

BUKU AJAR HYBRID DEEP LEARNING CNN, BIGRU, DAN ATTENTION UNTUK PREDIKSI SAHAM



Ibnu Rasyid Munthe, S.T., M.Kom
Bhakti Helvi Rambe, SE., M.Si
Shabrina Rasyid Munthe, S.Pd.I., MS

Buku Ajar Hybrid Deep Learning CNN, BiGRU, dan Attention Untuk Prediksi Saham

**Ibnu Rasyid Munthe, S.T., M.Kom
Bhakti Helvi Rambe, SE., M.Si
Shabrina Rasyid Munthe, S.Pd.I., MS
Selamat Subagio, S.Kom., M.Kom
Rahmad Aditiya
Eko Firman Susilo, M.Pd**



**BUKU AJAR HYBRID DEEP LEARNING CNN, BIGRU, DAN ATTENTION
UNTUK PREDIKSI SAHAM**

**Ibnu Rasyid Munthe, S.T., M.Kom
Bhakti Helvi Rambe, SE., M.Si
Shabrina Rasyid Munthe, S.Pd.I., MS**

Editor :
Sahat Parulian Sitorus

Desain Cover :
Nurhasanah

Sumber :
<https://isbn.jndi.my.id/products/buku-ajar-hybrid-deep-learning-cnn-bigru-dan-attention-untuk-prediksi-saham>

Tata Letak :
Ida Royani Sitorus

Ukuran :
Jml hal judul 149, Jml hal isi naskah 147, Uk: 18x25 cm

ISBN :
978-634-04-2491-1

Cetakan Pertama :
20 Agustus 2025

Hak Cipta 2025, Pada Penulis
Isi diluar tanggung jawab percetakan

Copyright © 2025 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

PRAKATA

Perkembangan teknologi kecerdasan buatan telah membuka peluang baru dalam menganalisis dan memprediksi pergerakan pasar saham yang dinamis. Buku ajar *Hybrid Deep Learning: CNN, BiGRU, dan Attention untuk Prediksi Saham* ini hadir untuk menjembatani teori pasar modal dengan penerapan praktis metode deep learning terkini. Disusun secara sistematis, buku ini memandu pembaca mulai dari konsep dasar pasar saham, teknik pengolahan data, hingga implementasi model hybrid yang menggabungkan kekuatan CNN dalam ekstraksi fitur, BiGRU dalam memahami konteks temporal dua arah, dan mekanisme Attention untuk fokus pada informasi yang paling relevan.

Kami berharap buku ini tidak hanya menjadi sumber belajar di kelas, tetapi juga panduan praktis bagi peneliti, mahasiswa, dan praktisi yang ingin mengembangkan solusi prediksi saham berbasis AI. Semoga karya ini bermanfaat dan menginspirasi inovasi di bidang keuangan digital.

Rantauprapat, Agustus 2025

Tim Penulis

KATA PENGANTAR

Perubahan lanskap pasar modal di era digital memaksa para pelaku industri dan akademisi untuk beradaptasi dengan perkembangan teknologi yang begitu cepat. Salah satu terobosan besar adalah penerapan *Artificial Intelligence* (AI), khususnya *Deep Learning*, dalam memprediksi pergerakan harga saham. Metode ini menawarkan kemampuan menganalisis pola yang kompleks dan dinamis—sesuatu yang sulit dicapai dengan pendekatan konvensional.

Buku Ajar *Hybrid Deep Learning: CNN, BiGRU, dan Attention untuk Prediksi Saham* disusun yang menggabungkan landasan teori pasar saham, teknik pengolahan data, hingga implementasi model hybrid modern. CNN dimanfaatkan untuk mengekstraksi pola lokal, BiGRU untuk memahami hubungan temporal dua arah, dan Attention untuk memfokuskan prediksi pada informasi yang paling relevan. Kami berharap buku ini menjadi jembatan antara pengetahuan akademis dan kebutuhan praktis di lapangan, membantu mahasiswa, peneliti, dan praktisi dalam mengembangkan solusi prediksi saham yang presisi dan adaptif.

Rantauprapat, Agustus 2025

Tim Penulis

DAFTAR ISI

PRAKATA	4
KATA PENGANTAR	5
RENCANA PEMBELAJARAN SEMESTER (RPS) Berbasis Outcome-Based Education (OBE)	9
A. Identitas Mata Kuliah	9
B. Profil Lulusan (CPL)	9
C. Capaian Pembelajaran Mata Kuliah (CPMK)	10
D. Peta CPL–CPMK	11
E. Rencana Pembelajaran Semester	11
F. Daftar Referensi	14
PENDAHULUAN & PERAN PASAR SAHAM	15
1. Learning Outcome (LO)	15
2. Materi	15
3. Soal Esai	220
Tugas Individu	21
5. Studi Kasus	25
6. Penilaian	25
TANTANGAN DALAM PREDIKSI HARGA SAHAM	26
1. Learning Outcome (LO)	26
2. Materi	26
3. Soal Esai	31
4. Tugas Individu	31
5. Studi Kasus	36
6. Penilaian	36
PENDEKATAN KONVENSIONAL DALAM PREDIKSI HARGA SAHAM	37
1. Learning Outcome (LO)	37
2. Materi	37
3. Soal Esai	42
4. Tugas Individu	42
5. Studi Kasus	45
6. Penilaian	45
PENDEKATAN MODERN BERBASIS AI UNTUK PREDIKSI HARGA SAHAM	46
1. Learning Outcome (LO)	46
2. Materi	46
3. Soal Esai	50
4. Tugas Individu	51
5. Studi Kasus	54
6. Penilaian	55
ARSITEKTUR CNN UNTUK DATA TIME SERIES HARGA SAHAM	56
1. Learning Outcome (LO)	56
2. Materi	56

3. Soal Esai	60
4. Tugas Individu.....	61
5. Studi Kasus	65
6. Penilaian	65
ARSITEKTUR GRU & BIGRU UNTUK DATA TIME SERIES HARGA	
SAHAM	66
1. Learning Outcome (LO)	66
2. Materi.....	66
3. Soal Esai	70
4. Tugas Individu.....	70
5. Studi Kasus	74
6. Penilaian	74
INTEGRASI CNN BIGRU DENGAN MEKANISME ATTENTION	75
1. Learning Outcome (LO)	75
2. Materi.....	75
3. Soal Esai	80
4. Tugas Individu.....	80
5. Studi Kasus	84
6. Penilaian	84
UJIAN TENGAH SEMESTER (UTS)	85
1. Learning Outcome (LO)	85
2. Materi yang Diujikan	85
4. Soal Esai	86
5. Soal Studi Kasus.....	86
6. Penilaian	86
PENGOLAHAN DATA PASAR SAHAM UNTUK MODEL DEEP	
LEARNING	87
1. Learning Outcome (LO)	87
2. Materi.....	87
3. Soal Esai	91
4. Tugas Individu.....	92
5. Studi Kasus	94
6. Penilaian	95
REKAYASA FITUR (<i>FEATURE ENGINEERING</i>) UNTUK PREDIKSI	
SAHAM	96
1. Learning Outcome (LO)	96
2. Materi.....	96
3. Soal Esai	100
4. Tugas Individu.....	101
5. Studi Kasus	104
6. Penilaian	104
PELATIHAN MODEL CNN BIGRU ATTENTION UNTUK PREDIKSI	
SAHAM	105
1. Learning Outcome (LO)	105
2. Materi.....	105

3. Soal Esai	108
4. Tugas Individu.....	109
5. Studi Kasus	113
6. Penilaian	113
PENANGANAN OVERFITTING & EVALUASI MODEL PREDIKSI	
SAHAM	114
1. Learning Outcome (LO)	114
2. Materi.....	114
3. Soal Esai	118
4. Tugas Individu.....	118
5. Studi Kasus	123
6. Penilaian	123
INTERPRETASI HASIL MODEL PREDIKSI SAHAM.....	124
1. Learning Outcome (LO)	124
2. Materi.....	124
3. Soal Esai	130
4. Tugas Individu.....	131
5. Studi Kasus	137
6. Penilaian	137
PERANCANGAN PROYEK PENGEMBANGAN MODEL PREDIKSI	
SAHAM	138
1. Learning Outcome (LO)	138
2. Materi.....	138
3. Soal Esai	139
4. Tugas Individu.....	139
5. Studi Kasus	140
6. Penilaian	140
PRESENTASI PROYEK AKHIR PREDIKSI SAHAM	141
1. Learning Outcome (LO)	141
2. Materi.....	141
4. Tugas Individu.....	142
5. Studi Kasus	142
6. Penilaian	142
UJIAN AKHIR SEMESTER (UAS).....	143
1. Learning Outcome (LO)	143
2. Materi yang Diujikan	143
3. Bentuk Ujian.....	143
4. Soal Esai	144
5. Soal Studi Kasus.....	144
6. Penilaian	144

RENCANA PEMBELAJARAN SEMESTER (RPS) Berbasis Outcome-Based Education (OBE)

A. Identitas Mata Kuliah

Nama Mata Kuliah	Pemodelan Prediksi Saham dengan Deep Learning
Kode Mata Kuliah	
SKS	3 SKS (Teori 2 SKS, Praktikum 1 SKS)
Semester	
Dosen Pengampu	
Prasyarat	Machine Learning, Analisis Data Keuangan

B. Profil Lulusan (CPL)

Kode CPL	Deskripsi CPL
CPL-P1	Menguasai konsep dan prinsip kecerdasan buatan untuk pemecahan masalah kompleks
CPL-P2	Mampu merancang dan mengimplementasikan model pembelajaran mesin/deep learning
CPL-P3	Mampu menganalisis dan memvisualisasikan data multivariat secara efektif
CPL-KK	Mampu mengelola proyek teknologi dan berkolaborasi dalam tim multidisiplin

C. Capaian Pembelajaran Mata Kuliah (CPMK)

Kode CPMK	Deskripsi CPMK	Keterkaitan CPL
CPMK-1	Menjelaskan konsep pasar saham, faktor penggerak harga, dan indikator teknikal	CPL-P1, CPL-P3
CPMK-2	Membedakan pendekatan konvensional dan modern (AI) dalam prediksi harga saham	CPL-P1
CPMK-3	Mendesain dan mengimplementasikan model CNN–BiGRU–Attention untuk data pasar saham multivariabel	CPL-P1, CPL-P2
CPMK-4	Mengevaluasi performa model menggunakan metrik kuantitatif dan interpretasi visual	CPL-P2, CPL-P3
CPMK-5	Mengelola proyek prediksi saham secara kolaboratif dan menyajikan hasilnya	CPL-KK

D. Peta CPL–CPMK

CPL	CPMK-1	CPMK-2	CPMK-3	CPMK-4	CPMK-5
CPL-P1	√	√	√		
CPL-P2			√	√	
CPL-P3	√			√	
CPL-KK					√

E. Rencana Pembelajaran Semester

Minggu	Sub-CPMK	Indikator Pencapaian	Materi	Metode Pembelajaran	Aktivitas Mahasiswa	Bentuk & Kriteria Penilaian	Bobot	Referensi
1	Memahami peran pasar saham & faktor penggeraknya (CPMK-1)	Mampu menjelaskan fungsi pasar saham, price discovery, dan indikator ekonomi	Bab 1: Pendahuluan – Pasar Saham & Perannya, Fungsi Penghimpunan Modal	Ceramah interaktif, diskusi	Membaca bab, diskusi kelas	Quiz, partisipasi	5%	[Buku Bab 1]
2	Mengidentifikasi tantangan prediksi harga saham (CPMK-1)	Mampu menjelaskan tantangan volatilitas, faktor psikologis, dan risiko	Bab 1: Tantangan & Faktor Pendorong Volatilitas	Studi kasus, diskusi kelompok	Analisis kasus pasar saham	Tugas ringkasan kasus	5%	[Buku Bab 1]
3	Membedakan pendekatan konvensional & modern (CPMK-2)	Menjelaskan karakteristik model klasik vs AI	Bab 3: Perbandingan Pendekatan	Ceramah, <i>case-based learning</i>	Membandingkan hasil model klasik dan AI	Laporan analisis	5%	[Buku Bab 3]
4	Mendesripsikan arsitektur CNN (CPMK-3)	Mampu menjelaskan peran CNN pada data time series	Bab 4: Peran CNN	Praktikum lab, simulasi	Implementasi CNN sederhana	Penilaian praktikum	5%	[Buku Bab 4]
5	Mendesripsikan GRU dan BiGRU (CPMK-3)	Mampu menjelaskan mekanisme GRU & BiGRU	Bab 4: GRU & BiGRU	Praktikum lab	Implementasi GRU & BiGRU	Penilaian praktikum	5%	[Buku Bab 4]

6	Mengintegrasikan CNN & BiGRU (CPMK-3)	Mampu menggabungkan CNN dan BiGRU	Bab 4: Sinergi CNN–BiGRU	<i>Project-based learning</i>	Mendesain arsitektur hybrid	Penilaian desain	5%	[Buku Bab 4]
7	Memahami mekanisme Attention (CPMK-3)	Mampu menjelaskan prinsip kerja attention	Bab 5: Mekanisme Attention	Ceramah, simulasi kode	Latihan implementasi attention	Penilaian praktikum	5%	[Buku Bab 5]
8	UTS	Mampu mengintegrasikan konsep minggu 1–7	-	Ujian tertulis & praktikum	Menyelesaikan soal & kode	Nilai UTS	20%	-
9	Mengolah data pasar saham (CPMK-3)	Mampu melakukan pembersihan & normalisasi data	Bab 6: Pengolahan Data	Praktikum lab	Preprocessing dataset	Penilaian praktikum	5%	[Buku Bab 6]
10	Rekayasa fitur (CPMK-3)	Mampu membuat indikator teknis & fitur lag	Bab 7: Rekayasa Fitur	Praktikum lab	Menambah fitur teknis	Penilaian praktikum	5%	[Buku Bab 7]
11	Melatih model CNN–BiGRU–Attention (CPMK-3)	Mampu melatih model hybrid	Bab 8: Pelatihan Model	Praktikum lab	Training model	Penilaian praktikum	5%	[Buku Bab 8]
12	Mengevaluasi kinerja model (CPMK-4)	Mampu menghitung metrik MAE, RMSE, dsb.	Bab 9: Evaluasi Model	Praktikum, analisis	Evaluasi hasil model	Laporan evaluasi	5%	[Buku Bab 9]

13	Menginterpretasi hasil model (CPMK-4)	Mampu menjelaskan hasil & visualisasi	Bab 9: Interpretasi Hasil	Diskusi, <i>presentation-based learning</i>	Presentasi hasil model	Penilaian presentasi	5%	[Buku Bab 9]
14	Merancang pengembangan model masa depan (CPMK-5)	Mampu merencanakan pengembangan model & integrasi ke sistem investasi	Bab 10: Prospek & Pemanfaatan Model	Diskusi, proyek	Rencana pengembangan model	Laporan proyek	5%	[Buku Bab 10]
15	Presentasi Proyek Akhir (CPMK-5)	Mampu mempresentasikan proyek dengan baik	Semua bab	<i>Project-based learning</i>	Presentasi kelompok	Penilaian presentasi & laporan	5%	Semua bab
16	UAS	Mengintegrasikan seluruh pengetahuan & keterampilan	Semua bab	Ujian teori & praktikum	Menyelesaikan soal & kode	Nilai UAS		

F. Daftar Referensi

- Alexander, J., & Beegam, M. (2023). *Capital formation and its role in economic growth*. Springer.
- Amalia, D. (2022). The impact of COVID-19 on global stock markets. *Journal of Financial Studies*, 15(2), 45–62.
- Bohyer, T., & Hayajneh, M. (2023). Algorithmic trading and smart contracts in financial markets. *IEEE Transactions on FinTech*, 1(1), 1–12.
- Chen, X., & Hu, Y. (2022). Hybrid LSTM models for volatility forecasting. *Journal of Computational Finance*, 25(4), 101–118.
- Han, S., Park, J., & Lee, K. (2022). Price discovery in global markets. *International Review of Economics & Finance*, 78, 234–249.
- Kaushik, R., & Kumar, P. (2023). Stock indices as economic indicators. *Economic Perspectives*, 37(5), 345–362.
- Lastauskas, P., & Minh, Q. (2021). US monetary policy shocks and global financial markets. *Journal of International Economics*, 131, 103–142.
- Laudon, K.C., & Laudon, J.P. (2021). *Management Information Systems*. Pearson.
- Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning architectures. *Journal of Machine Learning Research*, 22(1), 1–56.
- Lin, H. (2024). Transformer-based models for financial forecasting. *IEEE Access*, 12, 4567–4580.
- Sethy, P.K., & Tripathy, H.K. (2024). Network-based liquidity estimation in stock markets. *Finance Research Letters*, 55, 103234.
- Turban, E., & Volonino, L. (2019). *Information Technology for Management*. Wiley.
- Zuhroh, I., et al. (2021). Macroeconomic variables and stock market volatility in Indonesia. *Economic Modelling*, 94, 193–202.

PENDAHULUAN & PERAN PASAR SAHAM

1. Learning Outcome (LO)

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- a) Menjelaskan pengertian dan fungsi pasar saham dalam perekonomian.
- b) Menguraikan mekanisme kerja pasar saham dan price discovery.
- c) Mengidentifikasi peran pelaku pasar dalam menentukan harga saham.

2. Materi

Definisi Pasar Saham

Pasar saham merupakan suatu entitas terorganisir yang memfasilitasi kegiatan jual beli instrumen ekuitas dan surat berharga lainnya antara pihak-pihak yang membutuhkan dana dan pihak-pihak yang memiliki dana. Instrumen ekuitas mencakup saham biasa, saham preferen, dan instrumen lain yang mewakili kepemilikan modal dalam suatu perusahaan. Keberadaan pasar saham menjadi bagian penting dari sistem keuangan modern karena menyediakan mekanisme yang transparan, teratur, dan diawasi oleh otoritas resmi untuk memastikan kelancaran proses transaksi. Dalam praktiknya, pasar saham tidak hanya berfungsi sebagai sarana pertukaran aset, tetapi juga sebagai wadah pembentukan nilai perusahaan berdasarkan persepsi pelaku pasar terhadap kinerja dan prospek usaha.

Pasar saham bekerja dengan prinsip mempertemukan permintaan dan penawaran. Ketika calon pembeli saham bersedia membayar harga tertentu, dan pemegang saham bersedia menjual pada harga tersebut, terjadilah transaksi. Proses ini berlangsung secara terus-menerus selama jam perdagangan dan difasilitasi oleh sistem elektronik yang canggih. Transparansi informasi, kecepatan eksekusi transaksi, dan keandalan infrastruktur teknologi menjadi kunci bagi kepercayaan publik terhadap pasar saham.

Fungsi Utama

Fungsi pasar saham mencakup berbagai aspek yang berkontribusi langsung pada perekonomian nasional dan global. Setidaknya terdapat empat fungsi utama yang membedakan peran pasar saham dengan mekanisme keuangan lainnya.

Pertama, *capital formation* atau penghimpunan modal. Perusahaan yang membutuhkan pendanaan untuk ekspansi usaha, pengembangan produk, atau investasi dalam teknologi dapat menerbitkan saham dan menjualnya kepada publik. Melalui proses penawaran umum perdana atau *initial public offering* (IPO), dana segar dapat diperoleh untuk mendukung keberlangsungan dan pertumbuhan usaha. Misalnya, sebuah perusahaan teknologi yang berencana mengembangkan pusat data baru dapat mengumpulkan dana miliaran rupiah dari investor publik melalui pasar saham. Proses ini memberi keuntungan bagi perusahaan karena tidak perlu mengandalkan

pinjaman bank dengan kewajiban bunga tetap, sementara investor memperoleh kesempatan memiliki sebagian kepemilikan perusahaan.

Kedua, *price discovery* atau penemuan harga. Mekanisme ini memungkinkan harga saham ditentukan oleh kekuatan pasar, yaitu interaksi antara permintaan dan penawaran. Harga saham akan meningkat jika banyak pembeli bersedia membayar lebih, dan sebaliknya akan menurun jika banyak penjual bersedia melepas saham pada harga lebih rendah. Sebagai contoh, ketika sebuah perusahaan mengumumkan laporan keuangan dengan laba yang melebihi ekspektasi, permintaan terhadap saham tersebut biasanya meningkat sehingga harga pasar ikut naik. Proses ini mencerminkan penilaian kolektif investor terhadap nilai perusahaan berdasarkan informasi terbaru yang tersedia.

Ketiga, fungsi likuiditas. Likuiditas merujuk pada kemudahan mengubah saham menjadi uang tunai tanpa mempengaruhi harga pasar secara signifikan. Pasar saham yang likuid memudahkan investor untuk membeli atau menjual saham kapan pun dibutuhkan. Likuiditas juga mencerminkan kesehatan pasar karena menunjukkan adanya partisipasi aktif dari banyak pelaku pasar. Sebagai ilustrasi, saham-saham dengan kapitalisasi pasar besar di Bursa Efek Indonesia (BEI) cenderung memiliki volume perdagangan tinggi, sehingga transaksi dalam jumlah besar dapat dilakukan dengan cepat dan biaya transaksi yang rendah.

Keempat, pasar saham berfungsi sebagai barometer ekonomi. Pergerakan indeks harga saham gabungan (IHSG) di BEI sering dijadikan acuan untuk menilai kondisi ekonomi secara umum. Ketika IHSG mengalami tren kenaikan, hal ini dapat mengindikasikan optimisme investor terhadap prospek ekonomi, sedangkan penurunan tajam dapat mencerminkan kekhawatiran akan perlambatan ekonomi atau ketidakpastian politik. Fungsi ini menjadikan pasar saham sebagai salah satu indikator penting bagi pemerintah, pelaku usaha, dan masyarakat luas dalam mengambil keputusan strategis.

Pelaku Pasar

Pasar saham diisi oleh berbagai jenis pelaku dengan peran yang saling melengkapi. Emiten adalah perusahaan yang menerbitkan saham untuk dijual kepada publik. Mereka bertanggung jawab memberikan informasi yang akurat dan tepat waktu mengenai kinerja dan prospek usaha. Investor ritel adalah individu yang membeli saham untuk tujuan investasi pribadi, sementara investor institusional mencakup lembaga seperti dana pensiun, perusahaan asuransi, dan manajer investasi yang mengelola dana dalam jumlah besar. Pialang atau *broker* bertindak sebagai perantara yang menghubungkan penjual dan pembeli, sedangkan regulator seperti Otoritas Jasa Keuangan (OJK) dan BEI memastikan bahwa seluruh transaksi berjalan sesuai dengan peraturan dan etika pasar.

Interaksi antar pelaku pasar ini membentuk dinamika harga dan volume perdagangan harian. Misalnya, keputusan investor institusional untuk membeli saham dalam jumlah besar dapat memicu kenaikan harga, yang kemudian diikuti oleh investor ritel yang berharap memperoleh keuntungan dari tren tersebut.

Contoh Nyata

Salah satu contoh nyata fungsi dan peran pasar saham dapat dilihat pada Bursa Efek Indonesia (BEI). BEI menyediakan platform bagi perusahaan untuk menghimpun dana melalui penerbitan saham, sekaligus memberi kesempatan bagi investor untuk berpartisipasi dalam kepemilikan perusahaan. Indeks Harga Saham Gabungan (IHSG) sebagai tolok ukur kinerja pasar modal Indonesia menjadi indikator yang diawasi secara luas. Perubahan nilai IHSG mencerminkan sentimen pasar terhadap kondisi ekonomi, baik domestik maupun global.

Sebagai ilustrasi, ketika perekonomian menunjukkan tanda-tanda pertumbuhan, seperti peningkatan ekspor atau stabilnya nilai tukar rupiah, IHSG cenderung menguat karena investor optimis terhadap prospek perusahaan yang tercatat di bursa. Sebaliknya, krisis keuangan global atau gejolak politik domestik dapat memicu aksi jual besar-besaran yang menekan IHSG ke level lebih rendah. Guna memperjelas peran BEI dan IHSG, dapat dibuat tabel sederhana berikut.

Tabel 1. Fungsi Utama Bursa Efek Indonesia

Fungsi	Deskripsi
Penghimpunan Modal	Memfasilitasi perusahaan memperoleh dana dari publik.
Penemuan Harga	Menentukan harga saham melalui interaksi permintaan dan penawaran.
Likuiditas	Memudahkan transaksi jual beli saham dengan cepat dan efisien.
Barometer Ekonomi	Mencerminkan kondisi ekonomi melalui pergerakan indeks harga saham.

Dengan memahami definisi, fungsi utama, pelaku, dan contoh nyata pasar saham, mahasiswa diharapkan mampu melihat peran strategis pasar saham dalam perekonomian. Pemahaman ini menjadi landasan untuk mempelajari topik lanjutan seperti pengolahan data pasar, rekayasa fitur, serta implementasi model prediksi harga saham yang relevan dengan dinamika ekonomi modern.

3. Soal Esai

1. Jelaskan definisi pasar saham menurut Anda.
2. Sebutkan 3 fungsi utama pasar saham dan jelaskan masing-masing.
3. Apa yang dimaksud dengan *price discovery*?
4. Bagaimana peran pasar saham sebagai barometer ekonomi?
5. Siapa saja pelaku pasar saham? Jelaskan perannya.

6. Mengapa likuiditas menjadi faktor penting bagi investor?
7. Jelaskan hubungan antara pasar saham dan pertumbuhan ekonomi.
8. Berikan contoh instrumen yang diperdagangkan di BEI selain saham biasa.
9. Apa perbedaan investor ritel dan investor institusional?
10. Bagaimana peran regulator dalam menjaga stabilitas pasar saham?

Tugas Individu

Cari data IHSG untuk 1 bulan terakhir, lalu buat grafik pergerakannya menggunakan Excel atau Python. Sertakan analisis singkat (maksimal 1 halaman) mengenai tren yang terlihat.

```
# =====
# IHSG: 1 Bulan Terakhir (robust fix untuk kolom ^JKSE)
# =====
!pip -q install yfinance pandas matplotlib openpyxl

import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt
from datetime import timedelta

pd.set_option('display.float_format', lambda x: f'{x:,.2f}')

# --- Helper untuk ekstrak kolom Close secara robust ---
def extract_close_series(df, ticker="^JKSE"):

    if df is None or df.empty:
        raise ValueError("Data kosong dari yfinance.")

    cols = df.columns

    # CASE 1: kolom tunggal
    if not isinstance(cols, pd.MultiIndex):
        if "Close" in cols:
            return df["Close"].rename("Close")
        if "Adj Close" in cols:
            return df["Adj Close"].rename("Close")
        raise KeyError(f"Tidak ada 'Close'/'Adj Close'. Kolom tersedia: {list(cols)}")

    # CASE 2: MultiIndex
    lv0 = list(cols.get_level_values(0))
    lv1 = list(cols.get_level_values(1))
```

```

# 2a) struktur (field -> ticker)
if "Close" in lv0 or "Adj Close" in lv0:
    # Prioritaskan Close
    if "Close" in df.columns.get_level_values(0):
        sub = df["Close"]
    else:
        sub = df["Adj Close"]

    # sub bisa Series (jika hanya satu ticker) atau DataFrame
    if isinstance(sub, pd.Series):
        return sub.rename("Close")
    # DataFrame: pilih ticker jika ada, kalau tidak ambil kolom pertama
    if ticker in sub.columns:
        return sub[ticker].rename("Close")
    return sub.iloc[:, 0].rename("Close")

# 2b) struktur (ticker -> field)
if ticker in lv0:
    sub = df[ticker]
    # sub sekarang harus punya kolom 'Close' atau 'Adj Close'
    if "Close" in sub.columns:
        return sub["Close"].rename("Close")
    if "Adj Close" in sub.columns:
        return sub["Adj Close"].rename("Close")
    # kalau tidak ada, coba field 'Close' pada level 1 di semua ticker, ambil pertama
    if "Close" in lv1:
        sub2 = df.xs("Close", level=1, axis=1)
        return sub2.iloc[:, 0].rename("Close")
    if "Adj Close" in lv1:
        sub2 = df.xs("Adj Close", level=1, axis=1)
        return sub2.iloc[:, 0].rename("Close")

```

```

# 2c) fallback terakhir: ambil field 'Close' di level 1 (atau 'Adj Close')
if "Close" in lv1:
    sub = df.xs("Close", level=1, axis=1)
    return sub.iloc[:, 0].rename("Close")
if "Adj Close" in lv1:
    sub = df.xs("Adj Close", level=1, axis=1)
    return sub.iloc[:, 0].rename("Close")

raise KeyError(f"Gagal menemukan 'Close'/'Adj Close'. Contoh kolom: {list(cols)[:6]}")

# --- Unduh data (pakai period biar selalu ke tanggal terakhir yang tersedia) ---
ticker = "^JKSE"
hist_days = 60 # ambil lebih panjang agar 30 hari kalender bersih
raw = yf.download(
    ticker,
    period=f"{hist_days}d",
    interval="1d",
    auto_adjust=False,
    group_by="column",
    progress=False
)

# Ekstrak Close secara robust
close = extract_close_series(raw, ticker=ticker)

# Rapikan ke DataFrame
px = close.to_frame(name="Close_IHSG").copy()
px.index = pd.to_datetime(px.index)
px = px.sort_index()
px.index.name = "Date"

```

```

# Filter 30 hari kalender terakhir dari tanggal trading terakhir
end_date = px.index.max().date()
start_date = end_date - timedelta(days=30)
px_1m = px.loc[px.index.date >= start_date].copy()
if px_1m.empty:
    raise ValueError("Tidak ada data untuk 30 hari terakhir setelah penyaringan.")

# --- Grafik ---
plt.figure(figsize=(10,5))
plt.plot(px_1m.index, px_1m["Close_IHSG"])
plt.title("IHSG (Close) - 1 Bulan Terakhir")
plt.xlabel("Tanggal")
plt.ylabel("Indeks")
plt.grid(True)
plt.tight_layout()
plt.show()

# --- Ekspor ke Excel & CSV ---
out_xlsx = "IHSG_1bulan.xlsx"
out_csv = "IHSG_1bulan.csv"
px_1m.to_excel(out_xlsx)
px_1m.to_csv(out_csv, index=True)
print("Tersimpan:", out_xlsx, "dan", out_csv)

# --- Ringkasan & Analisis ---
def to_float(x):
    if hasattr(x, "item"):
        try:
            return float(x.item())
        except Exception:
            pass
    if isinstance(x, (pd.Series, pd.Index, np.ndarray)):
        x = x.iloc[0] if hasattr(x, "iloc") else x[0]
    return float(x)

```

```

if len(px_1m) >= 2:
    rets = px_1m["Close_IHSG"].pct_change().dropna()
    total_return = px_1m["Close_IHSG"].iloc[-1] / px_1m["Close_IHSG"].iloc[0] - 1
    avg_day = rets.mean()
    vol_day = rets.std()
    up_days = int((rets > 0).sum())
    down_days = int((rets < 0).sum())
else:
    rets = pd.Series(dtype=float)
    total_return = np.nan
    avg_day = np.nan
    vol_day = np.nan
    up_days = 0
    down_days = 0

ma5 = px_1m["Close_IHSG"].rolling(5).mean()
ma10 = px_1m["Close_IHSG"].rolling(10).mean()

level_awal = to_float(px_1m["Close_IHSG"].iloc[0])
level_akhir = to_float(px_1m["Close_IHSG"].iloc[-1])
total_return_pct = None if pd.isna(total_return) else to_float(total_return * 100.0)
avg_day_pct = None if pd.isna(avg_day) else to_float(avg_day * 100.0)
vol_day_pct = None if pd.isna(vol_day) else to_float(vol_day * 100.0)

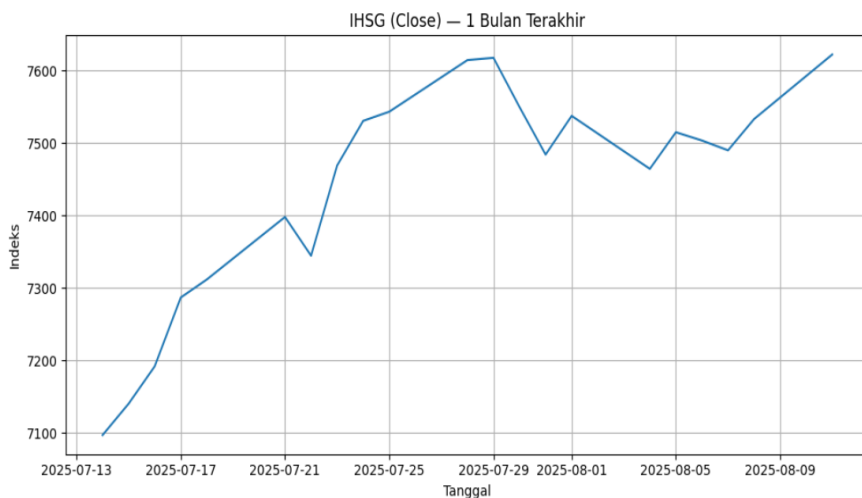
```

```
summary = {
    "Periode": f"{px_1m.index.min().date()} s.d. {px_1m.index.max().date()}",
    "Level awal": level_awal,
    "Level akhir": level_akhir,
    "Total return (≈%)": total_return_pct,
    "Rata-rata harian (≈%)": avg_day_pct,
    "Vol harian (≈%)": vol_day_pct,
    "Hari naik": up_days,
    "Hari turun": down_days,
}
display(pd.DataFrame([summary]))

trend_hint = "Tidak ada sinyal jelas."
if len(px_1m) >= 10 and not (ma5.isna().iloc[-1] or ma10.isna().iloc[-1]):
    last = px_1m["Close_IHSG"].iloc[-1]
    if last > ma5.iloc[-1] > ma10.iloc[-1]:
        trend_hint = "Bias naik jangka sangat pendek (Close > MA5 > MA10).\"
    elif last < ma5.iloc[-1] < ma10.iloc[-1]:
        trend_hint = "Bias turun jangka sangat pendek (Close < MA5 < MA10).\"
    else:
        trend_hint = "Pergerakan campuran/konsolidasi (MA5 & MA10 saling silang).\"

print("\nRingkasan singkat (≤1 halaman):\")
parts = [
    f\"Dalam 1 bulan terakhir ({summary['Periode']}), IHSG bergerak dari sekitar {level_awal:,.0f} ke {level_akhir:,.0f}.\",
    f\"Perubahan total sekitar {total_return_pct:.2f}%\" if total_return_pct is not None else \"\",
    f\"Rerata harian ~{avg_day_pct:.2f}% dengan volatilitas harian ~{vol_day_pct:.2f}%\"
    if (avg_day_pct is not None and vol_day_pct is not None) else \"\",
    f\"Indikator sederhana (MA5 & MA10): {trend_hint}\",
    \"Faktor relevan: data makro global/regional, valas/komoditas, arus dana asing, rilis data domestik.\"
]
print(\" \".join([p for p in parts if p]))
```

Output Program



Tersimpan: IHSG_1bulan.xlsx dan IHSG_1bulan.csv

	Periode	Level awal	Level akhir	Total return (≈%)	Rata-rata harian (≈%)	Vol harian (≈%)	Hari naik	Hari turun
0	2025-07-14 s.d. 2025-08-11	7,097.15	7,622.49	7.40	0.36	0.79	14	6

5. Studi Kasus

Kelompokkan diri menjadi 4–5 orang. Analisis sebuah peristiwa ekonomi (misalnya, pengumuman suku bunga, krisis global, atau berita besar) dan jelaskan pengaruhnya terhadap pergerakan harga saham di BEI. Presentasikan hasilnya dalam 5–7 slide.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

TANTANGAN DALAM PREDIKSI HARGA SAHAM

1. Learning Outcome (LO)

Setelah mengikuti pertemuan ini, mahasiswa mampu:

- a. Mengidentifikasi faktor-faktor yang memengaruhi volatilitas harga saham.
- b. Menjelaskan risiko dan ketidakpastian dalam pasar saham.
- c. Menganalisis dampak faktor fundamental, teknikal, dan psikologis terhadap harga saham.

2. Materi Volatilitas

Volatilitas dalam pasar saham dapat diibaratkan seperti *"cuaca di lautan yang kadang tenang namun bisa tiba-tiba berubah menjadi badai"*. Istilah ini merujuk pada tingkat perubahan harga yang cepat dan signifikan dalam suatu periode waktu tertentu. Kondisi tersebut menjadi perhatian penting bagi para pelaku pasar karena berhubungan langsung dengan risiko dan potensi keuntungan investasi. Sama seperti *"nakhoda kapal yang harus peka terhadap perubahan arah angin"*, investor juga harus peka terhadap sinyal pasar. Perubahan harga yang tajam dapat disebabkan oleh berbagai faktor, baik yang berasal dari kondisi internal perusahaan maupun pengaruh eksternal yang bersifat makroekonomi. Volatilitas dapat diukur menggunakan indikator statistik seperti

deviasi standar dari return harga saham dalam periode tertentu. Semakin besar nilai volatilitas, semakin tinggi pula ketidakpastian pergerakan harga yang dihadapi investor.

Fenomena ini menjadi karakteristik alami pasar saham yang tidak dapat dihindari. Pergerakan harga dapat berubah secara tiba-tiba akibat sentimen pasar, informasi yang bersifat darurat, atau perubahan fundamental pada suatu emiten. Sebagai contoh, laporan keuangan yang menunjukkan penurunan laba secara drastis dapat memicu penjualan besar-besaran, seperti "*penonton yang serentak meninggalkan stadion saat hujan deras*". Sebaliknya, berita positif seperti peluncuran produk baru atau kerja sama strategis dapat memicu lonjakan harga dalam waktu singkat, layaknya "*loncatan penonton saat tim favorit mencetak gol*".

Faktor Fundamental

Faktor fundamental adalah elemen-elemen yang berkaitan dengan kondisi internal perusahaan serta kinerjanya dalam jangka panjang. Laba perusahaan yang konsisten meningkat biasanya menjadi sinyal positif bagi investor, bagaikan "*tanaman yang tumbuh subur karena dirawat dengan baik*". Selain laba, laporan keuangan secara keseluruhan menjadi rujukan penting untuk menilai kesehatan finansial. Informasi seperti rasio utang terhadap ekuitas, arus kas, dan margin keuntungan memberikan gambaran stabilitas perusahaan.

Kebijakan dividen juga termasuk dalam faktor fundamental. Perusahaan yang rutin membagikan dividen dengan jumlah stabil atau meningkat sering dipandang positif, seperti "*pohon buah yang tiap musim berbuah*". Sebaliknya, pemotongan dividen dapat menimbulkan sentimen negatif. Pertumbuhan sektor tempat perusahaan beroperasi turut memengaruhi daya tarik saham; sektor yang berkembang pesat layaknya "*kota yang sedang membangun banyak infrastruktur*" akan lebih menarik bagi "pendatang" baru.

Faktor Teknikal

Faktor teknikal berkaitan dengan pola dan tren yang diamati melalui harga dan volume perdagangan. Analisis teknikal memanfaatkan data historis seperti "*seorang meteorolog yang memprediksi hujan dari pola awan*". Tren harga dapat berbentuk *uptrend*, *downtrend*, atau *sideways*. Volume perdagangan yang tinggi biasanya mencerminkan minat besar terhadap saham, mirip "*antrean panjang di toko yang memberi sinyal produk sedang digemari*". Indikator seperti *moving averages*, *relative strength index* (RSI), dan *Bollinger Bands* digunakan untuk memberi sinyal kekuatan tren atau potensi pembalikan arah.

Sebagai contoh, jika harga berada di atas *moving average* jangka panjang dan RSI di bawah 70, kondisi ini dapat diibaratkan sebagai "*peluang berlayar saat angin sedang bersahabat*". Sebaliknya, RSI di atas 70 menandakan kondisi *overbought*, seperti "*kapal yang terlalu sarat muatan dan rawan terguling*".

Faktor Eksternal

Faktor eksternal mencakup peristiwa di luar kendali perusahaan yang memengaruhi harga saham. Krisis keuangan, resesi, atau kenaikan harga komoditas dapat berdampak luas; seperti *"gelombang pasang yang memengaruhi semua perahu di pelabuhan"*. Misalnya, kenaikan harga minyak dunia menguntungkan sektor energi namun membebani sektor transportasi. Perubahan kebijakan pemerintah seperti penurunan suku bunga dapat mendorong investasi, ibarat *"membuka pintu air yang membuat arus modal mengalir lebih deras"*.

Sentimen pasar akibat berita dan media sosial di era digital ibarat *"api yang cepat menyebar di padang rumput kering"*; informasi positif menciptakan euforia, sementara informasi negatif memicu kepanikan.

Risiko dalam Prediksi

Memprediksi pergerakan harga saham adalah tantangan kompleks. Data historis tidak selalu mencerminkan masa depan, sama seperti *"cuaca kemarin tidak menjamin cuaca besok"*. Perubahan mendadak akibat *black swan events* bagaikan *"gempa bumi yang datang tanpa peringatan"*, seperti krisis keuangan 2008 atau pandemi COVID-19. Peristiwa ini dapat menyebabkan volatilitas ekstrem dalam waktu singkat.

Sebagai ilustrasi matematis, volatilitas dapat dianalogikan seperti *"mengukur seberapa besar gelombang"*

yang terjadi di laut investasi" dan dihitung dengan deviasi standar return harian:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (R_i - \bar{R})^2}{n}}$$

dengan R_i adalah return harian ke- i , $\bar{R}$ adalah rata-rata return harian, dan n adalah jumlah periode pengamatan.

dengan adalah return harian ke- i , rata-rata return harian, dan jumlah periode. Nilai yang tinggi diibaratkan seperti "*ombak yang besar dan bergejolak*" yang menunjukkan risiko pergerakan harga besar.

Tabel 2. Faktor yang Mempengaruhi Volatilitas Saham

Faktor	Deskripsi	Contoh
Fundamental	Kondisi internal perusahaan yang mempengaruhi nilai saham	Laba meningkat, dividen stabil
Teknikal	Pola harga dan volume perdagangan yang memberikan sinyal pasar	<i>Uptrend</i> , volume tinggi, RSI rendah
Eksternal	Peristiwa di luar kendali perusahaan	Krisis global, kenaikan harga minyak, perubahan suku bunga
Risiko Prediksi	Ketidakpastian yang membuat prediksi sulit dilakukan	Perubahan mendadak akibat <i>black swan events</i> , data historis tidak relevan dengan kondisi baru

Pemahaman menyeluruh mengenai volatilitas dan faktor-faktor yang mempengaruhinya membantu mahasiswa maupun praktisi menganalogikan pasar saham seperti "*ekosistem hidup yang dinamis*". Dengan memahami interaksi faktor

fundamental, teknikal, dan eksternal, strategi investasi dapat disesuaikan lebih adaptif terhadap perubahan pasar.

3. Soal Esai

1. Apa yang dimaksud dengan volatilitas harga saham?
2. Sebutkan 3 faktor fundamental yang memengaruhi harga saham.
3. Bagaimana peran volume perdagangan dalam analisis teknikal?
4. Jelaskan contoh peristiwa global yang memengaruhi pasar saham.
5. Apa perbedaan risiko sistematis dan risiko tidak sistematis?
6. Mengapa sentimen pasar dapat memengaruhi harga saham?
7. Jelaskan istilah *black swan event* dalam konteks pasar keuangan.
8. Bagaimana hubungan inflasi dengan harga saham?
9. Mengapa data historis tidak selalu relevan untuk prediksi masa depan?
10. Berikan contoh indikator teknikal yang umum digunakan.

4. Tugas Individu

Pilih satu perusahaan yang terdaftar di BEI. Analisis pergerakan harga sahamnya selama 3 bulan terakhir dan

identifikasi minimal 3 faktor yang memengaruhi pergerakan tersebut. Buat laporan maksimal 2 halaman.

```
# =====
# Analisis Saham BEI: 3 Bulan Terakhir + Faktor Pendorong
# =====
!pip -q install yfinance pandas numpy matplotlib openpyxl

import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt
from datetime import timedelta

pd.set_option("display.float_format", lambda x: f"{x:,.4f}")

# ----- PARAMETER: PILIH SAHAM -----
TICKER = "BBCA.JK" # <-- Ganti sesuai pilihanmu, contoh: TLKM.JK, ASII.JK, ICBP.JK

# ----- Helper: ekstrak Close robust -----
def extract_close_series(df, ticker=None):
    """
    Ambil Series 'Close' (fallback 'Adj Close') dari beragam struktur kolom yfinance.
    """
    if df is None or df.empty:
        raise ValueError("Data kosong dari yfinance.")

    cols = df.columns
    # Kasus kolom datar
    if not isinstance(cols, pd.MultiIndex):
        if "Close" in cols:
            return df["Close"].rename("Close")
        if "Adj Close" in cols:
            return df["Adj Close"].rename("Close")
        raise KeyError(f"Tidak ada 'Close'/'Adj Close'. Kolom: {list(cols)}")

    # MultiIndex
    lv0 = list(cols.get_level_values(0))
    lv1 = list(cols.get_level_values(1))

    # (field -> ticker)
    if "Close" in lv0 or "Adj Close" in lv0:
        sub = df["Close"] if "Close" in lv0 else df["Adj Close"]
        if isinstance(sub, pd.Series):
            return sub.rename("Close")
        if ticker and ticker in sub.columns:
            return sub[ticker].rename("Close")
        return sub.iloc[:, 0].rename("Close")

    # (ticker -> field)
    if ticker and ticker in lv0:
        sub = df[ticker]
        if "Close" in sub.columns:
            return sub["Close"].rename("Close")
        if "Adj Close" in sub.columns:
            return sub["Adj Close"].rename("Close")

    # Fallback: ambil level=1 'Close' (atau 'Adj Close') across semua ticker
    if "Close" in lv1:
        return df.xs("Close", level=1, axis=1).iloc[:, 0].rename("Close")
    if "Adj Close" in lv1:
        return df.xs("Adj Close", level=1, axis=1).iloc[:, 0].rename("Close")
```

```

raise KeyError("Gagal menemukan 'Close'/'Adj Close'.")

# ----- Helper: indikator teknikal -----
def rsi(series, period=14):
    delta = series.diff()
    up = delta.clip(lower=0)
    down = -delta.clip(upper=0)
    gain = up.rolling(period).mean()
    loss = down.rolling(period).mean()
    rs = gain / loss
    return 100 - (100 / (1 + rs))

def drawdown(series):
    cummax = series.cummax()
    dd = series / cummax - 1.0
    return dd

# ----- Unduh data 3 bulan -----
# Ambil 90 hari agar mencakup 3 bulan kalender
period_days = 90

# Saham terpilih
raw = yf.download(
    TICKER, period=f"{period_days}d", interval="1d",
    group_by="column", auto_adjust=False, progress=False
)
close = extract_close_series(raw, ticker=TICKER)
px = close.to_frame("Close").copy()
px.index = pd.to_datetime(px.index)
px = px.sort_index()
px.index.name = "Date"

# Data volume (opsional, jika tersedia)
vol = None
try:

```

```

    if isinstance(raw.columns, pd.MultiIndex):
        # Cari Volume di berbagai bentuk
        if "Volume" in raw.columns.get_level_values(0):
            vol = raw["Volume"]
            if isinstance(vol, pd.DataFrame):
                if TICKER in vol.columns:
                    vol = vol[TICKER]
                else:
                    vol = vol.iloc[:, 0]
            elif TICKER in raw.columns.get_level_values(0):
                sub = raw[TICKER]
                if "Volume" in sub.columns:
                    vol = sub["Volume"]
            else:
                if "Volume" in raw.columns:
                    vol = raw["Volume"]
        except Exception:
            vol = None
    if vol is not None:
        vol = vol.reindex(px.index).astype("float")

# ----- Indikator utama -----
px["MA20"] = px["Close"].rolling(20).mean()
px["MA50"] = px["Close"].rolling(50).mean()
px["RSI14"] = rsi(px["Close"], 14)
px["Ret"] = px["Close"].pct_change()
px["DD"] = drawdown(px["Close"])

```

```

# ----- Grafik Harga + MA & Volume -----
plt.figure(figsize=(11,5))
plt.plot(px.index, px["Close"], label="Close")
plt.plot(px.index, px["MA20"], label="MA20")
plt.plot(px.index, px["MA50"], label="MA50")
plt.title(f"{TICKER} - Harga & MA (3 Bulan Terakhir)")
plt.xlabel("Tanggal"); plt.ylabel("Harga (IDR)")
plt.legend(); plt.grid(True); plt.tight_layout()
plt.show()

if vol is not None:
    plt.figure(figsize=(11,3.5))
    plt.bar(vol.index, vol.values)
    plt.title(f"{TICKER} - Volume (3 Bulan Terakhir)")
    plt.xlabel("Tanggal"); plt.ylabel("Volume")
    plt.tight_layout()
    plt.show()

# ----- Grafik RSI -----
plt.figure(figsize=(11,3.5))
plt.plot(px.index, px["RSI14"])
plt.axhline(70, linestyle="--"); plt.axhline(30, linestyle="--")
plt.title(f"{TICKER} - RSI(14)")
plt.xlabel("Tanggal"); plt.ylabel("RSI")
plt.grid(True); plt.tight_layout()
plt.show()

```

```

# ----- Ringkasan Statistik -----
def to_float(x):
    try:
        return float(x)
    except Exception:
        if hasattr(x, "item"):
            try:
                return float(x.item())
            except Exception:
                pass
        return np.nan

summary = {}
summary["Periode"] = f"{px.index.min().date()} s.d. {px.index.max().date()}"
summary["Harga awal"] = to_float(px["Close"].iloc[0])
summary["Harga akhir"] = to_float(px["Close"].iloc[-1])
summary["Total return %"] = to_float((px["Close"].iloc[-1]/px["Close"].iloc[0]-1)*100)
summary["Rata-rata harian %"] = to_float(px["Ret"].mean()*100)
summary["Vol harian %"] = to_float(px["Ret"].std()*100)
summary["Max drawdown %"] = to_float(px["DD"].min()*100)

print("Ringkasan (3 bulan):")
display(pd.DataFrame([summary]))

# =====
# Analisis Faktor (Korelasi Rolling 20 Hari)
# =====
# Faktor proxy:
factors = {
    "IHSG": "^JKSE",          # pasar Indonesia
    "USIDIDR": "USIDIDR=X",   # kurs USD/IDR
    "Brent": "BZ=F",          # minyak Brent
    "Gold": "GC=F",           # emas

```

```

}

# Unduh data faktor
fac_frames = {}
for name, sym in factors.items():
    f_raw = yf.download(sym, period=f"{period_days}d", interval="1d",
                        group_by="column", auto_adjust=False, progress=False)
    try:
        f_close = extract_close_series(f_raw, ticker=sym).rename(name)
    except Exception as e:
        print(f"Peringatan: gagal ambil {name} ({sym}): {e}")
        continue
    fac_frames[name] = f_close

if not fac_frames:
    raise RuntimeError("Tidak ada data faktor yang berhasil diunduh.")

fac_df = pd.concat(fac_frames.values(), axis=1).sort_index()
fac_df = fac_df.reindex(px.index) # sinkron ke tanggal saham
fac_ret = fac_df.pct_change()

# Return saham & faktor selaras
ret_all = pd.concat([px["Ret"].rename(TICKER), fac_ret], axis=1).dropna(how="any")

# Korelasi rolling 20-hari
win = 20
rolling_corrs = {}
for col in ret_all.columns:
    if col == TICKER:
        continue
    rolling_corrs[col] = ret_all[TICKER].rolling(win).corr(ret_all[col])

corr_df = pd.DataFrame(rolling_corrs)

```

```

# Tampilkan korelasi terakhir (snapshot)
last_corr = corr_df.dropna().iloc[-1].sort_values(key=lambda s: s.abs(), ascending=False)
print(f"Korelasi rolling {win} hari (snapshot terbaru) antara {TICKER} dan faktor:")
display(last_corr.to_frame("corr").T)

# Plot korelasi rolling
plt.figure(figsize=(11,5))
for c in corr_df.columns:
    plt.plot(corr_df.index, corr_df[c], label=c)
plt.title(f"{TICKER} - Korelasi Rolling {win} Hari dgn Faktor")
plt.axhline(0, linestyle="--")
plt.xlabel("Tanggal"); plt.ylabel("Korelasi")
plt.legend(); plt.grid(True); plt.tight_layout()
plt.show()

# Interpretasi cepat: ambil 3 faktor teratas (berdasarkan |korelasi| terbaru)
top3 = list(last_corr.index[:3])
print("Tiga faktor paling berpengaruh (statistik, berdasarkan |korelasi| terbaru):")
for i, f in enumerate(top3, 1):
    sign = "searah" if last_corr[f] > 0 else "berlawanan arah"
    print(f"{i}) {f}: korelasi {last_corr[f]:.2f} ({sign})")

# =====
# Ekspor Excel (multi-sheet)
# =====
out_xlsx = f"{TICKER.replace('.', '_')}_Analisis_3Bulan.xlsx"
with pd.ExcelWriter(out_xlsx, engine="openpyxl") as writer:
    px.to_excel(writer, sheet_name="Harga_Indikator")

```

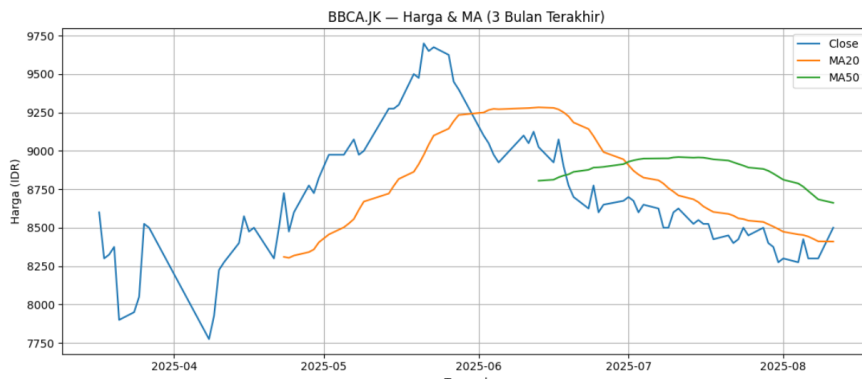
```

if vol is not None:
    vol.to_frame("Volume").to_excel(writer, sheet_name="Volume")
    fac_df.to_excel(writer, sheet_name="Faktor_Harga")
    fac_ret.to_excel(writer, sheet_name="Faktor_Return")
    corr_df.to_excel(writer, sheet_name=f"Korelasi_Rolling_{win}")
    pd.DataFrame([summary]).to_excel(writer, sheet_name="Ringkasan", index=False)
    last_corr.to_frame("Korelasi_Snapshot").to_excel(writer, sheet_name="Topline_Korelasi")

print("File Excel tersimpan:", out_xlsx)

```

Output Program



5. Studi Kasus

Setiap kelompok memilih satu peristiwa besar (misalnya, pandemi COVID-19, krisis finansial 2008, atau perang dagang AS–China) dan menganalisis dampaknya terhadap IHSG serta 3 saham unggulan. Hasil analisis disajikan dalam bentuk poster atau infografis.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

PENDEKATAN KONVENSIONAL DALAM PREDIKSI HARGA SAHAM

1. Learning Outcome (LO)

- a. Menjelaskan metode konvensional dalam prediksi harga saham.
- b. Menguraikan asumsi, kelebihan, dan keterbatasan metode ARIMA, GARCH, dan regresi linear.
- c. Mengimplementasikan model konvensional sederhana pada data harga saham.

2. Materi

Regresi Linear

Regresi linear adalah metode statistik yang digunakan untuk memodelkan hubungan antara variabel independen dan harga saham sebagai variabel dependen. Hubungan ini diasumsikan berbentuk garis lurus, di mana perubahan pada variabel independen menghasilkan perubahan yang proporsional pada harga saham. Dalam konteks pasar modal, variabel independen dapat berupa indikator ekonomi, rasio keuangan perusahaan, atau variabel teknis seperti volume perdagangan. Model regresi linear sering dianggap sebagai langkah awal dalam memahami keterkaitan antara faktor-faktor penentu dan pergerakan harga saham.

Sebagai ilustrasi, jika kita menganalogikan harga saham sebagai "*tingkat air dalam waduk*" dan variabel independen sebagai "*jumlah air yang masuk melalui saluran*", maka semakin besar jumlah air yang masuk, semakin tinggi pula

permukaan air. Demikian pula, peningkatan variabel ekonomi tertentu dapat memicu kenaikan harga saham. Rumus umum regresi linear sederhana adalah:

$$Y = a + bX + e$$

dengan Y adalah harga saham, X variabel independen, a intercept, b koefisien regresi, dan e error. Kelebihan regresi linear terletak pada kemudahan interpretasi koefisien. Misalnya, jika koefisien b bernilai positif, maka kenaikan satu unit pada variabel X diasosiasikan dengan kenaikan harga saham sebesar b unit, asumsi faktor lainnya tetap.

ARIMA (Autoregressive Integrated Moving Average)

Model *Autoregressive Integrated Moving Average* atau ARIMA adalah pendekatan *time series* yang digunakan untuk memodelkan data yang bersifat stasioner atau telah dibuat stasioner melalui proses diferensiasi. Model ini menggabungkan tiga komponen utama: autoregresif (AR), integrasi (I), dan *moving average* (MA). AR mengacu pada penggunaan nilai masa lalu variabel untuk memprediksi nilai saat ini, I menggambarkan proses pembedaan data untuk menghilangkan tren, dan MA menggunakan kesalahan prediksi masa lalu untuk memprediksi nilai saat ini. Misalnya, dalam mengamati pola harga saham, ARIMA dapat diibaratkan seperti "*navigator kapal yang memanfaatkan catatan perjalanan sebelumnya untuk menentukan arah selanjutnya*". ARIMA efektif digunakan saat data menunjukkan pola yang konsisten dari waktu ke waktu, seperti siklus musiman atau tren stabil. Sebagai contoh, data harga bulanan saham suatu sektor yang bergerak mengikuti pola musiman dapat dianalisis dengan

ARIMA untuk memperkirakan harga di bulan-bulan berikutnya.

Formula umum **ARIMA** (p, d, q) memuat parameter p untuk orde **Autoregressive (AR)**, d untuk tingkat diferensiasi, dan q untuk orde **Moving Average (MA)**.

Bentuk persamaan umum ARIMA:

$$\phi_p(B)(1 - B)^d y_t = \theta_q(B)\varepsilon_t$$

dengan:

- B adalah operator *backshift* ($By_t = y_{t-1}$)
- $\phi_p(B)$ adalah polinomial AR orde p
- $\theta_q(B)$ adalah polinomial MA orde q
- d adalah tingkat diferensiasi untuk membuat data stasioner
- ε_t adalah komponen error (*white noise*)

Model ARIMA efektif digunakan saat data memiliki pola konsisten dari waktu ke waktu, seperti siklus musiman atau tren stabil. Sebagai contoh, data harga bulanan saham suatu sektor yang bergerak mengikuti pola musiman dapat dianalisis menggunakan ARIMA untuk memproyeksikan harga pada bulan-bulan berikutnya.

GARCH (Generalized Autoregressive Conditional Heteroskedasticity)

Model *Generalized Autoregressive Conditional*

Heteroskedasticity atau GARCH dirancang untuk memodelkan dan memprediksi varians dari suatu *time series*. Berbeda dengan regresi linear atau ARIMA yang berfokus pada nilai prediksi, GARCH lebih menekankan pada perubahan volatilitas dari waktu ke waktu. Hal ini penting karena volatilitas harga saham cenderung berkelompok; periode volatilitas tinggi diikuti oleh periode serupa, begitu pula volatilitas rendah.

GARCH dapat diibaratkan seperti "*prakiraan cuaca yang tidak hanya memprediksi suhu, tetapi juga kemungkinan badai*". Model ini menggunakan varians masa lalu dan kesalahan prediksi sebelumnya untuk mengestimasi varians di periode mendatang.

Rumus umum GARCH(1,1) adalah:

$$\sigma_t^2 = \omega + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

dengan:

σ_t^2 adalah varians pada waktu t ,

ε_{t-1}^2 adalah error kuadrat pada periode sebelumnya,

ω, α, β adalah parameter yang diperkirakan dari data.

Model GARCH(1,1) sering digunakan untuk memodelkan volatilitas yang berubah-ubah (*volatility clustering*) pada data deret waktu keuangan, misalnya pergerakan harga saham atau nilai tukar.

Kelebihan Metode Konvensional

Metode konvensional seperti regresi linear, ARIMA, dan GARCH memiliki kelebihan utama pada kemudahan interpretasi. Hasil estimasi dapat langsung dihubungkan dengan hubungan sebab-akibat yang jelas. Selain itu, metode ini tidak memerlukan data berukuran sangat besar untuk memberikan hasil yang signifikan secara statistik. Dalam dunia akademik dan praktis, sifat transparan dari model ini membuatnya ideal untuk pengajaran dan pengambilan keputusan berbasis data yang terstruktur. Sebagai contoh, seorang analis dapat dengan cepat memahami dampak kenaikan suku bunga terhadap harga saham melalui koefisien regresi, atau mengidentifikasi periode risiko tinggi berdasarkan prediksi varians dari model GARCH.

Keterbatasan

Keterbatasan utama dari metode konvensional adalah kesulitan dalam menangkap pola non-linear yang sering muncul di pasar saham. Hubungan antarvariabel di pasar keuangan sering kali tidak bersifat linier, melainkan kompleks

dan berubah-ubah. Selain itu, model ini kurang responsif terhadap perubahan mendadak yang disebabkan oleh kejadian tak terduga (*black swan events*). Dalam analogi, metode ini seperti "*peta jalan yang akurat untuk rute biasa, tetapi kurang berguna saat terjadi longsor atau banjir yang memutus jalan*".

Untuk mengatasi keterbatasan ini, sering kali dibutuhkan kombinasi metode atau pengembangan model yang lebih adaptif, seperti penggunaan model non-linear atau teknik pembelajaran mesin (*machine learning*) yang mampu mengenali pola kompleks.

Tabel 3. Perbandingan Metode Konvensional untuk Harga Saham

Metode	Fokus Utama	Kelebihan	Keterbatasan
Regresi Linear	Hubungan linier variabel	Mudah diinterpretasikan, cepat diaplikasikan	Tidak menangkap pola non-linear
ARIMA	Prediksi nilai time series	Cocok untuk data stasioner, mengenali pola	Kurang efektif untuk data non-stasioner dengan lonjakan tak terduga
GARCH	Prediksi varians/volatilitas	Menangkap pola volatilitas berkelompok	Tidak memprediksi nilai, sensitif pada asumsi model

Dengan memahami ketiga metode ini, mahasiswa S1 diharapkan mampu memilih pendekatan yang sesuai berdasarkan sifat data dan tujuan analisis. Integrasi antara pemahaman teoretis dan aplikasi praktis akan mempermudah

dalam merumuskan strategi investasi yang efektif serta adaptif terhadap dinamika pasar.

3. Soal Esai

1. Jelaskan prinsip dasar regresi linear.
2. Apa yang dimaksud dengan data stasioner dalam analisis time series?
3. Sebutkan tiga komponen utama model ARIMA.
4. Jelaskan fungsi model GARCH dalam analisis harga saham.
5. Mengapa model konvensional sulit menangkap pola non-linear?
6. Apa keuntungan utama ARIMA dibandingkan regresi linear untuk prediksi harga saham?
7. Berikan contoh situasi di mana model GARCH lebih tepat digunakan.
8. Bagaimana cara menentukan orde p , d , q pada ARIMA?
9. Mengapa interpretasi menjadi keunggulan model konvensional?
10. Sebutkan keterbatasan penggunaan data historis pada model konvensional.

4. Tugas Individu

Ambil data harga penutupan harian satu saham selama 6 bulan terakhir. Buat model ARIMA sederhana dan tampilkan hasil prediksi 5 hari ke depan beserta grafiknya.

```

# =====
# ARIMA – TLKM.JK (6 Bulan) → Prediksi 5 Hari (FIX freq)
# =====
!pip -q install yfinance pandas numpy matplotlib statsmodels openpyxl
import warnings
warnings.filterwarnings("ignore")
import pandas as pd
import numpy as np
import yfinance as yf
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from pandas.tseries.offsets import BDay
from datetime import timedelta
pd.set_option("display.float_format", lambda x: f"{x:,.4f}")
# ----- PARAMETER -----
TICKER = "TLKM.JK" # ganti jika mau

# ----- Helper: ekstrak Close robust -----
def extract_close_series(df, ticker=None):
    if df is None or df.empty:
        raise ValueError("Data kosong dari yfinance.")
    cols = df.columns
    if not isinstance(cols, pd.MultiIndex):
        if "Close" in cols: return df["Close"].rename("Close")
        if "Adj Close" in cols: return df["Adj Close"].rename("Close")
        raise KeyError(f"Tidak ada 'Close'/'Adj Close'. Kolom: {list(cols)}")
    lv0 = list(cols.get_level_values(0))
    lv1 = list(cols.get_level_values(1))
    if "Close" in lv0 or "Adj Close" in lv0:
        sub = df["Close"] if "Close" in lv0 else df["Adj Close"]
        if isinstance(sub, pd.Series): return sub.rename("Close")
        if ticker and ticker in sub.columns: return sub[ticker].rename("Close")
        return sub.iloc[:, 0].rename("Close")

    if ticker and ticker in lv0:
        sub = df[ticker]
        if "Close" in sub.columns: return sub["Close"].rename("Close")
        if "Adj Close" in sub.columns: return sub["Adj Close"].rename("Close")
    if "Close" in lv1:
        return df.xs("Close", level=1, axis=1).iloc[:, 0].rename("Close")
    if "Adj Close" in lv1:
        return df.xs("Adj Close", level=1, axis=1).iloc[:, 0].rename("Close")
    raise KeyError("Gagal menemukan 'Close'/'Adj Close'.")

# ----- 1) Unduh 6 bulan data -----
period_days = 185 # ~6 bulan + buffer
raw = yf.download(
    TICKER, period=f"{period_days}d", interval="1d",
    group_by="column", auto_adjust=False, progress=False
)
close = extract_close_series(raw, ticker=TICKER).dropna()
close.index = pd.to_datetime(close.index)
close = close.sort_index()
close.name = "Close"

# Ambil tepat ~6 bulan terakhir dari last available date
end_date = close.index.max().date()
start_date = end_date - timedelta(days=183)
close_6m = close.loc[close.index.date >= start_date].copy()

```

```

if close_6m.empty():
    raise ValueError("Tidak ada data 6 bulan terakhir.")

# ----- 2) SET FREKUENSI: Business Day + Ffill -----
y = close_6m.asfreq("B") # buat kalender hari kerja
y = y.ffill() # isi libur bursa dengan nilai hari kerja terakhir
assert y.index.freqstr == "B"

# ----- 3) Pilih order ARIMA via AIC (grid kecil) -----
p_values, d_values, q_values = [0,1,2], [0,1], [0,1,2]
best_aic, best_order, best_model = np.inf, None, None

for p in p_values:
    for d in d_values:
        for q in q_values:
            try:
                res = ARIMA(y, order=(p,d,q)).fit(method_kwargs={"maxiter": 500})
                if res.aic < best_aic:
                    best_aic, best_order, best_model = res.aic, (p,d,q), res
            except Exception:
                # coba lanjut kombinasi lain
                pass

# fallback jika belum dapat model
if best_model is None:
    best_order = (0,1,0)
    best_model = ARIMA(y, order=best_order).fit(method_kwargs={"maxiter": 500})

print(f"Order ARIMA terbaik: {best_order} | AIC = {best_aic:.2f}")

# ----- 4) Prediksi 5 hari ke depan dgn index tanggal kerja -----
steps = 5
forecast_res = best_model.get_forecast(steps=steps)
mean_forecast = forecast_res.predicted_mean
conf_int = forecast_res.conf_int(alpha=0.05)

# Buat index tanggal kerja untuk 5 hari ke depan
future_idx = pd.bdate_range(y.index[-1] + BDay(1), periods=steps)

pred_df = pd.DataFrame({
    "Forecast": mean_forecast.values,
    "Lower_95": conf_int.iloc[:, 0].values,
    "Upper_95": conf_int.iloc[:, 1].values,
}, index=future_idx)
pred_df.index.name = "Date"

print("\nPrediksi 5 hari ke depan:")
display(pred_df)

# ----- 5) Grafik: history + prediksi -----
plt.figure(figsize=(11,5))
history_tail = y.tail(90) # 90 hari terakhir untuk konteks
plt.plot(history_tail.index, history_tail.values, label="History (Close)")
plt.plot(pred_df.index, pred_df["Forecast"], marker="o", label="Forecast 5D")
plt.fill_between(pred_df.index, pred_df["Lower_95"], pred_df["Upper_95"],
                 alpha=0.2, label="95% CI")
plt.title(f"{TICKER} - ARIMA{best_order} Forecast 5 Hari")
plt.xlabel("Tanggal"); plt.ylabel("Harga (IDR)")
plt.grid(True); plt.legend(); plt.tight_layout()
plt.show()

```

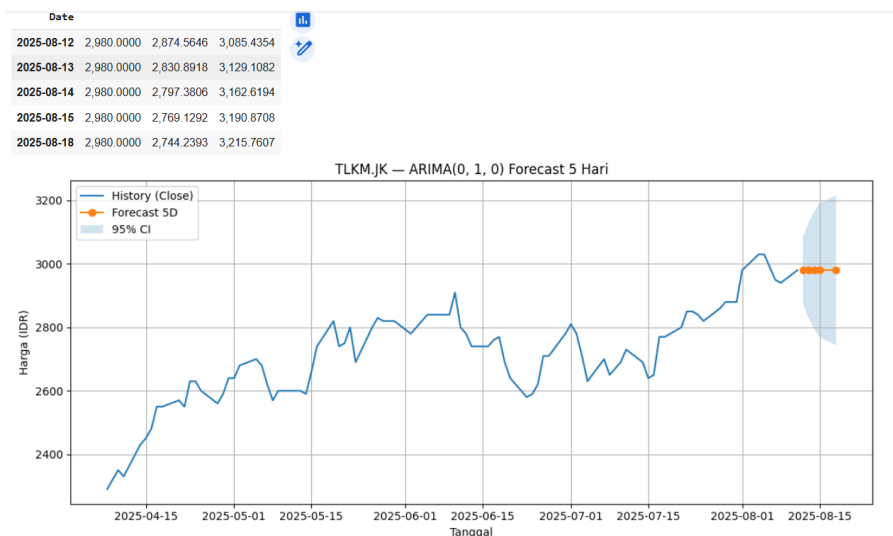
```

# ----- 6) Ekspor Excel -----
out_xlsx = f"{TICKER.replace('.', '_')}_ARIMA_6Bulan_Forecast.xlsx"
with pd.ExcelWriter(out_xlsx, engine="openpyxl") as writer:
    close_6m.to_frame("Close").to_excel(writer, sheet_name="Close_6Bulan")
    pd.DataFrame({"order": [str(best_order)],
                  "AIC": [best_aic]}).to_excel(writer,
                                                sheet_name="Model_Info", index=False)

    pred_df.to_excel(writer, sheet_name="Forecast_5D")
print("File Excel tersimpan:", out_xlsx)

```

Output Program



5. Studi Kasus

Setiap kelompok memilih dua metode (misalnya ARIMA dan GARCH), menerapkannya pada dataset harga saham yang sama, kemudian membandingkan hasil prediksi dan menguraikan perbedaan performa kedua metode.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

PENDEKATAN MODERN BERBASIS AI UNTUK PREDIKSI HARGA SAHAM

1. Learning Outcome (LO)

- a. Menjelaskan metode modern berbasis AI dalam prediksi harga saham.
- b. Menguraikan karakteristik LSTM, GRU, CNN, dan Attention.
- c. Membandingkan kelebihan dan kekurangan metode AI dengan metode konvensional.

2. Materi

LSTM (Long Short-Term Memory)

Long Short-Term Memory atau LSTM adalah jenis jaringan *Recurrent Neural Network* (RNN) yang dirancang untuk mengatasi kelemahan RNN tradisional dalam mengingat dependensi jangka panjang pada data *time series*. LSTM menggunakan *memory cell* khusus yang mampu menyimpan informasi dalam jangka waktu lama, sehingga sangat efektif untuk mempelajari pola yang terjadi dalam urutan data yang panjang. Komponen kunci dalam LSTM adalah *gate*—terdiri dari *input gate*, *forget gate*, dan *output gate*—yang mengatur kapan informasi baru masuk, kapan informasi lama dilupakan, dan kapan informasi dikeluarkan.

Sebagai contoh, bayangkan LSTM seperti "*seorang pustakawan yang menyimpan catatan penting di lemari arsip, menentukan kapan catatan lama dibuang atau disimpan kembali, dan memilih catatan yang tepat saat dibutuhkan*".

Dalam konteks pasar saham, LSTM dapat mengingat pola musiman atau tren harga yang berulang setiap beberapa bulan atau tahun, meskipun diselingi fluktuasi jangka pendek.

GRU (Gated Recurrent Unit)

Gated Recurrent Unit atau GRU adalah variasi dari LSTM yang lebih sederhana, namun memiliki kinerja yang sebanding untuk banyak kasus. GRU menggabungkan fungsi *forget gate* dan *input gate* menjadi satu *update gate*, serta menggabungkan *cell state* dan *hidden state* dalam satu representasi. Desain yang lebih ringkas ini membuat GRU lebih cepat dalam pelatihan dan sering kali membutuhkan sumber daya komputasi yang lebih sedikit dibandingkan LSTM.

Jika LSTM adalah "*pustakawan dengan tiga kunci untuk mengakses arsip dan membutuhkan prosedur lengkap*", maka GRU adalah "*pustakawan dengan satu kunci serbaguna yang dapat membuka semua arsip penting lebih cepat*". Dalam analisis harga saham, GRU sering digunakan ketika dataset tidak terlalu besar namun tetap memerlukan kemampuan untuk mengingat pola jangka panjang.

CNN (Convolutional Neural Network)

Convolutional Neural Network atau CNN awalnya dirancang untuk pengolahan citra, tetapi telah berkembang menjadi alat yang efektif dalam memproses data *time series*. CNN bekerja dengan mengekstraksi fitur penting dari data melalui operasi *convolution* dan *pooling*. Dalam konteks data

keuangan, CNN dapat mengidentifikasi pola spesifik seperti lonjakan harga, tren naik atau turun, dan anomali lainnya. CNN dapat diibaratkan seperti "*fotografer yang mengambil potongan gambar dari peristiwa pasar untuk menemukan detail yang berulang dan membandingkannya dengan album foto masa lalu*". Dengan memanfaatkan filter konvolusi, CNN mampu menangkap pola lokal dalam data yang mungkin terlewat oleh metode prediksi berbasis rata-rata.

Attention Mechanism

Attention mechanism adalah teknik yang memungkinkan model memfokuskan perhatiannya pada bagian-bagian data yang paling relevan untuk tugas tertentu. Dalam prediksi harga saham, *attention* membantu model untuk secara dinamis memilih informasi dari periode waktu tertentu yang paling berpengaruh terhadap prediksi saat ini.

Analogi yang tepat adalah "*seorang pembaca yang menyorot bagian penting dari buku sebelum ujian, lalu membacanya berulang kali untuk memastikan ia memahami poin-poin inti*". Mekanisme ini sangat berguna ketika pola dalam data tidak konsisten sepanjang waktu dan hanya periode tertentu yang memiliki relevansi tinggi terhadap hasil prediksi.

Perbandingan dengan Metode Konvensional

Metode berbasis *Artificial Intelligence* (AI) seperti LSTM, GRU, CNN, dan *attention mechanism* menawarkan fleksibilitas

yang lebih besar dibandingkan metode konvensional seperti regresi linear, ARIMA, dan GARCH. Keunggulan utamanya adalah kemampuan untuk menangkap pola non-linear dan interaksi kompleks antarvariabel yang sulit dijangkau oleh model tradisional. Selain itu, AI dapat memproses berbagai jenis data sekaligus, mulai dari harga historis, volume perdagangan, hingga berita dan sentimen media sosial.

Namun, keunggulan ini datang dengan kebutuhan akan jumlah data yang besar dan sumber daya komputasi yang signifikan. Latihan dan penyetelan parameter model AI memerlukan waktu dan keahlian khusus, serta infrastruktur yang memadai. Sebagai contoh, melatih model LSTM dengan data harga saham harian selama sepuluh tahun memerlukan kapasitas pemrosesan yang lebih tinggi dibandingkan menjalankan model ARIMA pada dataset yang sama.

Tabel 4. Perbandingan Metode AI untuk Harga Saham

Metode	Kelebihan	Kekurangan
LSTM	Mengingat pola jangka panjang, efektif untuk data sekuensial	Pelatihan lebih lambat, memerlukan data besar
GRU	Lebih sederhana dan cepat dibandingkan LSTM	Sedikit kurang fleksibel dalam mengelola memori
CNN	Menangkap pola lokal, efektif untuk ekstraksi fitur	Tidak dirancang khusus untuk data sekuensial
Attention	Memfokuskan pada informasi yang paling relevan	Kompleksitas tinggi, memerlukan interpretasi tambahan

Dengan memahami metode AI ini, mahasiswa S1 diharapkan mampu mengevaluasi dan memilih teknik yang tepat sesuai dengan karakteristik data dan tujuan analisis. Integrasi metode AI dengan pendekatan konvensional juga dapat menjadi strategi efektif, misalnya menggunakan CNN untuk ekstraksi fitur dan LSTM untuk memprediksi pergerakan harga, sambil menerapkan *attention mechanism* untuk memfokuskan prediksi pada informasi yang paling relevan.

3. Soal Esai

1. Jelaskan perbedaan utama antara LSTM dan GRU.
2. Apa fungsi utama *memory cell* pada LSTM?
3. Mengapa GRU sering dianggap lebih efisien daripada LSTM?
4. Bagaimana CNN dapat digunakan pada data time series?
5. Jelaskan konsep dasar *attention mechanism*.
6. Sebutkan dua kelebihan penggunaan AI untuk prediksi harga saham.
7. Sebutkan dua kelemahan penggunaan AI untuk prediksi harga saham.
8. Mengapa AI memerlukan dataset yang besar?
9. Bagaimana AI mengatasi masalah non-linearitas pada data pasar saham?
10. Berikan contoh kasus nyata penerapan AI dalam analisis pasar keuangan.

4. Tugas Individu

Gunakan dataset harga saham 1 tahun terakhir, lalu buat model LSTM sederhana untuk memprediksi harga penutupan 7 hari ke depan. Sertakan grafik hasil prediksi dan analisis singkat.

```
# LSTM Sederhana untuk Prediksi Harga Penutupan 7 Hari
# Dataset: 1 tahun terakhir (Yahoo Finance)
# =====
# 1) Install & Import
!pip -q install yfinance pandas numpy matplotlib scikit-learn tensorflow openpyxl
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras import layers, Sequential
from sklearn.preprocessing import MinMaxScaler
from pandas.tseries.offsets import BDay
from datetime import timedelta

# Reproducibility
np.random.seed(42)
tf.random.set_seed(42)
pd.set_option("display.float_format", lambda x: f"{x:,.4f}")

# ----- PARAMETER -----
TICKER = "BBCA.JK" # ganti sesuai kebutuhan, contoh: TLKM.JK, ASII.JK, ICBP.JK
LOOKBACK = 60 # panjang window historis (hari kerja)
FORECAST_HORIZON = 7 # prediksi 7 hari ke depan
TRAIN_RATIO = 0.8 # 80% train, 20% valid
# ----- Helper: ekstrak Close robust -----
def extract_close_series(df, ticker=None):
    """
    Ambil Series 'Close' (fallback 'Adj Close') dari beragam struktur kolom yfinance.
    """
    if df is None or df.empty:
        raise ValueError("Data kosong dari yfinance.")
    cols = df.columns

    # Kolom datar
    if not isinstance(cols, pd.MultiIndex):
        if "Close" in cols: return df["Close"].rename("Close")
        if "Adj Close" in cols: return df["Adj Close"].rename("Close")
        raise KeyError(f"Tidak ada 'Close'/'Adj Close'. Kolom: {list(cols)}")

    # MultiIndex
    lv0 = list(cols.get_level_values(0))
    lv1 = list(cols.get_level_values(1))

    # (field -> ticker)
    if "Close" in lv0 or "Adj Close" in lv0:
        sub = df["Close"] if "Close" in lv0 else df["Adj Close"]
        if isinstance(sub, pd.Series): return sub.rename("Close")
        if ticker and ticker in sub.columns: return sub[ticker].rename("Close")
        return sub.iloc[:, 0].rename("Close")
```

```

# (ticker -> field)
if ticker and ticker in lv0:
    sub = df[ticker]
    if "Close" in sub.columns: return sub["Close"].rename("Close")
    if "Adj Close" in sub.columns: return sub["Adj Close"].rename("Close")

# Fallback level=1
if "Close" in lv1:
    return df.xs("Close", level=1, axis=1).iloc[:, 0].rename("Close")
if "Adj Close" in lv1:
    return df.xs("Adj Close", level=1, axis=1).iloc[:, 0].rename("Close")

raise KeyError("Gagal menemukan 'Close'/'Adj Close'.")

# ----- 2) Unduh 1 tahun data -----
# Ambil ~400 hari untuk jaga libur akhir pekan +
# bursa -> filter 1y terakhir dari tanggal terakhir
raw = yf.download(
    TICKER, period="400d", interval="1d",
    group_by="column", auto_adjust=False, progress=False
)

close = extract_close_series(raw, ticker=TICKER).dropna()
close.index = pd.to_datetime(close.index)
close = close.sort_index()
close.name = "Close"

# Ambil tepat 1 tahun terakhir dari tanggal paling akhir yang tersedia
end_date = close.index.max().date()
start_date = end_date - timedelta(days=365)
close_1y = close.loc[close.index.date >= start_date].copy()
if close_1y.empty:
    raise ValueError("Tidak ada data 1 tahun terakhir untuk ticker ini.")

# Buat frekuensi Business Day dan isi tanggal libur dengan ffill (stabil untuk LSTM)
y = close_1y.asfreq("B").ffill()
assert y.index.freqstr == "B"

# ----- 3) Scaling & Pembuatan Sequence -----
scaler = MinMaxScaler(feature_range=(0, 1))
y_scaled = scaler.fit_transform(y.to_frame()) # shape: (n,1)

def make_sequences(arr, lookback=60):
    X, Y = [], []
    for i in range(lookback, len(arr)):
        X.append(arr[i - lookback:i, 0])
        Y.append(arr[i, 0])
    X = np.array(X)
    Y = np.array(Y)
    return X.reshape((X.shape[0], X.shape[1], 1)), Y.reshape((-1, 1))

X_all, y_all = make_sequences(y_scaled, lookback=LOOKBACK)

# Train/valid split (berdasarkan waktu, tanpa shuffle)
split_idx = int(len(X_all) * TRAIN_RATIO)
X_train, y_train = X_all[:split_idx], y_all[:split_idx]
X_val, y_val = X_all[split_idx:], y_all[split_idx:]

print(f"Samples: total={len(X_all)}, train={len(X_train)}, val={len(X_val)}")

```

```

▶ # ----- 4) Bangun & Latih Model LSTM -----
model = Sequential([
    layers.Input(shape=(LOOKBACK, 1)),
    layers.LSTM(64, return_sequences=True),
    layers.LSTM(32),
    layers.Dense(16, activation="relu"),
    layers.Dense(1)
])

model.compile(optimizer="adam", loss="mse")
early_stop = tf.keras.callbacks.EarlyStopping(
    monitor="val_loss", patience=10, restore_best_weights=True
)

history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=100,
    batch_size=32,
    callbacks=[early_stop],
    verbose=0
)

print("Training selesai. Epoch:", len(history.history["loss"]))

# ----- 5) Prediksi 7 hari ke depan (iteratif 1-step) -----
# Ambil window terakhir dari keseluruhan data untuk start prediksi
last_window = y_scaled[-LOOKBACK:].reshape(1, LOOKBACK, 1).copy()

future_scaled = []
for step in range(FORECAST_HORIZON):
    next_scaled = model.predict(last_window, verbose=0)[0, 0] # scalar
    future_scaled.append(next_scaled)
    # geser window: buang depan, tambahkan prediksi terbaru
    last_window = np.append(last_window[:, 1:, :], [[next_scaled]], axis=1)

▶ future_scaled = np.array(future_scaled).reshape(-1, 1)
future_pred = scaler.inverse_transform(future_scaled).ravel() # back to price

# Buat index tanggal kerja untuk 7 hari ke depan
future_idx = pd.bdate_range(y.index[-1] + BDay(1), periods=FORECAST_HORIZON)
pred_df = pd.DataFrame({"Forecast": future_pred}, index=future_idx)
pred_df.index.name = "Date"

print("\nPrediksi 7 hari ke depan:")
display(pred_df)

# ----- 6) Grafik: History + Prediksi -----
plt.figure(figsize=(11,5))
history_tail = y.tail(180) # 180 hari terakhir sebagai konteks
plt.plot(history_tail.index, history_tail.values, label="History (Close)")

plt.plot(pred_df.index, pred_df["Forecast"], marker="o", label="Forecast 7D")
plt.title(f"{TICKER} - LSTM Forecast 7 Hari")
plt.xlabel("Tanggal"); plt.ylabel("Harga (IDR)")
plt.grid(True); plt.legend(); plt.tight_layout()
plt.show()

```

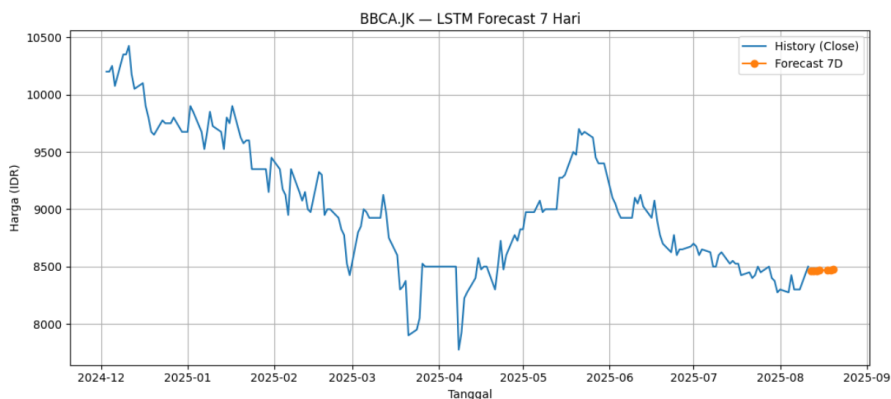
```
# ----- 7) (Optional) Simpan ke Excel -----
out_xlsx = f"{TICKER.replace('.', '_')}_LSTM_1Y_Forecast7D.xlsx"
with pd.ExcelWriter(out_xlsx, engine="openpyxl") as writer:
    y.to_frame("Close").to_excel(writer, sheet_name="Close_1Y")
    pd.DataFrame(history.history).to_excel(writer, sheet_name="Training_History")
    pred_df.to_excel(writer, sheet_name="Forecast_7D")
print("File Excel tersimpan:", out_xlsx)
```

Output program

Samples: total=201, train=160, val=41
Training selesai. Epoch: 14

Prediksi 7 hari ke depan:

Forecast	
Date	
2025-08-12	8,464.9688
2025-08-13	8,464.2881
2025-08-14	8,465.2168
2025-08-15	8,467.2070
2025-08-18	8,469.8828
2025-08-19	8,472.9805
2025-08-20	8,476.3262



5. Studi Kasus

Setiap kelompok membandingkan hasil prediksi antara model LSTM dan CNN untuk dataset yang sama. Analisis akurasi menggunakan metrik MAE dan RMSE, lalu simpulkan model mana yang lebih unggul.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

ARSITEKTUR CNN UNTUK DATA TIME SERIES HARGA SAHAM

1. Learning Outcome (LO)

- a. Menjelaskan prinsip kerja CNN (Convolutional Neural Network) pada data time series.
- b. Mengidentifikasi komponen utama CNN seperti convolution layer, pooling layer, dan activation function.
- c. Mengimplementasikan CNN 1D untuk ekstraksi fitur data harga saham.

2. Materi

Konsep CNN

Convolutional Neural Network atau CNN adalah jenis arsitektur jaringan saraf tiruan yang awalnya dikembangkan untuk keperluan *computer vision*, khususnya dalam mengenali objek pada gambar. Namun, seiring perkembangan riset dan aplikasi kecerdasan buatan, CNN juga terbukti efektif untuk memproses data *time series*, termasuk data harga saham. CNN bekerja dengan prinsip mengekstraksi fitur-fitur penting dari data mentah melalui serangkaian operasi matematis yang terstruktur, sehingga pola-pola tersembunyi yang relevan dapat diidentifikasi secara otomatis tanpa memerlukan rekayasa fitur manual dalam jumlah besar. CNN pada dasarnya beroperasi dengan cara yang menyerupai "*pemindai pola otomatis yang mencari pola tersembunyi di balik data mentah*". Seperti halnya mata manusia yang mengenali bentuk wajah melalui deteksi garis dan sudut, CNN memindai data untuk menemukan pola numerik tertentu yang konsisten. Pola ini

kemudian digunakan sebagai representasi yang lebih ringkas dan bermakna bagi model untuk melakukan prediksi.

Convolution Layer

Lapisan konvolusi (*convolution layer*) adalah inti dari CNN. Pada lapisan ini, data dipindai menggunakan *kernel* atau *filter*, yaitu matriks kecil yang bergerak secara sistematis di atas data untuk mendeteksi pola lokal. Setiap *kernel* memiliki bobot yang dilatih selama proses pembelajaran. Hasil dari operasi konvolusi menghasilkan *feature map*, yaitu representasi data yang menyoroti keberadaan pola tertentu.

Sebagai analogi, *convolution layer* dapat dibayangkan seperti "*sapu detektor logam yang menggerakkan sensor di sepanjang pantai untuk menemukan koin yang terkubur dan menandai lokasinya di peta*". Dalam konteks harga saham, *kernel* dapat belajar mengenali lonjakan mendadak, penurunan tajam, atau perubahan kecil yang berulang. Secara matematis, operasi konvolusi 1D pada data *time series* dapat dituliskan sebagai: dengan x adalah data input, w adalah bobot kernel berukuran k , dan p adalah posisi dalam data.

Pooling Layer

Setelah fitur diekstraksi oleh lapisan konvolusi, langkah berikutnya adalah *pooling layer*. Tujuan utama *pooling* adalah mereduksi dimensi data sekaligus mempertahankan informasi penting. Metode *pooling* yang umum digunakan meliputi *max pooling* (memilih nilai maksimum dalam jendela tertentu) dan

average pooling (mengambil nilai rata-rata dalam jendela tersebut).

Analoginya seperti "*meringkas buku menjadi poin-poin utama tanpa menghilangkan inti cerita, sehingga pembaca tetap memahami jalan cerita meski halaman dipangkas*". Dalam analisis harga saham, *pooling* membantu model tetap fokus pada pola signifikan sekaligus mengurangi beban komputasi.

Activation Function

Setelah proses konvolusi dan *pooling*, data melewati *activation function* untuk menambahkan sifat non-linear pada model. Tanpa fungsi aktivasi, CNN hanya akan menjadi model linier yang terbatas dalam kemampuannya menangkap pola kompleks. Fungsi aktivasi umum meliputi:

a) **ReLU (Rectified Linear Unit)**

efektif mengatasi masalah *vanishing gradient*.

b) **Sigmoid**

Menghasilkan keluaran antara 0 dan 1, sering digunakan untuk prediksi probabilitas.

c) **Tanh**

Menghasilkan keluaran antara -1 dan 1, memberikan normalisasi data secara alami.

Analogi fungsi aktivasi adalah seperti "*pintu geser yang memutuskan bagian mana dari informasi yang boleh masuk ke tahap berikutnya, dan bagian mana yang diblokir*".

Penerapan pada Time Series

CNN 1D sangat berguna untuk menangkap pola lokal dalam data *time series* seperti harga saham. Sebelum diproses oleh model sekuensial seperti LSTM atau GRU, CNN dapat digunakan sebagai tahap awal ekstraksi fitur. Pendekatan ini menggabungkan kekuatan CNN dalam mendeteksi pola lokal dengan kemampuan LSTM/GRU dalam mengingat pola jangka panjang. Sebagai contoh, data harga saham harian selama beberapa tahun dapat diolah dengan CNN 1D untuk mengidentifikasi pola *spike* harga mingguan, kemudian hasil ekstraksi fitur tersebut diberikan ke LSTM untuk memprediksi tren jangka panjang. Analogi sederhananya adalah "*menggunakan mikroskop untuk melihat detail serat kain sebelum memutuskan pola busana yang akan dibuat*".

Contoh Implementasi Python

```

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, LSTM, Dense
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(timesteps, features)))
model.add(MaxPooling1D(pool_size=2))
model.add(LSTM(50, activation='tanh'))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')

```

Tabel 5. Komponen CNN untuk Analisis Time Series

Komponen	Fungsi Utama	Analogi
Convolution Layer	Mengekstrak pola lokal dari data	"Sapu detektor logam di pantai untuk mencari koin"
Pooling Layer	Mengurangi dimensi data dan	"Ringkasan buku menjadi poin inti"

	mempertahankan fitur penting	
Activation Func.	Menambahkan non-linearitas	" <i>Pintu geser yang memilih informasi masuk</i> "
CNN 1D pada Time Series	Menangkap pola lokal sebelum pemrosesan sekuensial	" <i>Menyaring pola mingguan sebelum melihat tren tahunan</i> "

Dengan memahami konsep CNN, lapisan konvolusi, *pooling*, fungsi aktivasi, dan penerapannya pada *time series*, mahasiswa S1 diharapkan mampu mengintegrasikan pendekatan ini ke dalam analisis prediksi harga saham. Pemahaman ini tidak hanya bermanfaat untuk meningkatkan akurasi model, tetapi juga untuk mengembangkan strategi prediksi yang memadukan kekuatan berbagai arsitektur jaringan saraf.

3. Soal Esai

1. Jelaskan prinsip kerja CNN secara umum.
2. Apa fungsi convolution layer dalam CNN?
3. Jelaskan kegunaan pooling layer.
4. Sebutkan dua jenis pooling yang umum digunakan.
5. Apa peran activation function dalam CNN?
6. Mengapa CNN 1D cocok untuk data time series?
7. Bagaimana pemilihan ukuran kernel mempengaruhi hasil ekstraksi fitur?
8. Apa perbedaan CNN untuk gambar dan CNN untuk data time series?

9. Jelaskan bagaimana CNN dapat dikombinasikan dengan RNN untuk prediksi saham.
10. Berikan contoh kasus nyata penggunaan CNN dalam analisis data keuangan.

4. Tugas Individu

Buat model CNN 1D sederhana untuk memproses data harga penutupan harian satu saham selama 1 tahun terakhir. Tampilkan hasil ekstraksi fitur sebelum masuk ke layer prediksi.

```
# =====  
# CNN 1D - BBRI.JK | Prediksi Multi-step 30 Hari (Business Days)  
# Dataset: Harga penutupan 1 tahun terakhir (Yahoo Finance)  
# =====  
|  
!pip -q install yfinance pandas numpy matplotlib scikit-learn tensorflow openpyxl  
  
import os, warnings  
warnings.filterwarnings("ignore")  
os.environ["TF_CPP_MIN_LOG_LEVEL"] = "2"  
  
import numpy as np  
import pandas as pd  
import yfinance as yf  
import matplotlib.pyplot as plt  
  
import tensorflow as tf  
from tensorflow.keras import layers, Model  
from sklearn.preprocessing import MinMaxScaler  
from pandas.tseries.offsets import BDay  
from datetime import timedelta  
  
# -- Stabilkan re-run --  
tf.keras.backend.clear_session()  
np.random.seed(42)  
tf.random.set_seed(42)  
  
pd.set_option("display.float_format", lambda x: f"{x:,.4f}")
```

```

# ----- PARAMETER -----
TICKER = "BBRI.JK" # ganti jika perlu (mis. ^JKSE untuk IHSG)
LOOKBACK = 60 # panjang jendela historis (hari kerja)
FORECAST_HORIZON = 30 # prediksi 30 hari kerja ke depan
TRAIN_RATIO = 0.8
EPOCHS = 80
BATCH_SIZE = 32

# ----- Helper: ekstrak Close robust -----
def extract_close_series(df, ticker=None):
    """Ambil Series 'Close' (fallback 'Adj Close') dari beragam struktur kolom yfinance."""
    if df is None or df.empty:
        raise ValueError("Data kosong dari yfinance.")
    cols = df.columns
    # Kolom datar
    if not isinstance(cols, pd.MultiIndex):
        if "Close" in cols: return df["Close"].rename("Close")
        if "Adj Close" in cols: return df["Adj Close"].rename("Close")
        raise KeyError(f"Tidak ada 'Close'/'Adj Close'. Kolom: {list(cols)}")
    # MultiIndex
    lv0 = list(cols.get_level_values(0))
    lv1 = list(cols.get_level_values(1))
    # (field -> ticker)
    if "Close" in lv0 or "Adj Close" in lv0:
        sub = df["Close"] if "Close" in lv0 else df["Adj Close"]
        if isinstance(sub, pd.Series): return sub.rename("Close")
        if ticker and ticker in sub.columns: return sub[ticker].rename("Close")
        return sub.iloc[:, 0].rename("Close")
    # (ticker -> field)
    if ticker and ticker in lv0:
        sub = df[ticker]
        if "Close" in sub.columns: return sub["Close"].rename("Close")
        if "Adj Close" in sub.columns: return sub["Adj Close"].rename("Close")
    # Fallback level=1
    if "Close" in lv1:

```

```

# Fallback level=1
    if "Close" in lv1:
        return df.xs("Close", level=1, axis=1).iloc[:, 0].rename("Close")
    if "Adj Close" in lv1:
        return df.xs("Adj Close", level=1, axis=1).iloc[:, 0].rename("Close")
    raise KeyError("Gagal menemukan 'Close'/'Adj Close'.")

# ----- 1) Unduh 1 tahun data -----
raw = yf.download(
    TICKER, period="400d", interval="1d",
    group_by="column", auto_adjust=False, progress=False
)
close = extract_close_series(raw, ticker=TICKER).dropna()
close.index = pd.to_datetime(close.index)
close = close.sort_index()
close.name = "Close"

# Ambil tepat 1 tahun terakhir dari tanggal paling akhir tersedia
end_date = close.index.max().date()
start_date = end_date - timedelta(days=365)
close_1y = close.loc[close.index.date >= start_date].copy()
if close_1y.empty:
    raise ValueError("Tidak ada data 1 tahun terakhir untuk ticker ini.")

# Frekuensi Business Day + ffill agar stabil untuk CNN
y = close_1y.asfreq("B").ffill()
assert y.index.freqstr == "B"

```

```

# ----- 2) Scaling & sequence (1-step ahead untuk training) -----
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))
y_scaled = scaler.fit_transform(y.to_frame()) # (n,1)

def make_sequences(arr, lookback=60):
    X, Y = [], []
    for i in range(lookback, len(arr)):
        X.append(arr[i-lookback:i, 0])
        Y.append(arr[i, 0])
    X = np.array(X); Y = np.array(Y)
    return X.reshape((X.shape[0], X.shape[1], 1)), Y.reshape((-1, 1))

X_all, y_all = make_sequences(y_scaled, LOOKBACK)
split_idx = int(len(X_all) * TRAIN_RATIO)
X_train, y_train = X_all[:split_idx], y_all[:split_idx]
X_val, y_val = X_all[split_idx:], y_all[split_idx:]
print(f"Samples total={len(X_all)}, train={len(X_train)}, val={len(X_val)}")

# ----- 3) Bangun CNN 1D -----
inp = layers.Input(shape=(LOOKBACK, 1))
x = layers.Conv1D(32, kernel_size=5, padding="causal", activation="relu", name="conv1")(inp)
x = layers.MaxPooling1D(pool_size=2, name="pool1")(x)
x = layers.Conv1D(64, kernel_size=3, padding="causal", activation="relu", name="conv2")(x)
x = layers.GlobalAveragePooling1D(name="gap")(x)
feat = layers.Dense(32, activation="relu", name="features")(x)
out = layers.Dense(1, name="prediction")(feat)
model = Model(inputs=inp, outputs=out)
model.compile(optimizer="adam", loss="mse", metrics=["mae"])
model.summary()

# ----- 4) Training -----
early_stop = tf.keras.callbacks.EarlyStopping(
    monitor="val_loss", patience=10, restore_best_weights=True
)
history = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=EPOCHS, batch_size=BATCH_SIZE,
    callbacks=[early_stop], verbose=0
)
print("Training selesai. Epoch:", len(history.history["loss"]))

# ----- 5) Prediksi multi-step 30 hari ke depan -----
# Ambil window terakhir dari data skala [0,1]
last_window = y_scaled[-LOOKBACK:].reshape(1, LOOKBACK, 1).copy()
future_scaled = []
for step in range(FORECAST_HORIZON):
    # panggil model tanpa retracing
    next_scaled = model(last_window, training=False).numpy()[0, 0]
    future_scaled.append(next_scaled)
    # geser window & tambahkan prediksi terbaru
    last_window = np.append(last_window[:, 1:, :], [[next_scaled]], axis=1)

future_scaled = np.array(future_scaled).reshape(-1, 1)
future_pred = scaler.inverse_transform(future_scaled).ravel()

# Index tanggal kerja untuk 30 hari ke depan
future_idx = pd.bdate_range(y.index[-1] + BDay(1), periods=FORECAST_HORIZON)
pred_df = pd.DataFrame({"Forecast": future_pred}, index=future_idx)
pred_df.index.name = "Date"

```

```

print("\nPrediksi 30 hari kerja ke depan (BBRI.JK):")
display(pred_df.head(10))
print("..."); display(pred_df.tail(10))

# ----- 6) Grafik: History + Forecast 30D -----
plt.figure(figsize=(11,5))
history_tail = y.tail(180) # 180 hari terakhir sebagai konteks
plt.plot(history_tail.index, history_tail.values, label="History (Close)")
plt.plot(pred_df.index, pred_df["Forecast"], marker="o", label="Forecast 30D")
plt.title(f"{TICKER} - CNN 1D Forecast 30 Hari (Business Days)")
plt.xlabel("Tanggal"); plt.ylabel("Harga (IDR)")
plt.grid(True); plt.legend(); plt.tight_layout()
plt.show()

# ----- 7) (Opsional) Simpan ke Excel -----
out_xlsx = f"{TICKER.replace('.', '_')}_CNN1D_1Y_Forecast30D.xlsx"
with pd.ExcelWriter(out_xlsx, engine="openpyxl") as writer:
    y.to_frame("Close").to_excel(writer, sheet_name="Close_1Y")
    pd.DataFrame(history.history).to_excel(writer, sheet_name="Training_History")
    pred_df.to_excel(writer, sheet_name="Forecast_30D")
print("File Excel tersimpan:", out_xlsx)

```

Output Program

Samples total=201, train=160, val=41
Model: "functional"

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 60, 1)	0
conv1 (Conv1D)	(None, 60, 32)	192
pool1 (MaxPooling1D)	(None, 30, 32)	0
conv2 (Conv1D)	(None, 30, 64)	6,208
gap (GlobalAveragePooling1D)	(None, 64)	0
features (Dense)	(None, 32)	2,080
prediction (Dense)	(None, 1)	33

Total params: 8,513 (33.25 KB)
Trainable params: 8,513 (33.25 KB)
Non-trainable params: 0 (0.00 B)
Training selesai. Epoch: 12

Prediksi 30 hari kerja ke depan (BBRI.JK):

	Forecast 
Date 	
2025-08-12	3,792.8738
2025-08-13	3,788.9124
2025-08-14	3,784.6736
2025-08-15	3,780.3713
2025-08-18	3,774.6238
2025-08-19	3,770.5266

5. Studi Kasus

Setiap kelompok mengimplementasikan dua model CNN 1D dengan ukuran kernel yang berbeda pada dataset harga saham yang sama, kemudian membandingkan akurasi prediksi. Presentasikan hasil perbandingan tersebut.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

ARSITEKTUR GRU & BIGRU UNTUK DATA TIME SERIES HARGA SAHAM

1. Learning Outcome (LO)

- a. Menjelaskan konsep dasar GRU (*Gated Recurrent Unit*) dan BiGRU (*Bidirectional GRU*).
- b. Menguraikan fungsi *update gate* dan *reset gate* dalam GRU.
- c. Mengimplementasikan BiGRU untuk memproses data harga saham.

2. Materi

GRU (Gated Recurrent Unit)

Gated Recurrent Unit atau GRU adalah salah satu varian dari *Recurrent Neural Network* (RNN) yang dikembangkan untuk mengatasi masalah *vanishing gradient* yang umum terjadi pada RNN tradisional. GRU dirancang lebih sederhana dibandingkan *Long Short-Term Memory* (LSTM) tetapi tetap mampu menangkap dependensi jangka panjang dalam data sekuensial. Kesederhanaan GRU terletak pada jumlah *gate* yang digunakan, yakni hanya dua: *update gate* dan *reset gate*. Arsitektur ini menghilangkan *cell state* terpisah seperti pada LSTM dan menggabungkannya dengan *hidden state*, sehingga jumlah parameter yang dilatih menjadi lebih sedikit.

GRU sering diibaratkan seperti "*pintu ganda otomatis di sebuah gedung yang memutuskan kapan membuka lebar untuk membiarkan informasi masuk dan kapan menutup untuk mempertahankan informasi lama*". Dengan mekanisme ini,

GRU dapat menyesuaikan berapa banyak informasi lama yang harus dipertahankan dan berapa banyak informasi baru yang harus ditambahkan pada setiap langkah waktu.

Update Gate

Update gate berfungsi untuk mengontrol seberapa banyak informasi lama yang dipertahankan dari langkah sebelumnya. Nilai *update gate* yang tinggi berarti model lebih cenderung mempertahankan informasi lama, sementara nilai yang rendah membuat model lebih fokus pada informasi baru.

Secara matematis, *update gate* dapat dihitung menggunakan fungsi sigmoid:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

dengan z_t adalah *update gate*, W_z bobot yang dilatih, h_{t-1} *hidden state* sebelumnya, dan x_t input saat ini.

Analogi sederhananya adalah seperti "*memilih berapa banyak catatan lama yang perlu disimpan di meja kerja sebelum menambahkan catatan baru*".

Reset Gate

Reset gate mengatur seberapa banyak informasi lama yang diabaikan. Nilai *reset gate* yang rendah membuat model mengabaikan sebagian besar informasi masa lalu, yang berguna saat data lama dianggap tidak relevan untuk prediksi saat ini.

Persamaan matematisnya adalah:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

dengan r_t adalah *reset gate* dan W_r bobot yang dilatih.

Analogi *reset gate* adalah seperti "*menghapus papan tulis untuk menulis informasi baru tanpa terganggu oleh catatan lama*".

BiGRU (Bidirectional GRU)

Bidirectional GRU atau BiGRU adalah pengembangan GRU yang memproses data dalam dua arah, yaitu maju (*forward*) dan mundur (*backward*). Tujuannya adalah untuk menangkap konteks dari masa lalu dan masa depan secara bersamaan. Hal ini sangat berguna ketika keseluruhan urutan data tersedia sebelum prediksi dilakukan, seperti pada analisis teks atau data historis harga saham. BiGRU dapat dianalogikan seperti "*membaca buku dari awal hingga akhir, lalu membacanya lagi dari akhir ke awal untuk mendapatkan pemahaman yang lebih lengkap*". Dengan memanfaatkan dua arah pemrosesan, BiGRU dapat mengenali pola yang mungkin terlewat jika hanya membaca data dalam satu arah.

Keunggulan GRU/BiGRU

Dibandingkan LSTM, GRU dan BiGRU memiliki beberapa keunggulan penting:

1. Komputasi Lebih Ringan

Dengan arsitektur yang lebih sederhana, jumlah parameter GRU lebih sedikit sehingga waktu pelatihan lebih cepat.

2. Performa Setara

Dalam banyak aplikasi *time series*, performa GRU setara dengan LSTM meskipun menggunakan lebih sedikit sumber daya.

3. Fleksibilitas

BiGRU dapat menangkap informasi dua arah yang bermanfaat dalam skenario prediksi di mana konteks penuh data tersedia.

Contoh Implementasi Python GRU dan BiGRU

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Bidirectional, Dense

# Model GRU
model_gru = Sequential()
model_gru.add(GRU(64, activation='tanh', input_shape=(timesteps, features)))
model_gru.add(Dense(1))
model_gru.compile(optimizer='adam', loss='mse')

# Model BiGRU
model_bigru = Sequential()
model_bigru.add(Bidirectional(GRU(64, activation='tanh'), input_shape=(timesteps, features)))
model_bigru.add(Dense(1))
model_bigru.compile(optimizer='adam', loss='mse')
```

Tabel 6. Perbandingan GRU dan BiGRU

Aspek	GRU	BiGRU
Arah Pemrosesan	Satu arah (maju)	Dua arah (maju dan mundur)
Kompleksitas	Lebih rendah dibanding LSTM	Lebih tinggi dari GRU tapi masih lebih ringan dari LSTM
Kebutuhan Data	Cocok untuk data sekuensial biasa	Cocok untuk data dengan konteks penuh masa lalu dan depan
Performa	Setara dengan LSTM untuk banyak kasus	Potensial lebih baik jika konteks dua arah relevan

Dengan memahami konsep GRU, *update gate*, *reset gate*, dan BiGRU, mahasiswa S1 diharapkan mampu menerapkan metode ini dalam berbagai permasalahan prediksi

time series. Pengetahuan ini memungkinkan pengembangan model yang efisien namun tetap akurat, serta membuka peluang integrasi dengan arsitektur lain seperti CNN atau mekanisme *attention* untuk meningkatkan kualitas prediksi

3. Soal Esai

1. Jelaskan perbedaan GRU dan LSTM.
2. Apa fungsi *update gate* pada GRU?
3. Apa fungsi *reset gate* pada GRU?
4. Mengapa BiGRU memproses data dari dua arah?
5. Apa keuntungan BiGRU dibandingkan GRU biasa?
6. Sebutkan satu keunggulan GRU dibanding LSTM dalam hal komputasi.
7. Mengapa GRU cocok untuk data harga saham?
8. Bagaimana BiGRU membantu meningkatkan akurasi prediksi?
9. Jelaskan potensi kelemahan penggunaan GRU/BiGRU.
10. Berikan contoh aplikasi BiGRU di luar prediksi harga saham.

4. Tugas Individu

Implementasikan model GRU dan BiGRU untuk dataset harga saham 1 tahun terakhir. Bandingkan waktu pelatihan dan akurasi kedua model.

```

# === GRU vs BiGRU untuk Prediksi Harga Saham (BRIS.JK) - 1 Tahun Terakhir ===
# Jalankan sel ini sekaligus. Disarankan Runtime -> Change runtime type -> GPU.

# 0) Setup & install
!pip -q install yfinance

import math, time, warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt
from IPython.display import display
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Bidirectional, Dense, Dropout
from tensorflow.keras.callbacks import EarlyStopping

# 1) Reproducibility + Info
SEED = 42
np.random.seed(SEED)
tf.random.set_seed(SEED)
print("TensorFlow:", tf.__version__)
print("GPU available:", len(tf.config.list_physical_devices('GPU')) > 0)

```

```

# 2) Parameter utama (silakan ubah kalau perlu)
ticker = 'BRIS.JK' # Bank Syariah Indonesia
period = '1y' # 1 tahun terakhir
price_col = 'Close' # target
window_size = 30 # panjang jendela input
test_ratio = 0.2 # porsi test dari total sequence
EPOCHS = 100
BATCH = 32
PATIENCE = 10

# 3) Ambil data harga penutupan harian
df = yf.download(ticker, period=period, interval='1d')
if df.empty:
    raise ValueError("Data dari yfinance kosong. Coba jalankan ulang, periksa koneksi, atau ganti periode.")

df = df[[price_col]].copy()
df[price_col] = df[price_col].ffill().bfill()
print(f"Jumlah hari trading: {len(df)} | Rentang: {df.index.min().date()} - {df.index.max().date()}")
display(df.tail())

# 4) Scaling & windowing (prediksi t+1 dari window_size hari terakhir)
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(df[[price_col]].values)

def make_sequences(series, win):
    X, y = [], []
    for i in range(win, len(series)):
        X.append(series[i-win:i])
        y.append(series[i])
    return np.array(X), np.array(y)

```

```

X_all, y_all = make_sequences(scaled, window_size)

n_total = X_all.shape[0]
if n_total < 10:
    raise ValueError("Sequence sangat sedikit. Kurangi window_size atau perpanjang periode data.")
n_train = int((1 - test_ratio) * n_total)

X_train, y_train = X_all[:n_train], y_all[:n_train]
X_test, y_test = X_all[n_train:], y_all[n_train:]

# Tambah dimensi fitur (1 fitur: Close)
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], 1))
X_test = X_test.reshape((X_test.shape[0], X_test.shape[1], 1))
print("Train size:", X_train.shape, "| Test size:", X_test.shape)

# 5) Arsitektur model: GRU & BiGRU (hyperparameter disamakan)
def build_gru(input_shape):
    model = Sequential([
        GRU(64, return_sequences=True, input_shape=input_shape),
        Dropout(0.2),
        GRU(32),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mse')
    return model

def build_bigru(input_shape):
    model = Sequential([
        Bidirectional(GRU(64, return_sequences=True), input_shape=input_shape),
        Dropout(0.2),
        Bidirectional(GRU(32)),
        Dense(1)
    ])

```

```

    ])
    model.compile(optimizer='adam', loss='mse')
    return model

input_shape = (X_train.shape[1], X_train.shape[2])
gru_model = build_gru(input_shape)
bigru_model = build_bigru(input_shape)

# 6) Latih & ukur waktu (pakai EarlyStopping untuk efisiensi)
early_stop = EarlyStopping(monitor='val_loss', patience=PATIENCE, restore_best_weights=True)
history = {}

t0 = time.perf_counter()
hist_gru = gru_model.fit(
    X_train, y_train,
    validation_split=0.2,
    epochs=EPOCHS,
    batch_size=BATCH,
    callbacks=[early_stop],
    verbose=0
)
time_gru = time.perf_counter() - t0
history['GRU'] = hist_gru
print(f"Waktu latih GRU : {time_gru:.2f} detik | Epochs: {len(hist_gru.history['loss'])}")

t0 = time.perf_counter()
hist_bigru = bigru_model.fit(
    X_train, y_train,
    validation_split=0.2,
    epochs=EPOCHS,
    batch_size=BATCH,
    callbacks=[early_stop],
    verbose=0
)

```

```

)
time_bigru = time.perf_counter() - t0
history['BiGRU'] = hist_bigru
print(f'Waktu latih BiGRU : {time_bigru:.2f} detik | Epochs: {len(hist_bigru.history['loss'])}')

# 7) Evaluasi: MAE, MSE, RMSE, MAPE, Directional Accuracy
def inverse_transform(arr):
    return scaler.inverse_transform(arr.reshape(-1, 1)).flatten()

def evaluate(model, X_te, y_te):
    preds_scaled = model.predict(X_te, verbose=0).flatten()
    y_scaled = y_te.flatten()
    preds = inverse_transform(preds_scaled)
    y_true = inverse_transform(y_scaled)

    mae = mean_absolute_error(y_true, preds)
    mse = mean_squared_error(y_true, preds)
    rmse = math.sqrt(mse)
    mape = np.mean(np.abs((y_true - preds) / (y_true + 1e-8))) * 100

    # Directional Accuracy: kecocokan arah Δy (naik/turun)
    sign_true = np.sign(np.diff(y_true))
    sign_pred = np.sign(np.diff(preds))
    dir_acc = (sign_true == sign_pred).mean() * 100 if len(sign_true) > 0 else np.nan

    return {
        'MAE': mae, 'MSE': mse, 'RMSE': rmse, 'MAPE_%': mape,
        'Directional_Accuracy_%': dir_acc,
        'y_true': y_true, 'preds': preds
    }

res_gru = evaluate(gru_model, X_test, y_test)
res_bigru = evaluate(bigru_model, X_test, y_test)

```

```

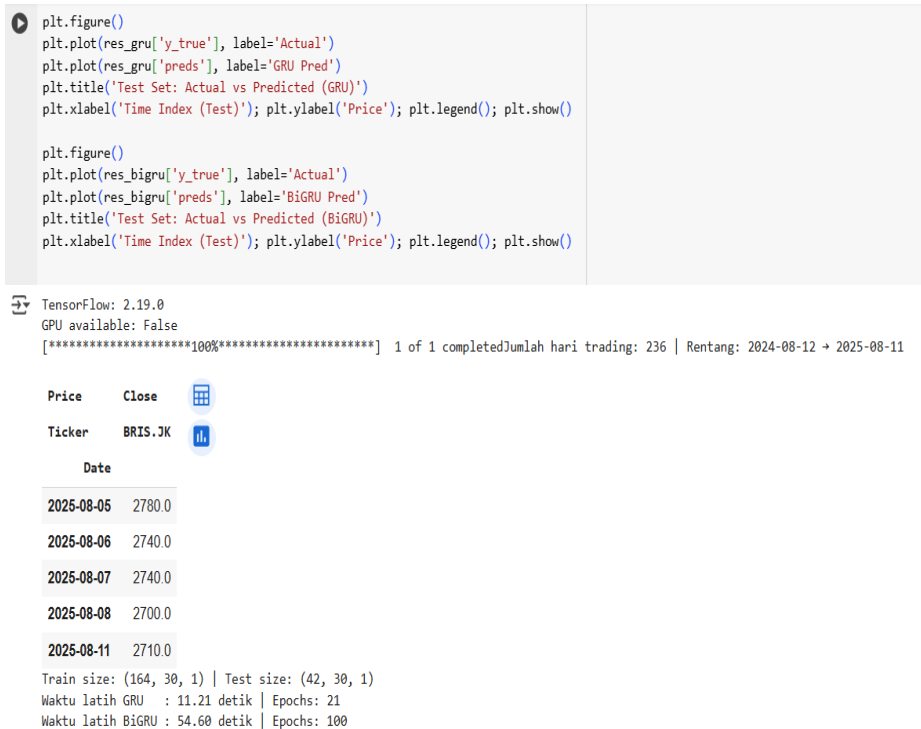
res_gru = evaluate(gru_model, X_test, y_test)
res_bigru = evaluate(bigru_model, X_test, y_test)

summary = pd.DataFrame([
    {
        'Model': 'GRU',
        'Train_Time_s': time_gru,
        **{k: v for k, v in res_gru.items() if k not in ['y_true', 'preds']}
    },
    {
        'Model': 'BiGRU',
        'Train_Time_s': time_bigru,
        **{k: v for k, v in res_bigru.items() if k not in ['y_true', 'preds']}
    }
])
display(summary)

# 8) Plot: history loss & kurva prediksi vs aktual (satu plot per gambar)
plt.figure()
plt.plot(history['GRU'].history['loss'], label='GRU - train loss')
plt.plot(history['GRU'].history['val_loss'], label='GRU - val loss')
plt.title('Training History - GRU')
plt.xlabel('Epoch'); plt.ylabel('Loss (MSE)'); plt.legend(); plt.show()

plt.figure()
plt.plot(history['BiGRU'].history['loss'], label='BiGRU - train loss')
plt.plot(history['BiGRU'].history['val_loss'], label='BiGRU - val loss')
plt.title('Training History - BiGRU')
plt.xlabel('Epoch'); plt.ylabel('Loss (MSE)'); plt.legend(); plt.show()

```



5. Studi Kasus

Setiap kelompok membangun model hybrid CNN–BiGRU untuk prediksi harga saham, kemudian membandingkan hasilnya dengan model BiGRU murni. Sajikan analisis perbedaan kinerja.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

INTEGRASI CNN BIGRU DENGAN MEKANISME ATTENTION

1. Learning Outcome (LO)

- a. Menjelaskan konsep integrasi CNN dan BiGRU dalam pemodelan data time series.
- b. Menguraikan fungsi mekanisme Attention untuk meningkatkan kinerja model.
- c. Mendesain arsitektur hybrid CNN–BiGRU–Attention untuk prediksi harga saham.

2. Materi

Kombinasi CNN–BiGRU

Kombinasi *Convolutional Neural Network* (CNN) dengan *Bidirectional Gated Recurrent Unit* (BiGRU) merupakan pendekatan hibrida yang memanfaatkan keunggulan masing-masing arsitektur untuk memproses data *time series*, khususnya harga saham. CNN berfungsi untuk mengekstraksi fitur-fitur lokal yang terdapat pada data, sedangkan BiGRU digunakan untuk memodelkan hubungan temporal dua arah. Pendekatan ini memungkinkan model memahami baik pola lokal maupun konteks jangka panjang secara bersamaan, ibarat "*tim arkeolog yang memetakan detail artefak sebelum menggabungkannya dalam urutan kronologis sejarah*".

CNN bekerja dengan memindai data menggunakan *filter* untuk menemukan pola numerik lokal yang berulang, seperti

tren mingguan atau pola lonjakan harga. Sementara itu, BiGRU memproses data hasil ekstraksi CNN dari dua arah: maju (*forward*) untuk menangkap pengaruh masa lalu, dan mundur (*backward*) untuk mempertimbangkan kemungkinan keterkaitan dengan data masa depan. Hasilnya adalah representasi yang kaya akan informasi spasial dan temporal, layaknya "*peta dua dimensi yang dilengkapi garis waktu maju dan mundur*".

BiGRU untuk Hubungan Temporal Dua Arah

BiGRU adalah pengembangan dari GRU yang memproses data secara dua arah. Dengan memanfaatkan informasi masa lalu dan masa depan, BiGRU memberikan pemahaman yang lebih utuh terhadap urutan data. Dalam konteks prediksi harga saham, ini berarti model dapat melihat bagaimana harga sebelumnya memengaruhi harga saat ini, sekaligus mempertimbangkan tren yang akan datang berdasarkan pola historis. Analogi yang dapat digunakan adalah "*membaca buku dari awal hingga akhir, lalu membacanya kembali dari akhir ke awal untuk memahami detail yang terlewat*". BiGRU memastikan bahwa konteks penuh data digunakan secara maksimal.

Mekanisme Attention

Mekanisme attention adalah lapisan yang memberikan bobot berbeda pada setiap langkah input, sehingga model dapat fokus pada bagian informasi yang paling relevan untuk

membuat prediksi. Dalam arsitektur ini, mekanisme attention ditempatkan setelah BiGRU untuk memperkuat kemampuan model dalam memilih fitur penting dari representasi temporal yang telah dihasilkan. Komponen utama mekanisme attention adalah *query*, *key*, dan *value*:

a) Query

Representasi dari elemen yang ingin kita fokuskan.

b) Key

Representasi dari setiap elemen dalam input.

c) Value

Informasi aktual yang terkait dengan setiap *key*.

Bobot perhatian dihitung dengan mengukur kesesuaian antara *query* dan *key*, kemudian diaplikasikan pada *value*. Proses ini memungkinkan model menyoroti informasi yang relevan dan mengabaikan yang kurang penting, seperti "*sorotan lampu panggung yang hanya menyorot aktor utama di tengah keramaian*".

Keunggulan Integrasi

Integrasi CNN, BiGRU, dan mekanisme attention memberikan keunggulan yang signifikan:

1. Ekstraksi Pola Spasial oleh CNN – CNN menangkap pola lokal dan fitur yang muncul dalam jangka waktu singkat.
2. Pemodelan Temporal oleh BiGRU – BiGRU mempertahankan dan memproses informasi jangka panjang dari dua arah.

3. Fokus pada Informasi Relevan oleh Attention – Mekanisme attention meningkatkan akurasi dengan memprioritaskan informasi yang paling berpengaruh.

Kombinasi ini menghasilkan model yang tidak hanya akurat, tetapi juga lebih mudah diinterpretasikan karena bobot attention dapat digunakan untuk memahami bagian mana dari data yang memengaruhi prediksi, layaknya "*peta panas yang menunjukkan area paling aktif pada sebuah pertandingan olahraga*".

Penerapan pada Prediksi Saham

Dalam penerapan pada prediksi harga saham, data *time series* harga saham harian pertama-tama diproses oleh CNN 1D untuk mengekstraksi fitur-fitur lokal. Output CNN kemudian diberikan ke BiGRU untuk memodelkan hubungan temporal dua arah. Selanjutnya, lapisan attention ditempatkan di atas BiGRU untuk memilih fitur temporal yang paling relevan terhadap prediksi harga di masa depan.

Sebagai contoh, model dapat belajar bahwa kenaikan volume perdagangan pada minggu tertentu memiliki pengaruh besar terhadap lonjakan harga pada minggu berikutnya. Mekanisme attention akan memberikan bobot tinggi pada periode waktu tersebut sehingga prediksi menjadi lebih akurat, seperti "*guru yang memberi tanda bintang pada jawaban esai yang paling berbobot*".

Contoh Implementasi Python CNN BiGRU dengan Attention

```
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Conv1D, MaxPooling1D, Bidirectional, GRU, Dense, Attention

inputs = Input(shape=(timesteps, features))
x = Conv1D(filters=64, kernel_size=3, activation='relu')(inputs)
x = MaxPooling1D(pool_size=2)(x)
x = Bidirectional(GRU(64, return_sequences=True))(x)
attn = Attention()([x, x])
output = Dense(1)(attn)

model = Model(inputs=inputs, outputs=output)
model.compile(optimizer='adam', loss='mse')
```

Tabel 7. Ringkasan Peran CNN BiGRU Attention

Komponen	Fungsi Utama	Analogi
CNN	Menangkap pola lokal dari data	"Fotografer yang memotret detail spesifik dari suatu kejadian"
BiGRU	Memodelkan hubungan temporal dua arah	"Membaca buku maju-mundur untuk pemahaman utuh"
Attention	Memfokuskan pada bagian data yang paling relevan	"Menyorot teks penting di buku sebelum ujian"

Dengan memahami kombinasi CNN–BiGRU–Attention, mahasiswa S1 dapat menguasai salah satu arsitektur modern yang efektif untuk memprediksi harga saham. Integrasi ini menggabungkan kekuatan ekstraksi fitur, pemahaman konteks temporal, dan fokus selektif pada informasi penting, sehingga menghasilkan prediksi yang lebih akurat dan dapat diinterpretasikan secara praktis.

3. Soal Esai

1. Jelaskan peran CNN dalam model hybrid CNN–BiGRU–Attention.
2. Apa fungsi utama BiGRU pada data time series?
3. Mengapa Attention diperlukan pada model prediksi saham?
4. Sebutkan tiga komponen utama Attention.
5. Bagaimana Attention membantu interpretabilitas model?
6. Jelaskan perbedaan Attention dan Self-Attention.
7. Apa dampak penggunaan Attention terhadap performa model?
8. Bagaimana CNN, BiGRU, dan Attention saling melengkapi?
9. Sebutkan salah satu tantangan dalam melatih model CNN–BiGRU–Attention.
10. Berikan contoh kasus di luar saham yang dapat menggunakan model ini.

4. Tugas Individu

Bangun model CNN–BiGRU dengan dan tanpa mekanisme Attention menggunakan dataset harga saham 1 tahun terakhir. Bandingkan akurasi keduanya menggunakan MAE dan RMSE.

```

# === CNN-BiGRU dgn & tanpa Attention | Data 1 Tahun (Yahoo Finance) ===
# Satu sel Colab. Disarankan Runtime -> Change runtime type -> GPU.

# 0) Setup & install
!pip -q install yfinance

import math, time, warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt
from IPython.display import display
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error

import tensorflow as tf
from tensorflow.keras import layers, Model, Input
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Dropout, Bidirectional, GRU, Dense
from tensorflow.keras.callbacks import EarlyStopping

# 1) Reproducibility + Info
SEED = 42
np.random.seed(SEED)
tf.random.set_seed(SEED)
print("TensorFlow:", tf.__version__)
print("GPU available:", len(tf.config.list_physical_devices('GPU')) > 0)

# 2) Parameter (ubah sesuai kebutuhan)
TICKER = "BRIS.JK" # Bank Syariah Indonesia – ganti jika mau saham lain
PERIOD = "1y" # 1 tahun terakhir
PRICE_COL = "Close" # target kolom

```

```

WINDOW_SIZE = 30 # panjang jendela input (hari)
TEST_RATIO = 0.2 # proporsi test dari total sequence
EPOCHS = 100
BATCH = 32
PATIENCE = 10

# 3) Ambil data
df = yf.download(TICKER, period=PERIOD, interval="1d")
if df.empty:
    raise ValueError("Data dari yfinance kosong. Coba ulangi, periksa koneksi, atau ganti TICKER/PERIOD.")
df = df[[PRICE_COL]].copy()
df[PRICE_COL] = df[PRICE_COL].ffill().bfill()
print(f"Jumlah hari trading: {len(df)} | Rentang: {df.index.min().date()} → {df.index.max().date()}")
display(df.tail())

# 4) Scaling & windowing (prediksi t+1 dari WINDOW_SIZE hari terakhir)
scaler = MinMaxScaler((0,1))
scaled = scaler.fit_transform(df[[PRICE_COL]].values)

def make_sequences(series, win):
    X, y = [], []
    for i in range(win, len(series)):
        X.append(series[i-win:i])
        y.append(series[i])
    return np.array(X), np.array(y)

X_all, y_all = make_sequences(scaled, WINDOW_SIZE)
n_total = X_all.shape[0]
if n_total < 10:
    raise ValueError("Sequence terlalu sedikit. Kurangi WINDOW_SIZE atau perpanjang PERIOD.")
n_train = int((1 - TEST_RATIO) * n_total)

X_train, y_train = X_all[:n_train], y_all[:n_train]
X_test, y_test = X_all[n_train:], y_all[n_train:]

```

```

# Tambah dimensi fitur (1 fitur: Close)
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], 1))
X_test = X_test.reshape((X_test.shape[0], X_test.shape[1], 1))
print("Train size:", X_train.shape, "| Test size:", X_test.shape)

# 5) Lapisan Attention sederhana (Bahdanau-style, time-wise)
class SimpleAttention(layers.Layer):
    def __init__(self, units=64, **kwargs):
        super().__init__(**kwargs)
        self.W = layers.Dense(units, activation="tanh")
        self.V = layers.Dense(1) # skor per time step

    def call(self, inputs, mask=None):
        # inputs: [batch, time, features]
        score = self.W(inputs) # [b, t, u]
        score = self.V(score) # [b, t, 1]
        weights = tf.nn.softmax(score, axis=1) # [b, t, 1]
        context = tf.reduce_sum(weights * inputs, axis=1) # [b, features]
        return context # (opsional: juga bisa return weights utk analisis)

# 6) Arsitektur model
def build_cnn_bigru_no_attention(input_shape):
    inp = Input(shape=input_shape)
    x = Conv1D(32, kernel_size=3, padding="causal", activation="relu")(inp)
    x = MaxPooling1D(pool_size=2)(x)
    x = Dropout(0.2)(x)
    x = Bidirectional(GRU(64))(x) # tanpa return_sequences
    x = Dense(32, activation="relu")(x)
    out = Dense(1)(x)
    model = Model(inp, out, name="CNN_BiGRU_NoAttention")
    model.compile(optimizer="adam", loss="mse")
    return model

def build_cnn_bigru_with_attention(input_shape):
    inp = Input(shape=input_shape)
    x = Conv1D(32, kernel_size=3, padding="causal", activation="relu")(inp)
    x = MaxPooling1D(pool_size=2)(x)
    x = Dropout(0.2)(x)
    x = Bidirectional(GRU(64, return_sequences=True))(x) # perlu sequences utk attention
    x = SimpleAttention(units=64)(x) # konteks ter-atensi
    x = Dense(32, activation="relu")(x)
    out = Dense(1)(x)
    model = Model(inp, out, name="CNN_BiGRU_Attention")
    model.compile(optimizer="adam", loss="mse")
    return model

input_shape = (X_train.shape[1], X_train.shape[2])
model_noatt = build_cnn_bigru_no_attention(input_shape)
model_att = build_cnn_bigru_with_attention(input_shape)

# 7) Latih kedua model (pakai EarlyStopping terhadap val_loss)
cb = EarlyStopping(monitor="val_loss", patience=PATIENCE, restore_best_weights=True)

t0 = time.perf_counter()
hist_noatt = model_noatt.fit(
    X_train, y_train,
    validation_split=0.2,
    epochs=EPOCHS, batch_size=BATCH,
    callbacks=[cb], verbose=0
)
t_noatt = time.perf_counter() - t0
print(f"Waktu latih TANPA Attention : {t_noatt:.2f} detik | epochs: {len(hist_noatt.history['loss'])}")

```

```
t0 = time.perf_counter()
hist_att = model_att.fit(
    X_train, y_train,
    validation_split=0.2,
    epochs=EPOCHS, batch_size=BATCH,
    callbacks=[cb], verbose=0
)
t_att = time.perf_counter() - t0
print(f"Waktu latih DENGAN Attention: {t_att:.2f} detik | epochs: {len(hist_att.history['loss'])}")

# 8) Evaluasi: MAE & RMSE di test set (dibalik ke skala harga asli)
def inverse_transform(arr):
    return scaler.inverse_transform(arr.reshape(-1,1)).flatten()

def evaluate(model, X_te, y_te):
    preds_scaled = model.predict(X_te, verbose=0).flatten()
    y_scaled = y_te.flatten()
    preds = inverse_transform(preds_scaled)
    y_true = inverse_transform(y_scaled)
    mae = mean_absolute_error(y_true, preds)
    rmse = math.sqrt(mean_squared_error(y_true, preds))
    return {"MAE": mae, "RMSE": rmse, "y_true": y_true, "preds": preds}

res_noatt = evaluate(model_noatt, X_test, y_test)
res_att = evaluate(model_att, X_test, y_test)

summary = pd.DataFrame([
    {"Model": "CNN-BiGRU (No Attention)", "MAE": res_noatt["MAE"], "RMSE": res_noatt["RMSE"]},
    {"Model": "CNN-BiGRU (+ Attention)", "MAE": res_att["MAE"], "RMSE": res_att["RMSE"]},
])
display(summary)

# 9) (Optional) Plot ringkas
plt.figure()
plt.plot(res_noatt["y_true"], label="Actual")
plt.plot(res_noatt["preds"], label="No-Att Pred")
plt.title("Test: Actual vs Pred (No Attention)")
plt.xlabel("Time Index (Test)"); plt.ylabel("Price"); plt.legend(); plt.show()

plt.figure()
plt.plot(res_att["y_true"], label="Actual")
plt.plot(res_att["preds"], label="Att Pred")
plt.title("Test: Actual vs Pred (Attention)")
plt.xlabel("Time Index (Test)"); plt.ylabel("Price"); plt.legend(); plt.show()
```

TensorFlow: 2.19.0
GPU available: False
[*****100%*****] 1 of 1 completedJumlah hari trading: 236 | Rentang: 2024-08-12 → 2025-08-11

Price	Close	
Ticker	BRIS.JK	
Date		
2025-08-05	2780.0	
2025-08-06	2740.0	
2025-08-07	2740.0	
2025-08-08	2700.0	
2025-08-11	2730.0	

Train size: (164, 30, 1) | Test size: (42, 30, 1)
Waktu latih TANPA Attention: 19.01 detik | epochs: 100
Waktu latih DENGAN Attention: 6.03 detik | epochs: 13

5. Studi Kasus

Setiap kelompok mendesain arsitektur CNN–BiGRU–Attention, mendokumentasikan hyperparameter yang digunakan, dan mempresentasikan hasil serta analisis performa model.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

UJIAN TENGAH SEMESTER (UTS)

1. Learning Outcome (LO)

- a. Mengintegrasikan pemahaman konsep, metode, dan implementasi prediksi harga saham yang telah dipelajari pada pertemuan 1–7.
- b. Menerapkan analisis fundamental, teknikal, dan metode pemodelan AI untuk kasus nyata.

2. Materi yang Diujikan

Materi UTS mencakup:

1. **Pertemuan 1** – Pendahuluan & Peran Pasar Saham.
2. **Pertemuan 2** – Tantangan dalam Prediksi Harga Saham.
3. **Pertemuan 3** – Pendekatan Konvensional (Regresi, ARIMA, GARCH).
4. **Pertemuan 4** – Pendekatan Modern Berbasis AI (LSTM, GRU, CNN, Attention).
5. **Pertemuan 5** – Arsitektur CNN untuk Data Time Series.
6. **Pertemuan 6** – Arsitektur GRU & BiGRU.
7. **Pertemuan 7** – Integrasi CNN–BiGRU–Attention.

3. Bentuk Ujian

- a) **Bagian A – Soal Esai (5 soal)**
Menguji pemahaman konsep dan analisis teoretis.
- b) **Bagian B – Soal Studi Kasus (2 soal)**
Menguji kemampuan mengidentifikasi masalah, merancang solusi, dan melakukan analisis hasil.

c) **Bagian C – Praktikum**

Implementasi model sederhana sesuai spesifikasi soal (misalnya ARIMA atau CNN–BiGRU pada dataset yang diberikan).

4. Soal Esai

1. Jelaskan perbedaan mendasar antara ARIMA dan LSTM.
2. Mengapa BiGRU dapat meningkatkan akurasi prediksi harga saham dibandingkan GRU biasa?
3. Sebutkan dan jelaskan tiga fungsi utama pasar saham.
4. Bagaimana Attention membantu meningkatkan interpretabilitas model prediksi?
5. Apa perbedaan fungsi pooling layer dan convolution layer pada CNN?

5. Soal Studi Kasus

Kasus 1

Diberikan data harga saham PT XYZ selama 1 tahun. Analisis faktor fundamental dan teknikal yang memengaruhi harga saham tersebut, lalu berikan rekomendasi strategi investasi.

Kasus 2

Gunakan potongan data time series yang diberikan untuk membuat prediksi 7 hari ke depan menggunakan metode konvensional dan metode AI. Bandingkan hasilnya.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Studi Kasus	30%
Praktikum	40%

PENGOLAHAN DATA PASAR SAHAM UNTUK MODEL DEEP LEARNING

1. Learning Outcome (LO)

- a) Mengidentifikasi sumber dan jenis data pasar saham.
- b) Melakukan pembersihan (*data cleaning*) dan penanganan *missing values*.
- c) Menerapkan normalisasi dan transformasi data untuk keperluan pelatihan model deep learning.

2. Materi

Sumber Data

Pemilihan sumber data yang tepat merupakan langkah fundamental dalam membangun model prediksi harga saham yang andal. Beberapa sumber data yang umum digunakan antara lain *Yahoo Finance*, *Bursa Efek Indonesia* (BEI), dan berbagai *Application Programming Interface* (API) penyedia data keuangan. Sumber-sumber ini dapat diibaratkan seperti "*perpustakaan dengan koleksi buku yang berbeda-beda, di mana peneliti harus memilih perpustakaan mana yang koleksinya paling sesuai dengan topik penelitiannya*".

Yahoo Finance menawarkan kemudahan akses dengan antarmuka yang ramah pengguna dan API gratis yang dapat diintegrasikan ke dalam bahasa pemrograman seperti Python. Data yang disediakan meliputi harga historis, laporan keuangan, dan indikator pasar. *Bursa Efek Indonesia* sebagai sumber resmi menyediakan data transaksi dan informasi emiten yang lebih detail dan akurat untuk pasar domestik.

Sementara itu, API data keuangan seperti Alpha Vantage, IEX Cloud, atau Quandl menawarkan fleksibilitas dan pembaruan data yang cepat, seringkali dalam format yang mudah diolah secara langsung.

Memahami kelebihan dan keterbatasan setiap sumber membantu peneliti memilih kombinasi yang optimal. Sebagai contoh, *Yahoo Finance* dapat digunakan untuk mendapatkan data harga historis global, sementara BEI digunakan untuk mendapatkan laporan emiten lokal yang mendalam.

Jenis Data

Data yang digunakan dalam prediksi harga saham dapat dikelompokkan menjadi tiga kategori utama:

1. OHLCV

Merupakan singkatan dari *Open, High, Low, Close, Volume*. Data ini merepresentasikan harga pembukaan, harga tertinggi, harga terendah, harga penutupan, dan volume perdagangan dalam periode tertentu. OHLCV adalah fondasi analisis teknikal, ibarat "*peta jalan yang menunjukkan jalur perjalanan harga setiap hari*".

2. Data Fundamental

Informasi terkait kinerja keuangan perusahaan seperti laba bersih, pendapatan, rasio keuangan, dan kebijakan dividen. Data ini penting untuk analisis jangka panjang, layaknya "*laporan kesehatan lengkap perusahaan yang membantu investor menilai kesehatannya*".

3. Data Sentimen

Informasi yang berasal dari berita, media sosial, atau laporan analis, yang menggambarkan persepsi pasar terhadap suatu emiten atau sektor. Ini dapat dianalogikan sebagai "*suasana hati pasar yang dapat memengaruhi keputusan kolektif investor*".

Tabel 8. Contoh Data OHLCV Harian

Tanggal	Open	High	Low	Close	Volume
2025-07-01	5000	5100	4950	5080	2,300,000
2025-07-02	5080	5150	5000	5120	2,100,000

Data Cleaning

Sebelum data digunakan untuk pelatihan model, diperlukan proses *data cleaning* untuk memastikan kualitas dan konsistensi. Dua tahap utama dalam proses ini adalah:

a. Menghapus atau Mengisi Nilai Hilang

Nilai yang hilang dapat menyebabkan bias atau kesalahan dalam prediksi. Teknik imputasi seperti rata-rata (*mean imputation*), median, atau interpolasi linear dapat digunakan untuk mengisi nilai tersebut.

b. Mengoreksi Data Anomali

Anomali dapat terjadi karena kesalahan input, gangguan sistem, atau peristiwa pasar yang tidak wajar. Identifikasi anomali dapat dilakukan dengan metode statistik, seperti deteksi nilai *z-score* yang ekstrem.

Sebagai analogi, *data cleaning* seperti "*membersihkan lensa kamera sebelum mengambil foto agar hasilnya jelas*".

Tanpa pembersihan, model dapat 'melihat' data yang kabur dan menghasilkan prediksi yang kurang akurat.

Normalisasi

Normalisasi adalah proses mengubah skala data agar berada dalam rentang tertentu atau memiliki distribusi yang seragam.

Dua metode umum adalah:

Min-Max Scaling (0–1) – Mengubah data sehingga nilai minimum menjadi 0 dan nilai maksimum menjadi 1. Rumusnya:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad \# \text{ Rumus Min-Max Scaling untuk mengubah nilai ke rentang 0 hingga 1}$$

Standard Scaling – Mengubah data sehingga memiliki rata-rata 0 dan standar deviasi 1:

$$x' = \frac{x - \mu}{\sigma} \quad \# \text{ Rumus Standard Scaling untuk mengubah data agar memiliki rata-rata 0 dan standar deviasi 1}$$

Normalisasi memastikan setiap fitur memiliki kontribusi yang seimbang pada model, ibarat "*memberikan panggung yang sama rata untuk semua pemain dalam pertunjukan*".

Transformasi Data

Transformasi data diperlukan untuk menyesuaikan format data dengan kebutuhan model pembelajaran mesin. Salah satu teknik umum adalah *sliding window*, yang digunakan untuk membuat data sekuensial dari data *time series*. Teknik ini memotong data menjadi jendela (*window*) yang berisi sejumlah langkah waktu tertentu, kemudian menggeser jendela tersebut satu langkah untuk membentuk set data berikutnya. Sebagai ilustrasi, jika kita memiliki deret harga [10,

11, 12, 13, 14] dan menggunakan *window size* = 3, maka akan terbentuk:

Input 1: [10, 11, 12], Target: 13

Input 2: [11, 12, 13], Target: 14

Selain itu, penyesuaian dimensi fitur dilakukan agar data sesuai dengan format input model. Misalnya, CNN memerlukan data dalam bentuk tiga dimensi (sampel, langkah waktu, fitur), sementara model feedforward hanya memerlukan dua dimensi.

Contoh Implementasi Python untuk Data Preprocessing

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

# Mengambil data dari Yahoo Finance
import yfinance as yf
data = yf.download('BBCA.JK', start='2024-01-01', end='2025-01-01')

# Data Cleaning
data = data.dropna() # Menghapus nilai hilang

# Normalisasi
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(data[['Open', 'High', 'Low', 'Close', 'Volume']])

# Sliding Window
window_size = 3
X, y = [], []
for i in range(len(scaled_data) - window_size):
    X.append(scaled_data[i:i+window_size])
    y.append(scaled_data[i+window_size, 3]) # Close sebagai target
```

Pengelolaan sumber data, jenis data, *data cleaning*, normalisasi, dan transformasi adalah fondasi utama dalam membangun model prediksi harga saham yang efektif. Proses ini memastikan bahwa model menerima input yang berkualitas, terstruktur, dan siap diproses, sehingga meningkatkan akurasi dan keandalan prediksi.

3. Soal Esai

1. Jelaskan perbedaan data OHLCV dan data fundamental.

2. Apa tujuan utama *data cleaning* pada data harga saham?
3. Berikan 2 metode penanganan *missing values*.
4. Jelaskan perbedaan *Min-Max Scaling* dan *Standard Scaling*.
5. Mengapa normalisasi penting dalam deep learning?
6. Apa fungsi *sliding window* dalam data time series?
7. Bagaimana cara mengidentifikasi *outlier* pada data harga saham?
8. Sebutkan 2 sumber data pasar saham yang populer digunakan peneliti.
9. Mengapa dimensi fitur harus disesuaikan sebelum pelatihan model?
10. Apa risiko jika normalisasi tidak dilakukan pada data input?

4. Tugas Individu

Unduh data harga saham 1 tahun terakhir dari Yahoo Finance. Lakukan *data cleaning* (penanganan *missing values*), normalisasi dengan *Min-Max Scaling*, dan buat *sliding window* dengan panjang 30 hari.

```

# === Pipeline: Download 1y ▶ Cleaning ▶ Min-Max Scaling ▶ Sliding Window (30 hari) ===
# Jalankan sekali di Colab.

!pip -q install yfinance

import numpy as np
import pandas as pd
import yfinance as yf
from sklearn.preprocessing import MinMaxScaler

# ---- Parameter (boleh diubah) ----
TICKER      = "BRIS.JK"  # contoh: "AAPL", "BBCA.JK"
PERIOD      = "1y"
INTERVAL    = "1d"
WINDOW_SIZE = 30         # panjang jendela (hari) untuk fitur X

# ---- 1) Download data dari Yahoo Finance ----
df = yf.download(TICKER, period=PERIOD, interval=INTERVAL, auto_adjust=False)
if df.empty:
    raise ValueError("Data dari yfinance kosong. Coba ulangi atau ganti TICKER/PERIOD.")

# Pastikanurut waktu dan pilih kolom target (Adj Close jika ada, else Close)
df = df.sort_index()
target_col = "Adj Close" if "Adj Close" in df.columns else "Close"
df = df[[target_col]].copy()

```

```

# ---- 2) Data cleaning: tangani missing & duplikat ----
# Hitung missing pakai numpy agar hasilnya int murni (tanpa FutureWarning)
missing_before = int(np.count_nonzero(pd.isna(df[target_col].to_numpy())))
# Isi celah: forward-fill lalu backward-fill
df[target_col] = df[target_col].ffill().bfill()
missing_after = int(np.count_nonzero(pd.isna(df[target_col].to_numpy())))

# Jika masih ada NaN (jarang), buang barisnya
if missing_after > 0:
    df = df.dropna(subset=[target_col])

# Singkirkan duplikasi index tanggal jika ada
df = df[~df.index.duplicated(keep="last")]

print(f"[INFO] {TICKER} | Rentang: {df.index.min().date()} → {df.index.max().date()}")
print(f"[INFO] Missing sebelum: {missing_before} | sesudah cleaning: {int(np.count_nonzero(pd.isna(df[target_col].to_numpy())))}")
display(df.tail())

# ---- 3) Normalisasi Min-Max ke [0, 1] ----
scaler = MinMaxScaler(feature_range=(0, 1))
scaled = scaler.fit_transform(df[[target_col]].to_numpy()) # shape: (N, 1)

```

```
# ---- 4) Sliding window (prediksi hari t+1 dari 30 hari terakhir) ----
def make_sliding_window(series_2d, window=30):
    s = series_2d.reshape(-1) # jadi vektor 1D
    X, y = [], []
    for i in range(window, len(s)):
        X.append(s[i-window:i]) # 30 hari terakhir
        y.append(s[i]) # target: hari berikutnya
    X = np.array(X, dtype=np.float32).reshape(-1, window, 1) # (samples, 30, 1 fitur)
    y = np.array(y, dtype=np.float32) # (samples,)
    return X, y

if len(df) < WINDOW_SIZE + 1:
    raise ValueError("Data terlalu pendek untuk window 30 hari. Perpanjang PERIOD atau kecilkan WINDOW_SIZE.")

X, y = make_sliding_window(scaled, window=WINDOW_SIZE)

print(f"[SHAPE] X: {X.shape} | y: {y.shape} (window={WINDOW_SIZE})")
print(f"[SAMPLE] 3 nilai y (skala 0-1):", np.round(y[:3], 4))

# ---- (Optional) Simpan untuk tahap modeling ----
out_name = f"windows_{TICKER.replace('.', '_')}_{PERIOD}_w{WINDOW_SIZE}.npz"
np.savez_compressed(out_name, X=X, y=y, window_size=WINDOW_SIZE, target_col=target_col, ticker=TICKER)
print(f"[SAVE] Disimpan ke:", out_name)

# Catatan:
# - Untuk modeling serius, biasakan fit scaler hanya di TRAIN (hindari data leakage).
# - Di sini scaler dipasang ke seluruh data supaya fokusnya ke pipeline yang diminta.
```

Output Program

[*****100%*****] 1 of 1 completed[INFO] BRIS.JK | Rentang: 2024-08-12 → 2025-08-11
[INFO] Missing sebelum: 0 | sesudah cleaning: 0

Price	Adj Close	
Ticker	BRIS.JK	

Date	
2025-08-05	2780.0
2025-08-06	2740.0
2025-08-07	2740.0
2025-08-08	2700.0
2025-08-11	2730.0

[SHAPE] X: (206, 30, 1) | y: (206,) (window=30)
[SAMPLE] 3 nilai y (skala 0-1): [0.9537 0.9167 0.9444]
[SAVE] Disimpan ke: windows_BRIS_JK_1y_w30.npz

5. Studi Kasus

Setiap kelompok mengambil 2 saham dari sektor berbeda, melakukan preprocessing data sesuai standar deep learning (cleaning, normalisasi, *sliding window*), dan membandingkan distribusi nilai fiturnya. Presentasikan hasilnya.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

REKAYASA FITUR (*FEATURE ENGINEERING*) UNTUK PREDIKSI SAHAM

1. Learning Outcome (LO)

- a) Menjelaskan konsep rekayasa fitur dalam analisis data pasar saham.
- b) Menghitung indikator teknikal yang relevan untuk prediksi harga saham.
- c) Membuat variabel lag untuk menangkap pola temporal.
- d) Mengevaluasi pengaruh fitur terhadap kinerja model.

2. Materi

Definisi Feature Engineering

Feature engineering adalah proses menambah, memodifikasi, atau mengubah fitur pada data agar model pembelajaran mesin dapat bekerja lebih efektif. Fitur di sini merujuk pada variabel input yang digunakan untuk memprediksi target. Proses ini menjadi krusial karena kualitas dan relevansi fitur dapat memengaruhi secara signifikan kinerja model.

Secara sederhana, *feature engineering* dapat dianalogikan seperti "*menambahkan bumbu yang tepat pada masakan agar cita rasanya meningkat*" atau "*menambahkan rempah rahasia pada resep tradisional agar rasa lebih kaya*". Dalam konteks prediksi harga saham, penambahan indikator teknikal, variabel lag, atau transformasi lain dapat membuat model lebih sensitif terhadap pola yang relevan.

Indikator Teknikal Umum

Indikator teknikal adalah perhitungan matematis berdasarkan data harga dan volume perdagangan yang digunakan untuk membantu mengidentifikasi tren, momentum, dan potensi pembalikan arah harga. Beberapa indikator teknikal yang umum digunakan meliputi:

1. Moving Average (MA)

Rata-rata harga selama periode waktu tertentu yang digunakan untuk menghaluskan fluktuasi harga. Rumus MA sederhana adalah:

$$MA_t = \frac{\sum_{i=0}^{n-1} P_{t-i}}{n}$$

di mana P adalah harga penutupan dan n adalah periode.

Analogi: "*melihat suhu rata-rata seminggu terakhir untuk memperkirakan cuaca umum*".

2. Exponential Moving Average (EMA)

Serupa dengan MA tetapi memberikan bobot lebih besar pada harga terbaru, sehingga lebih responsif terhadap perubahan. Rumusnya:

$$EMA_t = (P_t - EMA_{t-1}) \times k + EMA_{t-1}$$

dengan $k = \frac{2}{n+1}$.

Analogi: "*memprioritaskan kabar terbaru yang lebih relevan daripada berita lama*".

3. Relative Strength Index (RSI)

Mengukur kekuatan pergerakan harga untuk mengidentifikasi kondisi *overbought* atau *oversold*. Nilainya berada pada rentang 0–100. Analogi:

"menilai seberapa lelah seorang atlet dalam sebuah pertandingan".

4. Moving Average Convergence Divergence (MACD)

Mengukur hubungan antara dua EMA (biasanya EMA-12 dan EMA-26) dan digunakan untuk mengidentifikasi momentum. Analogi: "*membandingkan kecepatan dua pelari di lintasan*".

5. **Bollinger Bands** – Menggunakan MA sebagai garis tengah dan menambahkan serta mengurangi kelipatan deviasi standar untuk membentuk pita atas dan bawah. Digunakan untuk mengukur volatilitas. Analogi: "*mengamati seberapa lebar gelombang di laut*".

Variabel Lag

Variabel lag adalah penambahan data masa lalu sebagai prediktor. Misalnya, harga penutupan hari sebelumnya (lag-1) atau dua hari sebelumnya (lag-2) dapat digunakan sebagai input model untuk memprediksi harga saat ini. Contoh: Jika data harga adalah [100, 102, 101], maka dengan lag-1, kita akan memiliki pasangan (harga kemarin $\rightarrow$ harga hari ini) seperti (100 $\rightarrow$ 102) dan (102 $\rightarrow$ 101).

Analogi: "*mengingat hasil pertandingan sebelumnya untuk memperkirakan performa pertandingan berikutnya*" atau "*menggunakan catatan nilai ujian sebelumnya untuk memperkirakan hasil ujian berikutnya*".

Evaluasi Fitur

Evaluasi fitur dilakukan untuk menentukan seberapa besar kontribusi suatu fitur terhadap kinerja model. Dua metode yang umum digunakan adalah:

1. Feature Importance

Mengukur tingkat pengaruh fitur berdasarkan algoritma pembelajaran mesin seperti *Random Forest* atau *Gradient Boosting*. Semakin tinggi nilainya, semakin besar peran fitur tersebut.

2. Permutation Importance

Mengukur dampak perubahan acak pada suatu fitur terhadap akurasi model. Jika merombak nilai fitur menyebabkan penurunan signifikan pada akurasi, berarti fitur tersebut penting.

Evaluasi ini ibarat "*mengetes komponen mobil satu per satu untuk melihat bagian mana yang paling memengaruhi kecepatan*" atau "*mematikan lampu di ruangan satu per satu untuk melihat mana yang paling penting dalam penerangan*".

Dampak Fitur Tambahan pada Akurasi Model

Penambahan fitur yang relevan dapat meningkatkan akurasi model, terutama jika fitur tersebut menangkap pola yang sebelumnya tidak terlihat. Namun, terlalu banyak fitur yang tidak relevan justru dapat menurunkan performa karena menyebabkan *overfitting*.

Contoh implementasi dalam Python

```
import pandas as pd
import talib

# Contoh data harga penutupan
close = pd.Series([100, 102, 101, 103, 105, 104, 106])

# Moving Average
ma = talib.SMA(close, timeperiod=3)

# RSI
rsi = talib.RSI(close, timeperiod=14)

# MACD
macd, macdsignal, macdhist = talib.MACD(close, fastperiod=12, slowperiod=26, signalperiod=9)

# Bollinger Bands
upperband, middleband, lowerband = talib.BBANDS(close, timeperiod=20)
```

Tabel 9. Ringkasan Indikator Teknikal dan Fungsi

Indikator	Fungsi Utama	Analogi
MA	Menghaluskan harga dan menunjukkan tren	"Rata-rata suhu seminggu terakhir untuk gambaran umum"

EMA	Lebih responsif terhadap harga terbaru	"Menitikberatkan pada kabar terbaru dalam berita"
RSI	Mengidentifikasi kondisi jenuh beli/jual	"Mengukur seberapa lelah pemain dalam pertandingan"
MACD	Mengukur momentum dengan selisih dua EMA	"Membandingkan dua pelari di lintasan"
Bollinger Band	Mengukur volatilitas harga	"Mengamati seberapa lebar gelombang di laut"

Dengan memahami *feature engineering*, mahasiswa S1 diharapkan mampu merancang fitur yang relevan untuk meningkatkan akurasi model prediksi harga saham. Pemahaman yang baik tentang indikator teknikal, variabel lag, dan evaluasi fitur akan membantu dalam membangun model yang tidak hanya akurat tetapi juga dapat diinterpretasikan.

3. Soal Esai

1. Apa yang dimaksud dengan *feature engineering*?
2. Jelaskan fungsi Moving Average dalam analisis teknikal.
3. Apa perbedaan MA dan EMA?
4. Bagaimana RSI membantu mengidentifikasi kondisi *overbought* atau *oversold*?
5. Jelaskan fungsi MACD dalam analisis teknikal.
6. Apa manfaat penggunaan Bollinger Bands?

7. Mengapa variabel lag penting dalam prediksi saham?
8. Apa itu *feature importance* dan bagaimana cara mengukurnya?
9. Bagaimana penambahan fitur bisa meningkatkan akurasi model?
10. Berikan contoh indikator teknikal lain selain yang disebutkan di atas.

4. Tugas Individu

Pilih satu saham, hitung indikator MA(20), EMA(20), RSI(14), dan MACD untuk data harga 6 bulan terakhir. Sertakan grafik dan interpretasi singkat.

```
# === Indikator Teknis 6 Bulan: MA(20), EMA(20), RSI(14), MACD + Grafik & Interpretasi ===
# Jalankan sel ini di Google Colab.

!pip -q install yfinance

import numpy as np
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt

# ----- Parameter -----
TICKER = "BRIS.JK" # ganti: "AAPL", "BBCA.JK", dll.
PERIOD = "6mo"
INTERVAL = "1d"

# ----- Download & Cleaning -----
df = yf.download(TICKER, period=PERIOD, interval=INTERVAL, auto_adjust=False)
if df.empty:
    raise ValueError("Data tidak tersedia. Coba ganti TICKER atau periksa koneksi.")

price_col = "Adj Close" if "Adj Close" in df.columns else "Close"
prices = df[[price_col]].copy().rename(columns={price_col: "Close"}).sort_index()
prices["Close"] = prices["Close"].ffill().bfill()
```

```

# ----- Indikator -----
# MA(20) & EMA(20)
prices["MA20"] = prices["Close"].rolling(window=20, min_periods=20).mean()
prices["EMA20"] = prices["Close"].ewm(span=20, adjust=False).mean()

# RSI(14) (Wilder-ish dgn EWM) – tambah eps agar aman dari /0
delta = prices["Close"].diff()
gain = delta.clip(lower=0)
loss = -delta.clip(upper=0)
avg_gain = gain.ewm(alpha=1/14, adjust=False).mean()
avg_loss = loss.ewm(alpha=1/14, adjust=False).mean()
eps = 1e-10
rs = avg_gain / (avg_loss + eps)
prices["RSI14"] = 100 - (100 / (1 + rs))

# MACD (12, 26, 9)
ema12 = prices["Close"].ewm(span=12, adjust=False).mean()
ema26 = prices["Close"].ewm(span=26, adjust=False).mean()
prices["MACD"] = ema12 - ema26
prices["Signal"] = prices["MACD"].ewm(span=9, adjust=False).mean()
prices["Hist"] = prices["MACD"] - prices["Signal"]

# buang baris awal yang NaN karena rolling/EMA
ind = prices.dropna().copy()
display(ind.tail())

# ----- Grafik -----
fig, axes = plt.subplots(3, 1, figsize=(12, 10), sharex=True)

# Harga + MA/EMA
axes[0].plot(ind.index, ind["Close"], label="Close")
axes[0].plot(ind.index, ind["MA20"], label="MA20 (20 hari)")
axes[0].plot(ind.index, ind["EMA20"], label="EMA20 (20 hari)")
axes[0].set_title(f"{TICKER} – Harga Penutupan dengan MA20 & EMA20 (6 bulan)")
axes[0].set_ylabel("Harga")
axes[0].legend()
axes[0].grid(True, linestyle="--", alpha=0.3)

# RSI
axes[1].plot(ind.index, ind["RSI14"], label="RSI(14)")
axes[1].axhline(70, linestyle="--", alpha=0.5)
axes[1].axhline(30, linestyle="--", alpha=0.5)
axes[1].set_title("RSI(14)")
axes[1].set_ylabel("RSI")
axes[1].grid(True, linestyle="--", alpha=0.3)

# MACD
axes[2].plot(ind.index, ind["MACD"], label="MACD")
axes[2].plot(ind.index, ind["Signal"], label="Signal")
axes[2].bar(ind.index, ind["Hist"], alpha=0.5, label="Histogram")
axes[2].set_title("MACD (EMA12-EMA26, Signal=9)")
axes[2].set_ylabel("Nilai")
axes[2].legend()
axes[2].grid(True, linestyle="--", alpha=0.3)

plt.tight_layout()
plt.show()

```

```

# ----- Interpretasi Singkat (otomatis, bukan saran investasi) -----
# Helper ambil skalar aman dari Series/DataFrame
def last_val(col):
    return float(np.asarray(col.to_numpy())[-1])

def prev_val(col):
    arr = np.asarray(col.to_numpy())
    return float(arr[-2]) if arr.size >= 2 else None

last_close = last_val(ind["Close"])
last_ma20 = last_val(ind["MA20"])
last_ema20 = last_val(ind["EMA20"])
last_rsi = last_val(ind["RSI14"])
last_macd = last_val(ind["MACD"])
last_sig = last_val(ind["Signal"])

prev_macd = prev_val(ind["MACD"])
prev_sig = prev_val(ind["Signal"])

trend_notes = []
trend_notes.append("di atas MA20 (bias naik jangka pendek)"
                    if last_close > last_ma20 else "di bawah MA20 (bias turun jangka pendek)")
trend_notes.append("di atas EMA20" if last_close > last_ema20 else "di bawah EMA20")

if last_rsi >= 70:
    rsi_note = "overbought (>70)"
elif last_rsi <= 30:
    rsi_note = "oversold (<30)"
else:
    rsi_note = "netral (30-70)"

macd_note = "MACD di atas signal (bias bullish)" if last_macd > last_sig else "MACD di bawah signal (bias bearish)"
cross_note = ""
if (prev_macd is not None) and (prev_sig is not None):
    if (prev_macd <= prev_sig) and (last_macd > last_sig):
        cross_note = " • Ada bullish crossover terbaru."
    elif (prev_macd >= prev_sig) and (last_macd < last_sig):
        cross_note = " • Ada bearish crossover terbaru."

print("\n=== Interpretasi (ringkas) ===")
print(f"Tanggal terakhir: {ind.index[-1].date()}")
print(f"Harga vs MA/EMA : {', '.join(trend_notes)}")
print(f"RSI(14) : {last_rsi:.2f} → {rsi_note}")
print(f"MACD/Signal : {last_macd:.4f} / {last_sig:.4f} → {macd_note}{cross_note}")

```

Output Program

[*****100%*****] 1 of 1 completed

Price	Close	MA20	EMA20	RSI14	MACD	Signal	Hist
Ticker	BRIS.JK						
Date							
2025-08-05	2780.0	2767.0	2747.671654	54.928521	31.140965	37.774391	-6.633426
2025-08-06	2740.0	2773.5	2746.941020	50.643429	26.734230	35.566359	-8.832129
2025-08-07	2740.0	2775.0	2746.279971	50.643429	22.976995	33.048486	-10.071491
2025-08-08	2700.0	2771.0	2741.872355	46.441595	16.580564	29.754902	-13.174337
2025-08-11	2730.0	2771.5	2740.741654	49.805312	13.773325	26.558586	-12.785261

5. Studi Kasus

Setiap kelompok menambahkan minimal 3 indikator teknikal dan 2 variabel lag pada dataset harga saham yang sama. Latih model sederhana (misalnya regresi linear atau LSTM) sebelum dan sesudah penambahan fitur, lalu bandingkan akurasinya.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

PELATIHAN MODEL CNN BIGRU ATTENTION UNTUK PREDIKSI SAHAM

1. Learning Outcome (LO)

1. Menyiapkan data latih, validasi, dan uji untuk model CNN–BiGRU–Attention.
2. Mengimplementasikan arsitektur CNN–BiGRU–Attention menggunakan Python.
3. Menentukan dan menyesuaikan *hyperparameter* model.

2. Materi

Persiapan Dataset

Persiapan dataset adalah langkah awal yang menentukan kualitas hasil model pembelajaran mesin, khususnya pada prediksi harga saham. Proses ini dimulai dengan membagi data menjadi tiga bagian: **train set** untuk melatih model, **validation set** untuk menyesuaikan parameter dan mencegah *overfitting*, serta **test set** untuk mengukur performa akhir. Pembagian ini dapat dianalogikan seperti "*melatih tim sepak bola di sesi latihan, menguji strategi dalam laga persahabatan, lalu mengukur performa di pertandingan resmi*" atau "*menyusun skripsi dengan tahapan riset awal, revisi tengah, dan ujian akhir*".

Penanganan data *imbalance* juga menjadi aspek penting. Jika distribusi target tidak seimbang, seperti pada kasus prediksi kenaikan dan penurunan harga saham, model cenderung bias ke kelas mayoritas. Teknik seperti *oversampling*, *undersampling*, atau penggunaan bobot kelas

dapat diterapkan untuk mengatasi masalah ini. Hal ini ibarat *"menyesuaikan jumlah pemain dalam dua tim agar pertandingan tetap adil"*.

Implementasi Arsitektur

Arsitektur yang digunakan memadukan tiga komponen utama:

1. CNN

Digunakan untuk ekstraksi fitur lokal dari data *time series*. CNN mampu mengenali pola jangka pendek seperti lonjakan harga atau penurunan tajam, layaknya *"lensa pembesar yang fokus pada detail kecil di sebuah peta"*.

2. BiGRU

Memodelkan hubungan temporal dua arah, sehingga dapat mempertimbangkan informasi dari masa lalu dan masa depan secara simultan, seperti *"membaca buku dari awal dan dari akhir untuk mendapatkan pemahaman penuh cerita"*.

3. Attention

Memberikan bobot lebih pada informasi yang paling relevan untuk prediksi, ibarat *"lampu sorot yang menyorot aktor utama di panggung"* atau *"stabilizer kamera yang menjaga fokus pada objek penting"*.

Integrasi ketiga komponen ini memungkinkan model menangkap pola kompleks yang melibatkan interaksi spasial dan temporal secara bersamaan.

Hyperparameter Tuning

Pengaturan *hyperparameter* sangat memengaruhi performa model. Beberapa parameter yang umum dioptimalkan antara lain:

a) **Learning rate**

Menentukan seberapa besar langkah pembaruan bobot pada setiap iterasi, seperti "*mengatur kecepatan mengayuh sepeda agar tidak terlalu lambat atau terlalu cepat*".

b) **Batch size**

Jumlah sampel yang diproses sebelum pembaruan bobot.

c) **Jumlah filter CNN**

Mempengaruhi banyaknya pola yang dapat ditangkap.

d) **Kernel size**

Menentukan ukuran jendela konvolusi.

e) **Jumlah neuron BiGRU**

Mengatur kapasitas memori model.

Penggunaan teknik seperti *grid search*, *random search*, atau *Bayesian optimization* membantu menemukan kombinasi optimal. **Early stopping** digunakan untuk menghentikan pelatihan ketika performa *validation set* tidak lagi meningkat, mencegah *overfitting*, seperti "*menghentikan lomba maraton saat mencapai garis finish agar tidak menguras tenaga sia-sia*".

Monitoring Pelatihan

Monitoring pelatihan dilakukan dengan memantau kurva *loss* dan metrik evaluasi seperti akurasi atau *mean squared error*. Kurva ini memberikan gambaran apakah model sedang belajar dengan baik atau mengalami *overfitting/underfitting*. Visualisasi kurva *loss* dapat dianalogikan seperti "*memantau grafik detak jantung atlet untuk memastikan kondisinya optimal selama latihan*" atau "*melihat grafik kemajuan proyek untuk mengetahui apakah berjalan sesuai rencana*". Dengan memahami setiap tahap persiapan dataset, implementasi arsitektur, tuning *hyperparameter*, dan monitoring pelatihan, mahasiswa dapat membangun model prediksi harga saham yang efektif, akurat, dan dapat diinterpretasikan.

3. Soal Esai

1. Apa tujuan pembagian dataset menjadi train, validation, dan test set?
2. Jelaskan fungsi CNN dalam arsitektur CNN–BiGRU–Attention.
3. Mengapa BiGRU digunakan setelah CNN?
4. Apa manfaat mekanisme Attention pada model hybrid ini?
5. Sebutkan tiga contoh *hyperparameter* yang dapat diatur dalam model ini.
6. Mengapa *learning rate* penting untuk pelatihan model?
7. Apa itu *early stopping* dan kapan digunakan?
8. Mengapa perlu memantau kurva loss selama pelatihan?

9. Bagaimana *batch size* mempengaruhi pelatihan model?
10. Jelaskan dampak *overfitting* pada hasil prediksi.

4. Tugas Individu

Implementasikan model CNN–BiGRU–Attention pada dataset harga saham selama 2 tahun terakhir. Gunakan 70% data untuk pelatihan, 15% validasi, dan 15% pengujian.

```
# === CNN-BiGRU-Attention untuk Prediksi Harga Saham (2 Tahun) ===
# Jalankan sel ini di Google Colab (disarankan aktifkan GPU: Runtime → Change runtime type → GPU)

# 0) Setup
!pip -q install yfinance

import math, time, warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error

import tensorflow as tf
from tensorflow.keras import layers, Model, Input
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Dropout, Bidirectional, GRU, Dense
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau

# 1) Reproducibility + Info
SEED = 42
np.random.seed(SEED)
tf.random.set_seed(SEED)
print("TensorFlow:", tf.__version__)
print("GPU available:", len(tf.config.list_physical_devices('GPU')) > 0)

# 2) Parameter (silakan ubah)
TICKER      = "BRIS.JK"  # contoh lain: "AAPL", "BBCA.JK"
PERIOD      = "2y"       # 2 tahun terakhir
INTERVAL    = "1d"
TARGET_COL  = "Adj Close" # jika tidak ada, otomatis pakai Close
WINDOW_SIZE = 60         # panjang jendela input (hari). Bisa 30/60 dsb
EPOCHS      = 100
BATCH       = 32
PATIENCE    = 10

# 3) Download data & cleaning
df = yf.download(TICKER, period=PERIOD, interval=INTERVAL, auto_adjust=False)
if df.empty:
    raise ValueError("Data dari yfinance kosong. Coba ulangi atau ganti TICKER/PERIOD.")
df = df.sort_index()
```

```

▶ target_col = TARGET_COL if TARGET_COL in df.columns else "close"
prices = df[[target_col]].copy().rename(columns={target_col: "close"})
# cleaning: isi celah dan drop sisa NaN
prices["close"] = prices["close"].ffill().bfill()
prices = prices.dropna()
# hapus duplikasi tanggal jika ada
prices = prices[~prices.index.duplicated(keep="last")]

print(f"[INFO] {TICKER} | Rentang: {prices.index.min().date()} → {prices.index.max().date()} | N={len(prices)}")

if len(prices) < WINDOW_SIZE + 10:
    raise ValueError("Data terlalu pendek untuk window & split. Kurangi WINDOW_SIZE atau perpanjang PERIOD.")

# 4) Split 70% / 15% / 15% (berdasar WAKTU), fit scaler HANYA di train
N = len(prices)
N_train = int(0.70 * N)
N_val = int(0.15 * N)
N_test = N - N_train - N_val

train_df = prices.iloc[:N_train]
val_df = prices.iloc[N_train:N_train+N_val]
test_df = prices.iloc[N_train+N_val:]

print(f"[SPLIT] Train={train_df.shape[0]} | Val={val_df.shape[0]} | Test={test_df.shape[0]}")

scaler = MinMaxScaler(feature_range=(0, 1))
scaler.fit(train_df[["close"]].to_numpy()) # fit di TRAIN saja (hindari leakage)

▶ # Transform seluruh seri dengan scaler dari train
prices_scaled = scaler.transform(prices[["close"]].to_numpy()).reshape(-1)

# 5) Buat sliding window di SELURUH seri, lalu petakan ke split berdasarkan indeks TARGET
def make_all_sequences(series_id, window):
    X, y, tgt_idx = [], [], []
    for i in range(window, len(series_id)):
        X.append(series_id[i-window:i])
        y.append(series_id[i])
        tgt_idx.append(i) # indeks target (t)
    X = np.array(X, dtype=np.float32).reshape(-1, window, 1)
    y = np.array(y, dtype=np.float32)
    tgt_idx = np.array(tgt_idx)
    return X, y, tgt_idx

X_all, y_all, idx_all = make_all_sequences(prices_scaled, WINDOW_SIZE)

# fungsi bantu untuk memotong ke set, berdasarkan posisi target i terhadap batas split waktu
def split_by_index(X_all, y_all, idx_all, n_train, n_val):
    # target i < n_train -> train; n_train <= i < n_train+n_val -> val; lainnya -> test
    train_mask = idx_all < n_train
    val_mask = (idx_all >= n_train) & (idx_all < n_train + n_val)
    test_mask = idx_all >= (n_train + n_val)
    return (X_all[train_mask], y_all[train_mask],
            X_all[val_mask], y_all[val_mask],
            X_all[test_mask], y_all[test_mask])

X_train, y_train, X_val, y_val, X_test, y_test = split_by_index(
    X_all, y_all, idx_all, N_train, N_val
)

```

```

print("[SEQ] X_train:", X_train.shape, "| X_val:", X_val.shape, "| X_test:", X_test.shape)

# 6) Definisikan Attention (sederhana, time-wise)
class SimpleAttention(layers.Layer):
    def __init__(self, units=64, **kwargs):
        super().__init__(**kwargs)
        self.W = layers.Dense(units, activation="tanh")
        self.V = layers.Dense(1)

    def call(self, inputs):
        # inputs: [batch, time, features]
        score = self.W(inputs) # [b, t, u]
        score = self.V(score) # [b, t, 1]
        weights = tf.nn.softmax(score, axis=1) # softmax over time
        context = tf.reduce_sum(weights * inputs, axis=1) # [b, features]
        return context

# 7) Bangun model CNN-BiGRU-Attention
def build_cnn_bigru_attention(input_shape):
    inp = Input(shape=input_shape)
    # CNN feature extractor
    x = Conv1D(32, kernel_size=3, padding="causal", activation="relu")(inp)
    x = MaxPooling1D(pool_size=2)(x)
    x = Dropout(0.2)(x)
    x = Conv1D(64, kernel_size=3, padding="causal", activation="relu")(x)
    x = Dropout(0.2)(x)
    # BiGRU (kembalikan sequence untuk attention)
    x = Bidirectional(GRU(64, return_sequences=True))(x)
    # Attention di sepanjang time steps keluaran BiGRU
    x = SimpleAttention(units=64)(x)

# Head regresi
x = Dense(64, activation="relu")(x)
out = Dense(1)(x)
model = Model(inp, out, name="CNN_BiGRU_Attention")
model.compile(optimizer="adam", loss="mse", metrics=["mae"])
return model

input_shape = (WINDOW_SIZE, 1)
model = build_cnn_bigru_attention(input_shape)
model.summary()

# 8) Training (dengan early stopping dan LR scheduler), validasi 15%
early = EarlyStopping(monitor="val_loss", patience=PATIENCE, restore_best_weights=True)
plateau = ReduceLROnPlateau(monitor="val_loss", patience=5, factor=0.5, verbose=1)

t0 = time.perf_counter()
hist = model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=EPOCHS, batch_size=BATCH,
    callbacks=[early, plateau],
    verbose=0
)
train_time = time.perf_counter() - t0
print(f"[TRAIN] waktu: {train_time:.2f} detik | epochs: {len(hist.history['loss'])}")

# 9) Evaluasi di test set (MAE & RMSE dalam SKALA HARGA)
def inverse_transform_1d(arr_scaled):
    arr_scaled = np.asarray(arr_scaled).reshape(-1,1)
    return scaler.inverse_transform(arr_scaled).reshape(-1)

```

```

▶ # prediksi test
pred_test_scaled = model.predict(X_test, verbose=0).reshape(-1)
y_test_scaled = y_test.reshape(-1)

pred_test = inverse_transform_1d(pred_test_scaled)
y_test_true = inverse_transform_1d(y_test_scaled)

mae = mean_absolute_error(y_test_true, pred_test)
rmse = math.sqrt(mean_squared_error(y_test_true, pred_test))
print(f"[TEST] MAE={mae:.4f} | RMSE={rmse:.4f}")

# 10) Plot: history loss & Actual vs Predicted (Test)
plt.figure(figsize=(6,4))
plt.plot(hist.history["loss"], label="train loss")
plt.plot(hist.history["val_loss"], label="val loss")
plt.title("Training History (MSE)")
plt.xlabel("Epoch"); plt.ylabel("Loss"); plt.legend(); plt.tight_layout(); plt.show()

plt.figure(figsize=(10,4))
plt.plot(y_test_true, label="Actual")
plt.plot(pred_test, label="Predicted")
plt.title("Test Set: Actual vs Predicted")
plt.xlabel("Time Index (Test)"); plt.ylabel("Price"); plt.legend(); plt.tight_layout(); plt.show()

```

Output Program

```

[*****100%*****] 1 of 1 completed
[INFO] BRIS.JK | Rentang: 2023-08-11 → 2025-08-11 | N=473
[SPILT] Train=331 | Val=70 | Test=72
[SEQ] X_train: (271, 60, 1) | X_val: (70, 60, 1) | X_test: (72, 60, 1)
Model: "CNN_BiGRU_Attention"

```

Layer (type)	Output Shape	Param #
input_layer (InputLayer)	(None, 60, 1)	0
conv1d (Conv1D)	(None, 60, 32)	128
max_pooling1d (MaxPooling1D)	(None, 30, 32)	0
dropout (Dropout)	(None, 30, 32)	0
conv1d_1 (Conv1D)	(None, 30, 64)	6,208
dropout_1 (Dropout)	(None, 30, 64)	0
bidirectional (Bidirectional)	(None, 30, 128)	49,920
simple_attention (SimpleAttention)	(None, 128)	8,321
dense_2 (Dense)	(None, 64)	8,256
dense_3 (Dense)	(None, 1)	65

```

Total params: 72,898 (284.76 KB)
Trainable params: 72,898 (284.76 KB)
Non-trainable params: 0 (0.00 B)

```

5. Studi Kasus

Setiap kelompok membuat dua versi model CNN–BiGRU–Attention dengan *hyperparameter* berbeda, lalu membandingkan performa keduanya menggunakan metrik MAE dan RMSE.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

PENANGANAN OVERFITTING & EVALUASI MODEL PREDIKSI SAHAM

1. Learning Outcome (LO)

- a) Mengidentifikasi tanda-tanda *overfitting* dan *underfitting* pada model.
- b) Menerapkan teknik regularisasi untuk mencegah *overfitting*.
- c) Menghitung metrik evaluasi seperti MAE, RMSE, dan MAPE.

2. Materi Overfitting

Overfitting adalah kondisi ketika model pembelajaran mesin terlalu menyesuaikan diri dengan data latih, sehingga kinerjanya pada data uji menurun secara signifikan. Model yang mengalami *overfitting* memiliki performa luar biasa saat memproses data latih, namun gagal memprediksi data baru karena cenderung menghafal detail dan noise alih-alih memahami pola yang mendasarinya.

Fenomena ini dapat dianalogikan seperti "*siswa yang menghafal jawaban latihan ujian tanpa memahami materi, sehingga kesulitan menjawab soal yang berbeda pada ujian sebenarnya*" atau "*pemain catur yang hanya hafal satu skenario permainan, tetapi bingung saat lawan mengubah strategi*". *Overfitting* sering terjadi pada model dengan kompleksitas tinggi dan jumlah parameter yang besar, terutama ketika data latih terbatas.

Underfitting

Sebaliknya, *underfitting* terjadi ketika model terlalu sederhana sehingga tidak mampu menangkap pola dalam data secara memadai. Akibatnya, model menunjukkan performa rendah baik pada data latih maupun data uji.

Analogi untuk underfitting adalah "*menggunakan teropong mainan untuk melihat pemandangan jauh — terlalu sederhana sehingga detail penting terlewat*" atau "*memasak dengan resep yang terlalu ringkas sehingga hasilnya hambar dan tidak sesuai harapan*". Penyebab umum underfitting meliputi arsitektur model yang terlalu dangkal, jumlah fitur yang tidak memadai, atau waktu pelatihan yang terlalu singkat.

Teknik Penanganan Overfitting

Beberapa teknik yang dapat digunakan untuk mengurangi risiko overfitting meliputi:

1. Dropout

Metode regulasi yang secara acak menonaktifkan sejumlah neuron selama pelatihan untuk mencegah model menjadi terlalu bergantung pada fitur tertentu. Misalnya, jika dropout rate 0,5, maka setengah neuron akan dinonaktifkan pada setiap iterasi pelatihan, ibarat "*mengacak formasi pemain dalam latihan agar semua anggota tim fleksibel di berbagai posisi*".

2. Early Stopping

Menghentikan pelatihan lebih awal ketika performa pada *validation set* mulai menurun, meskipun performa pada

train set terus meningkat. Teknik ini ibarat "*menghentikan lari maraton tepat sebelum kelelahan, sehingga tetap bugar untuk lomba berikutnya*".

3. **Regularisasi L1/L2**

Menambahkan penalti pada bobot model agar tetap kecil, sehingga mengurangi kompleksitas model. L1 mendorong bobot menjadi nol, menghasilkan model yang lebih jarang (*sparse*), sementara L2 mengurangi besar bobot secara proporsional.

4. **Menambah Data Latih (Data Augmentation)**

Memperluas jumlah dan variasi data latih untuk membantu model belajar pola yang lebih umum. Dalam prediksi harga saham, ini dapat dilakukan dengan menambah data historis atau membuat simulasi data yang realistis, seperti "*memberi atlet berbagai skenario latihan agar siap menghadapi kondisi pertandingan yang beragam*".

Metrik Evaluasi

Evaluasi performa model dilakukan menggunakan metrik yang sesuai. Beberapa metrik umum dalam regresi untuk prediksi harga saham antara lain:

1. **MAE (Mean Absolute Error)**

Mengukur rata-rata kesalahan absolut antara nilai prediksi dan nilai aktual. Rumusnya:

2. **RMSE (Root Mean Squared Error)**

Menghitung akar dari rata-rata kesalahan kuadrat, memberikan penalti lebih besar untuk kesalahan besar:

3. MAPE (Mean Absolute Percentage Error)

Mengukur kesalahan relatif dalam persen, berguna untuk memahami skala kesalahan terhadap nilai aktual:

Tabel 10. Ringkasan Metrik Evaluasi

Metrik	Fungsi Utama	Kelebihan	Kekurangan
MAE	Rata-rata kesalahan absolut	Mudah diinterpretasi	Tidak membedakan kesalahan kecil/besar
RMSE	Akar rata-rata kesalahan kuadrat	Memberi penalti lebih besar pada kesalahan besar	Kurang intuitif dibanding MAE
MAPE	Kesalahan relatif dalam persen	Memperhatikan skala data	Tidak cocok untuk nilai aktual mendekati nol

Contoh Implementasi Python untuk Metrik Evaluasi

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np

# Nilai aktual dan prediksi
actual = np.array([100, 102, 98, 105])
predicted = np.array([101, 101, 99, 104])

mae = mean_absolute_error(actual, predicted)
rmse = np.sqrt(mean_squared_error(actual, predicted))
mape = np.mean(np.abs((actual - predicted) / actual)) * 100

print(f"MAE: {mae}")
print(f"RMSE: {rmse}")
print(f"MAPE: {mape}%")
```

Dengan memahami perbedaan antara overfitting dan underfitting, teknik penanganannya, serta metrik evaluasi yang tepat, mahasiswa S1 dapat membangun model prediksi harga saham yang lebih andal dan dapat diinterpretasikan. Pengetahuan ini akan membantu mereka mengidentifikasi

masalah, menerapkan solusi yang sesuai, dan mengevaluasi performa model secara objektif.

3. Soal Esai

1. Apa definisi *overfitting*?
2. Apa definisi *underfitting*?
3. Sebutkan dua tanda model mengalami *overfitting*.
4. Bagaimana dropout mencegah *overfitting*?
5. Apa perbedaan regularisasi L1 dan L2?
6. Mengapa *early stopping* dapat membantu pelatihan model?
7. Jelaskan kelebihan MAE dibandingkan RMSE.
8. Dalam kondisi apa RMSE lebih cocok digunakan dibandingkan MAE?
9. Apa kelemahan MAPE dalam evaluasi model?
10. Berikan contoh nilai MAE dan interpretasinya dalam prediksi harga saham.

4. Tugas Individu

Gunakan model CNN–BiGRU–Attention yang sudah dilatih. Tambahkan teknik dropout 0.3 dan *early stopping* dengan *patience* = 5. Bandingkan hasil MAE dan RMSE sebelum dan sesudah perubahan.

```

▶ # === CNN-BiGRU-Attention: Perbandingan sebelum vs sesudah (dropout 0.3 + patience=5) ===
# Jalankan sel ini di Google Colab. Disarankan aktifkan GPU (Runtime → Change runtime type → GPU).
# 0) Setup
!pip -q install yfinance

import math, time, warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import yfinance as yf
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error

import tensorflow as tf
from tensorflow.keras import layers, Model, Input
from tensorflow.keras.layers import Conv1D, MaxPooling1D, Dropout, Bidirectional, GRU, Dense
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau

# 1) Reproducibility + info
SEED = 42
np.random.seed(SEED)
tf.random.set_seed(SEED)
print("TensorFlow:", tf.__version__)
print("GPU available:", len(tf.config.list_physical_devices('GPU')) > 0)

```

```

▶ # 1) Reproducibility + info
SEED = 42
np.random.seed(SEED)
tf.random.set_seed(SEED)
print("TensorFlow:", tf.__version__)
print("GPU available:", len(tf.config.list_physical_devices('GPU')) > 0)

# 2) Parameter (ubah sesuai kebutuhan)
TICKER      = "BRIS.JK" # contoh: "AAPL", "BBCA.JK"
PERIOD      = "2y"      # pakai 2 tahun supaya data cukup
INTERVAL    = "1d"
TARGET_COL  = "Adj Close" # fallback ke Close
WINDOW_SIZE = 60        # jendela input (mis. 30/60/90)
EPOCHS      = 100
BATCH       = 32

# Konfigurasi baseline vs modifikasi
BASELINE_DROPOUT = 0.2
BASELINE_PATIENCE = 10
NEW_DROPOUT = 0.3
NEW_PATIENCE = 5

```

```

▶ # 3) Download & cleaning
df = yf.download(TICKER, period=PERIOD, interval=INTERVAL, auto_adjust=False)
if df.empty:
    raise ValueError("Data yfinance kosong. Ganti TICKER/PERIOD atau coba lagi.")
df = df.sort_index()

target_col = TARGET_COL if TARGET_COL in df.columns else "Close"
prices = df[[target_col]].copy().rename(columns={target_col: "Close"})
prices["Close"] = prices["Close"].ffill().bfill()
prices = prices.dropna()
prices = prices[~prices.index.duplicated(keep="last")]

print(f"[INFO] {TICKER} | Range: {prices.index.min().date()} → {prices.index.max().date()} | N={len(prices)}")
if len(prices) < WINDOW_SIZE + 30:
    raise ValueError("Data terlalu pendek untuk WINDOW_SIZE ini. Kecilkan WINDOW_SIZE atau perpanjang PERIOD.")

# 4) Split time-series 70% / 15% / 15% (tanpa acak), scaler hanya di TRAIN
N = len(prices)
N_train = int(0.70 * N)
N_val = int(0.15 * N)
N_test = N - N_train - N_val

train_df = prices.iloc[:N_train]
val_df = prices.iloc[N_train:N_train+N_val]
test_df = prices.iloc[N_train+N_val:]

print(f"[SPLIT] Train={len(train_df)} | Val={len(val_df)} | Test={len(test_df)}")

scaler = MinMaxScaler((0,1))
scaler.fit(train_df[["Close"]].to_numpy()) # fit hanya di train

▶ series_scaled = scaler.transform(prices[["Close"]].to_numpy()).reshape(-1) # 1D array

# 5) Sliding window di seluruh seri, lalu dipotong ke set berdasarkan index target
def make_all_sequences(series_1d, window):
    X, y, tgt_idx = [], [], []
    for i in range(window, len(series_1d)):
        X.append(series_1d[i-window:i])
        y.append(series_1d[i])
        tgt_idx.append(i) # posisi target
    X = np.array(X, dtype=np.float32).reshape(-1, window, 1)
    y = np.array(y, dtype=np.float32)
    return X, y, np.array(tgt_idx)

X_all, y_all, idx_all = make_all_sequences(series_scaled, WINDOW_SIZE)

def split_by_index(X_all, y_all, idx_all, n_train, n_val):
    train_mask = idx_all < n_train
    val_mask = (idx_all >= n_train) & (idx_all < n_train + n_val)
    test_mask = idx_all >= (n_train + n_val)
    return (X_all[train_mask], y_all[train_mask],
            X_all[val_mask], y_all[val_mask],
            X_all[test_mask], y_all[test_mask])

X_train, y_train, X_val, y_val, X_test, y_test = split_by_index(X_all, y_all, idx_all, N_train, N_val)
print(f"[SEQ] X_train:", X_train.shape, " | X_val:", X_val.shape, " | X_test:", X_test.shape)

# 6) Attention layer (sederhana)
class SimpleAttention(layers.Layer):
    def __init__(self, units=64, **kwargs):
        super().__init__(**kwargs)
        self.W = layers.Dense(units, activation="tanh")

```

```

        self.V = layers.Dense(1)
    def call(self, inputs):
        score = self.W(inputs)          # [b, t, u]
        score = self.V(score)           # [b, t, 1]
        weights = tf.nn.softmax(score, axis=1)
        context = tf.reduce_sum(weights * inputs, axis=1) # [b, features]
        return context

# 7) Arsitektur CNN-BiGRU-Attention (dropout sebagai argumen)
def build_model(input_shape, dropout_rate=0.2):
    inp = Input(shape=input_shape)
    x = Conv1D(32, kernel_size=3, padding="causal", activation="relu")(inp)
    x = MaxPooling1D(pool_size=2)(x)
    x = Dropout(dropout_rate)(x)
    x = Conv1D(64, kernel_size=3, padding="causal", activation="relu")(x)
    x = Dropout(dropout_rate)(x)
    x = Bidirectional(GRU(64, return_sequences=True))(x)
    x = SimpleAttention(units=64)(x)
    x = Dense(64, activation="relu")(x)
    out = Dense(1)(x)
    model = Model(inp, out)
    model.compile(optimizer="adam", loss="mse", metrics=["mae"])
    return model

# 8) Fungsi latih & evaluasi (kembalikan MAE/RMSE skala harga asli)
def inverse_transform_1d(arr_scaled):
    arr_scaled = np.asarray(arr_scaled).reshape(-1,1)
    return scaler.inverse_transform(arr_scaled).reshape(-1)

```

```

def train_and_eval(dropout_rate, patience, label):
    model = build_model((WINDOW_SIZE,1), dropout_rate=dropout_rate)
    early = EarlyStopping(monitor="val_loss", patience=patience, restore_best_weights=True)
    plateau = ReduceLROnPlateau(monitor="val_loss", patience=3, factor=0.5, verbose=0)
    t0 = time.perf_counter()
    hist = model.fit(
        X_train, y_train,
        validation_data=(X_val, y_val),
        epochs=EPOCHS, batch_size=BATCH,
        callbacks=[early, plateau],
        shuffle=False, verbose=0
    )
    t_spent = time.perf_counter() - t0

    # Prediksi test
    pred_scaled = model.predict(X_test, verbose=0).reshape(-1)
    y_scaled = y_test.reshape(-1)
    pred = inverse_transform_1d(pred_scaled)
    y_true = inverse_transform_1d(y_scaled)

    mae = mean_absolute_error(y_true, pred)
    rmse = math.sqrt(mean_squared_error(y_true, pred))

```

```

print(f"[{label}] time={t_spent:.2f}s | epochs={len(hist.history['loss'])} | TEST MAE={mae:.4f} RMSE={rmse:.4f}")
return {
    "label": label,
    "dropout": dropout_rate,
    "patience": patience,
    "time_s": t_spent,
    "epochs": len(hist.history["loss"]),
    "MAE": mae,
    "RMSE": rmse,
    "y_true": y_true,
    "y_pred": pred,
    "history": hist.history
}

# 9) LATIH baseline (simulasi "model sudah dilatih") vs konfigurasi baru
res_base = train_and_eval(BASELINE_DROPOUT, BASELINE_PATIENCE, label=f"Baseline (dropout={BASELINE_DROPOUT}, patience={BASELINE_PATIENCE})")
res_new = train_and_eval(NEW_DROPOUT, NEW_PATIENCE, label=f"Modified (dropout={NEW_DROPOUT}, patience={NEW_PATIENCE})")

# 10) Perbandingan MAE & RMSE
summary = pd.DataFrame([
    {"Model": res_base["label"], "MAE": res_base["MAE"], "RMSE": res_base["RMSE"]},
    {"Model": res_new["label"], "MAE": res_new["MAE"], "RMSE": res_new["RMSE"]}
])
display(summary)

# 11) Plot ringkas: history loss & kurva prediksi (test) utk model baru
plt.figure(figsize=(6,4))
plt.plot(res_base["history"]["loss"], label="base train")
plt.plot(res_base["history"]["val_loss"], label="base val")
plt.plot(res_new["history"]["loss"], label="new train")

```

```

plt.plot(res_new["history"]["val_loss"], label="new val")
plt.title("Training History (MSE)")
plt.xlabel("Epoch"); plt.ylabel("Loss"); plt.legend(); plt.tight_layout(); plt.show()

plt.figure(figsize=(10,4))
plt.plot(res_base["y_true"], label="Actual")
plt.plot(res_base["y_pred"], label="Base Pred")
plt.plot(res_new["y_pred"], label="New Pred", alpha=0.8)
plt.title("Test Set: Actual vs Predicted")
plt.xlabel("Time Index (Test)"); plt.ylabel("Price"); plt.legend(); plt.tight_layout(); plt.show()

```

```

TensorFlow: 2.19.0
GPU available: False
[*****100%*****] 1 of 1 completed
[INFO] BRIS.K | Range: 2023-08-11 → 2025-08-11 | N=473
[SPLIT] Train=331 | Val=70 | Test=72
[SEQ] X_train: (271, 60, 1) | X_val: (70, 60, 1) | X_test: (72, 60, 1)
[Baseline (dropout=0.2, patience=10)] time=13.70s | epochs=11 | TEST MAE=153.6594 RMSE=186.9498
[Modified (dropout=0.3, patience=5)] time=12.29s | epochs=7 | TEST MAE=163.8339 RMSE=198.6629

```

	Model	MAE	RMSE
0	Baseline (dropout=0.2, patience=10)	153.659409	186.949766
1	Modified (dropout=0.3, patience=5)	163.833923	198.662903

5. Studi Kasus

Setiap kelompok menguji tiga metode penanganan *overfitting* (dropout, early stopping, dan regularisasi L2) pada dataset harga saham yang sama, lalu membandingkan performanya. Presentasikan hasilnya.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

INTERPRETASI HASIL MODEL PREDIKSI SAHAM

1. Learning Outcome (LO)

- a) Menginterpretasikan hasil prediksi harga saham berdasarkan output model.
- b) Menggunakan visualisasi untuk membandingkan hasil prediksi dan data aktual.
- c) Menjelaskan peran bobot Attention dalam interpretabilitas model.

2. Materi

Analisis Hasil Prediksi

Evaluasi kinerja model prediksi harga saham berbasis *hybrid deep learning* (CNN–BiGRU–Attention) memerlukan pendekatan sistematis yang mencakup perbandingan hasil prediksi dengan harga aktual, identifikasi tren yang berhasil ditangkap model, serta analisis kesalahan prediksi. Langkah pertama adalah menyusun dataset uji yang merepresentasikan kondisi pasar sebenarnya. Dataset ini biasanya mencakup data *Open, High, Low, Close, dan Volume* (OHLCV) dalam periode tertentu yang tidak digunakan selama proses pelatihan model. Pengujian dilakukan untuk memastikan bahwa model tidak hanya menghafal pola historis, melainkan mampu menggeneralisasi prediksi pada data baru, *"ibarat siswa yang diuji dengan soal berbeda dari latihan sebelumnya untuk melihat pemahaman sejati"*.

Perbandingan antara harga prediksi dan harga aktual dapat dilakukan dengan mengukur *error* menggunakan metrik seperti *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), atau *Mean Absolute Percentage Error* (MAPE). Misalnya, jika model memprediksi harga penutupan (closing price) saham sebesar Rp5.120, sementara harga aktual adalah Rp5.150, maka selisih prediksi sebesar Rp30 dapat dihitung dan diakumulasikan pada seluruh dataset uji untuk memperoleh nilai rata-rata kesalahan. Nilai metrik yang rendah menunjukkan model memiliki akurasi tinggi, "*seperti jarum kompas yang hampir selalu menunjuk tepat ke arah utara*".

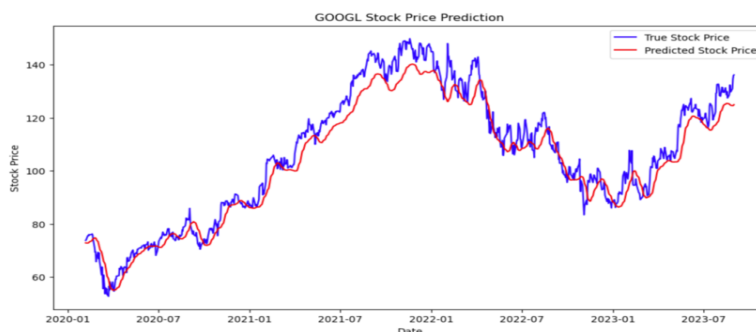
Selain nilai numerik, analisis visual terhadap perbedaan prediksi dan aktual dapat mengungkap pola penting. Model yang baik tidak hanya mendekati nilai aktual, tetapi juga mampu mengikuti arah tren harga, baik tren naik (*uptrend*), tren turun (*downtrend*), maupun pergerakan mendatar (*sideways*). Sebagai contoh, jika harga aktual mengalami kenaikan konsisten selama tujuh hari berturut-turut, model yang efektif akan menunjukkan pola prediksi dengan kecenderungan naik pada periode yang sama, meskipun mungkin terdapat perbedaan kecil pada nilai absolutnya, "*layaknya pelukis yang mampu meniru goresan kuas aslinya meski ada sedikit variasi warna*".

Kesalahan prediksi (*prediction error*) menjadi fokus berikutnya. Kesalahan dapat dikategorikan menjadi *overestimation* (prediksi lebih tinggi dari aktual) dan *underestimation* (prediksi lebih rendah dari aktual). Analisis

distribusi kesalahan ini bermanfaat untuk memahami bias model. Misalnya, model yang cenderung *overestimate* dapat memberikan sinyal positif palsu (*false positive*) pada strategi investasi, "*mirip ramalan cuaca yang memprediksi hujan padahal langit cerah*", sementara model yang sering *underestimate* dapat melewatkan peluang keuntungan, "*seperti pedagang yang ragu menaikkan harga walau permintaan sedang membludak*".

Visualisasi Hasil

Visualisasi merupakan sarana penting untuk memahami perilaku model prediksi harga saham. Salah satu teknik yang umum digunakan adalah *grafik garis prediksi vs aktual*. Grafik ini menampilkan dua kurva pada satu bidang koordinat: garis biru mewakili harga aktual, sedangkan garis oranye mewakili prediksi model. Kedekatan kedua garis secara visual mencerminkan akurasi model. Apabila garis prediksi mengikuti pola fluktuasi harga aktual dengan perbedaan minimal, dapat disimpulkan bahwa model berhasil mempelajari dinamika pasar.



Gambar 1. Grafik Perbandingan Harga Prediksi dan Aktual Saham GOOGL

Gambar ini menampilkan perbandingan antara harga saham aktual (True Stock Price, garis biru) dan harga saham hasil prediksi model (Predicted Stock Price, garis merah) untuk saham GOOGL pada periode Januari 2020 hingga Juli 2023. Sumbu horizontal menunjukkan waktu (tanggal), sedangkan sumbu vertikal menunjukkan harga saham dalam satuan USD.

Terlihat bahwa garis prediksi mengikuti pola fluktuasi harga aktual secara konsisten, baik dalam tren naik, tren turun, maupun periode pergerakan mendatar. Kedekatan kedua garis ini mengindikasikan bahwa model memiliki kemampuan yang baik dalam menangkap dinamika pasar dan memproyeksikan harga di masa mendatang, "layaknya bayangan yang bergerak hampir serentak dengan objek aslinya". Visualisasi ini menjadi bukti intuitif atas akurasi model dan membantu pembaca mengidentifikasi momen di mana model berhasil maupun meleset dari prediksi harga sebenarnya.

Selain itu, *plot distribusi error* digunakan untuk menganalisis sebaran kesalahan prediksi. Distribusi yang terkonsentrasi di sekitar nol menunjukkan bahwa model tidak memiliki bias signifikan ke arah positif atau negatif. Sebaliknya, distribusi yang miring atau memiliki ekor panjang (*long tail*) menandakan adanya periode atau kondisi pasar tertentu yang sulit diprediksi oleh model. Analisis ini dapat membantu perbaikan model, misalnya dengan menambahkan fitur teknikal atau indikator makroekonomi.

Visualisasi lainnya yang bermanfaat adalah *scatter plot* antara nilai aktual dan nilai prediksi. Titik-titik yang tersebar rapat di sekitar garis diagonal menandakan kesesuaian tinggi antara prediksi dan harga sebenarnya. Scatter plot ini sering dilengkapi dengan koefisien determinasi (*R-squared*) sebagai indikator kekuatan hubungan linier antara kedua variabel.

Bobot Attention

Mekanisme *attention* dalam arsitektur CNN-BiGRU-Attention memungkinkan model memfokuskan perhatiannya pada bagian-bagian data yang paling relevan terhadap prediksi. Bobot *attention* yang dihasilkan dapat divisualisasikan dalam bentuk *heatmap*, di mana intensitas warna menunjukkan tingkat kepentingan suatu langkah waktu terhadap output akhir. Sebagai contoh, jika model memprediksi lonjakan harga pada tanggal 15 Agustus, *heatmap* bobot *attention* mungkin menunjukkan warna merah pekat pada tanggal 10–12 Agustus, mengindikasikan bahwa periode tersebut memiliki pengaruh besar terhadap prediksi. Hal ini dapat diinterpretasikan bahwa kenaikan volume perdagangan atau *breakout* harga pada periode tersebut menjadi sinyal utama yang dipertimbangkan model.

Interpretasi bobot *attention* penting tidak hanya untuk validasi teknis, tetapi juga sebagai alat komunikasi hasil kepada pihak non-teknis seperti investor atau manajer portofolio. Dengan menunjukkan bagian data yang menjadi fokus model, pengguna dapat memahami alasan di balik

prediksi tertentu. Hal ini meningkatkan *trustworthiness* model dan memudahkan integrasi hasil prediksi ke dalam proses pengambilan keputusan.

Penggunaan Hasil Prediksi

Hasil prediksi harga saham dapat dimanfaatkan sebagai rekomendasi pendukung keputusan investasi. Misalnya, jika model memproyeksikan kenaikan harga yang konsisten selama beberapa hari ke depan, investor dapat mempertimbangkan untuk membeli saham terkait. Sebaliknya, prediksi penurunan harga dapat menjadi sinyal untuk menjual atau melakukan lindung nilai (*hedging*) terhadap posisi yang ada.

Selain peluang keuntungan, hasil prediksi juga berperan dalam penilaian risiko. Dengan menggabungkan prediksi harga dan volatilitas yang diperkirakan, investor dapat menghitung rasio risiko terhadap imbal hasil (*risk-return ratio*). Sebagai contoh, prediksi kenaikan harga 5% dalam seminggu dengan volatilitas rendah mungkin lebih menarik dibandingkan potensi kenaikan 10% dengan volatilitas sangat tinggi. Dalam konteks manajemen portofolio, hasil prediksi dapat digunakan untuk mengatur alokasi aset. Portofolio dapat dimodifikasi secara dinamis sesuai dengan sinyal prediksi model, sehingga eksposur terhadap risiko pasar dapat dioptimalkan. Misalnya, saat model mendeteksi tren bearish pada sektor teknologi, manajer portofolio dapat mengurangi proporsi aset di sektor

tersebut dan mengalihkan dana ke sektor yang diprediksi mengalami pertumbuhan.

Integrasi hasil prediksi ke dalam sistem perdagangan algoritmik (*algorithmic trading system*) juga memungkinkan eksekusi otomatis berdasarkan sinyal model. Dengan demikian, keputusan beli atau jual dapat dilakukan secara real-time tanpa keterlambatan manusia, yang krusial pada kondisi pasar dengan volatilitas tinggi.

3. Soal Esai

1. Apa tujuan interpretasi hasil model prediksi saham?
2. Bagaimana cara membandingkan hasil prediksi dan data aktual?
3. Mengapa visualisasi penting dalam interpretasi model?
4. Sebutkan dua jenis grafik yang digunakan untuk evaluasi hasil prediksi.
5. Apa yang dimaksud dengan bobot Attention?
6. Bagaimana bobot Attention membantu interpretabilitas model?
7. Jelaskan perbedaan error positif dan error negatif dalam prediksi.
8. Mengapa analisis tren penting dalam evaluasi model?
9. Bagaimana hasil prediksi dapat digunakan untuk strategi investasi?
10. Berikan contoh risiko jika interpretasi hasil model dilakukan secara keliru.

4. Tugas Individu

Gunakan model CNN–BiGRU–Attention yang telah dilatih. Buat grafik prediksi vs aktual, analisis perbedaan, dan berikan interpretasi singkat dalam 1 halaman.

```
# =====
# Prediksi Saham BSI (BRIS.JK) dari Yahoo Finance
# CNN–BiGRU–Attention: Inference + Grafik + Interpretasi
# Satu sel Colab (gabungan + patch Matplotlib)
# =====

# (Opsional) pasang paket di Colab kalau perlu:
#!pip install -q yfinance tensorflow==2.15.0

import os, math, warnings, datetime as dt
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error

import tensorflow as tf
from tensorflow.keras import layers, Model
from tensorflow.keras.models import load_model
from tensorflow.keras.layers import (
    Input, Conv1D, BatchNormalization, Activation,
    MaxPooling1D, Dropout, Bidirectional, GRU, Dense
)

import yfinance as yf

# ===== 1) Konfigurasi =====
TICKER      = "BRIS.JK"      # Bank Syariah Indonesia
START       = "2015-01-01"
END         = dt.date.today().strftime("%Y-%m-%d")

TARGET_COL  = "Close"       # target prediksi
FEATURE_COLS = ["Open", "High", "Low", "Close", "Volume"] # fitur default
TIME_STEPS  = 60            # panjang jendela input (lookback)
FORECAST_HORIZON = 1        # 1 langkah ke depan
TEST_SIZE_RATIO = 0.2       # porsi test (terakhir)
SCALER_TYPE = "minmax"      # 'minmax' atau 'standard'

# Model terlatih (opsional). Jika file tidak ada, script akan melatih singkat.
LOAD_SAVED_MODEL = True
MODEL_PATH       = "model_cnn_bigru_attention.h5" # unggah ini kalau sudah ada
MODEL_WEIGHTS_PATH = "weights_cnn_bigru_attention.h5" # jika LOAD_SAVED_MODEL=False
```

```

# ===== 2) Ambil data dari Yahoo Finance =====
print(f"Mengunduh data {TICKER} dari Yahoo Finance...")
df = yf.download(TICKER, start=START, end=END, auto_adjust=False, progress=False).reset_index()
if df.empty:
    raise ValueError("Data kosong dari Yahoo Finance. Periksa ticker/interval tanggal.")

# pastikan kolom Volume tidak NaN
if "Volume" in df.columns:
    df["Volume"] = df["Volume"].fillna(0)

# simpan CSV (opsional)
csv_path = f"{TICKER.replace('.', '_')}.csv"
df.to_csv(csv_path, index=False)
print("Contoh 5 baris:\n", df.head().to_string(index=False))

# ===== 3) Layer Attention =====
class TemporalAttention(layers.Layer):
    """
    Attention sederhana di atas dimensi waktu.
    Input: (batch, time_steps, hidden)
    Output: context (batch, hidden), attn_weights (batch, time_steps)
    """
    def __init__(self, **kwargs):
        super().__init__(**kwargs)

    def build(self, input_shape):
        self.W = self.add_weight(
            name='attn_W',
            shape=(input_shape[-1], 1),
            initializer='glorot_uniform',

            trainable=True
        )
        self.b = self.add_weight(
            name='attn_b',
            shape=(input_shape[1],),
            initializer='zeros',
            trainable=True
        )
        super().build(input_shape)

    def call(self, inputs, **kwargs):
        # inputs: (B, T, H)
        e = tf.squeeze(tf.tensordot(inputs, self.W, axes=[[2], [0]]), axis=-1) # (B, T)
        e = tf.keras.activations.tanh(e + self.b)
        a = tf.nn.softmax(e, axis=1) # (B, T)
        a_exp = tf.expand_dims(a, axis=-1) # (B, T, 1)
        context = tf.reduce_sum(inputs * a_exp, axis=1) # (B, H)
        return context, a

    def get_config(self):
        return super().get_config()

# ===== 4) Helper: Prepro & Visual =====
def choose_scaler(name='minmax'):
    return StandardScaler() if name.lower() == 'standard' else MinMaxScaler()

def create_sequences(X, y, dates, time_steps=60, horizon=1):
    Xs, ys, ds = [], [], []
    for i in range(len(X) - time_steps - horizon + 1):
        Xs.append(X[i:i+time_steps])

```

```

        ys.append(y[i+time_steps + horizon - 1])
        ds.append(dates[i+time_steps + horizon - 1])
    return np.array(Xs), np.array(ys), np.array(ds)

def time_series_train_test_split(X, y, dates, test_ratio=0.2):
    n_total = len(X)
    n_test = int(np.ceil(n_total * test_ratio))
    n_train = n_total - n_test
    return (
        X[:n_train], y[:n_train], dates[:n_train],
        X[n_train:], y[n_train:], dates[n_train:]
    )

def build_cnn_bigru_attention(input_shape):
    inp = Input(shape=input_shape)
    x = Conv1D(64, kernel_size=3, padding='causal')(inp)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = MaxPooling1D(pool_size=2)(x)

    x = Conv1D(128, kernel_size=3, padding='causal')(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = Dropout(0.2)(x)

    x = Bidirectional(GRU(128, return_sequences=True), name='bigru')(x)
    context, attn_weights = TemporalAttention(name='temporal_attention')(x)
    out = Dense(1, name='y_hat')(context)

```

```

model = Model(inputs=inp, outputs=out)
model.attn_model = Model(inputs=inp, outputs=attn_weights)
model.compile(optimizer='adam', loss='mse')
return model

def plot_pred_vs_actual(dates, y_true, y_pred, title='Prediksi vs Aktual'):
    plt.figure(figsize=(12,5))
    plt.plot(dates, y_true, label='Aktual')
    plt.plot(dates, y_pred, label='Prediksi')
    plt.title(title); plt.xlabel('Tanggal'); plt.ylabel('Harga')
    plt.legend(); plt.grid(True); plt.tight_layout(); plt.show()

def residual_analysis_plots(dates, y_true, y_pred):
    residuals = y_true - y_pred

    plt.figure(figsize=(12,3))
    plt.plot(dates, residuals)
    plt.title('Residual (Aktual - Prediksi) sepanjang waktu')
    plt.xlabel('Tanggal'); plt.ylabel('Residual'); plt.grid(True)
    plt.tight_layout(); plt.show()

    plt.figure(figsize=(6,4))
    plt.hist(residuals, bins=30)
    plt.title('Distribusi Residual')
    plt.xlabel('Residual'); plt.ylabel('Frekuensi')
    plt.tight_layout(); plt.show()

```

```

plt.figure(figsize=(5,5))
plt.scatter(y_pred, residuals, s=10)
plt.axhline(0, linestyle='--')
plt.title('Prediksi vs Residual')
plt.xlabel('Prediksi'); plt.ylabel('Residual')
plt.tight_layout(); plt.show()

def metrics_report(y_true, y_pred):
    mae = mean_absolute_error(y_true, y_pred)
    mse = mean_squared_error(y_true, y_pred)
    rmse = math.sqrt(mse)
    mape = np.mean(np.abs((y_true - y_pred) / (y_true + 1e-8))) * 100
    print('\n=== METRIK ERROR ===')
    print(f'MAE : {mae:.4f}')
    print(f'RMSE: {rmse:.4f}')
    print(f'MAPE: {mape:.2f}%')
    return {'MAE': mae, 'RMSE': rmse, 'MAPE': mape}

def build_attention_probe_from_loaded(model):
    # Coba temukan layer TemporalAttention pada model yang di-load (.h5)
    try:
        attn_layer = None
        for lyr in model.layers:
            if isinstance(lyr, TemporalAttention):
                attn_layer = lyr
                break
        if attn_layer is None:
            try:
                attn_layer = model.get_layer('temporal_attention')
            except Exception:

```

```

        attn_layer = None
    if attn_layer is not None:
        outs = attn_layer.output
        if isinstance(outs, (list, tuple)) and len(outs) == 2:
            attn_weights = outs[1]
            return Model(inputs=model.input, outputs=attn_weights)
    except Exception as e:
        print('Gagal membangun attn_model dari model .h5:', e)
    return None

# --- PATCH Matplotlib >=3.8: tanpa argumen use_line_collection ---
def attention_visualization(model, X_sample, dates_window=None, title='Bobot Attention (sampel)':
    if not hasattr(model, 'attn_model') or model.attn_model is None:
        print('Model tidak memiliki attn_model. Lewati visualisasi attention.')
        return
    attn_w = model.attn_model.predict(X_sample, verbose=0).reshape(-1)

    # Stem plot tanpa use_line_collection
    plt.figure(figsize=(12,3))
    plt.stem(range(len(attn_w)), attn_w, basefmt=" ")
    plt.title(title); plt.xlabel('Langkah waktu (dalam jendela)'); plt.ylabel('Bobot')
    plt.tight_layout(); plt.show()

    # Plot terhadap tanggal (sinkronkan panjang jika jendela diperkecil oleh pooling)
    if dates_window is not None:
        dates_window = np.array(dates_window)
        if len(dates_window) >= len(attn_w):
            dates_window = dates_window[-len(attn_w):]
            plt.figure(figsize=(12,3))
            plt.plot(dates_window, attn_w)

```

```

plt.title(title + ' - terhadap tanggal')
plt.xlabel('Tanggal dalam jendela input'); plt.ylabel('Bobot'); plt.grid(True)
plt.tight_layout(); plt.show()

# ===== 5) Preprocessing =====
# urutkan tanggal
assert "Date" in df.columns, "Kolom tanggal 'Date' tidak ditemukan."
df = df.sort_values("Date").reset_index(drop=True)

# validasi fitur & target
missing_feats = [c for c in FEATURE_COLS if c not in df.columns]
if missing_feats:
    print("Peringatan fitur hilang & di-skip:", missing_feats)
    FEATURE_COLS = [c for c in FEATURE_COLS if c in df.columns]
assert TARGET_COL in df.columns, f"Kolom target {TARGET_COL} tidak ada di data."
assert len(FEATURE_COLS) > 0, "Tidak ada fitur valid."

# nilai mentah
X_raw = df[FEATURE_COLS].values.astype("float32")
y_raw = df[TARGET_COL].values.astype("float32")
dates_all = df["Date"].values

# scaling
x_scaler = choose_scaler(SCALER_TYPE)
y_scaler = choose_scaler(SCALER_TYPE)
X_scaled = x_scaler.fit_transform(X_raw)
y_scaled = y_scaler.fit_transform(y_raw.reshape(-1,1)).ravel()

# sequence
X_seq, y_seq, seq_dates = create_sequences(
    X_scaled, y_scaled, dates_all,
    time_steps=TIME_STEPS, horizon=FORECAST_HORIZON
)
print("X_seq shape:", X_seq.shape, "| y_seq shape:", y_seq.shape)

# split train/test
X_train, y_train, d_train, X_test, y_test, d_test = time_series_train_test_split(
    X_seq, y_seq, seq_dates, test_ratio=TEST_SIZE_RATIO
)
print(f"Train: {X_train.shape}, Test: {X_test.shape}")

# ===== 6) Muat / Latih Model =====
input_shape = (TIME_STEPS, X_train.shape[-1])
model = None

if LOAD_SAVED_MODEL and os.path.exists(MODEL_PATH):
    print("Memuat model terlatih dari .h5 ...")
    model = load_model(MODEL_PATH, custom_objects={'TemporalAttention': TemporalAttention})
    probe = build_attention_probe_from_loaded(model)
    model.attn_model = probe
elif not LOAD_SAVED_MODEL and os.path.exists(MODEL_WEIGHTS_PATH):
    print("Bangun arsitektur + muat weights ...")
    model = build_cnn_bigru_attention(input_shape)
    model.load_weights(MODEL_WEIGHTS_PATH)
else:
    # fallback: latih singkat agar pipeline jalan
    print("Model .h5/weights tidak ditemukan. Melatih singkat (demo ...)")
    model = build_cnn_bigru_attention(input_shape)

```

```

EPOCHS = 5
HISTORY = model.fit(
    X_train, y_train,
    validation_data=(X_test, y_test),
    epochs=EPOCHS, batch_size=32, verbose=1
)
model.save("model_cnn_bigru_attention_bris.h5")
print("Model demo disimpan ke model_cnn_bigru_attention_bris.h5")

model.summary()

# ===== 7) Inference & Visualisasi =====
y_pred_scaled = model.predict(X_test, verbose=0).ravel()
y_pred = y_scaler.inverse_transform(y_pred_scaled.reshape(-1,1)).ravel()
y_true = y_scaler.inverse_transform(y_test.reshape(-1,1)).ravel()

_ = metrics_report(y_true, y_pred)
plot_pred_vs_actual(d_test, y_true, y_pred, title=f"Prediksi vs Aktual - {TICKER} (Data Test)")
residual_analysis_plots(d_test, y_true, y_pred)

# Simpan hasil & tampilkan 5 error terbesar
out_df = pd.DataFrame({
    "date": d_test,
    "y_true": y_true,
    "y_pred": y_pred,
    "residual": y_true - y_pred
})
out_path = f"hasil_prediksi_{TICKER.replace('.', '_')}_cnn_bigru_attention.csv"
out_df.to_csv(out_path, index=False)
print("Hasil disimpan ke:", out_path)

```

```

out_df["abs_err"] = out_df["residual"].abs()
print("\n5 tanggal dengan error terbesar:")
print(out_df.sort_values("abs_err", ascending=False)[["date", "y_true", "y_pred", "residual"]].head().to_string(index=False))

# ===== 8) Interpretasi: Attention =====
if hasattr(model, "attn_model") and model.attn_model is not None:
    sample_idx = 0
    Xs = X_test[sample_idx:sample_idx+1]

    # tanggal jendela input (menyesuaikan panjang attention bila terpotong pooling)
    target_date = d_test[sample_idx]
    try:
        pos = int(np.where(dates_all == target_date)[0][0])
        start_pos = max(0, pos - TIME_STEPS)
        window_dates = dates_all[start_pos:pos]
        attention_visualization(model, Xs, window_dates, title="Robot Attention - 1 Sampel Test")
    except Exception:
        attention_visualization(model, Xs, None, title="Robot Attention - 1 Sampel Test")
else:
    print("Tidak bisa mengekstrak bobot attention (attn_model tidak tersedia).")

```

Output Program

```
Mengunduh data BRIS.JK dari Yahoo Finance...
Contoh 5 baris:
      Date Adj Close      Close      High      Low      Open      Volume
      BRIS.JK      BRIS.JK      BRIS.JK      BRIS.JK      BRIS.JK      BRIS.JK
2018-05-09 512.617737 531.543335 614.444580 507.160583 585.185303 282154012
2018-05-11 583.161316 604.691467 643.703857 536.419861 536.419861 292500171
2018-05-14 564.349731 585.185303 633.950745 585.185303 604.691467 127848119
2018-05-15 564.349731 585.185303 585.185303 585.185303 585.185303 0
2018-05-16 564.349731 585.185303 585.185303 585.185303 585.185303 0
X_seq shape: (1715, 60, 5) | y_seq shape: (1715,)
Train: (1372, 60, 5), Test: (343, 60, 5)
Model .h5/weights tidak ditemukan. Melatih singkat (demo) ...
Epoch 1/5
43/43 ————— 15s 118ms/step - loss: 0.1505 - val_loss: 0.3334
Epoch 2/5
43/43 ————— 4s 90ms/step - loss: 0.0102 - val_loss: 0.2538
Epoch 3/5
43/43 ————— 5s 124ms/step - loss: 0.0069 - val_loss: 0.1716
Epoch 4/5
43/43 ————— 4s 94ms/step - loss: 0.0053 - val_loss: 0.1237
Epoch 5/5
43/43 ————— 5s 95ms/step - loss: 0.0052 - val_loss: 0.1068
```

5. Studi Kasus

Setiap kelompok memilih 2 saham dari sektor berbeda. Bandingkan hasil prediksi model dan data aktual menggunakan grafik dan analisis tren, lalu simpulkan potensi investasi berdasarkan hasil tersebut.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

PERANCANGAN PROYEK PENGEMBANGAN MODEL PREDIKSI SAHAM

1. Learning Outcome (LO)

1. Mengidentifikasi peluang pengembangan model prediksi saham.
2. Merancang konsep proyek pengembangan model berbasis AI.
3. Mempertimbangkan risiko dan tantangan implementasi model di dunia nyata.

2. Materi

Peluang Pengembangan Model

- a) Integrasi data sentimen media sosial.
- b) Penggunaan model transformer (BERT, GPT) untuk analisis teks.
- c) Optimisasi *hyperparameter* dengan AutoML.

Risiko Implementasi

- a) Perubahan kondisi pasar yang cepat.
- b) Data bias atau tidak representatif.
- c) Risiko hukum dan etika investasi berbasis AI.

Tahapan Perancangan Proyek:

- a) Identifikasi masalah dan tujuan.
- b) Pemilihan data dan metode.
- c) Perancangan arsitektur model.
- d) Uji coba dan evaluasi.
- e) Implementasi dan monitoring.

Output yang Diharapkan:

Model prediksi yang siap diintegrasikan ke sistem rekomendasi investasi.

3. Soal Esai

1. Sebutkan dua peluang pengembangan model prediksi saham di masa depan.
2. Mengapa data sentimen media sosial dapat meningkatkan akurasi prediksi?
3. Apa keunggulan model transformer dibandingkan RNN tradisional?
4. Jelaskan manfaat AutoML dalam pengembangan model AI.
5. Sebutkan dua risiko penggunaan AI untuk investasi.
6. Mengapa kondisi pasar yang berubah cepat menjadi tantangan utama?
7. Apa saja tahap awal yang dilakukan dalam perancangan proyek AI?
8. Bagaimana memastikan data yang digunakan representatif?
9. Jelaskan alasan mengapa evaluasi berkala model itu penting.
10. Berikan contoh bentuk output yang diharapkan dari proyek pengembangan model prediksi saham.

4. Tugas Individu

Buat proposal singkat (maksimal 2 halaman) konsep pengembangan model prediksi saham dengan satu fitur

tambahan (misalnya data berita atau data sentimen). Sertakan tujuan, metode, dan data yang digunakan.

5. Studi Kasus

Setiap kelompok merancang proyek pengembangan model CNN–BiGRU–Attention yang dioptimalkan dengan data tambahan (sentimen, indikator teknikal lanjutan, atau data fundamental). Presentasikan konsep proyek dalam 5–7 slide.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Tugas Individu	30%
Studi Kasus Kelompok	30%
Partisipasi Diskusi	10%

PRESENTASI PROYEK AKHIR PREDIKSI SAHAM

1. Learning Outcome (LO)

- a) Menyajikan hasil proyek pengembangan model prediksi saham secara terstruktur.
- b) Menjelaskan metodologi, data, dan hasil yang diperoleh.
- c) Menjawab pertanyaan dan memberikan argumen yang didukung data.

2. Materi

Tujuan Pertemuan

Menguji kemampuan presentasi dan komunikasi hasil riset/proyek.

Komponen Presentasi

1. Latar belakang dan tujuan proyek.
2. Data yang digunakan (sumber dan preprocessing).
3. Metodologi/model yang dipakai.
4. Hasil evaluasi model (MAE, RMSE, MAPE, grafik prediksi).
5. Interpretasi hasil dan rekomendasi.
6. Tantangan dan peluang pengembangan ke depan.

Etika Presentasi: Penyampaian jelas, visual menarik, waktu tepat, mampu menjawab pertanyaan audiens.

3. Soal Esai

1. Apa tujuan utama presentasi proyek akhir?
2. Sebutkan 3 poin penting yang harus ada di slide presentasi proyek prediksi saham.

3. Mengapa visualisasi hasil prediksi penting dalam presentasi?
4. Apa manfaat menjelaskan tantangan proyek kepada audiens?
5. Bagaimana menyusun struktur presentasi yang efektif?
6. Sebutkan 2 kesalahan umum dalam presentasi proyek dan cara menghindarinya.
7. Mengapa kejelasan komunikasi memengaruhi penilaian proyek?
8. Bagaimana cara menjawab pertanyaan kritis dari audiens dengan baik?
9. Apa peran kesimpulan dalam presentasi proyek akhir?
10. Berikan contoh strategi untuk meningkatkan daya tarik presentasi.

4. Tugas Individu

Setiap mahasiswa menyiapkan materi presentasi pribadi (minimal 5 slide) yang berisi kontribusi individu pada proyek kelompok, termasuk analisis dan hasil kerja yang dilakukan.

5. Studi Kasus

Setiap kelompok mempresentasikan proyek pengembangan model CNN–BiGRU–Attention (atau variasi lainnya) yang sudah dikerjakan. Durasi presentasi maksimal 15 menit dengan sesi tanya jawab 5 menit.

6. Penilaian

Komponen	Bobot
Soal Esai	20%
Tugas Individu	30%
Studi Kasus Kelompok (Presentasi)	40%
Partisipasi Diskusi/Tanya Jawab	10%

UJIAN AKHIR SEMESTER (UAS)

1. Learning Outcome (LO)

- a) Mengintegrasikan seluruh pengetahuan dan keterampilan yang telah dipelajari selama 15 pertemuan.
- b) Menerapkan metode prediksi saham mulai dari pengolahan data hingga interpretasi hasil.
- c) Menyajikan solusi analisis yang tepat berdasarkan data dan model yang digunakan.

2. Materi yang Diujikan

Materi UAS mencakup:

- 1. Konsep dasar pasar saham dan peranannya dalam ekonomi.
- 2. Faktor fundamental, teknikal, dan eksternal yang memengaruhi harga saham.
- 3. Metode konvensional (Regresi, ARIMA, GARCH).
- 4. Metode modern berbasis AI (LSTM, GRU, CNN, Attention).
- 5. Arsitektur CNN–BiGRU–Attention.
- 6. Pengolahan data dan rekayasa fitur.
- 7. Pelatihan, evaluasi, dan interpretasi model.
- 8. Perancangan dan presentasi proyek pengembangan model.

3. Bentuk Ujian

Bagian A – Soal Esai (5 soal)

Menguji pemahaman konsep dan analisis teoretis.

Bagian B – Studi Kasus (2 soal)

Menguji kemampuan menganalisis data dan merancang solusi.

Bagian C – Praktikum

Mengimplementasikan model prediksi saham sesuai spesifikasi yang diberikan.

4. Soal Esai

1. Jelaskan perbedaan mendasar antara ARIMA, LSTM, dan CNN–BiGRU–Attention.
2. Sebutkan langkah-langkah preprocessing data pasar saham untuk deep learning.
3. Apa peran bobot Attention dalam interpretasi hasil model?
4. Jelaskan manfaat penggunaan variabel lag dalam prediksi harga saham.
5. Apa tantangan terbesar dalam mengimplementasikan model prediksi saham di pasar nyata?

5. Soal Studi Kasus

Kasus 1

Diberikan dataset harga saham 2 tahun terakhir, lakukan preprocessing, rekayasa fitur, dan pilih metode yang sesuai. Jelaskan alasan pemilihan metode.

Kasus 2

Bandingkan performa model konvensional dan AI pada dataset yang sama menggunakan MAE dan RMSE.

6. Penilaian

Komponen	Bobot
Soal Esai	30%
Studi Kasus	30%
Praktikum	40%

Biografi Penulis



Ibnu Rasyid Munthe, ST., M.Kom., lahir di Rantau Prapat, Kabupaten Labuhanbatu, pada 1987. Beliau menyelesaikan pendidikan S1 di Universitas Nurtanio Bandung dan S2 di Universitas Putra Indonesia (YPTK) Padang. Aktif menulis artikel ilmiah dan buku, Ibnu Rasyid Munthe saat ini mengajar di Universitas Labuhanbatu. Selain itu, beliau mengelola kanal YouTube @manjaddawajada2022 sebagai media pembelajaran dan berbagi pengetahuan, memberikan kontribusi dalam dunia pendidikan dan teknologi.



Bhakti Helvi Rambe, SE., M.Si., lahir di Rantauprapat pada 28 September 1988, adalah sosok akademisi inspiratif yang memadukan dedikasi, pengetahuan, dan ketulusan dalam setiap langkahnya. Tumbuh di tanah kelahiran yang membentuk karakter gigih, ia menempuh pendidikan dasar hingga menengah di SD, MTsN, dan SMA Rantauprapat. Kecintaan mendalam pada ekonomi dan akuntansi mengantarkannya meraih gelar Sarjana dan Magister Akuntansi di Universitas Sumatera Utara. Memulai karier mengajar di LP3M Medan, kini ia menjadi dosen tetap di Fakultas Ekonomi dan Bisnis Universitas Labuhanbatu. Dengan semangat membimbing, ia terus menginspirasi generasi muda melalui ilmu dan teladan.



Shabrina Rasyid Munthe, S.Pd.I., MS, lahir di Rantauprapat pada 20 Februari 1992. Beliau meraih gelar S1 di bidang Tarbiyah dari Institut Agama Islam Negeri Padangsidempuan dan S2 Magister Sastra dari Universitas Islam Sumatera Utara (UISU). Sejak 2019, beliau menjabat dosen tetap di Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Al Washliyah Labuhanbatu. Shabrina juga aktif berkontribusi dalam pengembangan ilmu pengetahuan dan pendidikan, khususnya di bidang bahasa Inggris.

Buku Ajar Hybrid Deep Learning: CNN, BiGRU, dan Attention untuk Prediksi Saham menghadirkan panduan komprehensif untuk memahami dan mengimplementasikan teknologi kecerdasan buatan terkini dalam analisis pasar modal. Buku ini memadukan teori pasar saham, teknik pengolahan data, serta arsitektur *deep learning* modern yang menggabungkan CNN untuk ekstraksi pola lokal, BiGRU untuk pemodelan hubungan temporal dua arah, dan mekanisme Attention untuk fokus pada informasi yang paling relevan.

Disusun secara sistematis, buku ini membimbing pembaca mulai dari konsep dasar, tahapan *preprocessing* data, *feature engineering*, hingga penerapan model hybrid pada studi kasus nyata. Dilengkapi contoh kode dan visualisasi, buku ini memudahkan mahasiswa, peneliti, dan praktisi dalam membangun model prediksi yang akurat dan adaptif terhadap volatilitas pasar.

Dengan memadukan kekuatan teori dan praktik, buku ini menjadi referensi penting bagi siapa saja yang ingin memanfaatkan AI untuk meningkatkan strategi analisis dan pengambilan keputusan di dunia investasi.

