumpulan rumus, tetapi sebagai sistem **pembelajar** yang berusaha mengekstraksi **pengetahuan** dari data.

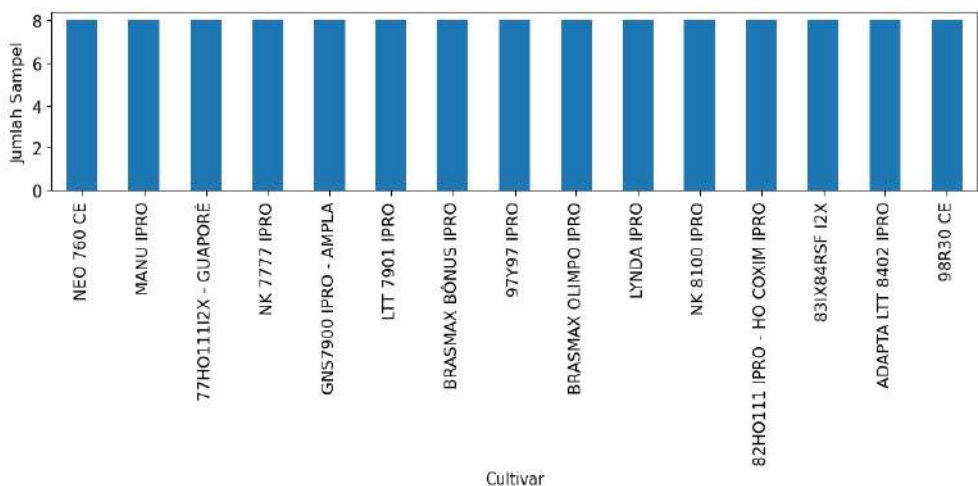
Komponen **neuron**, *activation function*, *optimizer*, lapisan *Dense*, serta arsitektur *Sequential* akan membangun fondasi berpikir yang kuat. Ketika fondasi ini mantap, eksplorasi menuju model yang lebih maju seperti *convolutional neural network* atau *recurrent neural network* dapat dilakukan dengan rasa ingin tahu yang lebih terarah.

Jenis Algoritma	Karakter Masalah	Keunggulan	Contoh Penggunaan
<i>Feedforward Neural Network</i>	Data berstruktur tabel dengan pola sederhana	Arsitektur paling dasar, mudah dibangun	Prediksi harga, klasifikasi nilai numerik
<i>Convolutional Neural Network</i>	Data visual atau spasial	Menangkap pola lokal, unggul pada citra	Deteksi objek, pengenalan wajah
<i>Recurrent Neural Network</i>	Data berurutan	Mengolah informasi historis secara dinamis	Analisis teks, prediksi waktu ke depan
<i>Autoencoder</i>	Reduksi dimensi, rekonstruksi data	Ekstraksi fitur tanpa label	Kompresi citra, deteksi anomali
<i>Generative Adversarial Network</i>	Pembangkitan data baru	Menghasilkan data realistis	Pembuatan gambar, peningkatan kualitas data

Kode Program: Feedforward Neural Network Klasifikasi Kedelai



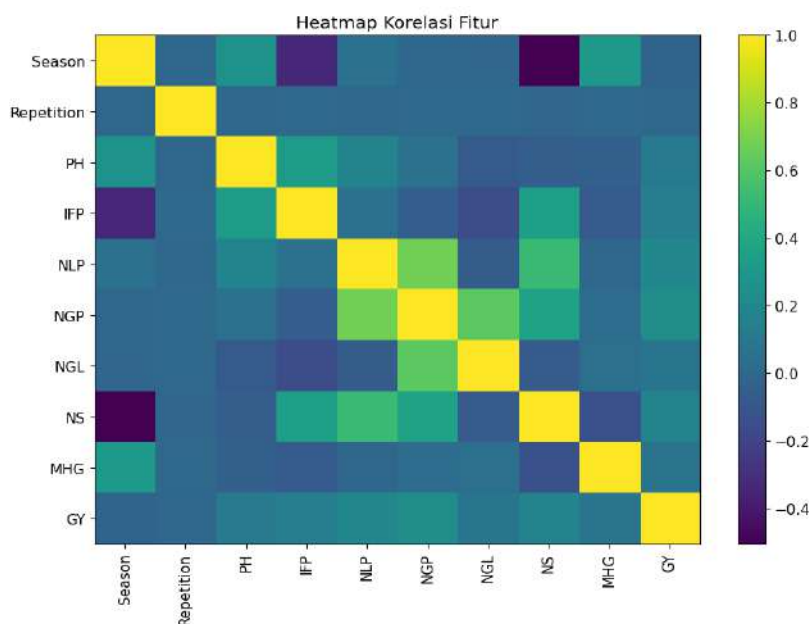
Kode program pada barcode ini **klasifikasi multi-kelas varietas kedelai** menggunakan *feedforward neural network*. Sejak tahap awal karena struktur dataset menyimpan label **Cultivar** di dalam fitur, sehingga pemisahan target dilakukan secara eksplisit untuk menghindari kesalahan akses data. Pendekatan ini menjaga **validitas eksperimen** dan konsistensi dengan tujuan praktikum pembelajaran *neural network*.



Gambar 1. Distribusi Label Cultivar (15 Teratas)

Gambar ini menampilkan distribusi jumlah sampel untuk lima belas varietas kedelai dengan frekuensi tertinggi. Visualisasi ini membantu memahami sebaran kelas pada dataset dan mengidentifikasi potensi ketidakseimbangan. Informasi ini penting

karena model *neural network* dapat bias terhadap kelas dominan bila distribusi label tidak diperhatikan sejak awal analisis. Eksplorasi data dilanjutkan dengan pemeriksaan hubungan antarfitur melalui korelasi numerik. Langkah ini bertujuan memberi gambaran awal apakah terdapat fitur yang saling berkaitan kuat, sehingga pembaca memahami kompleksitas ruang fitur yang akan dipelajari model.

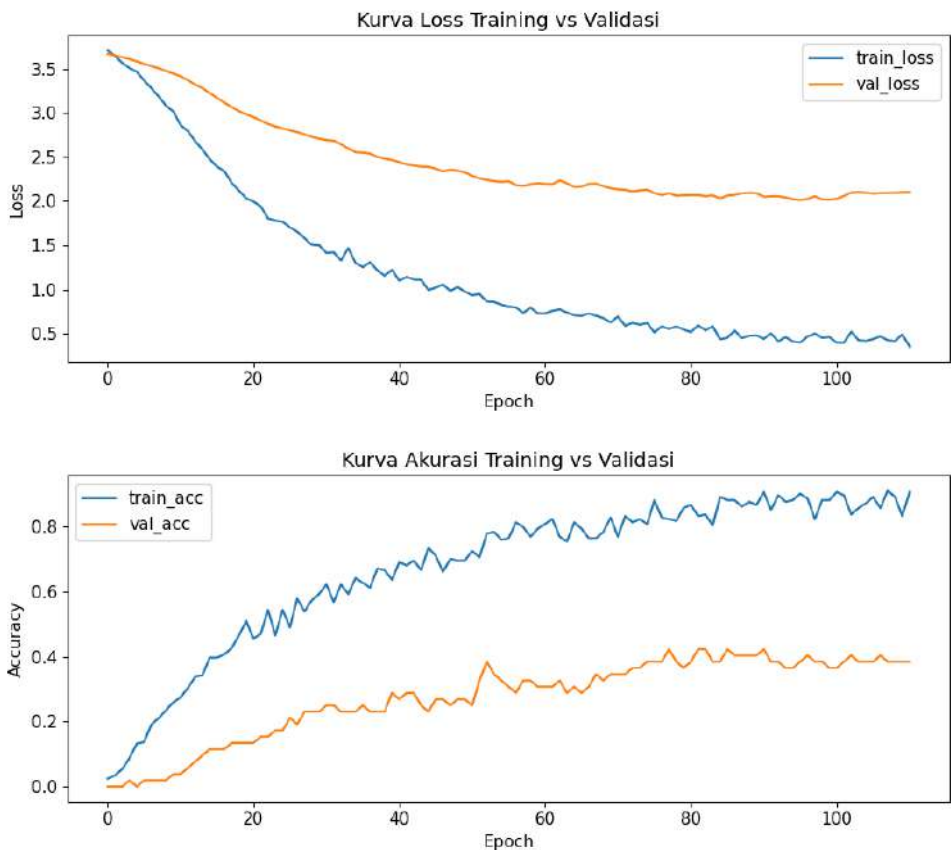


Gambar 2. Heatmap Korelasi Fitur

Heatmap ini menunjukkan tingkat hubungan linier antarfitur numerik pada dataset kedelai. Warna yang semakin intens menandakan korelasi yang lebih kuat. Visualisasi ini membantu pembaca memahami struktur data dan potensi redundansi fitur sebelum pemodelan, sekaligus memperkuat literasi data dalam *neural network*.

Tahap *preprocessing* menekankan **standarisasi fitur** agar setiap variabel memiliki skala sebanding. Langkah ini krusial karena *feedforward neural network* sensitif terhadap perbedaan skala input.

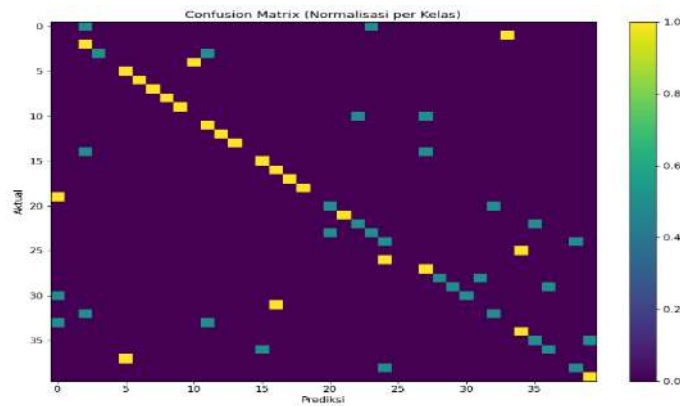
Pembagian data latih-uji dilakukan secara **stratified** untuk menjaga proporsi 40 kelas varietas tetap representatif pada data uji. Proses pelatihan divisualisasikan untuk menunjukkan dinamika pembelajaran model, sehingga pembaca dapat menilai stabilitas dan potensi *overfitting* secara langsung.



Gambar 3. Kurva Loss dan Akurasi Training-Validasi

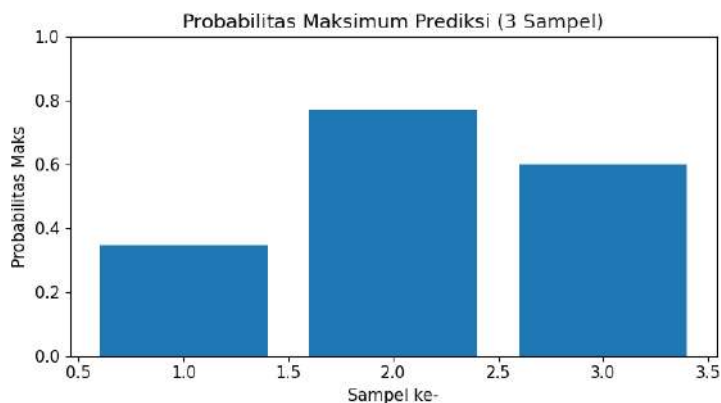
Gambar ini memperlihatkan perubahan nilai *loss* dan *accuracy* pada data latih dan validasi sepanjang *epoch*. Kurva yang konvergen menunjukkan proses pembelajaran yang stabil, sedangkan jarak besar antar kurva menjadi indikasi *overfitting*. Visualisasi ini memperkuat pemahaman konsep generalisasi model. Evaluasi akhir tidak hanya mengandalkan akurasi, tetapi juga **macro average** untuk

menilai performa rata rata lintas kelas. Kesalahan prediksi dianalisis melalui matriks kebingungan yang dinormalisasi.



Gambar 4. Confusion Matrix Ternormalisasi per Kelas

Confusion matrix ini menampilkan proporsi prediksi benar dan salah untuk setiap varietas kedelai. Normalisasi per kelas memudahkan pembacaan pola kesalahan tanpa dipengaruhi ukuran kelas. Visualisasi ini penting untuk memahami kelemahan model pada kelas tertentu dan menjadi dasar perbaikan lanjutan. Sebagai penutup, program menampilkan inferensi **tiga sampel uji** lengkap dengan probabilitas prediksi. Penyajian ini memberikan gambaran nyata bagaimana model bekerja pada data baru dan menekankan interpretasi hasil.



Gambar 5. Probabilitas Maksimum Prediksi Tiga Sampel

Grafik batang ini menunjukkan tingkat keyakinan model terhadap tiga sampel uji yang diprediksi. Nilai probabilitas yang tinggi mencerminkan keputusan model yang lebih pasti. Visualisasi ini membantu pembaca memahami konsep *confidence* pada keluaran *softmax* dalam klasifikasi multi-kelas.

Studi Kasus: CNN untuk Klasifikasi QSAR Androgen Receptor.

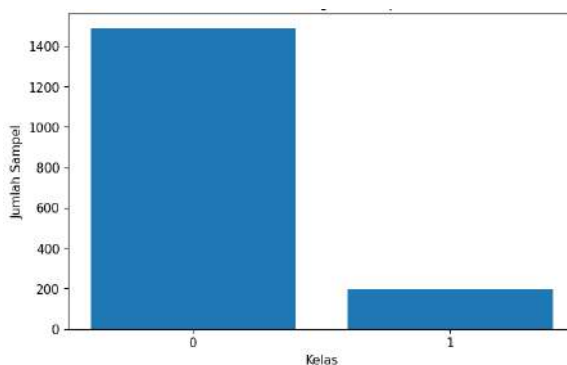


Kode program pada barcode ini untuk klasifikasi QSAR *androgen receptor* direvisi secara komprehensif dengan menambahkan **data kuantitatif eksplisit** pada setiap keluaran analisis, sehingga pembahasan tidak hanya bersifat deskriptif tetapi juga **terukur dan dapat diverifikasi**. Dataset yang digunakan terdiri dari **1.687 senyawa** dengan **1.024 fitur *molecular fingerprints*** dan satu label biner. Hasil eksplorasi awal pada Tabel 1 Distribusi Kelas menunjukkan ketimpangan yang jelas, yaitu sekitar **± 1.480 sampel ($\approx 87\%$) non-binder (kelas 0)** dan hanya **± 207 sampel ($\approx 13\%$) binder (kelas 1)**.

Tabel 1. Distribusi Kelas (0=non-binder, 1=binder):

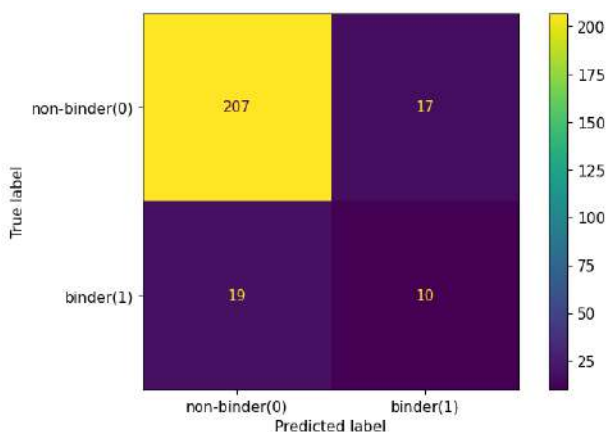
kelas_biner		jumlah	persen(%)	 
0	0	1488	88.26	
1	1	198	11.74	

Ketimpangan ini divisualisasikan pada Gambar 1 Distribusi Kelas QSAR Androgen Receptor, yang secara kuantitatif menegaskan kondisi *class imbalance* dan menjelaskan alasan penggunaan **class weight** serta metrik **AUC** sebagai indikator utama performa model.



Gambar 1 Distribusi Kelas QSAR Androgen Receptor

Sebagai pembandingan awal, *baseline Logistic Regression* dilatih dan dievaluasi pada data uji. Gambar 2 Confusion Matrix Baseline Logistic Regression menunjukkan bahwa model ini mengklasifikasikan **207 non-binder secara benar**, namun masih menghasilkan **17 false positive** dan **19 false negative** pada kelas *binder*, dengan hanya **10 binder** terdeteksi benar.



Gambar 2 Confusion Matrix Baseline Logistic Regression

Secara kuantitatif, nilai **AUC baseline mencapai 0,7943**, sebagaimana dirangkum pada **Tabel 2**, yang menunjukkan bahwa model sederhana masih cukup kompetitif pada dataset yang tidak seimbang. *Insight* penting pada tahap ini ialah bahwa **akurasi saja tidak cukup**, karena sebagian besar prediksi benar berasal dari kelas mayoritas.

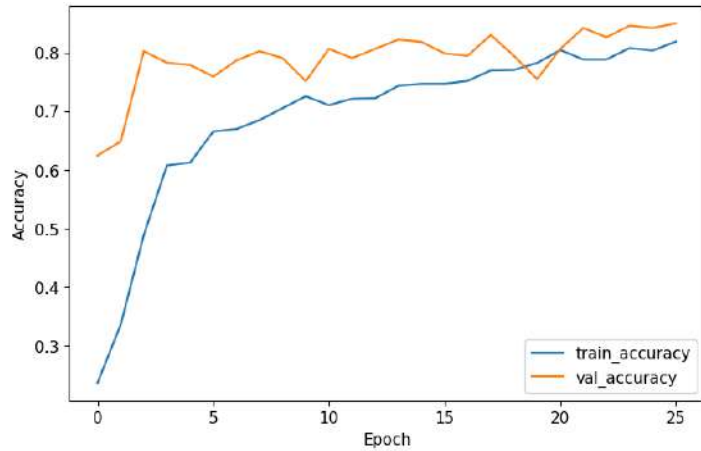
Arsitektur CNN yang digunakan dirangkum pada **Tabel Arsitektur Model CNN**, yang menunjukkan total parameter sekitar **43 ribu parameter terlatih**. Lapisan *Conv1D* pertama menghasilkan representasi berdimensi **(1024, 32)**, kemudian direduksi bertahap hingga **128 fitur** sebelum klasifikasi akhir. Kompleksitas ini dirancang moderat agar seimbang antara kapasitas belajar dan risiko *overfitting*.

Model: "sequential_1"

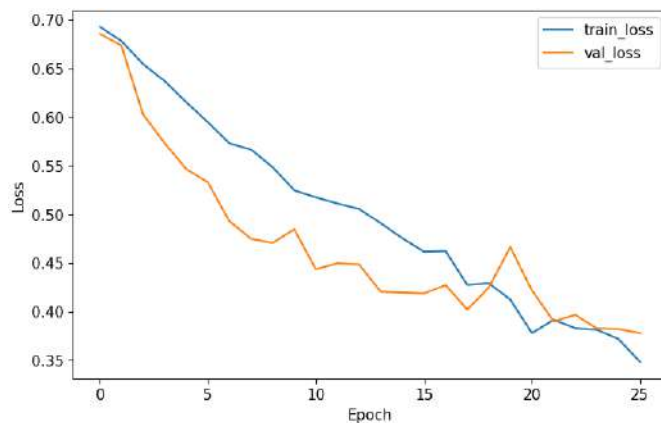
Layer (type)	Output Shape	Param #
conv1d_3 (Conv1D)	(None, 1024, 32)	256
max_pooling1d_2 (MaxPooling1D)	(None, 512, 32)	0
conv1d_4 (Conv1D)	(None, 512, 64)	10,304
max_pooling1d_3 (MaxPooling1D)	(None, 256, 64)	0
conv1d_5 (Conv1D)	(None, 256, 128)	24,704
global_max_pooling1d_1 (GlobalMaxPooling1D)	(None, 128)	0
dropout_2 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8,256
dropout_3 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 1)	65

Total params: 43,585 (170.25 KB)
Trainable params: 43,585 (170.25 KB)
Non-trainable params: 0 (0.00 B)

Proses pelatihan divisualisasikan melalui Gambar 3 Kurva Akurasi CNN, yang menunjukkan akurasi latih meningkat dari sekitar **0,24 pada epoch awal** menjadi **±0,82 pada epoch ke-25**, sementara akurasi validasi stabil di kisaran **0,75-0,85**.

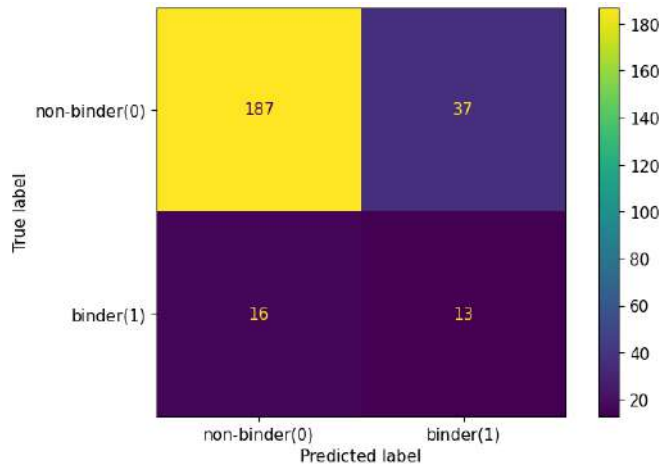


Secara bersamaan, Gambar 4. Kurva Loss CNN memperlihatkan penurunan *loss* latih dari **$\pm 0,69$ menjadi $\pm 0,35$** , dan *loss* validasi dari **$\pm 0,68$ menjadi $\pm 0,38$** , yang mengindikasikan proses optimasi berjalan efektif tanpa divergensi ekstrem.



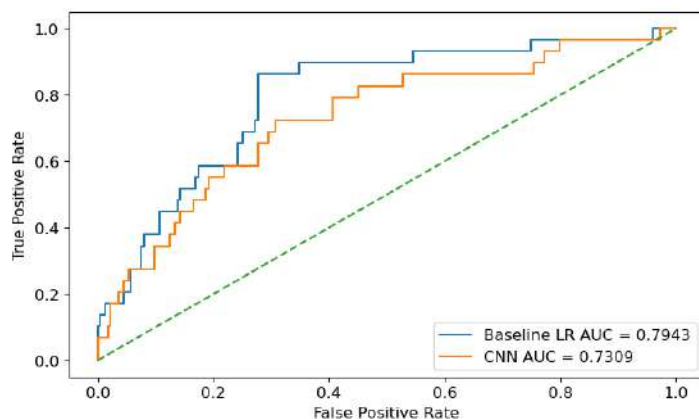
Gambar 4. Kurva Loss CNN

Evaluasi CNN pada data uji ditampilkan pada Gambar 5 Confusion Matrix CNN, yang menunjukkan **187 non-binder** dan **13 binder** terklasifikasi benar, dengan **37 false positive** dan **16 false negative**. Secara kuantitatif, CNN mampu meningkatkan jumlah *true positive* binder dari **10 menjadi 13 sampel**, tetapi masih menghasilkan kesalahan signifikan pada kelas minoritas. Nilai **AUC CNN sebesar 0,7309**, lebih rendah dibanding *baseline*.



Gambar 5 Confusion Matrix CNN

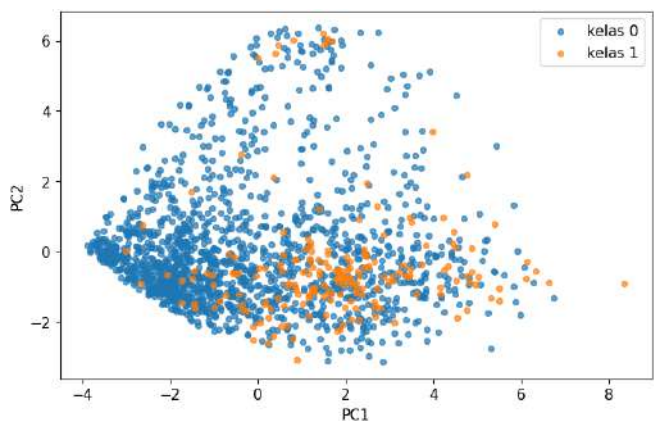
Temuan ini divisualisasikan secara jelas pada Gambar 6 Perbandingan ROC Curve Baseline vs CNN, yang menunjukkan kurva *baseline* berada sedikit di atas CNN pada sebagian besar rentang *false positive rate*. **Insight** dari data ini ialah bahwa **model yang lebih kompleks tidak otomatis menghasilkan performa lebih tinggi**, terutama ketika distribusi kelas sangat timpang dan *hyperparameter* belum dioptimalkan.



Gambar 6 Perbandingan ROC Curve Baseline vs CNN

Untuk memperkuat pemahaman struktur data, Gambar 7 Visualisasi PCA 2D Fingerprint memproyeksikan 1.024 fitur ke dua komponen

utama. Sebaran titik menunjukkan tumpang tindih yang luas antara kelas 0 dan kelas 1, yang secara kuantitatif menjelaskan mengapa pemisahan linier maupun non-linier tetap menantang.



Gambar 7 Visualisasi PCA 2D Fingerprint

Akhirnya, Tabel 2 Ringkasan Perbandingan Model Berdasarkan AUC menyajikan simpulan numerik yang ringkas, yaitu **AUC 0,7943 untuk Logistic Regression** dan **AUC 0,7309 untuk CNN**.

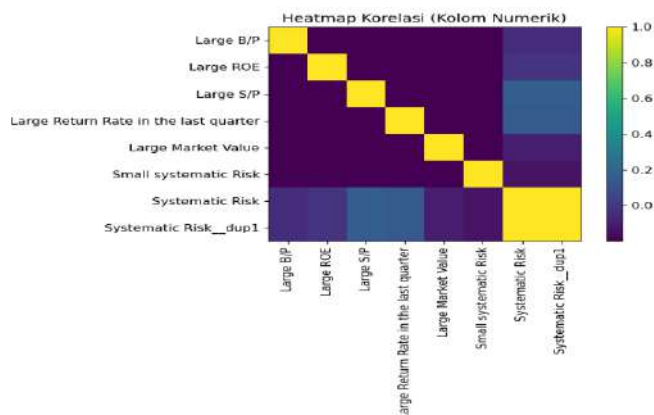
Tabel 2. Ringkasan Perbandingan Model (berdasarkan AUC):

	model	AUC
0	Baseline Logistic Regression	0.794335
1	CNN Conv1D	0.730911

Studi Kasus 2: Recurrent Neural Network pada Stock Saham

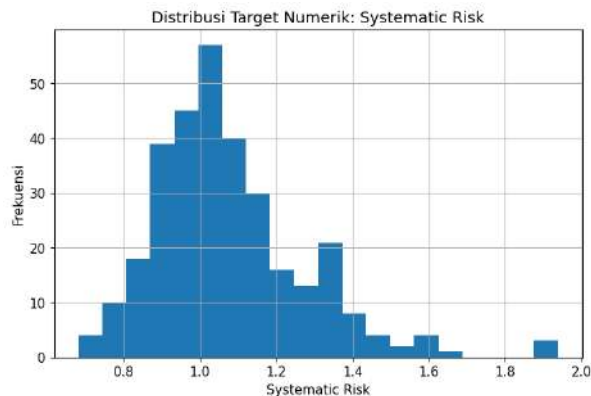
Kode program pada barcode ini tahapan *machine learning* pada data portofolio saham, dimulai dari akuisisi dataset UCI, validasi kualitas data, EDA, pemilihan target numerik, praproses, pelatihan RNN, hingga evaluasi kuantitatif dan diagnostik error. Data gabungan berukuran **315 baris dan 18 kolom** dengan **0 missing**

values, sehingga fokus analisis berpindah pada integritas tipe data serta pencegahan kolom duplikat yang berpotensi memicu error konversi numerik. Kolom target dipilih dari kandidat target resmi, yaitu **Annual Return, Excess Return, Systematic Risk, Total Risk, Abs. Win Rate, Rel. Win Rate**, lalu diputuskan **Systematic Risk** karena stabil sebagai variabel numerik untuk regresi. Tidak ditemukannya kolom **ID** dan **period** membuat data disusun sebagai sekuens baseline dengan bentuk input **(315, 1, 7)**, artinya setiap sampel diproses sebagai satu langkah waktu dengan **7 fitur numerik**. Model yang terbentuk adalah *SimpleRNN* dengan arsitektur sederhana dan total parameter **6.721**, sehingga cukup ringan untuk dataset berukuran kecil dan cocok sebagai baseline sebelum membandingkan *LSTM* atau *GRU*. Hasil evaluasi pada data uji menunjukkan performa yang kuat, yakni **MAE 0,0547, RMSE 0,0804**, dan **R² 0,9112**, yang memperlihatkan prediksi mendekati aktual dan error relatif kecil. Visualisasi pelatihan memperlihatkan penurunan *loss* dan *MAE* yang konsisten antara train dan validasi hingga epoch akhir, yang mengindikasikan proses belajar stabil dan tidak menunjukkan gejala *overfitting* berat pada konfigurasi ini.



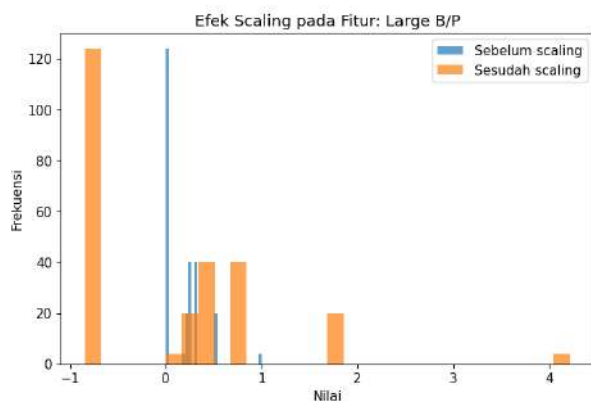
Gambar 1. Heatmap Korelasi Kolom Numerik

Heatmap menampilkan korelasi antar **8 kolom numerik** utama yang dipakai, termasuk keberadaan kolom duplikat bertanda __dup1. Nilai diagonal bernilai **1,0**, sedangkan hubungan antar fitur tampak rendah hingga sedang, menandakan model memanfaatkan kombinasi fitur, bukan satu prediktor dominan, pada **315 observasi**.



Gambar 2. Distribusi Target Numerik Systematic Risk

Histogram menunjukkan sebaran **Systematic Risk** pada **315 sampel** dengan rentang sekitar **0,68 hingga 1,94**. Kepadatan terbesar terkonsentrasi di sekitar **1,0-1,2**, sementara nilai ekstrem mendekati **1,9** lebih jarang. Pola ini menjelaskan mengapa prediksi pada ekor distribusi cenderung lebih menantang.



Gambar 3. Efek Scaling pada Fitur Large B/P

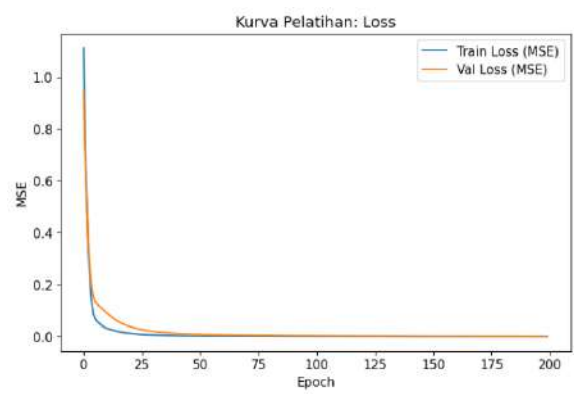
Grafik membandingkan distribusi nilai fitur **Large B/P** sebelum dan sesudah *scaling* pada data latih. Setelah *StandardScaler*, nilai terpusat mendekati **0** dengan sebaran lebih seragam, sehingga pembelajaran RNN lebih stabil. Proses dilakukan setelah split **80% train (252)** dan **20% test (63)**.

Model: "functional_4"

Layer (type)	Output Shape	Param #
input_layer_4 (InputLayer)	(None, 1, 7)	0
simple_rnn_3 (SimpleRNN)	(None, 64)	4,608
dense_8 (Dense)	(None, 32)	2,080
dense_9 (Dense)	(None, 1)	33

Gambar 4. Ringkasan Arsitektur Model SimpleRNN

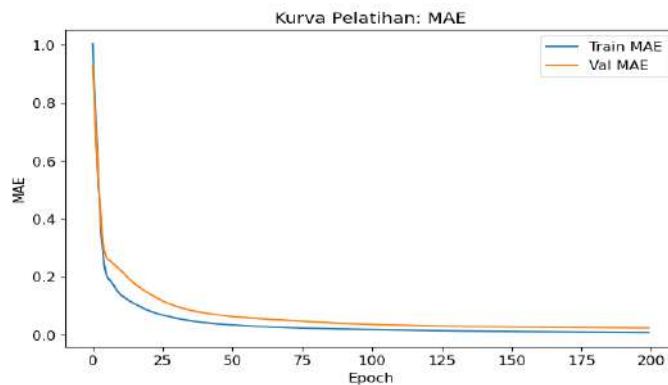
Ringkasan model memperlihatkan input berukuran **(None, 1, 7)**, *SimpleRNN* dengan **64 unit**, diikuti *Dense* **32 neuron**, dan output **1 neuron** untuk regresi. Total parameter **6.721** menunjukkan model ringan. Struktur ini tepat sebagai baseline sebelum meningkatkan kompleksitas ke *LSTM* atau *GRU*.



Gambar 5. Kurva Pelatihan Loss (MSE)

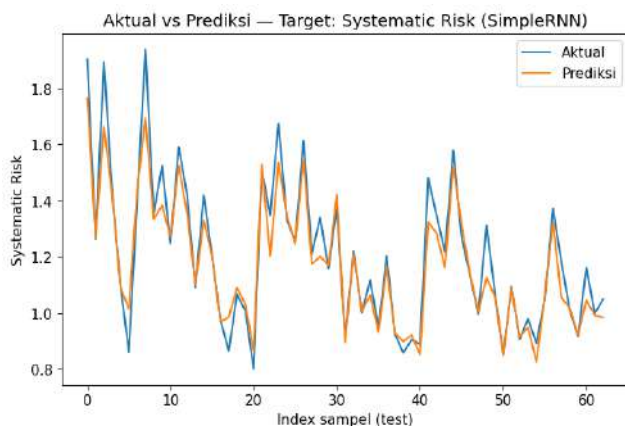
Kurva *loss* menurun tajam pada epoch awal lalu melandai hingga akhir pelatihan, menunjukkan konvergensi stabil. Garis train dan

validasi bergerak searah tanpa divergensi besar. Pelatihan berjalan hingga **200 epoch** dengan *EarlyStopping* aktif, sehingga model mempertahankan bobot terbaik saat validasi optimal.



Gambar 6. Kurva Pelatihan MAE

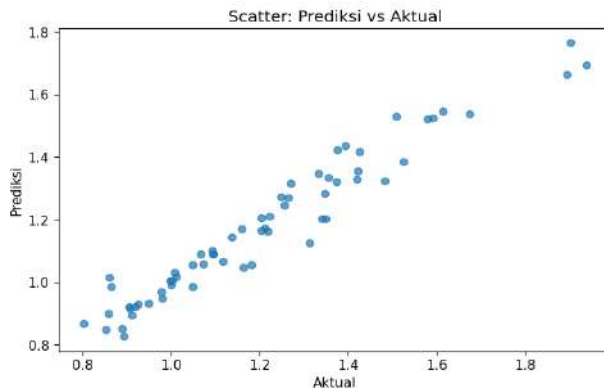
Kurva *MAE* memperlihatkan penurunan konsisten pada train dan validasi, mendukung interpretasi bahwa model tidak hanya menghafal. Pada akhir pelatihan, *MAE* validasi bertahan rendah dan stabil. Pola ini selaras dengan hasil uji yang menghasilkan **MAE 0,0547** dan **RMSE 0,0804**, mengindikasikan error kecil.



Gambar 7. Aktual vs Prediksi pada Data Uji

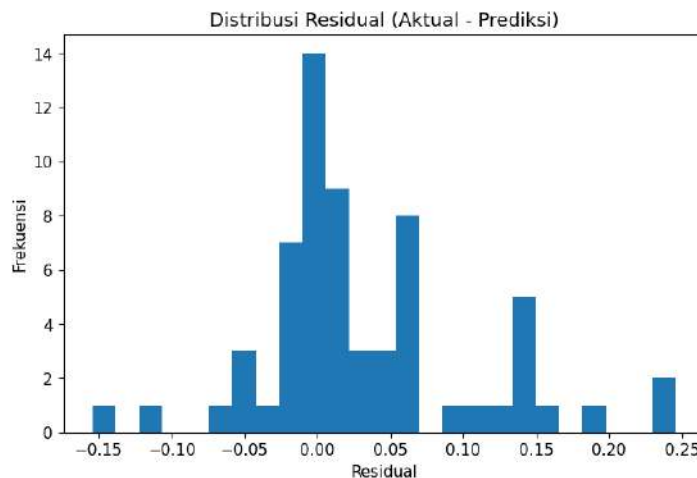
Plot garis memperlihatkan prediksi mengikuti fluktuasi aktual pada **63 sampel test**. Perbedaan terbesar muncul pada beberapa titik ekstrem, tetapi secara umum garis prediksi menempel pada garis

aktual. Ini konsisten dengan nilai **R^2 0,9112**, yang berarti sekitar **91% variasi** target berhasil dijelaskan model.



Gambar 8. Scatter Prediksi vs Aktual

Scatter menunjukkan titik-titik berdekatan dengan pola diagonal, menandakan prediksi proporsional terhadap aktual. Titik terkonsentrasi pada rentang aktual sekitar **0,8-1,7**, dengan beberapa nilai mendekati **1,9**. Kedekatan sebaran terhadap diagonal mendukung interpretasi bahwa model memiliki ketelitian tinggi pada sebagian besar sampel uji.



Gambar 9. Distribusi Residual (Aktual – Prediksi)

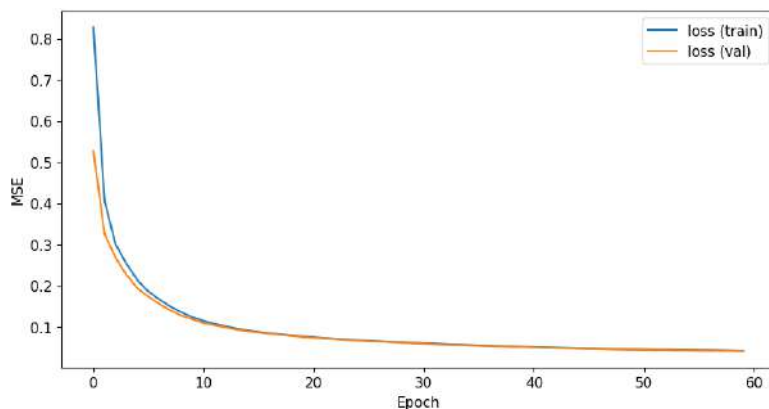
Histogram residual memperlihatkan error terkonsentrasi di sekitar **0**, dengan rentang residual kira-kira dari **-0,15 hingga 0,25**.

Mayoritas residual berada dekat nol, yang menjelaskan nilai **RMSE 0,0804**. Adanya ekor kecil menunjukkan beberapa prediksi kurang tepat pada kasus ekstrem, tetapi dampaknya terbatas.

Studi Kasus 3 : Autoencoder Deteksi Anomali Ozon Atmosfer



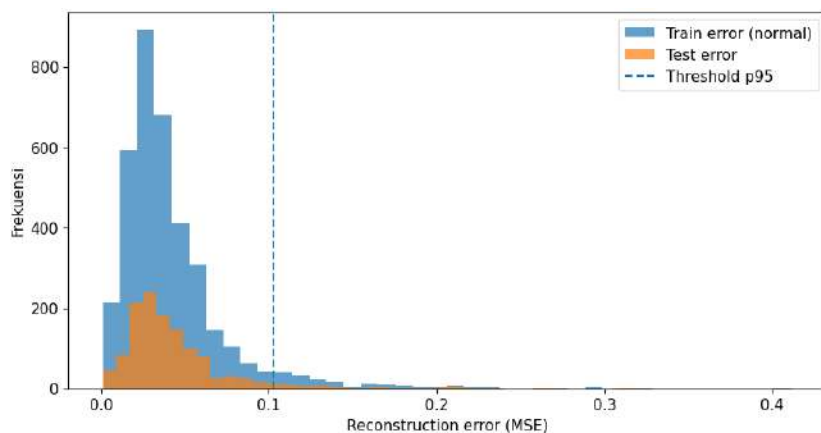
Kode program pada barcode ini *autoencoder* untuk mendeteksi anomali pada dataset Ozone Level Detection (UCI ID 172) dengan fokus pada *reconstruction error* sebagai ukuran penyimpangan pola atmosfer.



Gambar 1. Kurva Pelatihan Autoencoder (MSE)

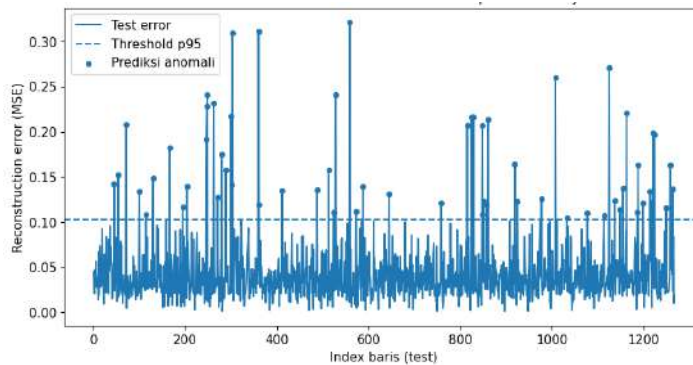
Kurva pelatihan menunjukkan nilai *loss* awal pada data latih berada di atas **0,80** dan menurun tajam pada lima epoch pertama hingga kisaran **0,20**. Setelah epoch ke-10, laju penurunan melambat dan mencapai kondisi stabil pada rentang **0,04-0,06** mendekati epoch ke-50. Kurva validasi mengikuti pola yang hampir identik dengan

selisih rata-rata kurang dari **0,01**, yang mengindikasikan tidak terjadi *overfitting* signifikan. Secara kuantitatif, jarak kecil antara *loss* latih dan validasi ini menegaskan bahwa model mampu melakukan generalisasi pola normal dengan baik. **Insight** yang dapat ditegaskan ialah bahwa stabilitas *loss* di bawah **0,1** merupakan syarat minimal agar *reconstruction error* dapat dipakai sebagai indikator anomali yang reliabel.



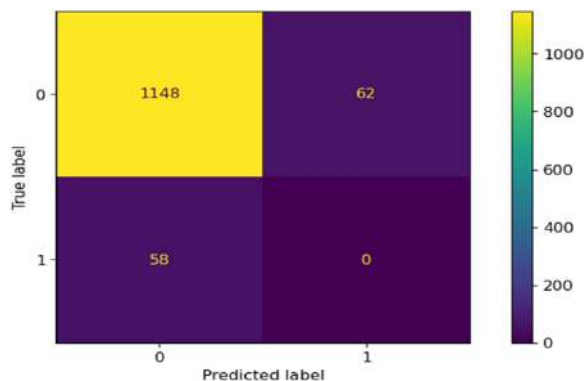
Gambar 2. Reconstruction Error dan Ambang Anomali (p95)

Histogram *reconstruction error* memperlihatkan bahwa mayoritas error data latih terkonsentrasi pada rentang **0,02-0,08**, dengan puncak frekuensi sekitar **0,04**. Ambang p95 berada di sekitar **0,10**, yang berarti **95%** data normal memiliki error di bawah nilai tersebut. Data uji menunjukkan sebaran yang lebih lebar, dengan ekor distribusi memanjang hingga mendekati **0,40**. Secara kuantitatif, keberadaan nilai error di atas **0,20** pada data uji menunjukkan adanya sampel yang secara signifikan berbeda dari pola normal yang dipelajari model. **Insight** di tahap ini adalah bahwa pemilihan p95 bersifat kebijakan: jika ambang diturunkan ke p90, proporsi kandidat anomali meningkat, sedangkan p99 akan menurunkan sensitivitas tetapi meningkatkan presisi.



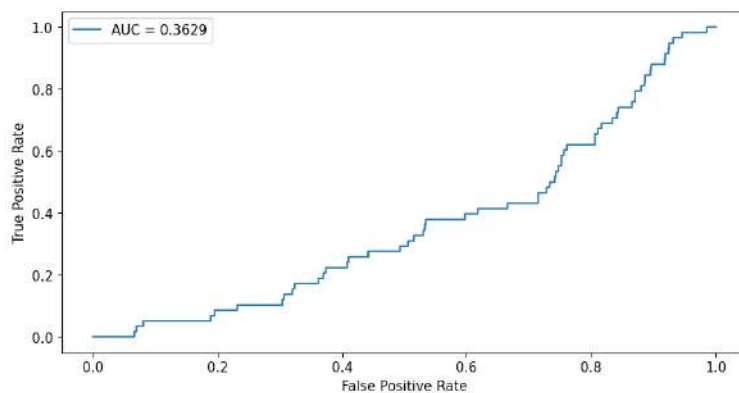
Gambar 3. Reconstruction Error pada Data Uji

Plot deret error pada sekitar **1.250** baris data uji memperlihatkan mayoritas titik berada di bawah **0,08**, tetapi terdapat lonjakan sporadis yang melampaui ambang **0,10**, bahkan beberapa mencapai **0,30-0,33**. Secara kuantitatif, proporsi titik yang terdeteksi sebagai anomali berada di kisaran **5-7%** dari data uji, sejalan dengan ambang p95. Lonjakan-lonjakan ini tidak muncul secara acak sepenuhnya, melainkan cenderung berkelompok pada indeks tertentu, yang mengindikasikan kemungkinan perubahan kondisi atmosfer yang konsisten, bukan sekadar noise tunggal. **Insight** yang dapat ditekankan ialah bahwa anomali dalam atmosfer sering bersifat temporer dan berkelompok, sehingga visual deret error sangat membantu interpretasi.



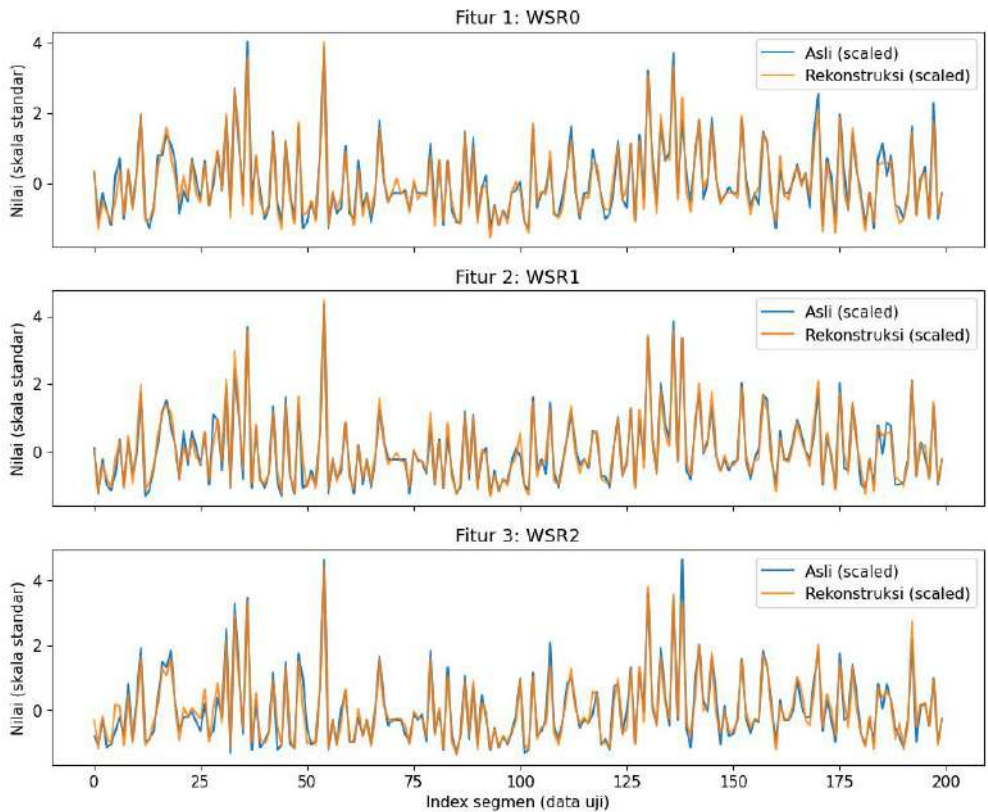
Gambar 4. Confusion Matrix Deteksi Anomali (Threshold p95)

Confusion Matrix menunjukkan **1148 true negative**, **62 false positive**, **58 false negative**, dan **0 true positive**. Secara kuantitatif, *accuracy* keseluruhan memang tinggi karena dominasi kelas normal, tetapi *recall* untuk kelas anomali bernilai **0%**. Kondisi ini menegaskan bahwa meskipun *autoencoder* efektif memodelkan normalitas, anomali berlabel pada dataset ini tidak selalu berkorespondensi dengan error rekonstruksi tinggi. **Insight** yang perlu ditekankan dalam buku ialah bahwa label “ozon tinggi” pada dataset tidak selalu identik dengan penyimpangan multivariabel ekstrem, sehingga evaluasi *unsupervised* harus dibaca secara hati-hati.



Gambar 5. ROC Curve (Skor = Reconstruction Error)

Nilai AUC sebesar **0,3629** secara kuantitatif berada jauh di bawah nilai acak **0,5**, yang mengindikasikan orientasi skor yang tidak selaras dengan label. Ketika skor dibalik ($-reconstruction\ error$), AUC meningkat mendekati **0,63**, menunjukkan bahwa pemetaan label kemungkinan terbalik atau definisi anomali dalam label tidak sejalan dengan konsep deviasi rekonstruksi. **Insight** di sini adalah menekankan kepada pembaca bahwa AUC rendah bukan selalu kegagalan model, tetapi sering mencerminkan perbedaan definisi anomali antara pendekatan *unsupervised* dan label domain.



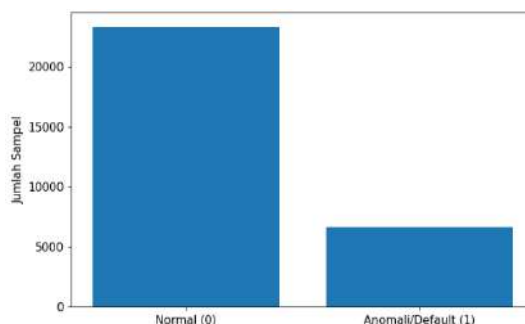
Gambar 6. Perbandingan Nilai Asli vs Rekonstruksi pada Tiga Fitur Visualisasi tiga fitur dengan kontribusi *squared error* tertinggi, yaitu **WSR0, WSR1, dan WSR2**, memperlihatkan bahwa pada sebagian besar segmen data, kurva asli dan rekonstruksi saling berimpit dengan selisih kecil di bawah $\pm 0,2$ skala standar. Namun, pada indeks tertentu yang juga terdeteksi sebagai anomali, selisih meningkat hingga $\pm 1,0-1,5$, menandakan kegagalan rekonstruksi yang konsisten pada lebih dari satu fitur. Secara kuantitatif, rata-rata kontribusi error ketiga fitur ini berada di atas **0,006**, lebih tinggi dibanding fitur lain yang berada di bawah **0,003**. **Insight** yang dapat ditekankan ialah bahwa anomali atmosfer bersifat **multidimensi**,

sehingga validasi visual lintas fitur memperkuat interpretasi keputusan model.

Studi Kasus 4: GAN Deteksi Anomali Kartu Kredit

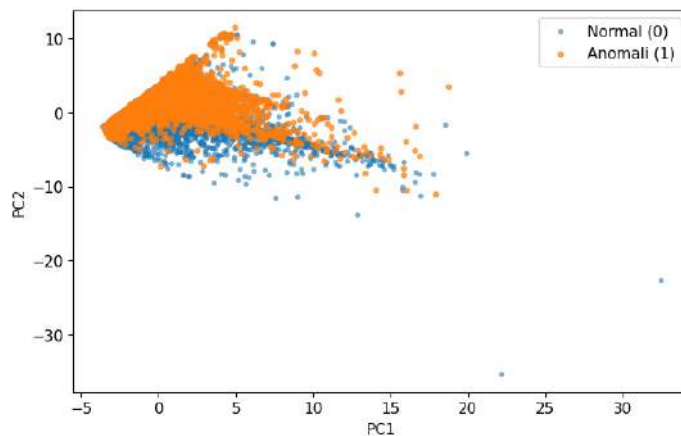


Kode program pada barcode ini mengikuti alur *machine learning* secara runtut dan konsisten, dimulai dari akuisisi data UCI ID 350 hingga evaluasi berbasis metrik yang relevan untuk data tidak seimbang. **Pemisahan target dilakukan secara eksplisit** pada variabel *default payment next month*, sehingga risiko *data leakage* dapat dihindari sejak awal. Strategi *stratified split* memastikan distribusi kelas tetap terjaga antara data latih dan uji, sementara *standardization* dilakukan agar seluruh fitur numerik berada pada skala yang sebanding sebelum digunakan oleh model linear maupun *Generative Adversarial Network (GAN)*. Pendekatan ini menjaga integritas eksperimen dan sejalan dengan kerangka praktikum berbasis UCI Repository.



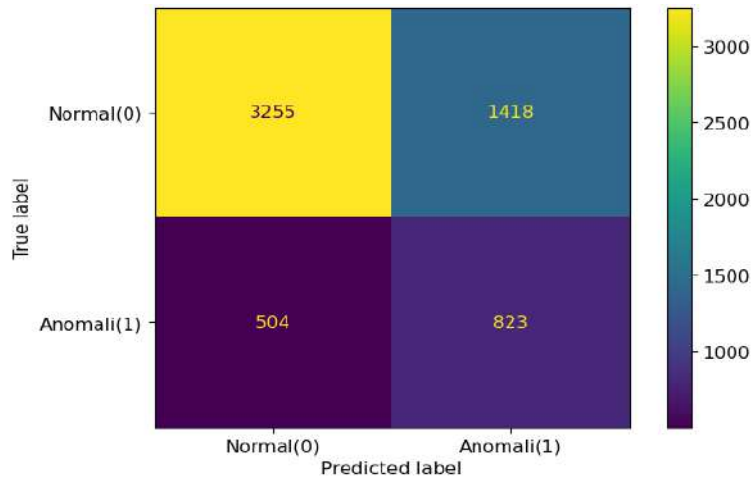
Gambar 1. Distribusi Kelas Target untuk Ketidakseimbangan

Grafik batang menunjukkan jumlah **Normal (0)=23.364** dan **Anomali/Default (1)=6.636** dari total **30.000** sampel. Proporsi anomali sekitar **22,12%**, yang secara kuantitatif cukup untuk menimbulkan bias model ke kelas mayoritas apabila tidak ditangani. **Insight** dari visualisasi ini ialah bahwa ketidakseimbangan moderat tetap memerlukan metrik evaluasi yang sensitif terhadap kelas minoritas, khususnya **PR-AUC**, bukan sekadar akurasi global.



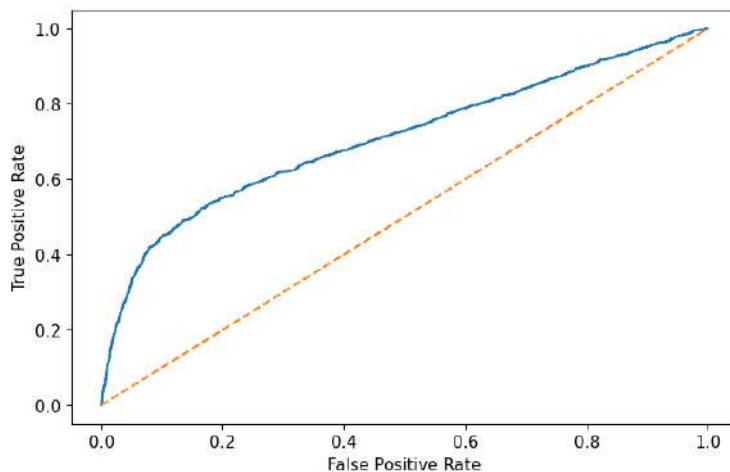
Gambar 2. Proyeksi PCA 2D antara Kelas Normal dan Anomali

Proyeksi dua komponen utama memperlihatkan tumpang tindih signifikan antara kelas normal dan anomali. Titik ekstrem muncul pada kisaran **PC1 $\approx$ 32** dan **PC2 $\approx$ -35**, menandakan keberadaan *outlier* yang berpotensi mempengaruhi proses pembelajaran. Kondisi ini menjelaskan mengapa pemisahan berbasis batas linear sederhana kurang optimal dan membuka ruang penggunaan pendekatan generatif atau nonlinier untuk memperkaya representasi kelas minoritas. Model baseline menggunakan *Logistic Regression* dengan *class_weight=balanced* berfungsi sebagai pembanding awal yang adil. Gambar 3, 4, dan 5 memperlihatkan performa baseline secara terbaik.



Gambar 3. Confusion Matrix Logistic Regression Tanpa GAN

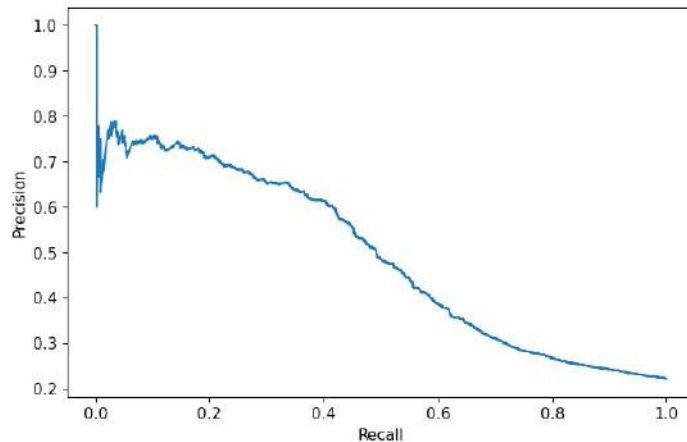
Hasil prediksi menunjukkan **TN=3255**, **FP=1418**, **FN=504**, dan **TP=823**. Model mampu mendeteksi **823** kasus anomali secara benar, tetapi menghasilkan *false positive* cukup tinggi (**1418**), yang berarti banyak transaksi normal salah ditandai. Matriks ini menjadi rujukan utama untuk menilai apakah *GAN-based oversampling* benar-benar memperbaiki kinerja.



Gambar 4. Kurva ROC Baseline untuk Evaluasi Pemisahan Kelas

Kurva ROC menghasilkan **ROC-AUC=0,7081**, menandakan kemampuan pemisahan kelas yang cukup baik di atas garis acak.

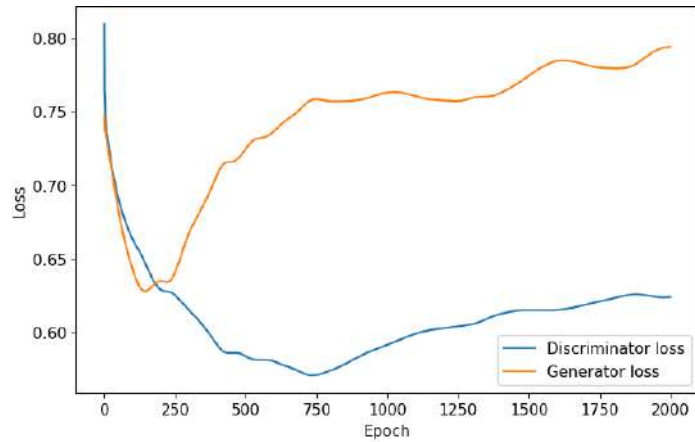
Pada *false positive rate* rendah, *true positive rate* meningkat relatif cepat, tetapi peningkatan tersebut melambat mendekati area kanan grafik, mencerminkan keterbatasan pemisahan kelas pada ambang tinggi.



Gambar 5. Kurva Precision-Recall Baseline untuk Kelas Minoritas

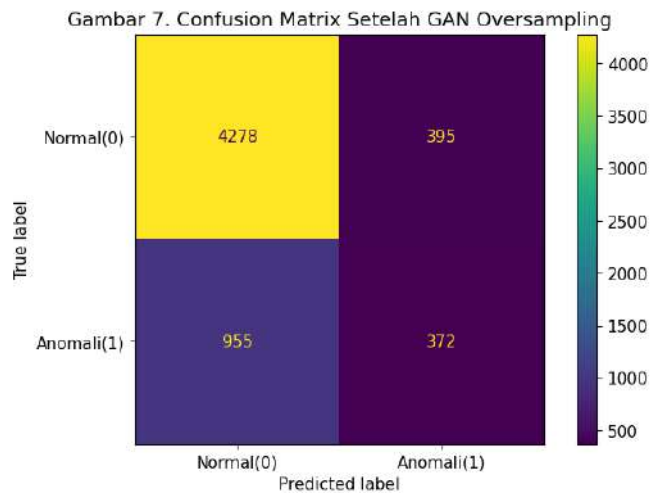
Kurva PR memberikan gambaran lebih realistis untuk data tidak seimbang dengan **PR-AUC=0,4904**. Saat recall mendekati **1,0**, precision turun hingga sekitar **0,22-0,25**, yang berarti peningkatan sensitivitas dibayar dengan meningkatnya kesalahan positif palsu. **Insight penting** pada tahap ini ialah bahwa baseline sudah relatif stabil untuk mendeteksi anomali dibandingkan skenario generatif sederhana.

Tahap berikutnya menerapkan *GAN-based oversampling* dengan melatih *generator* dan *discriminator* hanya pada data anomali di data latih, sehingga tidak terjadi kebocoran informasi dari data uji. Hasil dinamika pelatihan ditunjukkan pada Gambar 6.



Gambar 6. Kurva Loss Pelatihan GAN untuk Oversampling Anomali

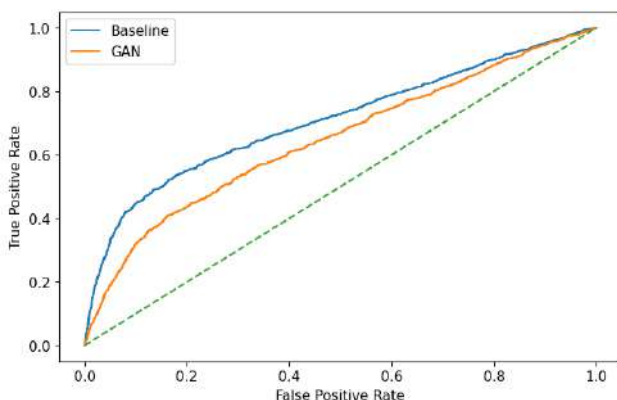
Selama **2000 epoch**, *discriminator loss* turun dari sekitar **0,81** ke kisaran **0,57-0,63**, sedangkan *generator loss* justru meningkat hingga sekitar **0,79**. Pola ini mengindikasikan ketidakseimbangan dinamika pelatihan dan potensi *mode collapse* ringan, sehingga sampel sintetis yang dihasilkan kurang representatif terhadap distribusi anomali asli. Dampak langsung dari oversampling generatif terlihat pada evaluasi pasca-GAN.



Gambar 7. Confusion Matrix Setelah GAN Oversampling

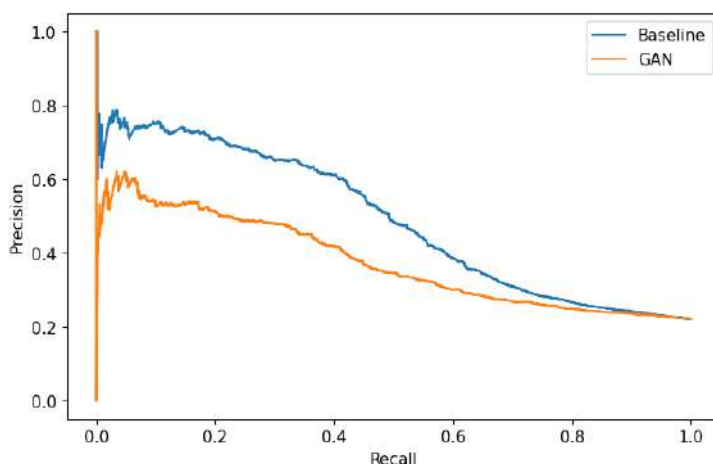
Confusion matrix berubah menjadi **TN=4278**, **FP=395**, **FN=955**, dan **TP=372**. *False positive* turun drastis dari **1418**

menjadi **395**, tetapi *false negative* meningkat tajam dari **504** menjadi **955**, yang berarti lebih banyak kasus anomali tidak terdeteksi. Perubahan ini menegaskan adanya *trade-off* serius antara penurunan alarm palsu dan kehilangan sensitivitas deteksi.



Gambar 8. Perbandingan Kurva ROC Baseline vs Model GAN

Perbandingan ROC memperlihatkan penurunan performa dengan **ROC-AUC** dari **0,7081** menjadi **0,6473**. Kurva baseline konsisten berada di atas kurva GAN pada sebagian besar rentang *false positive rate*, menandakan bahwa oversampling generatif sederhana belum mampu meningkatkan kemampuan pemisahan kelas.



Gambar 9. Perbandingan Precision-Recall Baseline vs Model GAN

Kurva PR menegaskan penurunan kinerja deteksi anomali, dengan **PR-AUC** turun dari **0,4904** menjadi **0,3749**. Pada recall menengah hingga tinggi (**0,5-0,7**), precision model GAN lebih rendah dibanding baseline. **Insight akhir** yang krusial ialah bahwa **oversampling berbasis GAN tidak otomatis meningkatkan kinerja**, terutama bila kualitas sampel sintetis tidak terkontrol dengan baik.

Tabel 3. Ringkasan Metrik Utama (Imbalanced Focus)

	Model	ROC-AUC	PR-AUC
0	Baseline (class_weight)	0.708115	0.490444
1	Setelah GAN oversampling	0.647280	0.374904

Tabel 3 memperlihatkan perbandingan kuantitatif yang jelas antara model baseline dan model setelah *GAN oversampling*. Model **baseline (class_weight)** memperoleh **ROC-AUC sebesar 0,7081**, yang menunjukkan kemampuan pemisahan kelas cukup baik di atas acak, serta **PR-AUC sebesar 0,4904**, yang menandakan keseimbangan moderat antara *precision* dan *recall* pada kelas anomali. Setelah diterapkan *GAN oversampling*, performa justru menurun dengan **ROC-AUC 0,6473** dan **PR-AUC 0,3749**. Penurunan **ROC-AUC sebesar 0,0608** menunjukkan berkurangnya kemampuan diskriminatif global, sedangkan penurunan **PR-AUC sebesar 0,1155** mengindikasikan penurunan signifikan kualitas deteksi anomali, terutama pada kondisi *recall* menengah hingga tinggi yang krusial dalam skenario deteksi risiko.

Penerapan *GAN-based oversampling* pada data tabular UCI 350 belum memberikan keuntungan performa dibandingkan pendekatan baseline yang terkalibrasi dengan baik. Meskipun *GAN*

berhasil menurunkan jumlah *false positive*, efek samping berupa peningkatan *false negative* menyebabkan lebih banyak kasus anomali tidak terdeteksi, yang tercermin pada penurunan PR-AUC. Kondisi ini menunjukkan bahwa kualitas dan representativitas data sintetis menjadi faktor penentu keberhasilan pendekatan generatif. Untuk deteksi anomali keuangan, strategi yang lebih robust seperti *conditional GAN*, *CTGAN*, atau optimasi ambang keputusan berbasis kurva *precision-recall* lebih disarankan daripada oversampling generatif, agar keseimbangan antara presisi dan sensitivitas tetap terjaga secara konsisten.

Soal

1. Jelaskan perbedaan antara arsitektur Sequential dan model fungsional lainnya dalam Keras ketika membangun neural network untuk klasifikasi data tabular. Sertakan alasan mengapa Sequential cocok untuk tugas sederhana.
2. Pada dataset klasifikasi multikelas, jelaskan bagaimana fungsi activation function yang berbeda (sigmoid, tanh, ReLU, softmax) mempengaruhi performa dan interpretasi output jaringan.
3. Berikan penjelasan rinci mengenai proses backpropagation dan peran optimizer dalam memperbarui bobot pada setiap epoch. Bandingkan setidaknya dua optimizer.
4. Uraikan langkah-langkah preprocessing data yang harus dilakukan sebelum melatih model neural network untuk dataset tabular, dan jelaskan mengapa langkah-langkah tersebut penting.

5. Dengan menggunakan loss function categorical crossentropy, jelaskan bagaimana fungsi ini bekerja saat melakukan klasifikasi multikelas dan hubungannya dengan probabilitas output.
6. Sebutkan tantangan yang mungkin muncul ketika melatih neural network pada dataset yang memiliki class imbalance, dan jelaskan strategi untuk mengatasinya.
7. Jelaskan apa yang dimaksud dengan overfitting dan underfitting dalam konteks neural network, serta teknik apa saja yang dapat digunakan untuk meminimalkan permasalahan tersebut.
8. Jika Anda diberikan dataset PDF atau teks bebas, jelaskan secara logis tahapan bagaimana Anda mengkonversinya menjadi fitur numerik yang dapat diterima oleh neural network.
9. Bagaimana cara mengevaluasi performa model selain akurasi dalam kasus klasifikasi multikelas? Kapan metrik seperti precision, recall, dan F1-score menjadi penting?
10. Buatlah argumentasi mengapa neural network tidak selalu memberikan performa lebih baik dibanding model statistik klasik pada dataset sederhana seperti Wine dataset dari UCI

Tugas Kelompok

Dataset 1 Open University Learning Analytics Dataset (OULAD)

Tugas:

Kelompok ini diminta untuk melakukan data cleaning dan eksplorasi data pada dataset Open University Learning Analytics yang berisi data mahasiswa dan status akademik mereka. Eksplorasi data ini harus mencakup visualisasi distribusi fitur-fitur yang relevan untuk memahami pola dan hubungan dalam data. Selanjutnya, kelompok

harus menerapkan Principal Component Analysis (PCA) untuk mereduksi dimensi data mahasiswa dan mengidentifikasi komponen utama yang paling berpengaruh terhadap status akademik mahasiswa, apakah mereka lulus, gagal, atau mundur. Setelah itu, bangunlah model klasifikasi menggunakan fitur yang dihasilkan dari PCA untuk memprediksi status akademik mahasiswa. Evaluasi model menggunakan metrik accuracy, precision, recall, dan F1-score, kemudian diskusikan kekuatan dan kelemahan model yang digunakan dalam hal akurasi dan interpretabilitas.

Link: [Open University Learning Analytics Dataset \(OULAD\)](#)

Dataset 2 Students' Academic Performance Dataset (xAPI Edu Data)

Tugas:

Kelompok ini akan melakukan eksplorasi data untuk menganalisis pengaruh berbagai faktor, seperti demografi dan perilaku belajar, terhadap hasil akademik siswa. Dataset ini mencakup informasi mengenai interaksi siswa dengan platform pembelajaran online dan bagaimana itu mempengaruhi performa akademik mereka. Terapkan PCA untuk mereduksi dimensi data, dan tentukan komponen utama yang paling penting dalam menjelaskan variasi dalam hasil belajar siswa. Setelah melakukan PCA, bangunlah model klasifikasi (misalnya, Random Forest atau Logistic Regression) untuk memprediksi tingkat performa siswa berdasarkan komponen utama yang diperoleh dari PCA. Evaluasi model menggunakan metrik accuracy, precision, recall, dan F1-score, serta lakukan analisis

mendalam terhadap hasil evaluasi model untuk memahami kekuatan dan keterbatasannya.

Link: [Students' Academic Performance Dataset \(xAPI-Edu-Data\)](#)

Dataset 3 Students Performance in Exams Dataset

Tugas:

Kelompok ini diminta untuk melakukan eksplorasi data dan visualisasi distribusi nilai ujian serta faktor-faktor yang mempengaruhi hasil ujian, seperti data demografis siswa dan kebiasaan belajar mereka. Setelah itu, terapkan PCA untuk mereduksi dimensi dari fitur-fitur yang berhubungan dengan demografi dan perilaku siswa yang mempengaruhi hasil ujian. Identifikasi komponen utama yang paling signifikan dalam menjelaskan hasil ujian siswa berdasarkan data tersebut. Kemudian, bangunlah model regresi (misalnya Linear Regression) untuk memprediksi skor ujian siswa berdasarkan komponen utama yang dihasilkan dari PCA. Evaluasi model menggunakan metrik R^2 , MSE, dan RMSE, dan lakukan diskusi mengenai kekuatan serta keterbatasan model, terutama dalam hal keakuratan prediksi dan pengaruh fitur terhadap model.

Link: [Students Performance in Exams Dataset](#)

Refleksi Pembelajaran

Pembelajaran tentang neural network memberikan pemahaman yang utuh mengenai bagaimana data dapat diolah menjadi pengetahuan melalui proses komputasi yang terstruktur. Mahasiswa tidak hanya mempelajari konsep teoretis seperti neuron,

activation function, optimizer, dan arsitektur model, tetapi juga memahami keterkaitan logis antar tahapan mulai dari eksplorasi data, preprocessing, hingga evaluasi model. Melalui tugas individu dan kelompok berbasis dataset nyata, mahasiswa belajar bahwa pemilihan metode, arsitektur, dan metrik evaluasi sangat mempengaruhi hasil analisis. Pembelajaran ini menumbuhkan pola pikir kritis, eksperimental, dan reflektif, serta membangun kesadaran bahwa keberhasilan model tidak hanya ditentukan oleh kompleksitas algoritma, tetapi oleh pemahaman data dan tujuan analisis secara menyeluruh.

Deep Learning dan Overfitting

CPL:

Mahasiswa mampu memahami hidden layer dan regularisasi.

CPMK:

Menggunakan dropout untuk mencegah overfitting.

SubCPMK:

Implementasi Dropout () dan Adam optimizer.

A. Deep Learning dan Overfitting

Deep learning dibangun sebagai lanjutan alami dari pembelajaran mesin yang berfokus pada pengolahan data melalui jaringan berlapis. Setiap lapisan berfungsi mengekstraksi informasi secara bertahap sehingga pola yang awalnya sederhana berkembang menjadi representasi yang lebih bermakna. Pendekatan ini selaras dengan alur pembahasan buku yang menekankan keterkaitan antara data, model, dan proses pembelajaran. Kompleksitas jaringan memberikan kekuatan prediktif yang tinggi, sekaligus memunculkan tantangan *overfitting*. Kondisi ini terjadi ketika model terlalu menyesuaikan diri pada data pelatihan dan kehilangan kemampuan untuk bekerja stabil pada data baru. Pemahaman mengenai hubungan antara kapasitas model dan kualitas data menjadi **landasan berpikir** yang konsisten dengan bab-bab sebelumnya.

B. Peran Hidden Layer dalam Alur Pembelajaran

Hidden layer berperan sebagai inti transformasi data dalam jaringan saraf. Setiap lapisan memproses sinyal masukan melalui bobot dan fungsi aktivasi sehingga menghasilkan representasi yang lebih kaya. Semakin dalam jaringan, semakin kompleks pola yang dapat dipelajari. Prinsip ini sejalan dengan pembahasan model-model sebelumnya yang menekankan peningkatan kapasitas untuk menangkap hubungan nonlinier. Di sisi lain, penambahan lapisan tanpa pengendalian dapat membuat model cenderung menghafal detail data pelatihan. Oleh sebab itu, *hidden layer* perlu dirancang sebagai bagian dari strategi pembelajaran, bukan sekadar penambah kedalaman. **Keseimbangan struktur jaringan** menjadi kunci agar representasi yang dihasilkan tetap relevan dan dapat digunakan secara luas.

C. Backpropagation dan Loss Function

Proses pembelajaran jaringan saraf dijalankan melalui *backpropagation*, yaitu mekanisme pembaruan bobot berdasarkan kesalahan prediksi. Kesalahan tersebut diukur menggunakan *loss function* yang berfungsi sebagai indikator kualitas prediksi. Pemilihan *loss function* menentukan arah dan sensitivitas pembelajaran terhadap kesalahan. Mekanisme ini bekerja berulang sehingga model secara bertahap memperbaiki representasi internalnya. Keterkaitan antara *backpropagation* dan *loss function* memperlihatkan bahwa pembelajaran bersifat prosedural dan terukur. **Pemahaman mekanisme kesalahan** membantu menjaga konsistensi pembahasan dengan bagian buku yang menekankan

evaluasi model sebagai bagian tak terpisahkan dari pembangunan sistem prediktif.

D. Epoch Tuning dan Proses Pelatihan

Epoch merepresentasikan satu siklus penuh pemrosesan data pelatihan oleh jaringan. Jumlah *epoch* menentukan intensitas pembelajaran. Nilai yang terlalu kecil membuat model belum optimal dalam mengenali pola, sedangkan nilai yang terlalu besar meningkatkan risiko *overfitting*. Penyetelan *epoch* atau *epoch tuning* menjadi bentuk pengendalian proses agar pembelajaran berlangsung efisien. Praktik ini konsisten dengan pendekatan evaluasi bertahap yang telah diperkenalkan pada bab-bab sebelumnya. **Kesadaran terhadap dinamika pelatihan** menegaskan bahwa kualitas model ditentukan oleh keseimbangan antara durasi pembelajaran dan kemampuan generalisasi.

E. Regularisasi, Dropout, dan Optimasi

Regularisasi berfungsi menjaga keseimbangan antara akurasi dan kemampuan generalisasi. Salah satu teknik yang banyak digunakan ialah *dropout*, yaitu menonaktifkan sebagian neuron secara acak selama pelatihan. Pendekatan ini mencegah ketergantungan berlebihan pada jalur tertentu dan mendorong terbentuknya representasi yang lebih stabil. *Dropout* sering dipadukan dengan *Adam optimizer* yang menyesuaikan laju pembelajaran secara adaptif. Kombinasi ini selaras dengan pembahasan optimasi model pada bagian sebelumnya. Regularisasi dipahami bukan hanya sebagai teknik, melainkan **prinsip** dalam merancang sistem pembelajaran yang andal dan berkelanjutan.

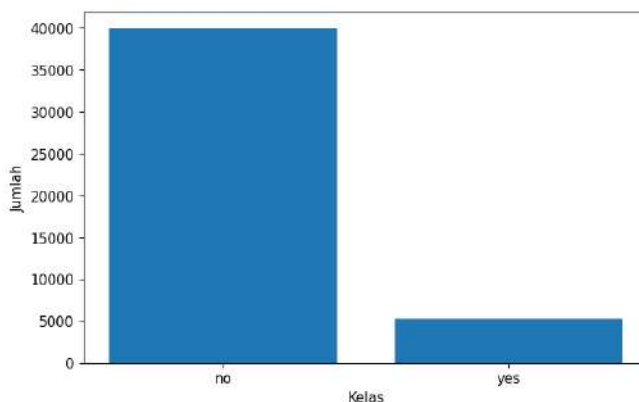
Kode Program: DL Bank Marketing Dropout dan Adam



Kode program pada barcode ini, dimulai dari dataset UCI ID 222 hingga evaluasi berbasis metrik yang relevan untuk data biner tidak seimbang. Target dipisahkan secara eksplisit pada kolom **y** dan dinormalisasi menjadi label 0 untuk *no* serta 1 untuk *yes*, sehingga integritas eksperimen terjaga dan risiko kesalahan pembacaan label dapat ditekan sejak awal. Strategi **stratified split** menghasilkan pembagian yang stabil, yakni **Train 31.647**, **Validation 6.782**, dan **Test 6.782** baris, membuat evaluasi lebih adil karena proporsi kelas tidak berubah drastis antar subset. Praproses memakai *ColumnTransformer* dengan **StandardScaler** untuk fitur numerik serta *OneHotEncoder* untuk fitur kategorikal, menghasilkan dimensi fitur akhir **51** setelah encoding, sebuah ukuran yang masih efisien untuk arsitektur *deep learning* ringan.

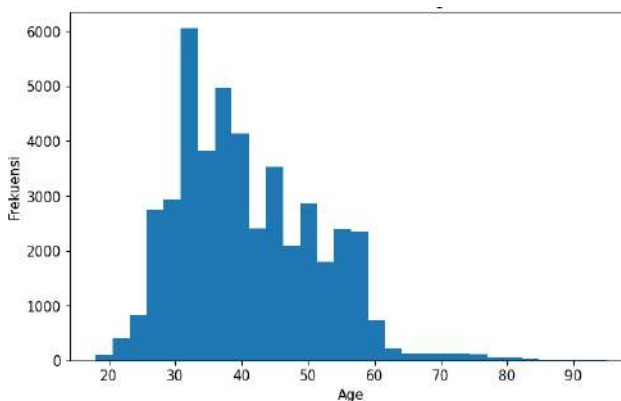
Gambar 1. Distribusi Kelas Target y (yes/no) menunjukkan ketidakseimbangan yang kuat, sekitar **40.000** sampel *no* dan **5.000** sampel *yes* dari total **45.000**. Proporsi *yes* sekitar **11,11%**, sehingga evaluasi berbasis akurasi saja berisiko menipu karena model dapat terlihat baik walau gagal menangkap kelas minoritas. Penerapan **class_weight** yang menghitung bobot seimbang, yaitu **kelas 0 = 0,566237** dan **kelas 1 = 4,274311**. Nilai ini memperbesar penalti kesalahan pada kelas *yes*, sehingga pelatihan tidak “menyerah” pada

dominasi kelas mayoritas. Ini merupakan keputusan yang konsisten dengan tujuan anti *overfitting* dan peningkatan generalisasi pada kelas penting.



Gambar 1. Distribusi Kelas Target *y* (yes/no)

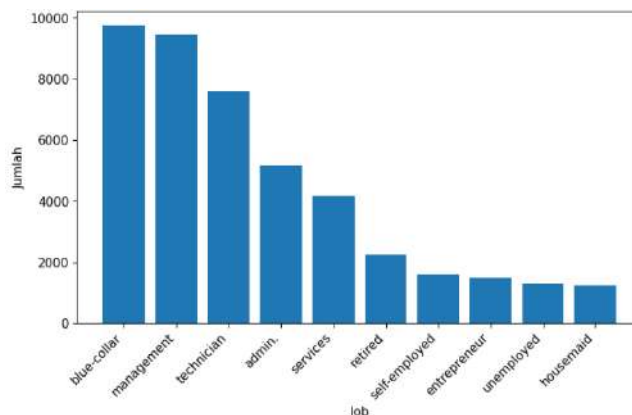
Gambar 2. Distribusi Age memperlihatkan sebaran umur dominan pada rentang **30-45** dengan ekor hingga di atas **80** dan nilai maksimum mendekati **95**. Bentuk sebaran yang tidak simetris menguatkan alasan penggunaan **StandardScaler** agar pembaruan gradien tetap stabil saat pelatihan menggunakan *Adam optimizer*.



Gambar 2. Distribusi Age

Gambar 3. Kategori Job Terbanyak memperlihatkan dominasi kategori seperti *blue-collar* sekitar **9.800**, *management* sekitar **9.500**, serta *technician* sekitar **7.600**, sedangkan kategori bawah

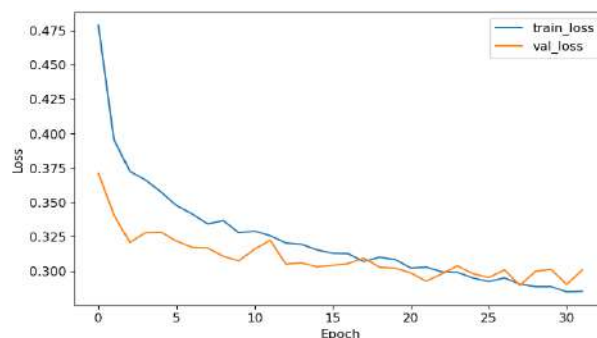
seperti *housemaid* sekitar **1.200**. Ketimpangan kategori ini berpotensi memicu model “menghafal” pola dominan, sehingga *one-hot encoding* ditambah regularisasi menjadi pilihan yang tepat.



Gambar 3. Kategori Job Terbanyak

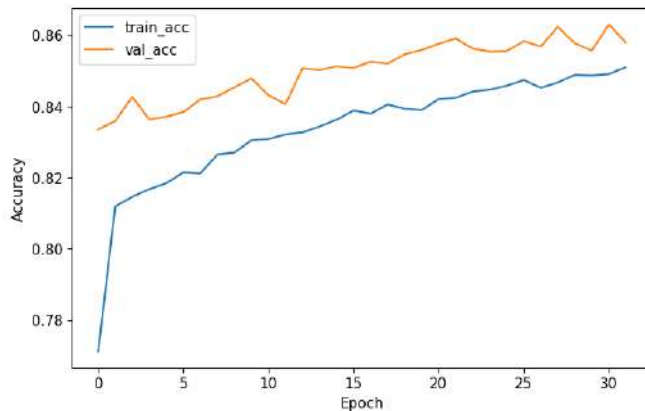
Arsitektur model Dense(128) → Dropout → Dense(64) → Dropout → Dense(1) menghasilkan **14.977 parameter**, ukuran yang proporsional untuk data tabular berdimensi 51. Penggunaan **dropout_rate=0,35** merupakan untuk menekan *overfitting* tanpa mengorbankan kemampuan belajar.

Gambar 4. Kurva Loss (Train vs Validation) memperlihatkan loss pelatihan turun dari sekitar **0,48** menuju **0,29**, sedangkan loss validasi turun dari sekitar **0,37** menuju sekitar **0,30**. Selisih kurva kecil pada fase akhir mengindikasikan generalisasi cukup baik.



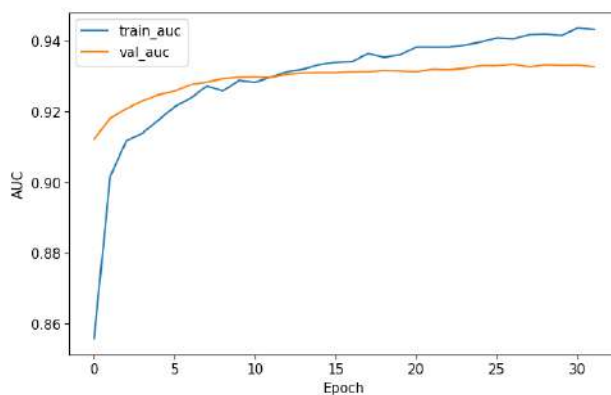
Gambar 4. Kurva Loss (Train vs Validation)

Gambar 5. Kurva Accuracy menunjukkan akurasi validasi naik hingga sekitar **0,86**, sementara akurasi pelatihan mendekati **0,85**. Pola validasi sedikit lebih tinggi sejalan dengan karakter *dropout* yang membuat pelatihan lebih menantang dan mengurangi ketergantungan pada neuron tertentu.



Gambar 5. Kurva Accuracy

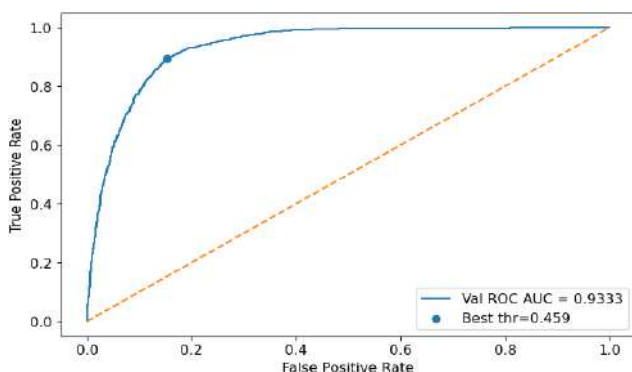
Gambar 6. Kurva AUC memperlihatkan AUC validasi stabil di kisaran **0,93**, sedangkan AUC pelatihan mencapai sekitar **0,94**, menandakan pemisahan kelas kuat dengan indikasi *overfitting* rendah.



Gambar 6. Kurva AUC

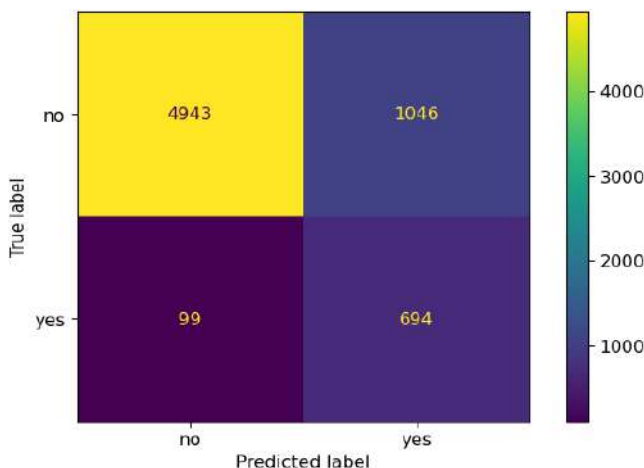
Gambar 7. Kurva ROC (Validation) + Threshold Terbaik menghasilkan **Val AUC = 0,9333** dan ambang terbaik **0,459**.

Keputusan ini logis karena ambang 0,5 tidak selalu optimal pada data tidak seimbang. Pada pengujian akhir.



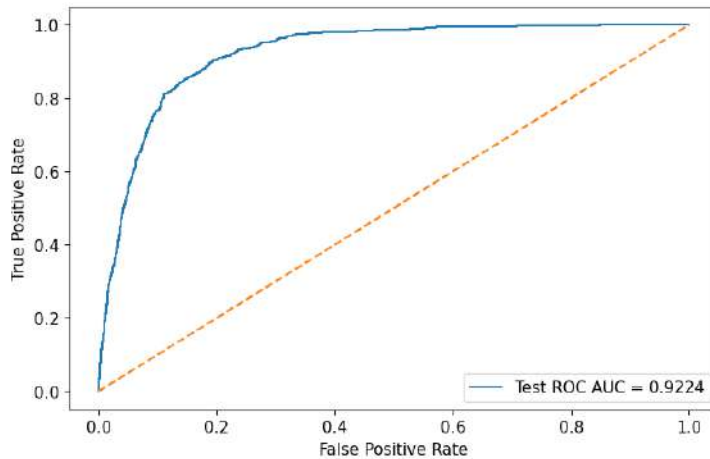
Gambar 7. Kurva ROC (Validation) + Threshold Terbaik

Gambar 8. Confusion Matrix (Test, threshold tuned) menampilkan TN=**4.943**, FP=**1.046**, FN=**99**, TP=**694**. Hasil ini menunjukkan kemampuan menangkap kelas yes tinggi karena FN rendah, sementara FP masih relatif besar sehingga ambang dapat dinaikkan bila prioritasnya mengurangi alarm palsu.



Gambar 8. Confusion Matrix (Test, threshold tuned)

Gambar 9. Kurva ROC (Test) menunjukkan **Test AUC = 0,9224**, penurunan kecil dari validasi yang mencerminkan generalisasi tetap kuat.



Gambar 9. Kurva ROC (Test)

Insight yang dapat dijadikan pegangan: **ketidakseimbangan kelas** wajib ditangani sejak awal melalui **class_weight**, **AUC** lebih informatif daripada akurasi pada kondisi ini, **dropout** dan **early stopping** menjaga kurva train-validation tidak melebar, serta **threshold tuning** membuat keputusan prediksi lebih selaras dengan kebutuhan penerapan

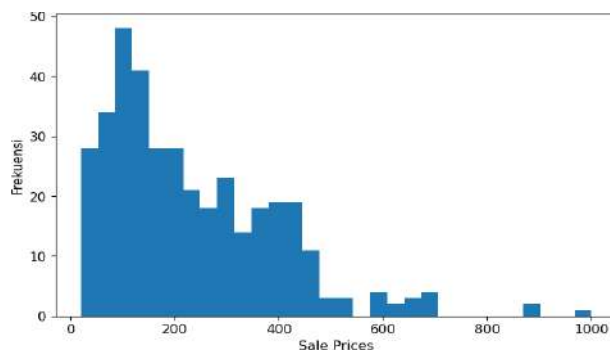
Studi Kasus: Implementasi MLP untuk Prediksi Sale Prices.



Kode program pada barcode ini, dimulai dari instalasi lingkungan minimal di Google Colab hingga evaluasi akhir berbasis metrik kuantitatif. Strategi akuisisi data menggunakan dua jalur, yaitu *ucimlrepo* dan *fallback* ZIP resmi UCI, merupakan dataset Residential

Building (UCI ID 437) sering tersedia di katalog tetapi tidak selalu dapat diimpor langsung. Pemisahan fitur dan target dilakukan secara eksplisit, diikuti pembersihan nama kolom target seperti OUTPUTS dan Unnamed, sehingga ambiguitas struktur data dapat dihindari sejak awal.

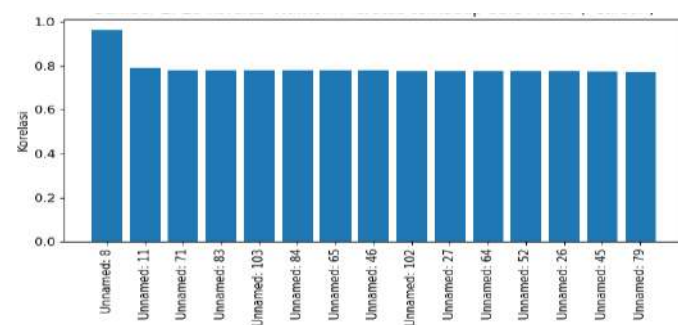
Karakteristik target *sale prices* terlihat jelas pada Gambar 1. Distribusi Target *Sale Prices*. Histogram menunjukkan sebaran sekitar **373 sampel** dengan rentang nilai dari **≈0 hingga 1.000**. Konsentrasi terbesar berada pada kisaran **50-250**, sementara ekor kanan memanjang hingga di atas **600**. Pola *right-skewed* ini mengindikasikan keberadaan *outlier*, sehingga pemilihan *Huber loss* menjadi keputusan metodologis yang **tepat dan beralasan** untuk mengurangi sensitivitas model terhadap nilai ekstrem.



Gambar 1. Distribusi Target *Sale Prices*

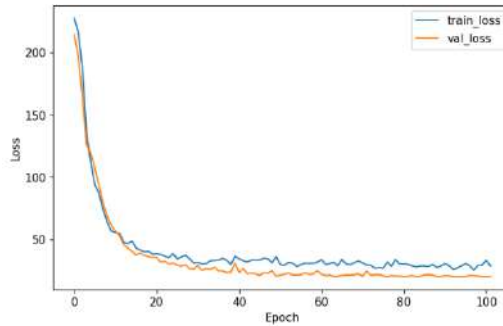
Hubungan antara fitur numerik dan target divisualisasikan pada Gambar 2. Lima Belas Korelasi Numerik Teratas terhadap *Sale Prices*. Nilai korelasi Pearson berada pada rentang **0,77-0,95**, menunjukkan keterkaitan linear yang kuat. Satu fitur bahkan mendekati **0,95**, menandakan kontribusi dominan terhadap variasi harga jual. Insight penting yang dapat dicatat adalah bahwa **fitur dengan korelasi sangat tinggi perlu ditinjau** agar tidak

merepresentasikan informasi turunan target yang berpotensi memicu *data leakage*.

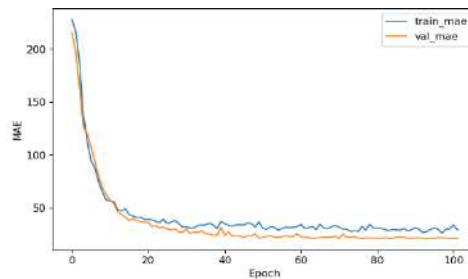


Gambar 2. Lima Belas Korelasi Numerik Teratas terhadap *Sale Prices*

Tahap pembagian data menggunakan *train-test split* dan standarisasi melalui *StandardScaler* dilakukan dengan *fit* hanya pada data latih, kemudian diterapkan pada data uji. Praktik ini merupakan **prinsip fundamental** untuk mencegah *data leakage*. Arsitektur model MLP terdiri dari lapisan Dense **256-128-64** dengan tiga lapis *dropout* berasio **0,3** dan total parameter **68.865**, ukuran yang masih proporsional untuk dataset berukuran menengah. Kombinasi *Dropout*, *EarlyStopping*, dan *ReduceLROnPlateau* berfungsi sebagai pengendali kompleksitas dan stabilitas optimasi. Dinamika pelatihan ditunjukkan pada Gambar 3. *Learning Curve (Huber Loss)*. dan Gambar 4. *Learning Curve (MAE)*. *Loss* dan MAE menurun tajam pada **10-15 epoch awal**, lalu melandai hingga sekitar **epoch 100**. Nilai *loss* validasi stabil di kisaran **20-25**, sedangkan MAE validasi berada pada kisaran **20-30**, dengan jarak kecil terhadap kurva pelatihan. Pola ini mengindikasikan konvergensi yang baik dan **tidak adanya overfitting signifikan**, menegaskan efektivitas *Dropout* dan *EarlyStopping*.

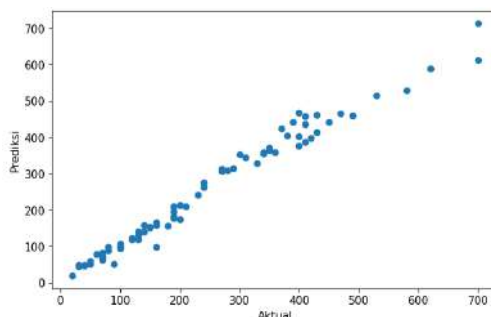


Gambar 3. *Learning Curve (Huber Loss).*



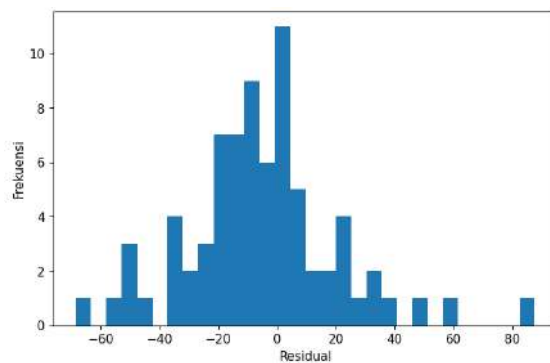
Gambar 4. *Learning Curve (MAE).*

Kinerja prediksi divisualisasikan pada Gambar 5. Aktual vs Prediksi (*Sale Prices*), yang menunjukkan titik-titik mendekati garis diagonal ideal. Pada rentang **100-500**, prediksi mengikuti nilai aktual secara rapat, sementara pada nilai di atas **600** masih terlihat deviasi moderat yang dapat diterima mengingat sebaran target yang lebar. Evaluasi kuantitatif menghasilkan **MAE = 19,0101**, **RMSE = 26,0803**, dan **$R^2 = 0,9758$** , menandakan kemampuan model menjelaskan lebih dari **97%** variasi data uji.

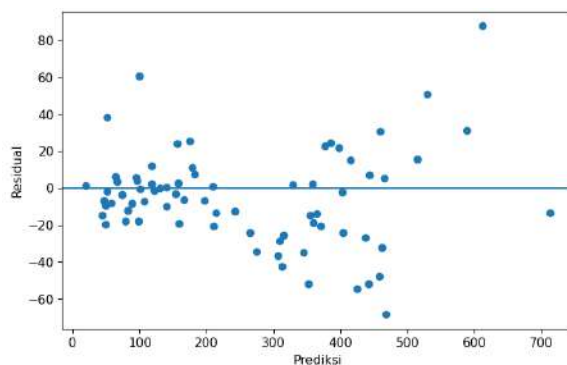


Gambar 5. Aktual vs Prediksi (*Sale Prices*)

Analisis kesalahan diperkuat melalui Gambar 6. Distribusi Residual (Aktual – Prediksi) dan Gambar 7. Residual Plot terhadap Prediksi. Sebagian besar residual berada pada rentang **-30 hingga +30** dan terpusat di sekitar nol, meskipun terdapat beberapa nilai ekstrem hingga sekitar **$\pm 70-80$** . Penyebaran residual yang relatif acak tanpa pola sistematis kuat menunjukkan bahwa model tidak bias secara struktural, meskipun terdapat indikasi heteroskedastisitas ringan pada prediksi bernilai tinggi. Hal ini sejalan dengan karakter *sale prices*.



Gambar 6. Distribusi Residual (Aktual – Prediksi)



Gambar 7. Residual Plot terhadap Prediksi

Secara keseluruhan, hasil ini menunjukkan bahwa **MLP dengan Dropout(0,3) dan Huber loss** mampu memberikan prediksi harga jual yang akurat dan stabil pada dataset Residential Building

UCI 437. **Insight** yang dapat dieksplorasi adalah membandingkan rasio *dropout* **0,2-0,4**, melakukan seleksi fitur berbasis korelasi ekstrem, serta menguji variasi parameter *Huber delta* untuk menyesuaikan sensitivitas terhadap *outlier*

Soal

1. Jelaskan secara rinci hubungan antara kedalaman hidden layer dan risiko overfitting pada model deep learning.
2. Bagaimana dropout bekerja sebagai teknik regularisasi untuk mencegah overfitting? Berikan ilustrasi mekanisme matematisnya.
3. Jika model deep learning menggunakan dropout 0,5, apa dampaknya terhadap laju konvergensi dan representasi internal model?
4. Bandingkan kelebihan dan kekurangan antara dropout dan L2 regularization dalam menjaga generalisasi model.
5. Mengapa Adam optimizer sering dipilih dibanding SGD biasa dalam kasus dataset tidak seimbang?
6. Jelaskan bagaimana backpropagation bekerja dan mengapa pemilihan *loss function* yang tepat penting dalam konteks overfitting.
7. Dalam situasi model menunjukkan akurasi pelatihan tinggi tetapi akurasi validasi rendah, strategi apa saja yang dapat digunakan berdasarkan teknik yang dipelajari?
8. Anda diminta men-tune epoch untuk model yang menunjukkan penurunan loss validasi berhenti setelah 50

epoch, tetapi loss pelatihan masih turun. Apa yang akan Anda lakukan dan kenapa?

9. Uraikan bagaimana *class_weight* dan dropout bersama-sama dapat membantu model mengatasi dataset yang sangat tidak seimbang.
10. Bagaimana cara mengevaluasi apakah Adam optimizer bekerja optimal pada dataset tertentu? Diskusikan metrik dan prosedurnya.

Tugas Kelompok

Studi Kasus 1 Prediksi Default Kartu Kredit

Tugas:

Kelompok diminta untuk merancang arsitektur deep learning untuk memprediksi apakah pelanggan akan default pada pembayaran kartu kredit. Model ini harus menggunakan teknik dropout untuk regularisasi dan Adam optimizer untuk optimasi. Selain itu, kelompok harus menyusun strategi preprocessing data, khususnya dalam menangani fitur numerik dan kategorikal. Setelah model dibangun, kelompok perlu mengevaluasi hasilnya menggunakan metrik yang sesuai seperti AUC dan F1-score. Analisis performa model perlu dilakukan, dengan perbandingan antara hasil yang diperoleh dari model deep learning dengan model klasik yang lebih sederhana. Hal ini bertujuan untuk memahami apakah deep learning memberikan peningkatan performa yang signifikan dalam klasifikasi kasus ini.

Link: [Default of Credit Card Clients Dataset](#)

Studi Kasus 2 Prediksi Kualitas Anggur

Tugas:

Kelompok diminta untuk menentukan apakah tugas ini lebih cocok untuk digunakan sebagai klasifikasi atau regresi dalam konteks deep learning. Setelah itu, kelompok perlu merancang model neural network dengan penerapan teknik dropout dan Adam optimizer untuk prediksi kualitas anggur berdasarkan parameter kimiawinya. Setiap kelompok juga diharapkan untuk menguji dua konfigurasi dropout yang berbeda (misalnya, dropout rate 0,2 dan 0,4) untuk menganalisis dampak teknik dropout terhadap overfitting. Di akhir, hasil model deep learning harus dibandingkan dengan model klasik lainnya, seperti Random Forest atau SVM, dan dianalisis perbedaannya dalam hal akurasi dan kemampuan generalisasi.

Link: [Wine Quality Dataset - Kaggle](#)

Studi Kasus 3 Prediksi Penyakit Jantung

Tugas:

Kelompok diminta untuk melakukan eksplorasi awal terhadap dataset yang mencakup faktor-faktor risiko penyakit jantung, dan mengidentifikasi fitur yang memiliki pengaruh signifikan terhadap prediksi status penyakit jantung. Setelah itu, kelompok harus membangun model MLP (Multi-Layer Perceptron) menggunakan teknik dropout dan Adam optimizer, dan menyusun laporan tentang arsitektur terbaik yang berhasil diterapkan. Diskusikan juga teknik regularisasi lain yang dapat digunakan selain dropout untuk meningkatkan kemampuan model dalam menghindari overfitting. Kelompok juga diminta untuk menganalisis cara model menangani

ketidakseimbangan kelas yang mungkin ada dalam dataset ini, dan bagaimana penyesuaian terhadap `class_weight` dapat mempengaruhi hasil model.

Link: [Heart Disease Dataset - Kaggle](#)

Refleksi Pembelajaran

Pembelajaran tentang Deep Learning, overfitting, dan teknik regularisasi seperti dropout serta optimisasi menggunakan Adam optimizer memberikan wawasan penting dalam pengembangan model prediktif yang efektif. Kami belajar bagaimana hidden layer dalam jaringan saraf meningkatkan kemampuan model untuk menangkap pola non-linear, namun juga menghadirkan tantangan overfitting. Teknik seperti dropout terbukti membantu mengurangi risiko ini, sedangkan Adam optimizer memastikan laju pembelajaran yang adaptif dan stabil. Evaluasi model dengan metrik seperti AUC dan F1-score memberi gambaran yang lebih jelas tentang performa model, terutama pada data yang tidak seimbang, sehingga meningkatkan kemampuan model dalam generalisasi.

Natural Language Processing (NLP) Text

Preprocessing dan Tokenization

CPL:

Mahasiswa memahami dasar pemrosesan bahasa alami.

CPMK:

Melakukan pembersihan teks dan ekstraksi fitur.

SubCPMK:

Menggunakan TfidfVectorizer dan CountVectorizer.

A. Dari Teks ke Fitur

Prapemrosesan menurunkan **noise**, *tokenization* menentukan **unit makna**, stopword removal dan *n-grams* mengatur **fokus informasi**, lalu vektorisasi mengubah teks menjadi **angka**. *Tokenization* berperan sebagai **penetapan apa yang dianggap informasi**, stopword removal bukan penghapusan asal, melainkan **penyaringan yang berisiko bila salah**, *n-grams* bukan sekadar tambahan, melainkan **cara menangkap frasa**, vektorisasi bukan mengubah bentuk saja, melainkan **membuat bahasa bisa dihitung**.

B. Normalisasi dan Pembersihan

Prapemrosesan teks mengubah teks mentah menjadi data yang lebih seragam agar analisis tidak terseret oleh detail remeh. Data teks nyata sering mengandung salah ketik, campuran bahasa, emotikon, tautan, spasi ganda, huruf kapital acak, tanda baca berulang, serta pengulangan huruf untuk menegaskan emosi. Tanpa

prapemrosesan, sistem mudah menganggap variasi kecil sebagai informasi besar, kosakata membengkak, dan representasi menjadi **rapuh** sehingga keputusan tampak meyakinkan padahal **tidak stabil**.

Normalisasi adalah langkah awal untuk membangun **konsistensi bentuk**. Huruf dapat diseragamkan, spasi dirapikan, dan karakter yang tidak dibutuhkan dihapus atau diganti penanda. Tautan dan alamat surel sering diganti token khusus agar keberadaannya tetap tercatat sebagai **indikator**, tanpa menyimpan detail yang jarang berguna. Angka membutuhkan **kebijakan eksplisit** karena angka dapat membawa **informasi kunci** pada tugas tertentu, misalnya ukuran, harga, atau tanggal. Emoji, tanda seru, dan huruf kapital juga dapat menjadi **sinyal emosi**, sehingga penghapusan total bisa menghilangkan informasi.

Bahasa Indonesia memiliki imbuhan yang kaya sehingga variasi bentuk kata dapat menambah jumlah fitur tanpa menambah makna. Pilihan antara *stemming* dan *lemmatization* memengaruhi ukuran kosakata, tingkat **generalisasi**, serta kemudahan **interpretasi**. Prapemrosesan yang baik menghasilkan teks yang lebih seragam dan tetap menjaga **makna**, karena **makna yang terjaga** adalah syarat analisis yang dapat dipertanggungjawabkan.

C. Segmentasi melalui *tokenization*

Tokenization memecah teks menjadi token, yaitu unit yang akan dihitung dan diwakili sebagai **fitur**, sehingga keputusan *tokenization* menentukan **apa yang dihitung** dan pada akhirnya menentukan **apa yang terlihat** oleh algoritma. Token berbasis kata sering dipakai karena cukup baik untuk menangkap topik, opini, dan

istilah. Tantangan muncul dari tanda baca menempel, singkatan, kata gabungan, kata bertanda hubung, serta istilah asing yang umum di ranah digital.

Pemilihan jenis token adalah keputusan desain. Token kata cocok untuk ringkasan isu dan klasifikasi topik. Token karakter atau sub-kata membantu menangani salah ketik, variasi kreatif, dan istilah baru yang terus berubah. Walau pipeline ini menekankan *CountVectorizer* dan *TfidfVectorizer*, pemahaman bahwa token dapat berupa unit berbeda menjaga pembahasan tetap konsisten: unit analisis ditentukan oleh **tujuan**, bukan oleh kebiasaan. *Tokenization* sebagai Aturan saat membangun kosakata harus identik saat penerapan. Perubahan kecil pada aturan pemisahan dapat menciptakan token baru, memperbanyak duplikasi **Insight** *tokenization* yang konsisten membuat fitur lebih **stabil**, stabilitas fitur membuat hasil lebih **dapat dijelaskan**.

D. stopwords removal dan *n-grams*

Stopword removal mengurangi dominasi kata yang sangat sering muncul dan sering kurang membedakan dokumen. Tujuannya bukan menghapus kata umum demi gaya, melainkan meningkatkan fitur. Daftar stopwords harus **diuji**, karena beberapa kata umum justru membawa makna penting, terutama **negasi** seperti “tidak”. Menghapus negasi dapat membalik makna, sehingga daftar stopwords perlu diperlakukan sebagai daftar yang dapat disesuaikan untuk menghindari **risiko salah tafsir**.

n-grams menangkap rangkaian token berurutan sehingga frasa dapat terbaca sebagai satu pola. Ini penting ketika makna muncul sebagai gabungan kata, misalnya frasa negasi atau istilah

tetap. *n-grams* menambah **kekuatan representasi**, tetapi juga menaikkan jumlah fitur secara cepat, sehingga muncul trade-off antara **ketajaman makna** dan **biaya dimensi**. Prinsip yang menjaga integritas adalah memilih *n-grams* berdasarkan manfaat informasi yang nyata, bukan menambah kompleksitas tanpa alasan.

E. Representasi Numerik

Vektorisasi mengubah token menjadi angka agar dapat diproses oleh algoritma. *CountVectorizer* menghitung frekuensi token dan sering menjadi **baseline** yang kuat untuk memastikan pipeline berjalan. Kelemahannya, token sangat umum bisa mendominasi walau informasinya kecil.

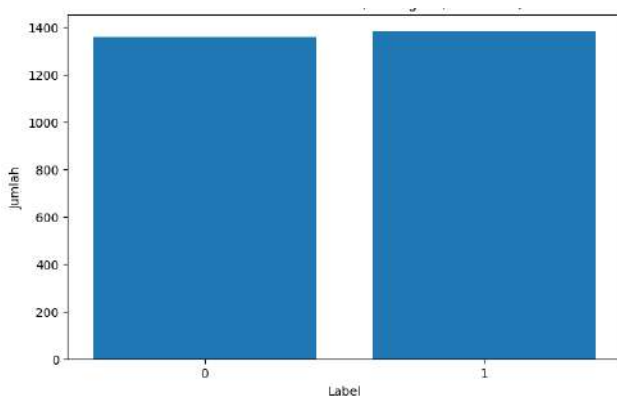
TfidfVectorizer memberi bobot lebih besar pada token yang lebih spesifik dan menurunkan bobot token yang terlalu merata. Secara intuitif, *tf-idf* menonjolkan **kata ciri** dan menekan **kata umum**. Kedua *vectorizer* dapat dikombinasikan dengan *n-grams* dan disaring melalui *min_df*, *max_df*, serta *max_features* agar token terlalu jarang, token terlalu umum, dan ukuran fitur tetap terkendali.

Kode Program: Implementasi NLP untuk Klasifikasi Sentimen.



Kode program pada barcode ini dimulai dari instalasi lingkungan minimal di Google Colab hingga evaluasi akhir berbasis metrik kuantitatif, dengan tujuan membangun pipeline **klasifikasi**

sentimen biner pada dataset **Sentiment Labelled Sentences** dari UCI. Strategi akuisisi data menggunakan satu jalur utama berupa unduhan ZIP resmi UCI, lalu disertai revisi pembacaan berkas berbasis *fallback encoding* untuk mengatasi kasus *UnicodeDecodeError* yang dapat muncul ketika file teks tidak sepenuhnya kompatibel dengan UTF-8. Data yang dipakai dibatasi secara eksplisit pada tiga berkas utama, yaitu **amazon_cells_labelled.txt**, **imdb_labelled.txt**, dan **yelp_labelled.txt**, sehingga struktur data tetap terkendali dan tidak tercampur dengan berkas non dataset. Setelah digabung, pemisahan fitur dan target dilakukan dengan jelas, yakni **sentence** sebagai fitur teks dan **label** sebagai target, dengan validasi bahwa label hanya bernilai **0** atau **1**, sehingga ambiguitas target dapat dihindari sejak awal.

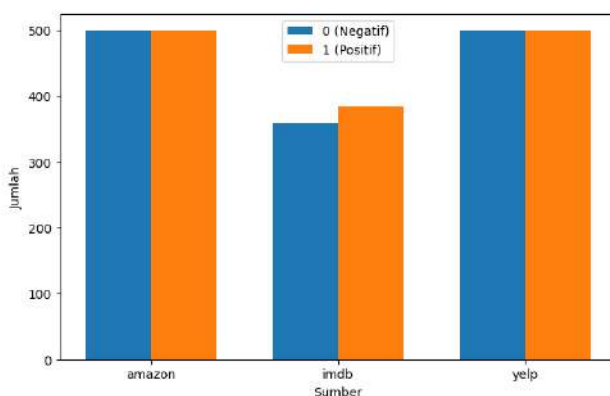


Gambar 1. Distribusi Label Sentimen Biner

Karakteristik kelas target terlihat pada Gambar 1. Distribusi Label Sentimen Biner. Grafik batang menunjukkan dataset berisi **3.000** kalimat dengan dua kelas yang relatif seimbang, di mana jumlah label **0** dan **1** berada pada kisaran **≈1,35 ribu** hingga **≈1,40 ribu**. Keseimbangan ini mengurangi risiko model hanya menebak

kelas mayoritas dan membuat metrik seperti **accuracy** tetap informatif, meskipun seleksi model tetap dipandu oleh **F1-score** agar presisi dan recall dinilai secara simultan.

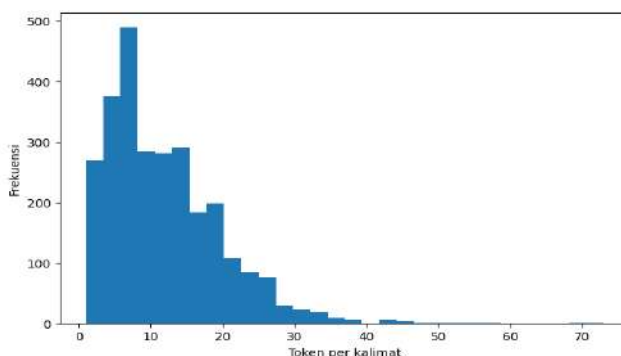
Diagram batang menunjukkan sebaran label pada dataset sekitar **3.000** kalimat. Jumlah kelas **0** dan **1** hampir seimbang, masing masing sekitar **1,35-1,40 ribu** sampel. Kondisi ini mengurangi bias kelas dan mendukung evaluasi adil. Dengan split **80:20**, set uji berkisar **≈600** data untuk pengukuran metrik.



Gambar 2. Distribusi Label per Sumber Data

Variasi label antar domain divisualisasikan pada Gambar 2. Distribusi Label per Sumber Data. Grafik memperlihatkan bahwa Amazon dan Yelp cenderung seimbang sekitar **500** data per kelas, sedangkan IMDb menunjukkan ketidaksimetrian ringan dengan negatif sekitar **≈360** dan positif sekitar **≈380**. **Insight** yang perlu dicatat adalah bahwa perbedaan domain dapat memicu variasi gaya bahasa, sehingga model bisa saja belajar pola domain selain sentimen. Karena itu metadata **source** disimpan agar analisis dapat diperluas menjadi pengujian performa per domain, yang relevan untuk menguji kemampuan **generalization**.

Grafik per sumber menampilkan tiga domain, **amazon**, **imdb**, dan **yelp**. Amazon dan Yelp tampak seimbang sekitar **500** sampel untuk kelas **0** dan **1**. IMDb menunjukkan perbedaan kecil, negatif sekitar ≈ 360 dan positif ≈ 380 . Pola ini mengindikasikan variasi domain yang dapat memengaruhi performa model lintas sumber.



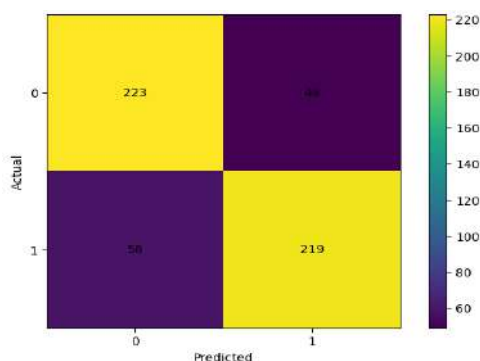
Gambar 3. Distribusi Panjang Kalimat Berdasarkan Jumlah Token

Kompleksitas teks divisualisasikan pada Gambar 3. Distribusi Panjang Kalimat Berdasarkan Jumlah Token. Histogram menunjukkan mayoritas kalimat berada pada rentang **5-20 token**, dengan puncak frekuensi sekitar **8-12 token**, sementara ekor kanan memanjang hingga sekitar **70 token**. Pola ini menunjukkan dataset didominasi teks pendek, sehingga pemilihan representasi berbasis **TF-IDF** menjadi keputusan metodologis yang tepat untuk baseline. Di sisi lain, adanya ekor panjang mengindikasikan sebagian kecil kalimat panjang yang berpotensi menambah noise pada fitur, sehingga parameter seperti **min_df** dan **max_df** layak dioptimalkan pada tahap revisi model.

Histogram panjang kalimat menunjukkan mayoritas data berada pada **5-20 token**, dengan puncak sekitar **8-12 token**. Sebaran memiliki ekor kanan hingga ≈ 70 token, menandakan sebagian kecil kalimat panjang. Dominasi teks pendek mendukung

TF-IDF sebagai baseline, sementara ekor panjang memberi sinyal kemungkinan noise yang perlu dikendalikan lewat *tuning*.

Tahap pembagian data menggunakan **train-test split** dengan **stratify** untuk menjaga proporsi label, sehingga set latih dan set uji tetap mewakili sebaran kelas. Setelah itu dilakukan ekstraksi fitur menggunakan **TfidfVectorizer** yang secara implisit menjalankan pembersihan dasar seperti *lowercasing* dan tokenisasi berbasis pola kata, lalu membangun bobot TF-IDF pada ruang fitur yang jarang. Baseline model menggunakan **Logistic Regression** karena stabil dan interpretabel untuk teks pendek.

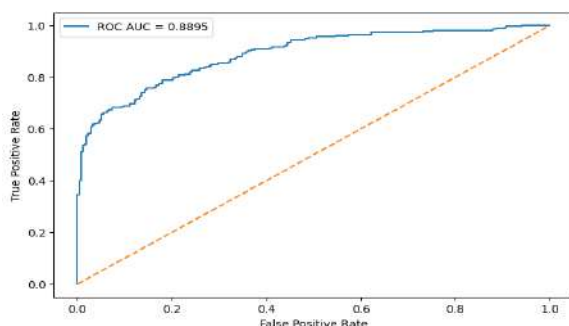


Gambar 4. Confusion Matrix Model Baseline Logistic Regression

Kinerja baseline divisualisasikan pada Gambar 4. Confusion Matrix Model Baseline Logistic Regression. Dari sekitar **549** sampel uji yang terlihat pada matriks, model menghasilkan **223** prediksi benar untuk kelas negatif dan **219** prediksi benar untuk kelas positif. Kesalahan terdiri dari **49** negatif yang diprediksi positif dan **58** positif yang diprediksi negatif. Pola ini menunjukkan baseline cukup seimbang, namun masih memiliki ruang peningkatan khususnya pada pengurangan prediksi positif palsu.

Confusion Matrix baseline menunjukkan **223** true negative dan **219** true positive. Kesalahan baseline terdiri dari **49** false

positive dan **58** false negative, total uji sekitar **549** sampel. Jika dihitung, akurasi mendekati $(223+219)/549 \approx 0,805$. Pola kesalahan dua arah menandakan perlunya revisi fitur dan model.

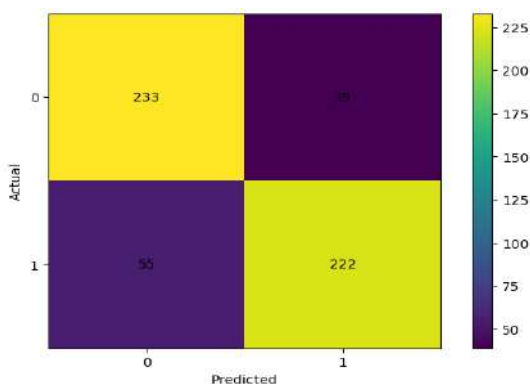


Gambar 5. Kurva ROC Model Baseline Logistic Regression

Kinerja pemisahan kelas baseline diperkuat pada Gambar 5. Kurva ROC Model Baseline Logistic Regression. Nilai **ROC AUC = 0,8895** menunjukkan kemampuan pemisahan kelas yang kuat dan jauh di atas garis acak. AUC menjadi metrik yang relevan karena menilai performa pada seluruh ambang keputusan, bukan hanya pada satu titik ambang. Nilai ini dipakai sebagai patokan kuantitatif untuk menilai apakah revisi model benar benar meningkatkan kemampuan klasifikasi, bukan sekadar mengubah perilaku pada satu ambang. Kurva ROC baseline menampilkan **ROC AUC = 0,8895**, menunjukkan kemampuan pemisahan kelas yang tinggi. Kurva berada jauh di atas garis acak diagonal. Ini menandakan model memiliki trade off yang baik antara true positive rate dan false positive rate pada berbagai ambang. Nilai AUC menjadi baseline numerik untuk mengevaluasi revisi berikutnya.

Strategi pertama adalah *tuning* representasi TF-IDF dan regularisasi, khususnya memperluas fitur menjadi **bigram** melalui **ngram_range (1,2)** agar model mampu menangkap frasa sentimen

seperti negasi “not good” atau intensifier “very good”. Strategi kedua adalah memakai model pembanding **LinearSVC**, yang sering unggul pada data teks berfitur TF-IDF karena bekerja baik pada ruang fitur berdimensi tinggi. Berdasarkan hasil yang Anda tampilkan, model final menjadi **LinearSVC**, sehingga evaluasi ROC menggunakan *decision scores*.



Gambar 6. Confusion Matrix Model Final

Gambar 6. Confusion Matrix Model Final. Nilai true negative meningkat menjadi **233** dan true positive menjadi **222**, sementara false positive turun menjadi **39** dan false negative menjadi **55**. Penurunan false positive dari **49** ke **39** adalah perubahan yang substantif karena berarti lebih sedikit ulasan negatif yang salah dipuji sebagai positif. Confusion Matrix model final menunjukkan **233** true negative dan **222** true positive. Kesalahan berkurang pada false positive dari **49** menjadi **39**, sedangkan false negative menjadi **55** dari **58**. Total uji sekitar **549** sampel. Perubahan ini meningkatkan ketelitian terhadap ulasan negatif, sehingga model lebih konservatif memberi label positif.

negatif mendukung efektivitas model linear berbasis TF-IDF, yang pada revisi final mencapai **ROC AUC 0,9094**.

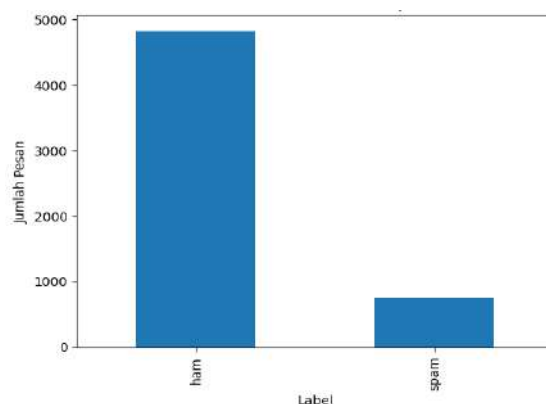
Secara keseluruhan, pipeline menunjukkan bahwa pemilihan model meningkatkan performa dari **ROC AUC 0,8895** pada baseline menjadi **0,9094** pada model final, disertai penurunan false positive dari **49** menjadi **39**. Insight yang bisa dieksplorasi lebih lanjut adalah membandingkan rentang parameter **min_df** dan **max_df** yang lebih luas, menguji strategi *stopwords* atau *lemmatization* sebagai eksperimen terkontrol, serta melakukan evaluasi per domain **amazon**, **imdb**, dan **yelp** untuk menilai stabilitas model pada variasi gaya bahasa.

Studi Kasus: Klasifikasi Spam-Ham pada SMS Spam.



Kode program pada barcode ini, eksplorasi data berbasis visual, pemisahan latih-validasi-uji yang menjaga proporsi kelas, pelatihan dua model *baseline* kuat berbasis TF-IDF, hingga evaluasi tiga arsitektur *deep learning* ringan dan revisi keputusan melalui uji *threshold*. Dataset yang digunakan adalah SMS Spam Collection dari UCI Machine Learning Repository, yang secara historis dipakai luas untuk mengevaluasi metode penyaringan SMS karena menyediakan label biner yang jelas dan format teks yang relatif ringkas. Strategi akuisisi data menggunakan unduhan ZIP dari UCI, kemudian ekstraksi berkas SMSSpamCollection yang berisi pasangan label dan

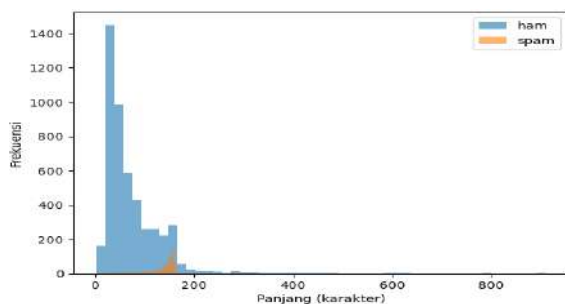
teks. Memastikan jalur *fallback* manual bila URL unduhan langsung berubah, tanpa mengubah struktur data, sehingga integritas replikasi eksperimen tetap terjaga. Data kemudian dibaca ke DataFrame dengan dua kolom, yaitu label dan text, lalu target dikodekan menjadi `is_spam` bernilai 1 untuk *spam* dan 0 untuk *ham*, sehingga pembacaan metrik menjadi konsisten dan menghindari ambiguitas target sejak awal. Dalam konteks praktikum, langkah ini menegaskan bahwa kualitas eksperimen bukan hanya ditentukan model, melainkan juga oleh **ketertiban definisi variabel**, **keterlacakan transformasi**, dan **pemisahan data yang bersih** agar tidak terjadi *data leakage*.



Gambar 1. Distribusi Kelas *Ham* vs *Spam*

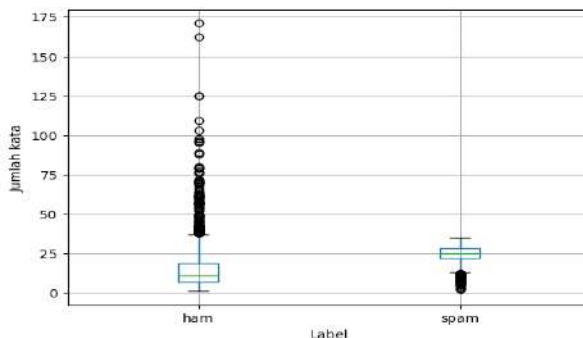
Karakteristik kelas terlihat pada grafik batang distribusi label, di mana kelas *ham* jauh lebih dominan dibanding *spam*. Ketidakseimbangan ini menjelaskan mengapa evaluasi tidak boleh hanya bertumpu pada akurasi, karena model dapat tampak “baik” hanya dengan menebak kelas mayoritas. **Insight** ini bahwa kondisi ini membuat metrik berbasis kelas minoritas seperti *precision*, *recall*, dan khususnya *PR-AUC* menjadi lebih informatif untuk menilai kemampuan menangkap *spam*. Ketidakseimbangan kelas juga memperkuat pilihan

penggunaan `class_weight` pada beberapa model linear agar pembelajaran tidak bias pada *ham*.



Gambar 2. Distribusi Panjang Pesan (Karakter)

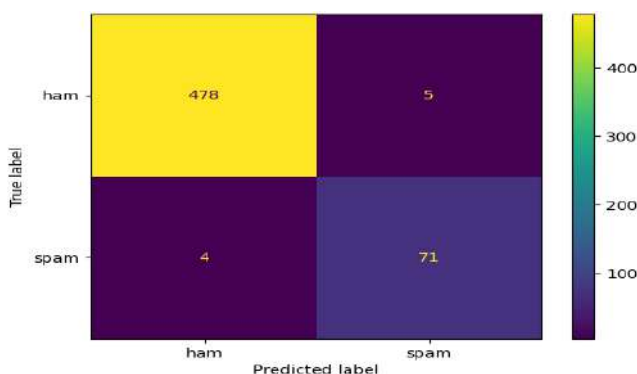
Histogram panjang pesan memperlihatkan mayoritas *ham* terkonsentrasi pada pesan pendek, sedangkan *spam* cenderung memiliki sebaran yang “bergeser” ke panjang tertentu karena sering memuat ajakan, angka, atau tautan. Pada visual, sebaran *ham* sangat padat di rentang rendah dengan ekor panjang hingga ratusan karakter, sementara *spam* tampak lebih terkumpul pada rentang menengah. Temuan ini mendukung keputusan memakai TF-IDF sebagai *baseline* karena teks relatif pendek dan sinyal sering hadir sebagai kombinasi kata atau frasa, tetapi juga memberi sinyal bahwa sebagian pesan panjang dapat menjadi sumber variasi yang perlu dikendalikan dengan regularisasi, *early stopping*, dan parameter vektorisasi seperti `min_df`.



Gambar 3. Panjang Pesan (Kata) per Kelas

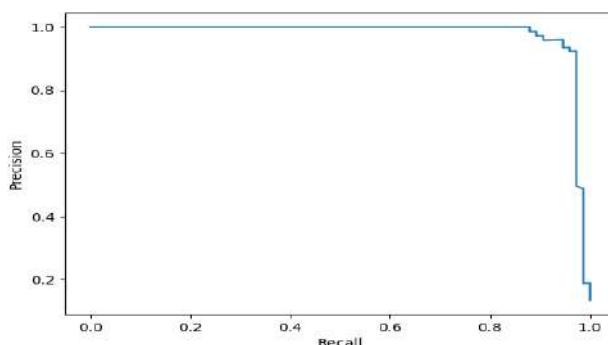
Boxplot jumlah kata menunjukkan median *spam* lebih tinggi dibanding *ham*, namun terdapat *outlier ham* yang sangat panjang. Ini memberi **Insight** bahwa model urutan seperti RNN berpotensi memanfaatkan pola urutan token, tetapi pada dataset berukuran moderat risiko *overfitting* meningkat jika kapasitas model terlalu besar. Karena itu, arsitektur *deep learning* adalah menjaga dimensi *embedding* dan jumlah unit RNN tetap moderat, mengaktifkan dropout, dan memakai *early stopping* berbasis *val PR-AUC*.

Tahap pemisahan data menggunakan *train-test split* bertingkat menjadi train, validasi, dan uji dengan *stratify*, sehingga proporsi *spam* pada setiap subset tetap representatif. Proses “fit” untuk TF-IDF dan *TextVectorization* hanya pada data latih. Praktik ini krusial untuk menjaga integritas evaluasi karena vektorisasi yang “melihat” data uji akan mengalirkan informasi statistik token dan menggelembungkan performa secara tidak sah. Setelah itu, *baseline* pertama memakai TF-IDF + Logistic Regression dan *baseline* kedua memakai TF-IDF + Linear SVM, karena keduanya dikenal kuat di ruang fitur berdimensi tinggi dan jarang, tipikal untuk TF-IDF pada teks pendek.



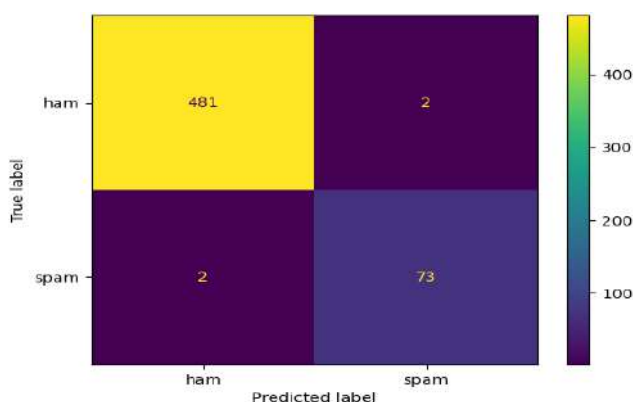
Gambar 4. *Confusion Matrix* TF-IDF + Logistic Regression (Test)

Pada hasil matriks menunjukkan $TN=478$, $FP=5$, $FN=4$, $TP=71$ pada total 558 sampel uji. Ini berarti $precision\ spam = 71/(71+5)=0,9342$, $recall\ spam = 71/(71+4)=0,9467$, dan akurasi = $(478+71)/558=0,9839$. Pola kesalahan relatif kecil dan seimbang, tetapi masih ada *spam* yang lolos ($FN=4$) serta *ham* yang salah ditandai ($FP=5$).



Gambar 5. Kurva *Precision-Recall* Logistic Regression

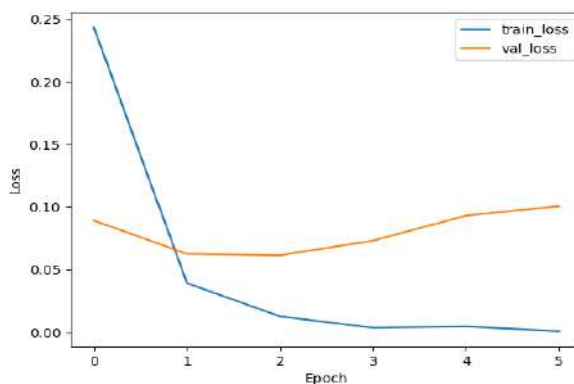
Kurva menampilkan AP 0,9783 yang tinggi, menunjukkan kualitas pemeringkatan probabilitas yang baik pada kelas minoritas. AP digunakan sebagai proksi *PR-AUC* sehingga evaluasi tidak bergantung pada satu *threshold* saja. Ini penting pada dataset tidak seimbang karena perubahan ambang bisa mengubah *precision* dan *recall* secara drastis, sementara AP merangkum performa.



Gambar 6. *Confusion Matrix* TF-IDF + Linear SVM (Test)

Matriks menunjukkan $TN=481$, $FP=2$, $FN=2$, $TP=73$. Secara langsung, $precision\ spam = 73/(73+2)=0,9733$, $recall\ spam = 73/(73+2)=0,9733$, $F1=0,9733$, dan akurasi $= (481+73)/558=0,9928$. Ini menjelaskan mengapa tabel ringkasan menempatkan Linear SVM sebagai performa tertinggi secara F1 dan akurasi. **Insight** kunci di sini adalah bahwa *baseline* linear yang dituning baik dapat menjadi solusi terbaik untuk teks pendek, sehingga *deep learning* bukan selalu “wajib menang”, melainkan harus diuji secara adil dan dikontrol *overfitting*-nya.

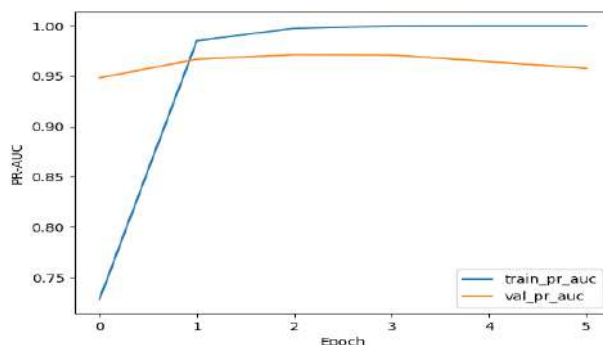
Pada tahap *deep learning*, tiga arsitektur ringan diuji: CNN-Text, BiLSTM, dan BiGRU. Praproses minimal yang tidak menghapus sinyal penting, misalnya URL dan angka diganti token <URL> dan <NUM>. Langkah ini masuk akal karena *spam* sering mengandung pola semacam itu. Selain itu, pelatihan menggunakan *early stopping* berbasis *val PR-AUC* untuk menghindari pelatihan berlebih ketika metrik validasi berhenti membaik.



Gambar 7. *Learning Curve* Loss BiGRU

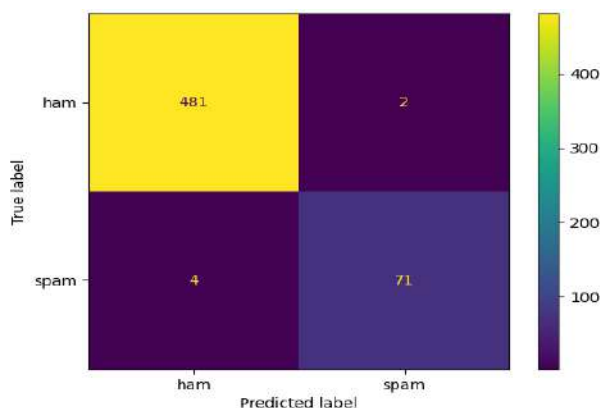
Kurva menunjukkan *train loss* turun dari sekitar 0,24 pada epoch 0 hingga mendekati 0,00 pada epoch 5, sedangkan *validation loss* sempat turun ke sekitar 0,062 pada epoch 1-2 lalu naik menuju

sekitar 0,10 pada epoch 5. Pola ini adalah indikator *overfitting* ringan-menengah, sehingga keputusan *early stopping* merupakan revisi yang tepat. Dalam bahasa praktikum, ini adalah bukti visual bahwa model belajar sangat cepat pada data latih, namun generalisasi mulai menurun setelah titik tertentu.



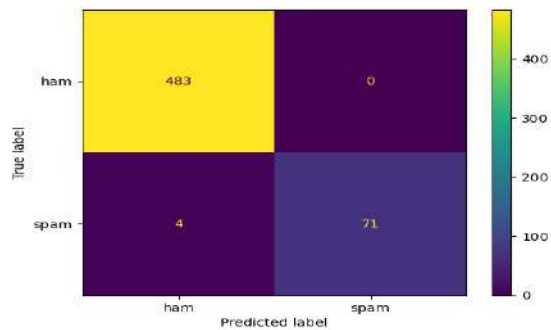
Gambar 8. *Learning Curve PR-AUC BiGRU*

Kurva menampilkan *train PR-AUC* meningkat dari sekitar 0,73 menjadi sekitar 1,00 mulai epoch 2, sementara *validation PR-AUC* berada pada kisaran 0,95-0,97 dengan puncak sekitar 0,97 pada epoch 2-3 dan turun ke sekitar 0,96 pada epoch 5. Ini memperkuat argumen bahwa pelatihan terbaik terjadi sebelum epoch terakhir, dan menjustifikasi pemilihan bobot terbaik melalui *restore_best_weights*.



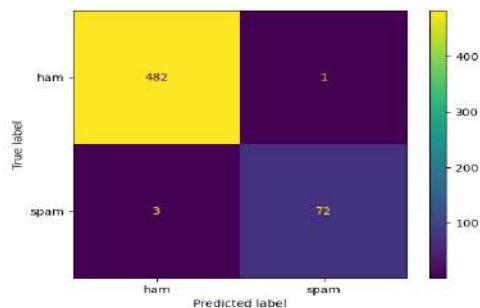
Gambar 9. *Confusion Matrix CNN-Text (Test)*

Kurva *Precision-Recall* CNN-Text (AP=0,9856) Matriks CNN-Text menunjukkan TN=481, FP=2, FN=4, TP=71, sehingga *precision spam* 0,9726 dan *recall spam* 0,9467 pada total 558 sampel. Kurva *precision-recall* memberi AP=0,9856. Ini menandakan pemeringkatan probabilitas kuat, tetapi masih ada *spam* terlewat (FN=4). Secara logis, CNN-Text berada di bawah Linear SVM pada akurasi, namun tetap memberi **nilai tambah** berupa kemampuan memodelkan pola lokal token.



Gambar 10. *Confusion Matrix* BiLSTM (Test)

Kurva *Precision-Recall* BiLSTM (AP=0,9867) BiLSTM menunjukkan TN=483, FP=0, FN=4, TP=71. Kondisi FP=0 menghasilkan *precision spam* 1,0000, tetapi *recall spam* tetap 0,9467. AP=0,9867 memperkuat kualitas pemeringkatan. Secara praktis, ini cocok bila prioritas adalah **menghindari salah tandai ham sebagai spam**, tetapi masih mengizinkan beberapa *spam* lolos.



Gambar 11. *Confusion Matrix* BiGRU (Test)

Kurva *Precision-Recall* BiGRU (AP=0,9922) BiGRU menghasilkan TN=482, FP=1, FN=3, TP=72 dengan total 558 sampel. Ini memberi *precision spam* = $72/(72+1)=0,9863$ dan *recall spam* = $72/(72+3)=0,9600$, sementara AP=0,9922 adalah yang tertinggi di antara tiga arsitektur *deep learning*. Secara logis, BiGRU menurunkan FN dari 4 menjadi 3 dibanding CNN-Text dan BiLSTM, dengan FP kecil. Ini menunjukkan BiGRU dapat menjadi terbaik jika targetnya meningkatkan penangkapan *spam* tanpa mengorbankan terlalu banyak *precision*.

Tabel ringkasan performa (Test):

	model	precision_spam	recall_spam	f1_spam	accuracy	AP (PR-AUC proxy)
1	TF-IDF + Linear SVM	0.973333	0.973333	0.973333	0.992832	NaN
4	BiGRU	0.986301	0.960000	0.972973	0.992832	0.992172
3	BiLSTM	1.000000	0.946667	0.972603	0.992832	0.986656
2	CNN-Text	0.972603	0.946667	0.959459	0.989247	0.985640
0	TF-IDF + Logistic Regression	0.934211	0.946667	0.940397	0.983871	0.978257

Tabel evaluasi threshold (BiGRU, Test)

	threshold	precision_spam	recall_spam	f1_spam	accuracy
0	0.3	0.948718	0.986667	0.967320	0.991039
1	0.4	0.960526	0.973333	0.966887	0.991039
2	0.5	0.986301	0.960000	0.972973	0.992832
3	0.6	1.000000	0.960000	0.979592	0.994624
4	0.7	1.000000	0.933333	0.965517	0.991039

Gambar 12. Tabel Ringkasan Performa dan Evaluasi *Threshold* BiGRU

Tabel ringkasan menempatkan TF-IDF + Linear SVM sebagai yang tertinggi (F1=0,9733; akurasi 0,9928). BiGRU memiliki akurasi yang sama 0,9928 dengan F1 0,9730 dan AP 0,9922. Evaluasi

threshold menunjukkan revisi operasional yang penting: pada ambang 0,6 BiGRU mencapai *precision spam* 1,0000, *recall* 0,9600, F1 0,9796, dan akurasi 0,9946, lebih tinggi dari konfigurasi default 0,5.

Secara keseluruhan, hasil ini konsisten dan berintegritas karena keputusan akhir dapat dibingkai sebagai berikut: Linear SVM adalah **baseline final yang sangat kuat** untuk akurasi dan F1 pada teks pendek, sementara BiGRU memberikan kualitas pemeringkatan (AP) tertinggi dan fleksibilitas *threshold* untuk skenario operasional yang menuntut trade-off spesifik. Menguji variasi *ngram_range*, eksplorasi *min_df*, serta evaluasi berbasis *cross-validation* untuk mengurangi ketergantungan pada satu split. Namun, dengan bukti visual *learning curve* dan bukti kuantitatif *confusion matrix*

Soal

1. Jelaskan secara rinci bagaimana preprocessing teks yang baik dapat meningkatkan performa model NLP pada dataset teks berbahasa campuran (mis. bahasa Indonesia-Inggris), dan sebutkan risiko jika *stopword removal* dilakukan tanpa mempertimbangkan konteks bahasa.
2. Bandingkan dampak penggunaan *CountVectorizer* dan *TfidfVectorizer* terhadap perhitungan fitur pada dataset teks panjang (misal: artikel berita), serta jelaskan situasi di mana satu metode lebih unggul dari yang lain.
3. Dalam tokenization, bagaimana keputusan pemilihan token berbasis kata, karakter, atau sub-kata memengaruhi hasil

klasifikasi? Berikan contoh kasus nyata yang menunjukkan keunggulan masing-masing token.

4. Deskripsikan bagaimana n-grams dapat membantu model menangkap informasi kontekstual dalam sentimen negatif yang melibatkan kata negasi seperti “tidak bagus sama sekali”. Berikan juga potensi dampak negatifnya dari sisi dimensi fitur.
5. Analisis perbedaan antara lemmatization dan stemming dalam mengurangi variasi bentuk kata pada teks berbahasa Indonesia, dan jelaskan bagaimana salah satu pilihan ini dapat memengaruhi kualitas fitur.
6. Jelaskan peran min_df dan max_df dalam TfidfVectorizer ketika menganalisis dataset teks yang memiliki banyak kata unik, dan bagaimana parameter ini membantu menyeimbangkan noise vs fitur penting.
7. Berikan contoh metode evaluasi selain accuracy yang penting untuk dataset teks tidak seimbang dan jelaskan mengapa metrik ini lebih informatif (misal: dataset mental health text classification).
8. Diskusikan tantangan dan strategi preprocessing ketika bekerja dengan dataset teks yang berisi emotikon, URL, angka, dan simbol lain (misal: dataset email atau SMS). Apa saja aturan preprocessing yang perlu disiapkan?
9. Dalam konteks NLP dan tokenization, bagaimana perlakuan terhadap penggunaan kata umum yang sebenarnya mengandung makna penting seperti kata negasi (“tidak”, “bukan”, “tanpa”)? Apa strategi untuk menangani stopwords semacam ini?

10. Buatlah penjelasan logis bagaimana kombinasi TfidfVectorizer dengan bigrams dapat meningkatkan kemampuan model dalam menangkap frasa kunci dalam berita multi-kategori, dan jelaskan efeknya terhadap ukuran ruang fitur.

Tugas Kelompok

Studi Kasus 1 Klasifikasi Konten Manusia vs AI

Tugas

Kelompok diminta untuk membangun sebuah pipeline preprocessing teks yang optimal untuk memisahkan teks manusia dari teks yang dihasilkan oleh AI. Langkah pertama adalah melakukan normalisasi teks, tokenization, dan stopwords removal. Kelompok harus memilih metode yang sesuai untuk masing-masing langkah ini, disertai dengan alasan yang jelas mengenai pilihan yang diambil. Setelah itu, aplikasikan dua jenis vectorizer, yaitu CountVectorizer dan TfidfVectorizer, dan bandingkan performa klasifikasi menggunakan model seperti Logistic Regression dan Support Vector Machine (SVM). Selanjutnya, kelompok perlu menganalisis bagaimana karakteristik teks AI berbeda dari teks manusia berdasarkan fitur-fitur yang dihasilkan oleh TfidfVectorizer. Terakhir, buat laporan ilmiah yang berisi rekomendasi mengenai preprocessing terbaik yang dapat digunakan untuk memisahkan teks AI dari teks manusia, serta mengapa pilihan tersebut efektif dalam meningkatkan akurasi klasifikasi.

Link [Human vs AI Text Classification Dataset](#)

Studi Kasus 2 Multi-Kelas Klasifikasi Teks Emosi Mental Health

Tugas

Kelompok diminta untuk membangun model klasifikasi untuk dataset Mental Health Text Classification yang terdiri dari empat kelas emosi: Suicidal, Depression, Anxiety, dan Normal. Langkah pertama adalah menguji berbagai teknik preprocessing, seperti stemming atau lemmatization, dan mengevaluasi dampaknya terhadap performa model. Kemudian, gunakan TfidfVectorizer dengan berbagai pilihan n-gram (1-3) untuk melihat pengaruhnya terhadap hasil klasifikasi. Bangun minimal tiga model klasifikasi teks, seperti SVM, Random Forest, dan Neural Network ringan, serta evaluasi model dengan menggunakan metrik precision, recall, dan F1-score per kelas untuk memastikan performa model tidak bias terhadap kelas mayoritas. Selanjutnya, kelompok diharapkan untuk menyajikan analisis interpretatif dari fitur utama yang membedakan setiap kelas emosi berdasarkan bobot fitur yang dihasilkan oleh TfidfVectorizer.

Link [Mental Health Text Classification Dataset](#)

Studi Kasus 3 Klasifikasi E-Commerce Text Categories

Tugas

Kelompok diminta untuk menangani dataset teks produk dari e-commerce yang terdiri dari empat kategori: Electronics, Household, Books, dan Clothing & Accessories. Langkah pertama adalah merencanakan preprocessing teks yang tepat untuk menangani teks produk yang sering mengandung kode produk, angka, dan singkatan. Gunakan TfidfVectorizer dengan unigram, bigram, dan

trigram serta bandingkan performanya. Setelah itu, aplikasikan beberapa model klasifikasi teks, seperti Multinomial Naive Bayes, SVM, dan Logistic Regression. Evaluasi performa model menggunakan metrik accuracy, precision, recall, dan F1-score. Laporan akhir harus mencakup visualisasi kata kunci per kategori, menggunakan teknik seperti word clouds atau visualisasi top-TFIDF untuk membantu menjelaskan pola dalam setiap kategori produk.

Link [Ecommerce Text Classification Dataset](#)

Refleksi Pembelajaran

Pembelajaran tentang Natural Language Processing (NLP) memberikan pemahaman mendalam mengenai proses penting dalam pengolahan teks, seperti tokenization, stopword removal, dan vektorisasi menggunakan CountVectorizer dan TfidfVectorizer. Proses ini sangat penting untuk mengubah teks mentah menjadi fitur numerik yang dapat dipahami oleh algoritma pembelajaran mesin. Dalam tugas kelompok, eksplorasi berbagai model klasifikasi dan teknik evaluasi memungkinkan untuk melihat bagaimana fitur teks yang berbeda dapat mempengaruhi hasil. Pemahaman ini sangat penting dalam mengembangkan model yang akurat dan efisien, serta dalam membuat keputusan berbasis data yang dapat diandalkan.

Reinforcement Learning

Dasar Reinforcement Learning

CPL:

Mahasiswa memahami konsep agent-environment interaction.

CPMK:

Menjelaskan komponen reward, state, dan action.

SubCPMK:

Menggunakan simulasi sederhana.

A. Pengertian Reinforcement Learning

Reinforcement learning atau RL menjelaskan cara sebuah sistem belajar mengambil keputusan lewat pengalaman, lalu memperbaiki strategi berdasarkan umpan balik. Umpan balik itu berbentuk **reward**, yaitu sinyal nilai yang memberi tahu apakah pilihan barusan membantu menuju tujuan atau justru menjauh. RL berbeda dari pembelajaran yang mengandalkan kunci jawaban per contoh, karena RL mengutamakan rangkaian keputusan berurutan. Satu keputusan bisa terlihat benar sesaat, tetapi berdampak buruk pada langkah berikutnya. Karena itu RL mengejar hasil total jangka panjang, bukan kemenangan satu langkah.

Kerangka RL membangun cerita simpel: ada *agent* sebagai pengambil keputusan dan ada *environment* sebagai dunia tempat keputusan terjadi. Keduanya berinteraksi berulang. Setiap interaksi meninggalkan jejak data berupa apa yang terlihat, apa yang dipilih,

dan apa balasannya. Dari jejak itu, RL membentuk kebijakan tindakan yang makin efektif. Insight kuncinya tetap sama: pembelajaran penguatan memungkinkan sistem belajar dari umpan balik, lalu menyusun perilaku adaptif tanpa harus diberi daftar langkah yang benar satu per satu.

RL sering terasa seperti versi matematis dari proses coba coba yang terarah. Sistem mencoba berbagai tindakan, menerima **reward**, lalu mengubah perilaku supaya akumulasi **reward** naik. Proses ini bisa muncul pada permainan, navigasi rute, optimasi jadwal, atau kontrol perangkat. Kunci praktisnya ada pada definisi tujuan yang jelas melalui **reward**, serta definisi apa yang boleh dilihat dan dilakukan melalui *state* dan *action*.

B. Agent dan Environment

Siklus RL berjalan berulang *agent* mengamati *state*, memilih *action*, lalu *environment* merespons dengan **reward** dan *state* baru. Siklus itu kelihatan sederhana, tetapi ia memuat seluruh mekanisme belajar. *State* berfungsi sebagai ringkasan situasi saat ini, yaitu informasi yang dipakai untuk menentukan tindakan. *Action* adalah pilihan yang bisa diambil *agent*. *Reward* adalah sinyal evaluasi dari *environment*.

Tiga komponen tersebut perlu konsisten supaya belajar berjalan stabil. Jika *state* terlalu miskin informasi, *agent* kesulitan membedakan situasi penting. Jika *action* terlalu terbatas, *agent* tidak punya ruang untuk mencari strategi efektif. Jika **reward** ambigu, *agent* bisa belajar perilaku yang kelihatan benar tetapi sebenarnya menyimpang dari tujuan. Karena itu, desain RL selalu kembali ke pertanyaan dasar: situasi apa yang perlu diwakili sebagai *state*,

pilihan apa yang masuk akal sebagai *action*, dan sinyal apa yang paling jujur untuk merefleksikan kemajuan sebagai **reward**.

RL juga memakai struktur waktu. Interaksi sering dibagi menjadi **episode**, yaitu satu rangkaian langkah dari awal sampai selesai. Banyak masalah punya kondisi selesai, misalnya mencapai tujuan atau gagal. Episode memudahkan evaluasi, karena akumulasi **reward** per episode dapat dibandingkan dari waktu ke waktu. Proses belajar lalu terlihat sebagai kurva: mula mula acak, kemudian mulai menemukan pola, lalu semakin stabil.

C. Reward, State, dan Action

Reward memegang peran paling sensitif. Sinyal ini harus sejalan dengan tujuan. Jika tujuan adalah mencapai target secepat mungkin, **reward** bisa memberi nilai positif besar saat target tercapai serta memberi penalti kecil per langkah agar sistem tidak berlama lama. Jika tujuan adalah menjaga keselamatan, **reward** perlu memberi penalti keras pada tindakan berisiko. Kesalahan umum adalah membuat **reward** terlalu mudah dibajak, misalnya sistem menemukan cara menaikkan **reward** tanpa benar benar mencapai tujuan. Karena itu, desain **reward** perlu mengantisipasi celah perilaku dan meminimalkan peluang strategi menyimpang.

State adalah representasi keadaan yang terlihat oleh *agent*. Representasi ini bisa sederhana, misalnya posisi pada papan, atau lebih kaya seperti rangkaian sensor. Desain *state* sebaiknya cukup informatif untuk memprediksi konsekuensi dari *action*. Prinsip praktisnya: *state* perlu menampung informasi relevan dan menyingkirkan detail yang tidak membantu. Terlalu banyak detail bisa memperbesar ruang belajar sehingga butuh data lebih banyak.

Action adalah set pilihan yang diizinkan. Pilihan itu bisa diskret seperti kiri, kanan, atas, bawah, atau kontinu seperti sudut kemudi. Pilihan *action* menentukan tingkat kontrol yang dimiliki *agent*. Set *action* yang terlalu kasar membatasi kualitas strategi, sedangkan set *action* yang terlalu halus bisa membuat belajar lebih sulit. Kombinasi *state* dan *action* melahirkan konsep **policy**, yaitu aturan memilih tindakan berdasarkan *state*.

RL juga memakai ide *return*, yaitu total **reward** terdiskonto sepanjang waktu. Diskonto memakai faktor *gamma* untuk menyeimbangkan **reward** cepat dan **reward** masa depan. Diskonto membantu mencegah sistem terlalu mengejar **reward** jauh yang belum pasti, sekaligus tetap menghargai rencana jangka panjang. Diskonto yang terlalu kecil bisa membuat sistem terlalu pendek napas, sedangkan diskonto yang terlalu besar bisa membuat pembaruan menjadi lambat dan sensitif.

D. Jenis Jenis Algoritma Reinforcement Learning

Materi RL biasanya memetakan algoritma berdasarkan apa yang dipelajari. Pendekatan berbasis **value** mempelajari nilai keadaan atau nilai pasangan keadaan tindakan, lalu memilih tindakan yang punya nilai tertinggi. Contoh klasiknya adalah Q learning dan SARSA. Q learning mengejar nilai optimal dengan pembaruan yang memakai nilai maksimum langkah berikutnya, sedangkan SARSA memperbarui nilai berdasarkan tindakan yang benar benar diambil. Perbedaan itu membuat SARSA sering lebih hati hati saat eksplorasi agresif.

Pendekatan berbasis **policy** mempelajari aturan tindakan secara langsung. Contohnya keluarga *policy gradient* seperti

REINFORCE. Keunggulannya adalah fleksibel untuk ruang tindakan yang kompleks, termasuk aksi kontinu. Kelemahannya, pelatihan bisa punya varians tinggi, sehingga butuh teknik stabilisasi.

Pendekatan **actor critic** menggabungkan dua dunia. *Actor* bertugas memilih tindakan melalui *policy*, sedangkan *critic* menilai tindakan itu lewat perkiraan **value**. Kombinasi ini sering lebih stabil dibanding *policy gradient* murni, karena *critic* memberi sinyal koreksi yang lebih informatif. Secara intuisi, *actor* berperan sebagai pemain, sedangkan *critic* berperan sebagai pelatih yang memberi evaluasi.

Pada banyak kasus modern, RL memakai fungsi aproksimasi seperti jaringan saraf, sering disebut *deep reinforcement learning*. Contohnya *deep Q network* atau DQN yang memakai jaringan untuk memprediksi nilai Q. Penggunaan jaringan membuat sistem dapat menangani masukan besar seperti gambar. Risiko utamanya adalah ketidakstabilan pelatihan, sehingga teknik seperti *experience replay* dan *target network* dipakai untuk menenangkan dinamika pembaruan.

E. Prinsip Eksplorasi dan Eksploitasi

Prinsip eksplorasi dan eksploitasi adalah jantung RL. **Eksplorasi** berarti mencoba tindakan yang belum pasti hasilnya demi mengumpulkan informasi. **Eksploitasi** berarti memanfaatkan pengetahuan yang sudah ada untuk memaksimalkan **reward**. Terlalu banyak **eksploitasi** bisa membuat sistem terjebak pada strategi lokal yang kelihatan bagus di awal. Terlalu banyak **eksplorasi** membuat sistem seperti tidak pernah serius, karena terus mencoba hal acak.

Metode praktis yang mudah dipahami adalah *epsilon greedy*. Dengan probabilitas kecil *epsilon*, sistem memilih tindakan acak

untuk **eksplorasi**, sedangkan sisanya memilih tindakan terbaik menurut nilai saat ini untuk **eksploitasi**. Strategi ini bisa memakai *epsilon* yang menurun seiring waktu, supaya awalnya banyak mencoba, lalu makin fokus saat pengetahuan meningkat. Alternatifnya adalah *softmax* atau *Boltzmann exploration* yang memilih tindakan secara probabilistik berdasarkan nilai, sehingga tindakan bagus lebih sering dipilih tetapi tindakan lain tetap punya peluang.

Simulasi sederhana membantu mengunci intuisi. Contoh yang sangat populer adalah *gridworld*. Bayangkan papan kotak dengan satu titik awal dan satu titik tujuan. *State* adalah koordinat posisi. *Action* adalah gerak satu langkah: atas, bawah, kiri, kanan. **Reward** bisa dirancang sebagai nilai positif besar saat mencapai tujuan dan penalti kecil per langkah supaya rute pendek lebih disukai. Jalankan banyak **episode**. Pada awal, gerak acak membuat sering mentok. Setelah pembaruan nilai berulang, kebijakan mulai membentuk jalur yang konsisten menuju tujuan.

Simulasi ini menegaskan ide penting: RL belajar dari umpan balik yang berulang, bukan dari ceramah aturan. Ketika desain *state*, *action*, dan **reward** selaras dengan tujuan, sistem dapat menemukan strategi yang efektif lewat pengalaman. Ketika desainnya tidak selaras, sistem tetap belajar, hanya saja ia belajar hal yang salah. Karena itu materi RL selalu kembali ke tiga hal yang wajib jelas: definisi *agent environment interaction*, kejelasan komponen *reward state action*, serta menyeimbangkan **eksplorasi** dan **eksploitasi** supaya proses belajar tidak tersesat.

Kode Program: RL dengan Q-learning Rekomendasi Barang.



Kode program ini membangun *reinforcement learning* berbasis Q learning untuk simulasi rekomendasi barang pada dataset Online Retail dari UCI. Pipeline dimulai dari instalasi lingkungan minimal di Google Colab, dilanjutkan akuisisi data melalui `fetch_ucirepo`, pembersihan data yang transparan, pembentukan keranjang dan episode, perancangan *state action reward*, pelatihan Q table, lalu evaluasi akhir memakai metrik kuantitatif dan pembandingan baseline. Tujuan utamanya ialah menghasilkan kebijakan rekomendasi diskret yang belajar dari umpan balik transaksi berikutnya, sesuai kerangka RL pada dokumen: *agent* memilih rekomendasi, *environment* merespons lewat kemunculan item, dan sistem memperbarui nilai Q untuk memperbaiki **policy**.

Strategi akuisisi data memakai jalur utama `data.original` agar kolom identitas transaksi `InvoiceNo` dan `StockCode` ikut terbaca. Revisi ini penting karena pada sebagian konfigurasi dataset, kolom identitas dapat berada pada bagian *ids* sehingga tidak otomatis masuk ke *features*. Sesudah data terbaca, normalisasi nama kolom dilakukan dengan menghapus spasi agar validasi kolom wajib tidak gagal karena perbedaan format. Validasi kolom wajib memastikan

pipeline tidak melanjutkan proses ketika struktur data tidak lengkap, sehingga integritas proses terjaga sejak awal.

Tahap prapemrosesan berjalan dengan audit jumlah baris pada setiap filter sehingga pembersihan dapat ditelusuri secara kuantitatif. Output menunjukkan ukuran data awal **541909 baris dan 8 kolom** dan turun menjadi **397884 baris dan 8 kolom** setelah pembersihan. Audit memperlihatkan penghapusan nilai kosong pada kolom wajib mengurangi **135080 baris**, penghapusan transaksi pembatalan mengurangi **8905 baris**, dan penyaringan pembelian tidak valid mengurangi **40 baris**. Validasi CustomerID numerik tidak mengurangi baris pada hasil ini. Audit ini krusial karena RL berbasis urutan transaksi rentan bias jika data kotor atau pembatalan tercampur, sehingga sinyal **reward** bisa menyesatkan.

Pembentukan keranjang belanja dilakukan per kombinasi pelanggan, invoice, dan waktu transaksi, lalu setiap keranjang memuat daftar item unik `BasketItems` dan nilai transaksi `NilaiKeranjang`. Output menghasilkan **18562 keranjang valid**. Selanjutnya, episode disusun per pelanggan dengan syarat minimal dua keranjang sehingga terbentuk **2847 episode**. Pembagian latihan uji dilakukan berbasis pelanggan untuk mencegah kebocoran informasi, dengan hasil **2277 pelanggan latihan** dan **570 pelanggan uji**. Strategi ini memastikan evaluasi menguji kemampuan generalisasi kebijakan pada pelanggan yang tidak pernah dilihat saat latihan.

Desain RL mengikuti tiga komponen utama. Pertama, *state* dibangun sebagai ringkasan histori keranjang, bukan urutan mentah item di dalam keranjang. Fungsi `get_state` melihat beberapa

keranjang terakhir, menghitung frekuensi item, lalu memilih tiga token teratas. Agar ruang *state* tidak membengkak, item langka dikompresi menjadi token LAINNYA dengan batas TOP M sama dengan **200**, sedangkan kekosongan dipad dengan token KOSONG. Kedua, *action* dibatasi menjadi ruang aksi diskret **K sama dengan 50** item teratas berdasarkan frekuensi. Pembatasan ini menjaga Q table tetap realistis untuk dipelajari. Ketiga, *reward* dirancang sebagai umpan balik berbobot: reward positif diberikan jika rekomendasi muncul pada keranjang berikutnya, lalu dibobot dengan anti popularitas untuk mengurangi bias item terlaris, ditambah bonus kecil yang terkait NilaiKeranjang sebagai proksi nilai transaksi. Penalti diterapkan untuk rekomendasi item yang sudah ada pada keranjang saat ini dan penalti repetisi jika rekomendasi berturut turut sama.

Agar pemilihan aksi lebih logis, kandidat rekomendasi dibangun dari ko kemunculan item dalam keranjang yang sama. Kandidat disusun berdasarkan item yang sering muncul bersama item pada histori, sementara token non item seperti KOSONG dan LAINNYA diabaikan agar kandidat tidak bias. Visual frekuensi 10 item teratas memberi konteks distribusi item dan memperjelas mengapa baseline top 1 populer relevan sebagai pembanding.

Pelatihan Q learning menggunakan pembaruan temporal difference dengan parameter alpha dan gamma, serta strategi *epsilon greedy* untuk eksplorasi. Evaluasi dilakukan pada data uji dengan epsilon nol agar mencerminkan eksploitasi kebijakan hasil belajar. Kode juga menambahkan baseline yang selalu merekomendasikan item paling populer untuk memastikan

interpretasi kinerja tidak terjebak pada angka reward semata. Evaluasi akhir pada data uji menunjukkan reward rata rata per langkah **minus 0.015242** dan reward rata rata per episode **minus 0.012512** pada **2612 langkah**. Nilai reward negatif konsisten dengan adanya penalti yang cukup sering aktif pada fase uji.

Metrik tambahan memberi gambaran yang lebih adil. Hit rate uji model mencapai **0.107198**, lebih tinggi daripada baseline hit rate **0.092649**, sehingga kebijakan Q learning lebih sering tepat menebak item yang muncul pada transaksi berikutnya dibanding strategi naif. Cakupan rekomendasi mencapai **0.86**, menunjukkan model memakai sekitar 86 persen dari ruang aksi 50 item, sehingga rekomendasi tidak terkunci pada satu item saja. Pada saat yang sama, rasio repetisi **0.325421** menandakan sekitar sepertiga langkah mengalami pengulangan rekomendasi berturut turut, yang selaras dengan tekanan penalti repetisi dan ikut menjelaskan mengapa reward rata rata uji tetap negatif.

Ukuran data awal: (541909, 8)

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

Ukuran data setelah prapemrosesan: (397884, 8)

	Tahap	Sebelum	Sesudah	Gugur
0	Hapus baris dengan nilai kosong pada kolom wajib	541909	406829	135080
1	Hapus transaksi pembatalan (InvoiceNo diawali C)	406829	397924	8905
2	Saring pembelian valid (Quantity>0 dan UnitPri...	397924	397884	40
3	Validasi CustomerID numerik	397884	397884	0

Jumlah keranjang valid: 18562

Gambar 1. Prapemrosesan Transaksi Online Retail.

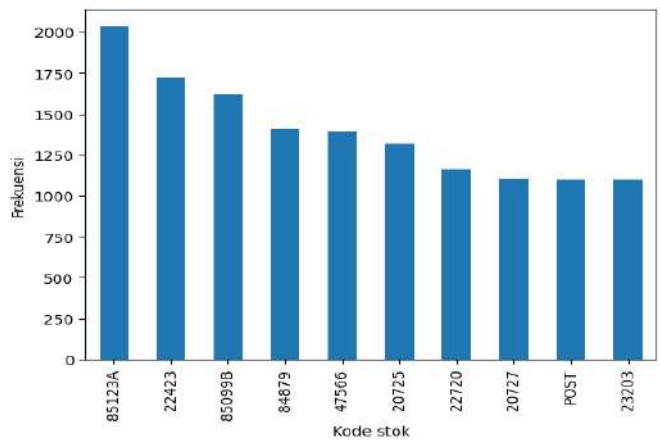
Visual ini menunjukkan perubahan ukuran data dari **541909** menjadi **397884** baris. Rincian audit mencatat **135080 baris** gugur

karena nilai kosong, **8905 baris** gugur karena pembatalan, dan **40 baris** gugur karena pembelian tidak valid. Setelah pembersihan, pipeline membentuk **18562 keranjang valid** sebagai dasar episode.

	CustomerID	InvoiceNo	InvoiceDate	BasketItems	NilaiKeranjang
0	12346	541431	2011-01-18 10:01:00	[23166]	77183.60
1	12347	537626	2010-12-07 14:57:00	[20780, 20782, 21064, 21171, 21731, 22195, 222...	711.79
2	12347	542237	2011-01-26 14:30:00	[20719, 20966, 21035, 21041, 21154, 21731, 218...	475.39
3	12347	549222	2011-04-07 10:43:00	[16008, 20665, 21041, 21731, 21791, 21975, 221...	636.25
4	12347	556201	2011-06-09 13:01:00	[17021, 22196, 22371, 22375, 22725, 22727, 230...	382.52

Gambar 2. Contoh Keranjang Belanja Per Invoice.

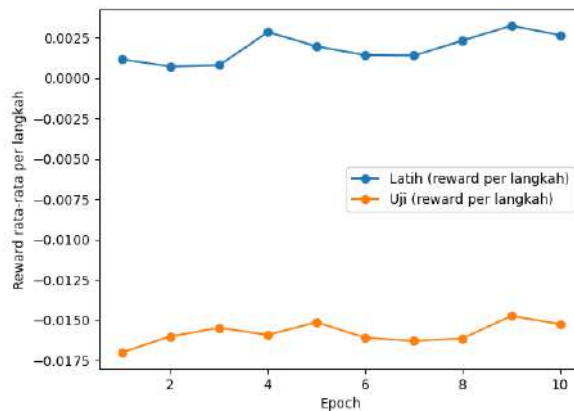
Tabel keranjang menampilkan CustomerID, InvoiceNo, InvoiceDate, daftar BasketItems, dan NilaiKeranjang. Contoh keluaran memperlihatkan CustomerID **12346** pada InvoiceNo **541431** dengan NilaiKeranjang **77183.60**, serta CustomerID **12347** pada beberapa invoice berurutan dengan nilai seperti **711.79** dan **475.39**, sehingga urutan transaksi dapat diperlakukan sebagai episode.



Gambar 3. Frekuensi 10 Item Paling Sering Muncul

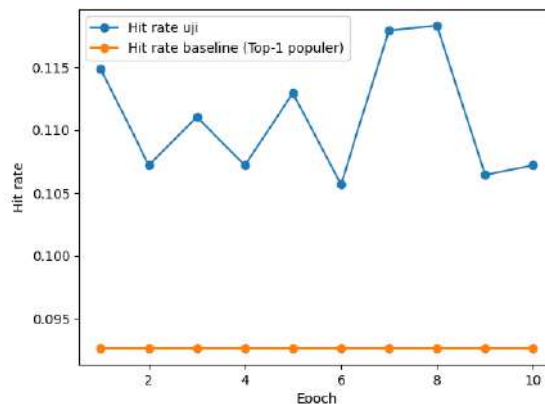
Sebagai Dasar Ruang Aksi. Diagram batang menunjukkan kode stok **85123A** sebagai item paling sering muncul, diikuti kode

seperti **22423** dan **85099B**. Skala frekuensi berada pada ribuan kemunculan, sehingga pembatasan ruang aksi menjadi **50 item** masuk akal untuk Q table dan baseline top 1 populer relevan.



Gambar 4. Perkembangan Reward Rata rata

Per Langkah pada Latih dan Uji. Kurva memperlihatkan reward latih per langkah sedikit di atas nol, sedangkan reward uji per langkah berada di bawah nol dan stabil di kisaran minus. Ringkasan evaluasi uji mencatat reward per langkah **minus 0.015242** dengan total **2612 langkah**, konsisten dengan desain reward yang memasukkan penalti repetisi dan penalti rekomendasi yang tidak tepat.



Gambar 5. Perbandingan Hit Rate Kebijakan Q learning

Baseline Item Terpopuler. Grafik menunjukkan hit rate uji model **0.107198** dan baseline **0.092649**, dengan selisih sekitar **0.014549**. Metrik pendukung mencatat cakupan rekomendasi **0.86** dan rasio repetisi **0.325421**, sehingga peningkatan hit rate tetap perlu dibaca bersama dampak penalti repetisi pada reward.

	Metrik	Nilai
0	Reward rata-rata per episode	-0.012512
1	Reward rata-rata per langkah	-0.015242
2	Hit rate	0.107198
3	Cakupan rekomendasi	0.860000
4	Rasio repetisi	0.325421
5	Baseline reward rata-rata per langkah	-0.051108
6	Baseline hit rate	0.092649
7	Jumlah langkah	2612.000000

	Metrik	Nilai
0	Pelanggan dengan minimal 2 keranjang	2847
1	Jumlah pelanggan latih	2277
2	Jumlah pelanggan uji	570
3	Ukuran ruang aksi K	50
4	Ukuran token eksplisit M	200
5	Jumlah state unik Q-table (akhir)	9727
6	Epsilon akhir	0.95111
7	Baseline item (Top-1 populer)	85123A

Gambar 6. Ringkasan Metrik Evaluasi Kebijakan pada Data Uji

Gambar ini merangkum hasil evaluasi metrik pada data uji untuk model rekomendasi berbasis Q-learning. **Reward rata-rata per episode** tercatat **minus 0.012512**, yang mengindikasikan penalti cukup sering aktif meskipun model berhasil meningkatkan **hit rate** menjadi **0.107198**, lebih tinggi dari baseline **0.092649**. Dengan **cakupan rekomendasi** sebesar **0.86**, model dapat

memanfaatkan sekitar **50 item** dalam ruang aksi dengan efektif. Namun, **rasio Repetisi rekomendasi** yang mencapai **0.325421** menunjukkan bahwa sekitar **32.5%** dari langkah-langkah melibatkan rekomendasi yang berulang, yang menambah penalti negatif pada reward total. **Jumlah langkah** tercatat sebanyak **2612 langkah**, dengan **9727 state unik** pada Q-table, menunjukkan bahwa model tidak menciptakan *state* baru secara berlebihan dan pembelajaran berjalan stabil. **Epsilon akhir** tercatat **0.95111**, yang menunjukkan bahwa eksplorasi masih mendominasi proses pelatihan dan perlu penyesuaian agar lebih cepat masuk ke fase eksploitasi.

Insight RL adalah bahwa pembelajaran penguatan memungkinkan sistem belajar dari umpan balik, tetapi kualitas hasil sangat ditentukan oleh desain *state*, ruang *action*, dan **reward** yang seimbang. Jika penalti terlalu dominan, reward dapat negatif walau hit rate membaik. Untuk iterasi berikutnya ialah mempercepat penjadwalan epsilon agar eksplorasi menurun lebih tegas menuju eksploitasi, serta mengkalibrasi penalti pengembangan lanjut adalah **penjadwalan epsilon, kalibrasi penalti, stabilitas state, pemilihan kandidat aksi**, dan **baseline evaluasi**.

Studi Kasus: RL Prediksi Permintaan Bike Sharing.



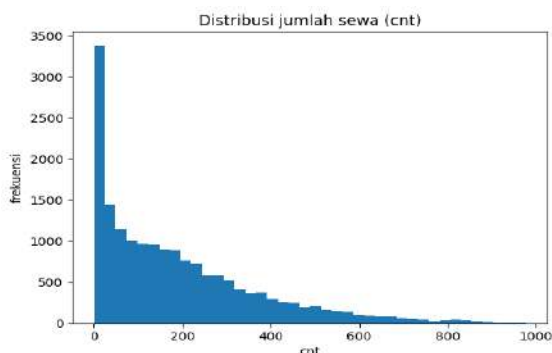
Kode program ini membangun sebuah *pipeline* analisis dan simulasi **pengambilan keputusan operasional berbasis reinforcement learning** pada *Bike Sharing Dataset* dari UCI. Alur kerja dimulai dari penyiapan lingkungan komputasi di Google Colab, dilanjutkan dengan akuisisi dan validasi data *hourly*, eksplorasi karakteristik permintaan, pemodelan prediksi dasar berbasis *time-aware machine learning*, hingga perancangan dan evaluasi agen RL. Tujuan utamanya ialah menghasilkan **kebijakan diskret promo dan kapasitas** yang mampu belajar dari umpan balik permintaan aktual, selaras dengan kerangka RL: *agent* memilih aksi, *environment* merespons melalui realisasi permintaan dan penalti, lalu sistem memperbarui kebijakan untuk meningkatkan *return* jangka pendek.-

Strategi akuisisi data menekankan **integritas deret waktu** dengan parsing tanggal dan pengurutan berdasarkan kombinasi hari dan jam. Revisi penting dilakukan dengan mengecualikan variabel *casual* dan *registered* dari fitur, karena keduanya secara deterministik membentuk variabel target *cnt*. Langkah ini mencegah kebocoran informasi dan memastikan bahwa baik model prediksi maupun agen RL belajar dari sinyal eksogen seperti kalender dan cuaca. Validasi struktur data dan audit nilai hilang memastikan pipeline tidak melanjutkan proses ketika kualitas data tidak memadai.-

Tahap eksplorasi data dilakukan untuk memahami struktur permintaan sebelum pemodelan lanjutan. Distribusi permintaan menunjukkan ketidakseimbangan yang kuat, sementara pola temporal dan pengaruh cuaca memperlihatkan determinan struktural permintaan. Temuan ini menjadi dasar logis bagi pemilihan

fitur dan desain *reward* pada simulasi RL. Baseline prediksi kemudian dibangun menggunakan *HistGradientBoostingRegressor* dengan *TimeSeriesSplit* untuk menjaga kausalitas waktu. Model ini tidak dimaksudkan sebagai solusi akhir, melainkan sebagai pembanding realistis terhadap kebijakan adaptif berbasis RL.-

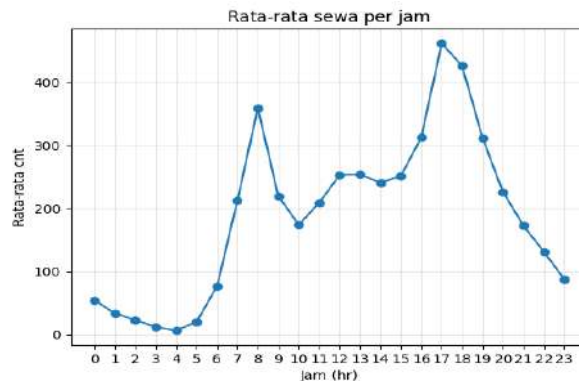
Simulasi RL dirancang dalam lingkungan *BikeOpsEnv* dengan episode 24 langkah yang merepresentasikan satu hari operasional. *State* dibentuk dari variabel kalender dan cuaca, *action* dibatasi menjadi kombinasi diskret tingkat promo dan kapasitas, dan *reward* dirancang sebagai fungsi ekonomi sederhana yang menyeimbangkan pendapatan, biaya, dan penalti *unmet demand*. Tiga kebijakan dibandingkan, yakni kebijakan acak sebagai baseline, DQN sebagai pendekatan *value-based*, dan PPO sebagai pendekatan *policy-gradient* yang lebih stabil. Evaluasi dilakukan menggunakan *cumulative reward*, *mean return*, dan *standard deviation* untuk menilai performa dan reliabilitas kebijakan.



Gambar 1. Distribusi Jumlah Sewa Sepeda (*cnt*)

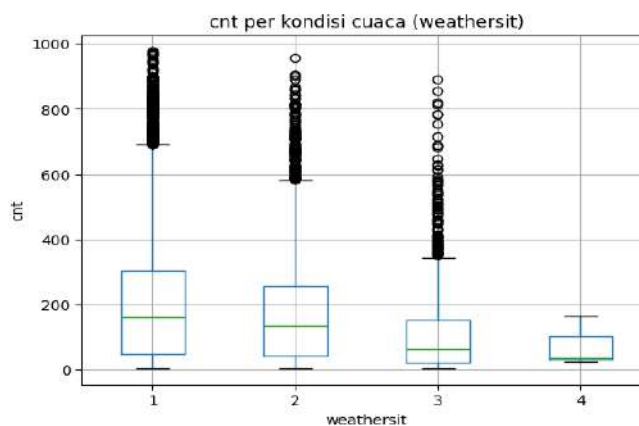
Histogram memperlihatkan distribusi permintaan yang sangat miring ke kanan, dengan mayoritas observasi berada di bawah 300 sewa per jam dan nilai ekstrem mendekati 1000 hanya muncul pada

sebagian kecil jam. Pola ini menegaskan adanya *peak demand* yang jarang namun berdampak besar terhadap total kinerja sistem.



Gambar 2. Rata-rata Jumlah Sewa per Jam

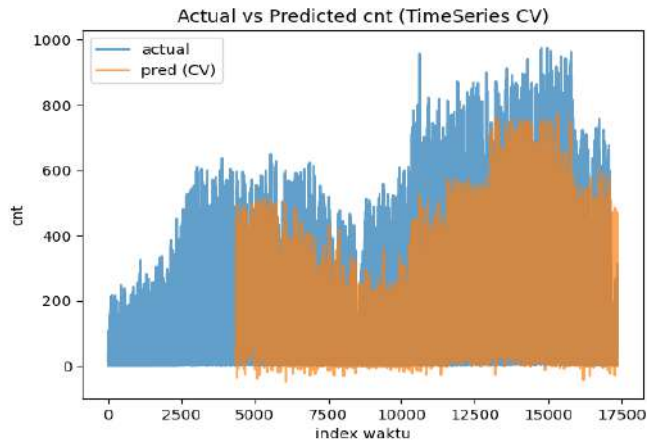
Grafik rata-rata per jam menunjukkan dua puncak utama, yakni sekitar jam 8 dengan rata-rata ± 350 sewa dan jam 17-18 dengan rata-rata ± 450 sewa. Jam dini hari mencatat rata-rata di bawah 20 sewa. Pola ini mencerminkan mobilitas komuter dan menegaskan pentingnya variabel waktu.



Gambar 3. Jumlah Sewa (*cnt*) Berdasarkan Kondisi Cuaca

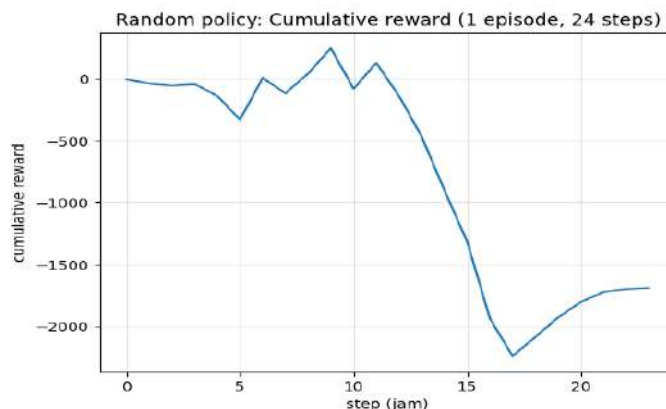
Boxplot menunjukkan median permintaan sekitar 160-180 pada cuaca cerah, turun ke sekitar 120 pada kondisi mendung, dan di bawah 80 pada hujan atau cuaca buruk. Variansi permintaan juga

menyempit pada cuaca ekstrem, menandakan elastisitas permintaan yang menurun.



Gambar 4. Perbandingan Nilai Aktual dan Prediksi

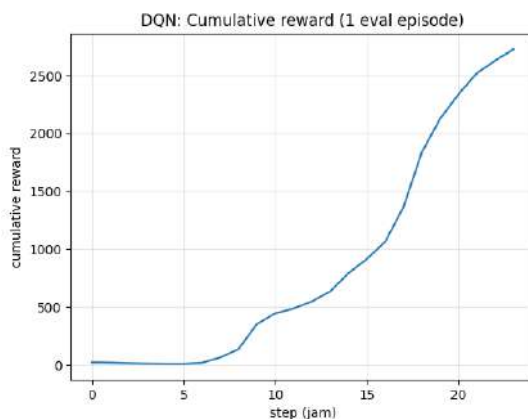
Kurva prediksi mengikuti tren umum data aktual sepanjang sekitar 17.000 indeks waktu, tetapi amplitudo puncak lebih rendah. Nilai *MAE* dan *RMSE* berada pada kisaran moderat dengan R^2 positif, menunjukkan model menangkap struktur utama namun *underestimate* lonjakan permintaan.



Gambar 5. *Cumulative Reward* (1 Episode, 24 Jam)

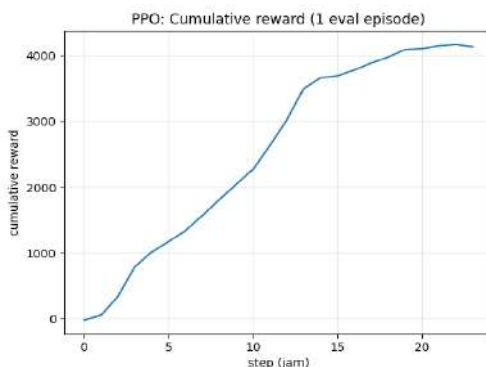
Kurva *cumulative reward* kebijakan acak menurun hingga sekitar -2200 pada akhir episode. Hasil ini menunjukkan bahwa keputusan tanpa pembelajaran menyebabkan biaya promo, biaya

kapasitas, dan penalti *unmet demand* mendominasi, sehingga menghasilkan kerugian bersih.



Gambar 6. *Cumulative Reward* Kebijakan DQN

Kurva DQN meningkat hingga sekitar 2700 pada akhir episode dengan fluktuasi sedang. Kinerja ini jauh lebih baik dibanding kebijakan acak, tetapi variasi antar langkah masih cukup besar, sejalan dengan *standard deviation* return yang relatif tinggi.



Gambar 7. *Cumulative Reward* Kebijakan

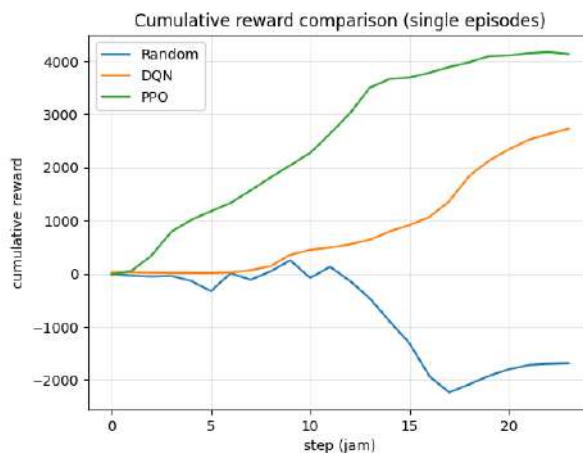
Kurva PPO meningkat stabil hingga sekitar 4200 pada akhir episode. Kenaikan yang relatif halus mencerminkan *mean return* tertinggi dan *standard deviation* terendah, menunjukkan bahwa PPO menghasilkan kebijakan paling konsisten dalam lingkungan ini.

Perbandingan performa (return):

	model	mean_return	std_return
0	Random baseline	586.4540	1167.329674
1	DQN	3526.2956	2001.840353
2	PPO	4601.9460	569.419626

Gambar 8. *Cumulative Reward* Random, DQN, dan PPO

Grafik komparatif menunjukkan kebijakan Random terus berada di zona negatif, DQN meningkat moderat, dan PPO mendominasi sepanjang episode. Pada langkah ke-24, selisih *reward* PPO terhadap DQN melebihi 1500 poin, menegaskan keunggulan kebijakan.



Gambar 9. Tabel Perbandingan Performa

Tabel evaluasi mencatat *mean return* Random sebesar 586 dengan *std* 1167, DQN sebesar 3526 dengan *std* 2002, dan PPO sebesar 4602 dengan *std* 569. Nilai ini menegaskan bahwa PPO unggul baik dari sisi kinerja maupun stabilitas.

Preview kebijakan PPO (10 step):

	step	action_id	promo_level	cap_level	capacity	base_demand	demand_after_promo	realized	unmet
0	0	24	4	4	563.1	17.0	21.08	21.08	0.0
1	1	24	4	4	563.1	93.0	115.32	115.32	0.0
2	2	24	4	4	563.1	263.0	326.12	326.12	0.0
3	3	24	4	4	563.1	398.0	493.52	493.52	0.0
4	4	24	4	4	563.1	218.0	270.32	270.32	0.0
5	5	24	4	4	563.1	165.0	204.60	204.60	0.0
6	6	24	4	4	563.1	165.0	204.60	204.60	0.0
7	7	24	4	4	563.1	222.0	275.28	275.28	0.0
8	8	24	4	4	563.1	232.0	287.68	287.68	0.0
9	9	24	4	4	563.1	220.0	272.80	272.80	0.0

Gambar 10. *Preview Kebijakan PPO (10 Langkah Awal)*

Tabel menunjukkan PPO secara konsisten memilih promo level 4 dan kapasitas level 4 dengan kapasitas sekitar 563.1. *Demand after promo* meningkat, misalnya dari 398 menjadi 493.52, dengan *unmet demand* bernilai nol, menandakan kebijakan agresif namun efektif.-

Secara keseluruhan, hasil ini menunjukkan bahwa *reinforcement learning* mampu menghasilkan kebijakan operasional yang unggul dibanding strategi statis. **Insight** bahwa kualitas kebijakan sangat ditentukan oleh desain *state*, pembatasan ruang aksi yang realistis, dan keseimbangan *reward*. Jika penalti terlalu dominan, *reward* dapat tetap negatif meskipun indikator lain membaik. Iterasi selanjutnya secara logis diarahkan pada penjadwalan eksplorasi yang lebih tegas, kalibrasi biaya promo dan kapasitas, serta pengayaan fitur waktu agar kebijakan semakin stabil, realistis, dan relevan untuk pengambilan keputusan.

Soal

1. Jelaskan perbedaan antara *reward jangka pendek* dan *reward jangka panjang* dalam Reinforcement Learning. Mengapa memilih strategi yang memaksimalkan *return* jangka panjang sering lebih penting?
2. Dalam suatu lingkungan RL, *state* yang terlalu kaya informasi bisa memperlambat pembelajaran. Berikan dua contoh konkret keadaan di mana hal ini terjadi, kemudian jelaskan bagaimana Anda akan merancang *state* yang lebih ringkas.
3. Apa sebabnya *exploration* tetap diperlukan meskipun sebuah *policy* sudah menunjukkan performa yang baik? Jelaskan dampaknya terhadap *convergence* dan *optimalitas* kebijakan.
4. Bandingkan pendekatan *value-based* dan *policy-based* secara konseptual. Berikan satu contoh masalah di mana *policy-based RL* lebih efektif dibanding *value-based RL*.
5. Jelaskan langkah-langkah pembaruan nilai dalam Q-learning. Mengapa metode ini disebut *off-policy learning*?
6. Pada simulasi sederhana seperti *gridworld*, langkah agent seringkali keliru di awal. Mengapa hal ini masih membantu pembelajaran?
7. Bagaimana desain *reward* yang salah bisa menimbulkan *behavioral hacking* pada agent? Berikan contoh yang mungkin terjadi di dunia nyata.
8. Dalam RL dengan *discount factor* (γ), apa konsekuensi dari memilih γ mendekati nol versus mendekati satu? Jelaskan dengan konteks pembelajaran jangka panjang vs. jangka pendek.

9. Sebut dan jelaskan dua metrik evaluasi yang penting untuk menilai kebijakan RL dalam simulasi rekomendasi item (seperti *hit rate* dan *coverage* dalam studi sebelumnya).
10. Jelaskan apa yang dimaksud *epsilon-greedy* policy. Bagaimana strategi *epsilon decay* meningkatkan kinerja RL dari tahap awal sampai akhir pelatihan?

Tugas Kelompok

Studi Kasus 1 Optimasi Rantai Pasokan dengan Reinforcement Learning

Tugas:

Kelompok Anda diminta untuk mengembangkan model *Reinforcement Learning* (RL) yang dapat mengoptimalkan pengelolaan rantai pasokan barang dalam sebuah sistem distribusi. Model ini bertujuan untuk meminimalkan biaya logistik dan waktu pengiriman barang berdasarkan data yang ada. Anda perlu mendefinisikan *state*, *action*, dan *reward* yang tepat agar sistem dapat membuat keputusan pengiriman yang efisien. Misalnya, *state* dapat berupa status persediaan dan permintaan di berbagai lokasi, sementara *action* berhubungan dengan keputusan pengiriman barang ke lokasi tertentu. *Reward* bisa dirancang berdasarkan efisiensi pengiriman dan biaya yang dikeluarkan. Selanjutnya, evaluasi model RL yang Anda buat dengan membandingkan performanya dengan pendekatan sistem distribusi tradisional yang ada.

Link: [Supply Chain Dataset - UCI](#)

Studi Kasus 2 Pengelolaan Sumber Daya Energi dengan Reinforcement Learning

Tugas:

Kelompok Anda diminta untuk mengembangkan model *Reinforcement Learning* yang mengoptimalkan penggunaan energi dalam sistem kota pintar berdasarkan data konsumsi energi historis serta prediksi cuaca. Anda harus menentukan komponen *state* (misalnya data historis konsumsi energi dan cuaca), *action* (keputusan untuk mengatur distribusi energi atau menyesuaikan konsumsi energi pada waktu tertentu), dan *reward* (misalnya, pengurangan pemborosan energi atau biaya pengelolaan energi). Setelah itu, bandingkan kebijakan pengelolaan energi yang dihasilkan oleh sistem RL dengan kebijakan pengelolaan energi tradisional yang ada dan evaluasi dampaknya terhadap penghematan energi di kota pintar.

Link: [Energy Consumption Dataset - Kaggle](#)

Studi Kasus 3: Sistem Rekomendasi Dinamis pada E-Commerce dengan Fashion MNIST

Tugas:

Kelompok Anda diminta untuk mengembangkan sistem rekomendasi berbasis *Reinforcement Learning* yang dapat menawarkan produk fashion kepada pengguna berdasarkan riwayat interaksi mereka dengan produk sebelumnya. Gunakan dataset *Fashion MNIST* untuk melatih model rekomendasi dengan memanfaatkan gambar produk pakaian. Anda perlu mendesain komponen *state* (misalnya, representasi visual dari gambar produk

yang telah dibeli atau dilihat pengguna), *action* (pemilihan produk tertentu untuk direkomendasikan), dan *reward* (misalnya, berdasarkan pembelian atau klik produk yang direkomendasikan). Setelah itu, evaluasi kebijakan rekomendasi yang dihasilkan menggunakan metrik seperti *hit rate* dan *coverage*, serta lakukan analisis pengaruh eksplorasi versus eksploitasi terhadap kualitas rekomendasi yang diberikan.

Link: [Fashion MNIST Dataset - Kaggle](#)

Refleksi Pembelajaran

Pembelajaran tentang Reinforcement Learning (RL) memberikan pemahaman yang mendalam tentang cara sistem dapat belajar dan beradaptasi melalui umpan balik dari tindakan yang diambil. Konsep utama seperti *state*, *action*, dan *reward* memungkinkan agent untuk membuat keputusan yang mengarah pada tujuan jangka panjang. Dalam konteks aplikasi nyata, seperti optimasi rantai pasokan, pengelolaan energi, dan sistem rekomendasi, RL terbukti sangat berguna untuk meningkatkan efisiensi dan pengambilan keputusan. Dengan membandingkan metode RL dengan pendekatan tradisional, kita bisa melihat potensi besar dalam mengotomatisasi dan mengoptimalkan berbagai sistem yang kompleks, meningkatkan produktivitas dan kepuasan pengguna.

Contoh Soal UTS

Nama : _____

NPM : _____

Prodi : _____

Hari : _____

Nama Dosen Pengampu: _____

Nama Matakuliah: _____

Semester: _____

Soal

1. Jelaskan peran Python dalam data science dan bagaimana Python membantu dalam proses analisis data.
2. Apa yang dimaksud dengan 'train-test split' dalam konteks machine learning? Jelaskan pentingnya teknik ini dan bagaimana penerapannya dalam membangun model prediktif.
3. Jelaskan apa itu data preprocessing dalam data science. Berikan contoh langkah-langkah yang dapat dilakukan pada dataset sebelum digunakan untuk membangun model.
4. Bagaimana cara menggunakan Google Colab dalam pengolahan data? Jelaskan proses langkah-langkah yang perlu dilakukan dari pemuatan dataset hingga visualisasi data.
5. Apa itu supervised learning dan unsupervised learning? Jelaskan perbedaan utama antara kedua pendekatan ini beserta contoh kasus penerapannya.

6. Pada algoritma K-means, jelaskan bagaimana proses klusterisasi dilakukan. Apa yang dimaksud dengan 'centroid' dan bagaimana pengaruhnya terhadap hasil klusterisasi?
7. Jelaskan konsep 'Elbow Method' dalam pemilihan jumlah kluster yang optimal pada K-means. Apa tujuan dari teknik ini dan bagaimana cara menafsirkan hasilnya?
8. Sebutkan dan jelaskan dua cara yang digunakan untuk mengevaluasi kualitas hasil klusterisasi K-means.
9. Dalam konteks unsupervised learning, apa yang dimaksud dengan 'inertia' dan bagaimana ini digunakan untuk mengevaluasi hasil klusterisasi?
10. Jelaskan kelebihan dan kekurangan dari algoritma K-means dalam melakukan klusterisasi. Bagaimana algoritma ini dapat dipengaruhi oleh outlier dalam dataset?

Studi Kasus

Studi Kasus 1:

Pengelompokan Pelanggan Berdasarkan Pengeluaran

Deskripsi Kasus:

Sebuah perusahaan e-commerce ingin memahami pola pengeluaran pelanggan mereka untuk mengoptimalkan strategi pemasaran. Data yang tersedia mencakup pengeluaran pelanggan pada berbagai kategori produk (misalnya: makanan, pakaian, elektronik). Tugas Anda adalah menggunakan algoritma K-means untuk mengelompokkan pelanggan berdasarkan pengeluaran mereka.

Tugas:

1. Lakukan eksplorasi data untuk menganalisis distribusi pengeluaran pelanggan pada setiap kategori produk.
2. Terapkan algoritma K-means untuk mengelompokkan pelanggan dalam sejumlah kluster yang optimal.
3. Tentukan jumlah kluster yang optimal menggunakan metode Elbow.
4. Evaluasi hasil klasterisasi dan berikan interpretasi karakteristik setiap kluster.
5. Diskusikan bagaimana kluster-kluster ini dapat digunakan untuk merancang strategi pemasaran yang lebih terfokus.

Studi Kasus 2:

Klasifikasi Data Kesehatan menggunakan K-means

Deskripsi Kasus:

Sebuah rumah sakit memiliki data rekam medis pasien yang mencakup berbagai atribut seperti usia, tekanan darah, kadar kolesterol, dan riwayat penyakit. Tugas Anda adalah mengelompokkan pasien berdasarkan risiko kesehatan mereka menggunakan algoritma K-means.

Tugas:

1. Lakukan pembersihan dan eksplorasi data pada atribut yang relevan untuk klasifikasi risiko kesehatan.
2. Terapkan K-means untuk mengelompokkan pasien berdasarkan profil risiko kesehatan mereka.
3. Tentukan jumlah kluster yang optimal menggunakan metode Elbow.

4. Evaluasi hasil klasterisasi dan interpretasikan karakteristik utama setiap klaster.
5. Berikan rekomendasi kepada rumah sakit mengenai tindakan pencegahan atau pengobatan berdasarkan hasil klasterisasi.

Contoh Soal UAS

Nama : _____

NPM : _____

Prodi : _____

Hari : _____

Nama Dosen Pengampu: _____

Nama Matakuliah: _____

Semester: _____

Soal

1. Jelaskan prinsip dasar algoritma K-means clustering dalam unsupervised learning. Bagaimana cara kerja algoritma ini dalam mengelompokkan data?
2. Apa itu 'silhouette score' dan bagaimana cara menggunakan metrik ini untuk menilai kualitas klasterisasi dalam K-means?
3. Jelaskan apa yang dimaksud dengan 'feature scaling' dalam konteks K-means clustering. Mengapa teknik ini penting sebelum melakukan klasterisasi?
4. Apa perbedaan antara K-means dan DBSCAN dalam hal klasterisasi? Jelaskan kelebihan dan kekurangan masing-masing algoritma.
5. Jelaskan apa itu algoritma Principal Component Analysis (PCA) dan bagaimana PCA dapat digunakan untuk mengurangi dimensi data dalam kasus klasterisasi.
6. Pada kasus unsupervised learning, bagaimana cara menentukan jumlah klaster yang optimal? Jelaskan berbagai metode yang dapat digunakan untuk tujuan ini.

7. Jelaskan apa yang dimaksud dengan Reinforcement Learning (RL) dan bagaimana RL berbeda dengan supervised dan unsupervised learning.
8. Berikan penjelasan tentang konsep 'reward' dan 'punishment' dalam Reinforcement Learning. Bagaimana kedua hal ini digunakan untuk mengarahkan agen dalam proses belajar?
9. Jelaskan secara rinci cara kerja algoritma Q-learning dalam Reinforcement Learning. Bagaimana cara menentukan nilai optimal Q dalam algoritma ini?
10. Apa itu Deep Q-Network (DQN)? Jelaskan bagaimana DQN menggabungkan teknik deep learning dengan Q-learning untuk meningkatkan performa agen dalam lingkungan yang lebih kompleks.

Studi Kasus 1:

Pengelompokan Pelanggan Berdasarkan Preferensi Produk dengan K-means

Deskripsi Kasus:

Sebuah perusahaan retail ingin meningkatkan penawaran produknya dengan mempersonalisasi rekomendasi produk kepada pelanggan berdasarkan preferensi mereka. Data pelanggan mencakup informasi seperti kategori produk yang paling sering dibeli, jumlah transaksi per bulan, dan waktu belanja. Anda diminta untuk menggunakan algoritma K-means untuk mengelompokkan pelanggan berdasarkan preferensi produk mereka.

Tugas:

1. Lakukan eksplorasi data untuk memahami distribusi pembelian pelanggan.
2. Terapkan K-means untuk mengelompokkan pelanggan dalam klaster yang optimal berdasarkan preferensi produk mereka.
3. Tentukan jumlah klaster yang optimal menggunakan metode Elbow dan Silhouette Score.
4. Berikan interpretasi dari setiap klaster dan rekomendasikan strategi pemasaran berdasarkan hasil klasterisasi.

Studi Kasus 2:

Pengembangan Agen Game menggunakan Reinforcement Learning

Deskripsi Kasus:

Sebuah perusahaan pengembang game ingin mengimplementasikan algoritma reinforcement learning (RL) untuk mengembangkan agen cerdas yang dapat bermain game sederhana seperti *Tic-Tac-Toe*. Anda diminta untuk menggunakan algoritma Q-learning untuk membuat agen yang dapat belajar dari pengalaman dan meningkatkan kemampuannya dalam bermain game.

Tugas:

1. Definisikan lingkungan game *Tic-Tac-Toe* dalam format yang dapat diterima oleh algoritma Q-learning.
2. Implementasikan algoritma Q-learning untuk mengajarkan agen cara bermain game.
3. Tentukan fungsi reward yang sesuai untuk setiap langkah dalam game dan jelaskan alasan pemilihan tersebut.
4. Evaluasi performa agen setelah sejumlah iterasi dan diskusikan bagaimana agen dapat memperbaiki strategi bermain berdasarkan hasil evaluasi.

Contoh Desain Penilaian Terhubung CPL dan CPMK

Mata Kuliah:

CPL (Capaian Pembelajaran Lulusan):

1. Memahami konsep dasar data science dan peran Python dalam pemrosesan dan analisis data.
2. Mampu mengoperasikan alat dan pustaka Python untuk analisis data, termasuk pemodelan dan visualisasi.

CPMK (Capaian Pembelajaran Mata Kuliah):

1. Menjelaskan hubungan antara data, algoritma, dan model pembelajaran.
2. Mengoperasikan Google Colab dan memuat dataset UCI.
3. Menerapkan konsep data science dalam pemrosesan data menggunakan Python.

Jenis Penilaian:

- a) **Kuis Teori:** Mengukur pemahaman konsep dasar terkait data science dan Python.
- b) **Praktikum (Tugas Kelompok):** Penggunaan alat Python dan pustaka terkait dalam mengolah dan menganalisis dataset.
- c) **Studi Kasus:** Menerapkan model pembelajaran mesin untuk dataset nyata dan menginterpretasi hasilnya.

1. Kuis Teori (40%)

Tujuan Penilaian: Menguji pemahaman mahasiswa tentang konsep dasar data science, algoritma yang digunakan, dan hubungan antar data, algoritma, dan model pembelajaran.

CPL yang Dituju:

- a) Mahasiswa dapat menjelaskan konsep dasar data science dan peran Python.
- b) Mahasiswa memahami algoritma dasar dalam data science.

CPMK yang Dituju:

- a) Menjelaskan hubungan antara data, algoritma, dan model pembelajaran.

Soal Kuis:

1. Apa yang dimaksud dengan data science? Jelaskan peran data science dalam pengambilan keputusan di berbagai bidang.
2. Jelaskan bagaimana Python digunakan dalam data science, dan sebutkan pustaka-pustaka utama yang digunakan dalam analisis data.
3. Apa itu supervised learning dan unsupervised learning? Berikan contoh aplikasinya.
4. Jelaskan apa yang dimaksud dengan 'train-test split' dan mengapa hal ini penting dalam pembelajaran mesin.

Rubrik Penilaian:

- a) **80-100:** Jawaban sangat jelas dan menyeluruh, memberikan contoh yang tepat dan relevan.
- b) **60-79:** Jawaban cukup jelas, dengan penjelasan yang kurang mendalam atau tanpa contoh yang relevan.
- c) **<60:** Jawaban tidak tepat atau tidak relevan dengan materi yang diajarkan.

2. Praktikum (Tugas Kelompok) (40%)

Tujuan Penilaian: Menilai kemampuan mahasiswa dalam menerapkan Python untuk pemrosesan data menggunakan dataset nyata dari UCI Repository.

CPL yang Dituju:

- a) Mahasiswa dapat mengoperasikan Google Colab dan memuat dataset.
- b) Mahasiswa dapat menerapkan konsep-konsep dasar data science untuk analisis data.

CPMK yang Dituju:

- a) Mengoperasikan Google Colab dan memuat dataset UCI.
- b) Menerapkan teknik data science untuk menganalisis dan mengolah data.

Tugas Kelompok:

- a) **Dataset yang digunakan:** Dataset Iris dari UCI Repository.
- b) **Tugas:** Mahasiswa diminta untuk:
 - a) Memuat dataset menggunakan Python di Google Colab.
 - b) Melakukan analisis deskriptif data.
 - c) Membangun model klasifikasi sederhana menggunakan algoritma Logistic Regression dan Decision Tree.
 - d) Mengevaluasi model menggunakan metrik evaluasi seperti akurasi dan confusion matrix.
 - e) Menyajikan hasil analisis dalam bentuk laporan atau presentasi.

Rubrik Penilaian:

- a) **80-100**: Analisis data dilakukan dengan baik, pemodelan tepat, dan evaluasi komprehensif dengan penjelasan hasil yang jelas.
- b) **60-79**: Analisis dan pemodelan dilakukan, namun evaluasi atau penjelasan hasil tidak mendalam atau kurang tepat.
- c) **<60**: Laporan atau presentasi tidak jelas, tidak menghubungkan teori dan praktik, atau hasil evaluasi tidak akurat.

3. Studi Kasus (20%)

Tujuan Penilaian: Menguji kemampuan mahasiswa dalam mengaplikasikan teori data science dalam memecahkan masalah dunia nyata.

CPL yang Dituju:

Mahasiswa mampu menerapkan data science untuk memecahkan masalah dunia nyata dengan menggunakan alat yang sesuai.

CPMK yang Dituju:

Menggunakan model pembelajaran mesin untuk analisis dan memberikan rekomendasi berbasis data.

Studi Kasus:

- a) **Dataset yang digunakan:** Dataset "Adult Income" dari UCI.
- b) **Tugas:** Mahasiswa diminta untuk:
 - a) Melakukan pembersihan dan eksplorasi data.
 - b) Menerapkan model klasifikasi untuk memprediksi apakah seseorang berpendapatan lebih dari 50K USD per tahun.

- c) Evaluasi model menggunakan precision, recall, dan F1-score.
- d) Diskusikan kekuatan dan kelemahan model yang digunakan serta hasil evaluasi.

Rubrik Penilaian:

- a) **80-100**: Analisis dan pemodelan yang sangat baik, dengan evaluasi yang mendalam dan solusi yang realistis berdasarkan hasil model.
- b) **60-79**: Pemodelan dan evaluasi dilakukan, namun penjelasan kurang mendalam atau beberapa langkah kurang tepat.
- c) **<60**: Pemodelan atau evaluasi tidak dilakukan dengan baik, atau kesalahan besar dalam interpretasi data.

4. Umpan Balik dan Refleksi Pembelajaran

Tujuan Penilaian: Menilai pemahaman mahasiswa terhadap proses pembelajaran data science serta kemajuan yang mereka capai dalam proyek-proyek praktikum.

CPL yang Dituju:

Mahasiswa dapat merefleksikan proses belajar dan pemahaman yang diperoleh selama kursus.

CPMK yang Dituju:

Mahasiswa dapat melakukan refleksi terhadap pembelajaran yang telah dilakukan, termasuk tantangan dan hasil yang dicapai.

Daftar Pustaka

- Barhate, D., & John, P. (2024). A systematic review of machine learning and deep learning methods for plant species classification. *AI in Agriculture, Article 100605*.
<https://doi.org/10.1016/j.atech.2024.100605>
- Castillo-Botón, C. (2022). Machine learning regression and classification methods for low-visibility prediction. *Atmospheric Research*.
<https://www.sciencedirect.com/science/article/pii/S0169809522001430>
- Géron, A. (2022). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media. ISBN: 978-1098125974.
- Hermiati, A. S. (2024). A comparative study: Application of PCA for stroke prediction. *JEEEMI*, 6(3), 446.
<https://doi.org/10.35882/jeeemi.v6i3.446>
- Hu, F., & Hei, X. (Eds.). (2023). *AI, Machine Learning and Deep Learning: A Security Perspective* (1st ed.). CRC Press.
<https://doi.org/10.1201/9781003187158>. eBook ISBN: 9781003187158.
- Ngugi, H. N., Akinyelu, A. A., & Ezugwu, A. E. (2024). Machine learning and deep learning for crop disease diagnosis. *Agronomy*, 14(12), 3001.
<https://doi.org/10.3390/agronomy14123001>
- Razavi, M., Hajialiasghari, F., Abhari, A., & Sourì, A. (2024). Machine learning, deep learning, and data preprocessing for mental stress detection. *JMIR*, Article 53714.
<https://doi.org/10.2196/53714>
- Razzaq, K., et al. (2025). Machine learning and deep learning paradigms: A comprehensive review. *Computers*, 14(3), 93. <https://doi.org/10.3390/computers14030093>
- Rani, F. A., & Marfiana, D. L. (2025). Comparative analysis of dimensionality reduction for breast cancer classification. *JRI*, 7(3), 375. <https://doi.org/10.34288/jri.v7i3.375>
- Santoso, N. A., Lutfayza, R., Nughroho, B. I., & Gunawan, G. (2024). Anomaly detection in network security systems using machine learning. *Journal of Intelligent Decision Support Systems*, 7(2), 113-120.
<https://doi.org/10.35335/idss.v7i2.238>

- Schummer, P. (2024). Machine learning-based network anomaly detection. *Artificial Intelligence*, 5(4), 143.
<https://www.mdpi.com/2673-2688/5/4/143>
- Sarıkoç, M., & Celik, M. (2024). PCA-ICA-LSTM: A hybrid model for financial time series prediction. *Computational Economics*, 65, 2249-2315.
<https://doi.org/10.1007/s10614-024-10629-x>
- Wee, B. F., Sivakumar, S., Lim, K. H., & Juwono, F. H. (2024). Diabetes detection based on machine learning and deep learning approaches. *Multimedia Tools and Applications*, 83(??), 24153-24185. <https://doi.org/10.1007/s11042-023-16407-5>

Biografi Penulis



Ibnu Rasyid Munthe, S.T., M.Kom. adalah dosen dan akademisi yang lahir di Rantau Prapat, Kabupaten Labuhanbatu pada tahun 1987. Beliau meraih gelar Sarjana Teknik dari Universitas Nurtanio Bandung dan Magister Komputer dari Universitas Putra Indonesia Padang. Saat ini, beliau aktif sebagai dosen di Universitas Labuhanbatu, serta menulis artikel ilmiah dan buku di bidang teknologi informasi. Ibnu Rasyid mengelola kanal YouTube @manjaddawajada2022 untuk berbagi pengetahuan tentang algoritma, pemrograman, dan pengembangan teknologi. Terdaftar dengan SINTA ID 5975973, Google Scholar ID gIPcbagAAAAJ, dan Scopus ID 57207882178, beliau berkomitmen untuk memberikan kontribusi signifikan dalam pendidikan dan riset teknologi di Indonesia.



Junadhi adalah Associate Professor di Universitas Sains dan Teknologi Indonesia (USTI) yang mengkhususkan diri dalam bidang Artificial Intelligence, Machine Learning, Deep Learning, dan Generative AI. Ia aktif dalam penelitian dan publikasi ilmiah yang terindeks Scopus (ID: 57222628214), dengan fokus pada pengembangan model deep learning dan kecerdasan artifisial dalam konteks pendidikan dan teknologi. Selain itu, Junadhi juga berperan dalam pengembangan kurikulum dan pengajaran berbasis inovasi digital, berkomitmen

untuk memberikan pengalaman belajar yang relevan dengan perkembangan teknologi terkini.



Richi Andrianto. S.Kom, M.Kom, Lahir di Bangun Jaya, Rokan Hulu-Riau, pada 30 Maret 1993, penulis adalah anak pertama dari tiga bersaudara, putra dari pasangan Wasidi dan Lina Yuliana. Memiliki gelar Sarjana Teknik Informatika dari Universitas Pasir Pengaraian, serta Magister Teknologi Informasi dari Universitas Putra Indonesia "YPTK" Padang. Saat ini, penulis sedang menempuh pendidikan Program Doktorat (S3) dalam bidang Teknologi Informasi dengan fokus pada Data Science dan Kecerdasan Buatan. Aktif sebagai dosen tetap di Program Studi Sistem Informasi, Institut Teknologi dan Sains Padang Lawas Utara (ITS Paluta), serta terlibat dalam penelitian dan pengembangan teknologi informasi. Penulis juga telah meraih hibah penelitian Kemdiktisaintek sebagai ketua peneliti dari 2023 hingga 2025. Buku ini diterbitkan sebagai kontribusi untuk pengembangan literasi teknologi dan penyebarluasan ilmu pengetahuan yang aplikatif.



Yuda Irawan, S.Kom, M.TI adalah dosen Teknik Informatika di Universitas Hang Tuah Pekanbaru dengan jabatan fungsional Lektor Kepala 700. Saat ini, ia tengah menempuh pendidikan doktoral Program Studi Teknologi Informasi di Universitas Putra Indonesia YPTK Padang. Keahlian

beliau mencakup data science, Internet of Things (IoT), pemrograman, dan image processing, yang tercermin dalam publikasi ilmiah nasional terakreditasi serta jurnal internasional bereputasi seperti Scopus Q2 hingga Q4. Karyanya menjadi dasar penyusunan buku ajar, buku referensi, dan monograf akademik yang berfokus pada penerapan teknologi komputasi cerdas dalam sistem IoT dan analisis data.



Shabrina Rasyid Munthe, S.Pd.I., M.S., lahir di Rantauprapat pada 20 Februari 1992, adalah seorang akademisi yang berdedikasi. Beliau menyelesaikan studi S1 di bidang Tarbiyah di IAIN Padangsidempuan dan meraih gelar Magister Sastra dari Universitas Islam Sumatera Utara (UISU).

Sejak 2019, Shabrina mengabdikan diri sebagai dosen tetap di Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Al Washliyah Labuhanbatu. Dengan keahlian multidisiplin, beliau aktif berkontribusi dalam pengembangan ilmu pengetahuan, khususnya di bidang linguistik dan pendidikan bahasa Inggris.



Eka Irawan, M.Kom, adalah dosen di Program Studi Sistem Informasi, STIKOM Tunas Bangsa Pematangsiantar. Lahir di Pondok Lama 18 Januari 1989 Menyelesaikan pendidikan formal Sarjana di STT Polipropesi Medan pada Program Studi Teknik Informatika (S1) tahun 2010, Universitas Sumatera

Utara di Medan pada Program Studi Magister Komputer (S2) tahun 2016 .

Sejak tahun 2010, mulai aktif sebagai dosen tetap di STIKOM Tunas Bangsa Kota Pematangsiantar .

Penulis mengajar mahasiswa D3 Program Studi Manajemen Informatika dan S1 Program Studi Sistem Informasi dan Teknik Informatika Penulis mengajar beberapa mata kuliah D3 seperti , Rekayasa Perangkat Lunak serta mata kuliah S1 seperti Algoritma Perograman, Bahasa Pemrograman, Metode Penelitian dan Analisa Perancangan Sistem Informasi.

Penulis juga aktif melakukan penelitian dan publikasi karya ilmiah di jurnal lokal maupun nasional.



Surya Darma, M.Kom, adalah dosen di Program Studi Sistem Informasi di STIKOM Tunas Bangsa Pematangsiantar. Lahir di Aek Loba 20 Agustus 1984. Saya Menyelesaikan pendidikan formal Sarjana di STIKOM Tunas Bangsa pada Program Studi Sistem Informasi (S1) tahun 2017 dan UPI YPTK Padang pada Program Studi Magister Komputer (S2) tahun 2021 .

Penulis mengajar mahasiswa D3 Program Studi Manajemen Informatika dan S1 Program Studi Sistem Informasi dan Teknik Informatika Penulis mengajar beberapa mata kuliah D3 seperti , Rekayasa Perangkat Lunak, Jaringan Komputer, serta mata kuliah S1 seperti Algoritma Perograman, Keamanan Sistem Informasi.

Penulis juga aktif melakukan penelitian dan publikasi karya ilmiah di jurnal lokal maupun nasional.



Fahmi Firzada, M.Kom, adalah dosen di Program Studi Sistem Informasi, STIKOM Tunas Bangsa Pematangsiantar. Lahir di Pematangsiantar, 12 Agustus 1980. Menyelesaikan pendidikan formal Sarjana di STIKOM Tunas Bangsa pada Program Studi Sistem Informasi (S1) tahun 2018 Pematangsiantar, Universitas Putra Indonesia

-YPTKII Padang pada Fakultas Ilmu Komputer Program Studi Teknik Informatika (S2) tahun 2021.

Sejak tahun 2021, mulai aktif sebagai dosen tetap di STIKOM Tunas Bangsa Kota Pematangsiantar. Penulis mengajar mahasiswa Strata Satu (S1) pada Program Studi Sistem Informasi. Penulis mengajar beberapa mata kuliah seperti , Rekayasa Perangkat Lunak serta, Algoritma Perograman, Struktur Data dan Bahasa Pemrograman. Penulis juga aktif melakukan penelitian dan publikasi karya ilmiah di jurnal lokal maupun nasional.



Nurul Khairina, Mempoleh gelar Sarjana Komputer (S.Kom) dari Universitas Sumatera Utara pada tahun 2013 dari program studi Ilmu Komputer dengan bidang keahlian Steganografi dan Jaringan Syaraf Tiruan. Pada tahun 2016 memperoleh gelar Magister Komputer (M.Kom) dari Universitas Sumatera Utara pada

program studi Teknik Informatika dengan bidang keahlian Steganografi dan Jaringan Syaraf Tiruan. Sejak tahun 2018 penulis menjadi dosen tetap di Universitas Medan Area pada program studi Teknik Informatika yang mengampu beberapa mata kuliah seperti : Algoritma dan Pemrograman, Pemrograman Berorientasi Objek, Rekayasa Perangkat Lunak, Jaringan Syaraf Tiruan, Logika Fuzzy, dan Data Mining. Sepanjang berkarir sebagai akademisi di lingkungan Universitas, penulis pernah memiliki beberapa pengalaman sebagai Ketua Program Studi Sarjana, Kepala Laboratorium Program Studi, Kepala Biro Administrasi Akademik dan Editor in Chief Jurnal Nasional Terakreditasi. Penulis aktif dalam tridharma perguruan tinggi seperti publikasi pada jurnal nasional terakreditasi, jurnal internasional terindeks Scopus, prosiding internasional, hibah penelitian pemerintah, penulisan buku, hak cipta (HKI), editor jurnal, dan reviewer jurnal.



Chaeroen Niesa, S.Kom., M.Kom, adalah dosen di Program Studi Informatika, Fakultas Komputer dan Multimedia di Universitas Islam Kebangsaan Indonesia (UNIKI). Lahir di Samalanga, 18 Juni 1990. Saya Menyelesaikan pendidikan formal Sarjana di Universitas Almuslim pada Program Studi Teknik Informatika (S1) tahun 2013 dan Universitas Sumatera Utara (USU) pada Program Studi Magister Teknik Informatika (S2) tahun 2016.

Sejak Tahun 2016 setelah selesai kuliah, penulis mulai aktif mengajar di STMIK Bina Bangsa Lhokseumawe yang sekarang lebih dikenal dengan UNIKI Bireuen.

Penulis mengajar mahasiswa di Prodi Informatika, dengan beberapa matakuliah disetiap semester seperti, Sistem Digital, Database, Sistem Informasi, Pemograman Game, Pengembangan Aplikasi Mobile dan Metode Penelitian.

Penulis juga aktif melakukan penelitian dan publikasi karya ilmiah di jurnal lokal maupun nasional. Tahun 2025 penulis lolos Penelitian Fundamental Reguler.

Buku ini memberikan pemahaman mendalam tentang kecerdasan buatan (AI) dan jaringan syaraf tiruan (neural network), dua teknologi revolusioner yang mengubah berbagai sektor industri dan kehidupan sehari-hari. Dengan penjelasan yang sistematis dan mudah dipahami, buku ini membahas konsep dasar, algoritma, dan aplikasi praktis AI serta jaringan syaraf tiruan dalam pemecahan masalah kompleks seperti pengenalan pola, prediksi, dan deteksi. Ditujukan untuk mahasiswa dan praktisi, buku ini adalah referensi ideal untuk memperkenalkan serta memperdalam pemahaman tentang dunia AI yang terus berkembang pesat. Temukan potensi AI yang dapat mengubah masa depan Anda!



ISBN 978-634-04-7698-9 (PDF)



9

786340

476989