

**THESIS**  
**OPTIMASI ADAM DALAM MODEL GATED RECURRENT**  
**UNIT (GRU) UNTUK PREDIKSI SAHAM**

Oleh:

R Daniel Hartanto

P31.2023.02570

Tesis diajukan sebagai salah satu syarat  
untuk memperoleh gelar  
Magister Komputer



**PROGRAM PASCA SARJANA**  
**MAGISTER TEKNIK INFORMATIKA**  
**UNIVERSITAS DIAN NUSWANTORO**

**SEMARANG**

**2025**



UNIVERSITAS DIAN NUSWANTORO

**PENGESAHAN STATUS TESIS**

JUDUL : OPTIMASI ADAM DALAM MODEL  
GATED RECURRENT UNIT (GRU)  
UNTUK PREDIKSI SAHAM

NAMA : R DANIEL HARTANTO

Mengijinkan Tesis Magister Komputer ini disimpan di Perpustakaan Universitas Dian Nuswantoro dengan syarat-syarat sebagai berikut:

1. Tesis adalah hak milik Universitas Dian Nuswantoro
2. Perpustakaan Universitas Dian Nuswantoro dibenarkan membuat salinan untuk tujuan referensi saja.
3. Perpustakaan juga dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antar institusi perguruan tinggi.
4. Berikan tanda  $\surd$  sesuai dengan kategori Tesis.

- Sangat Rahasia (Mengandung isi tentang keselamatan atau kepentingan Negara Republik Indonesia)
- Rahasia (Mengandung isi tentang kerahasiaan dari suatu organisasi/badan tempat penelitian tesis dikerjakan)
- Biasa

Disahkan oleh:

R Daniel Hartanto  
Penulis

Dr. Guruh Fajar Shidik S.Kom., M. Cs  
Pembimbing Utama

Alamat : Karanganyar Gunung 253 Semarang

Tanggal : Mei 2025



**UNIVERSITAS DIAN NUSWANTORO**

**PERNYATAAN PENULIS**

**JUDUL : OPTIMASI ADAM DALAM MODEL GATED RECURRENT UNIT (GRU) UNTUK PREDIKSI SAHAM**  
**NAMA : R DANIEL HARTANTO**  
**NIM : P31.2023.02570**

“Saya menyatakan dan bertanggungjawab dengan sebenarnya bahwa Tesis ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya. Jika pada waktu selanjutnya ada pihak lain yang mengklaim bahwa Tesis ini sebagai karyanya, yang disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk dibatalkan gelar Magister Komputer saya beserta segala hak dan kewajiban yang melekat pada gelar tersebut”.

Semarang, Mei 2025

  
R DANIEL HARTANTO  
Penulis



UNIVERSITAS DIAN NUSWANTORO

PERSETUJUAN TESIS


JUDUL : OPTIMASI ADAM DALAM MODEL GATED RECURRENT  
UNIT (GRU) UNTUK PREDIKSI SAHAM  
NAMA : R DANIEL HARTANTO  
NIM : P31.2023.02570

Tesis ini telah diperiksa dan disetujui,

Semarang, Mei 2025



Dr. Guruh Fajar Shidik S.Kom., M.Cs  
Pembimbing Utama



Dr. Farriz Al Zami, M.Kom  
Pembimbing Pembantu



**UNIVERSITAS DIAN NUSWANTORO**

**PENGESAHAN THESIS**

**JUDUL : OPTIMASI ADAM DALAM MODEL GATED RECURRENT UNIT (GRU) UNTUK PREDIKSI SAHAM**

**NAMA : R DANIEL HARTANTO**

**NPM : P31.2023.02570**

Tesis ini telah diuji dan dipertahankan di hadapan Dewan Penguji pada Sidang Tesis tanggal 22 Mei 2025. Menurut pandangan kami, Tesis ini memadai dari segi kualitas untuk tujuan penganugerahan gelar Magister Komputer (M.Kom)

Semarang, Mei 2025

Dewan Penguji

**Dr. A. Zainul Fanani, SSi, M.Kom**  
Anggota

**Prof. Dr. Aris Marjuni, S.Si., .Kom**  
Anggota

Mengetahui

**Sri Winarno, M.Kom., Ph.D.**  
Dekan

**Dr. Abdul Syukur,**  
Ketua Penguji

## ABSTRACT

The stock market, where shares are bought and sold, plays a vital role in the economy by providing capital access for companies and opportunities for investors to earn profits. The ability to predict stock market trends can offer insights into economic health and guide investment strategies aligned with market conditions.

Several deep learning methods have been developed for prediction, including traditional time series models such as ARIMA (Autoregressive Integrated Moving Average) and SARIMA (Seasonal ARIMA). The effectiveness of ARIMA and SARIMA models is often evaluated against other forecasting techniques, including machine learning models. Research indicates that while traditional time series models provide a solid foundation for forecasting, they may not always outperform more complex models such as Long Short-Term Memory (LSTM) networks or hybrid models that combine multiple data sources.

LSTM networks are particularly effective for time series forecasting due to their ability to learn long-term dependencies in sequential data, which is crucial for predicting stock prices influenced by historical trends. Another relatively simple yet powerful method is the use of deep learning with Gated Recurrent Units (GRU). GRU, a simplified version of LSTM, has also been employed in stock price prediction tasks.

To enhance the performance of GRU, one approach is optimization—specifically, using the ADAM optimizer. Although the ADAM optimizer is widely applied in LSTM models, there is limited literature discussing its application within GRU frameworks. Therefore, this study aims to explore the use of ADAM optimization in Gated Recurrent Unit (GRU) models for stock price prediction.

**Keywords:** Stock, Price, Share, Prediction

## ABSTRAK

Pasar saham, tempat saham dibeli dan dijual, memainkan peran penting dalam perekonomian dengan menyediakan akses modal bagi perusahaan dan peluang bagi investor untuk memperoleh laba. Kemampuan untuk memprediksi tren pasar saham dapat memberikan wawasan tentang kesehatan ekonomi dan memandu strategi investasi yang selaras dengan kondisi pasar. Metode deep learning untuk prediksi ada beberapa model menggunakan model deret waktu tradisional seperti *ARIMA* (*Autoregressive Integrated Moving Average*) dan *SARIMA* (*Seasonal ARIMA*). Efektivitas model ARIMA dan SARIMA sering dievaluasi terhadap teknik peramalan lainnya, termasuk model pembelajaran mesin. Penelitian menunjukkan bahwa meskipun model deret waktu tradisional memberikan dasar yang kuat untuk peramalan, model tersebut mungkin tidak selalu mengungguli model yang lebih kompleks seperti jaringan *LSTM* (*Long Short-Term Memory*) atau model hibrid yang menggabungkan beberapa sumber data jaringan LSTM sangat efektif untuk peramalan deret waktu karena dapat mempelajari ketergantungan jangka panjang dalam data sekuensial, yang sangat penting untuk memprediksi harga saham yang dipengaruhi oleh tren historis. Metode lain yang cukup sederhana namun kuat adalah penggunaan Deep Learning bersama *Gated Recurrent Units* (GRU). GRU, yang merupakan versi sederhana dari LSTM, juga telah digunakan dalam tugas prediksi harga saham. Untuk menaikkan performa GRU, salah satu pendekatan yang bisa digunakan adalah optimasi. Pengoptimal *ADAM*. Meskipun ADAM optimizer banyak digunakan di LSTM, literatur terkait penggunaan ADAM optimizer di GRU sangat sedikit. Oleh karena itu, penulis ingin mengangkat tema **optimasi adam dalam model gated recurrent unit (gru) untuk prediksi saham.**

**Kata kunci:** *Stock, Price, Share, Prediction*

## ACKNOWLEDGMENT

Tesis dengan judul "OPTIMASI ADAM DALAM MODEL GATED RECURRENT UNIT (GRU) UNTUK PREDIKSI SAHAM" ini dapat penulis selesaikan sesuai rencana karena dukungan dari berbagai pihak yang tidak ternilai besarnya. Oleh karena itu penulis menyampaikan terimakasih kepada:

1. Tuhan Yesus.
2. Kedua orang tua saya Alm. Bapak R. Hartono dan Ibu Chotidjah yang selalu memberi doa dan dukungannya.
3. Istri dan anak-anakku tercinta yang selalu mendukung dan mendoakan.
4. Bapak Prof. Dr. Ir. Edi Noersasongko, M.Kom selaku rektor Universitas Dian Nuswantoro.
5. Bapak Dr. Drs. Abdul Syukur selaku Dekan Fakultas Ilmu Komputer Universitas Dian Nuswantoro.
6. Bapak Dr. Arief Soeleman, M.Kom, selaku Kaprodi Fakultas Ilmu Komputer Universitas Dian Nuswantoro.
7. Bapak Dr. Guruh Fajar Shidik S.Kom., M.Cs dan Dr. Farikh Al Zami, M.Kom sebagai dosen dan pembimbing Tesis yang telah memberikan pengetahuan dan bimbingannya dalam penyusunan Tesis ini.
8. Staff pengajar pada Program Studi Magister Teknik Informatika, Universitas Dian Nuswantoro
9. Semua pihak yang telah membantu penulis dan tidak dapat disebutkan satu per satu.

Penulis menyadari adanya keterbatasan penelitian ini, maka kritik, saran, dan masukan yang membangun akan sangat membantu penulis dalam penelitian selanjutnya. Semoga tulisan ini dapat bermanfaat bagi ilmu pengetahuan dan pembaca.

Semarang, Mei 2025



R DANIEL HARTANTO

## DAFTAR ISI

THESIS .....	i
<b>PENGESAHAN STATUS TESIS</b> .....	ii
<b>PERNYATAAN PENULIS</b> .....	iii
<b>PERSETUJUAN TESIS</b> .....	iv
PENGESAHAN THESIS .....	v
ABSTRAK.....	vii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	xiii
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah .....	6
1.3 Tujuan Penelitian .....	7
1.4 Manfaat Penelitian .....	8
1.5 Manfaat Bagi Stakeholder.....	8
1.6 Manfaat Bagi Akademik .....	8
<b>BAB II TINJAUAN PUSTAKA</b> .....	9
2.1 Metode Tinjauan Pustaka.....	9
2.2 Hasil Tinjauan Pustaka.....	9
2.3 Gap pada penelitian terdahulu.....	35
2.4 SOTA ( State of Art ).....	36
2.5 Landasan Teori.....	36
2.5.1 Deep Learning adalah cabang dari Machine Learning .....	36
2.5.2 Konsep Prediksi dalam Deep Learning.....	36
2.5.3 GRU Gated Recurrent Unit ( GRU ).....	42
2.5.4 Optimasi ADAM.....	47

2.5.5	Metrik Evaluasi menggunakan RMSE, MAE, dan R <sup>2</sup> Akurasi .....	51
2.5.6	Prediksi Saham dengan Deep Learning .....	52
2.5.7	Analisis Evaluasi Metode di dalam Peramalan .....	53
BAB III METODE PENELITIAN .....		55
3.1	Metode Penelitian .....	55
3.2	Datashet dan Sumber Data .....	55
3.3	Tahapan Penelitian .....	56
3.4	Tahap Preprocessing Data .....	56
3.5	Arsitektur Model GRU.....	57
3.6	Algoritma Optimasi Adam.....	58
3.7	Evaluasi Model .....	58
3.8	Perangkat Lunak dan Pustaka .....	58
3.9	Jadwal.....	59
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....		60
4.1	Gambaran Umum Eksperimen.....	60
4.2	Akuisisi Data.....	60
4.3	Preprocessing Data.....	60
4.4	Perancangan Model GRU .....	62
4.5	Optimasi menggunakan Adam.....	64
4.6	Pelatihan dan Pengujian Model.....	65
4.7	Evaluasi Kinerja Model .....	65
4.8	Diskusi .....	70
BAB V KESIMPULAN DAN SARAN.....		72
5.1	Kesimpulan .....	72
5.2	Saran.....	73
DAFTAR PUSTAKA .....		74

## DAFTAR GAMBAR

Gambar 2.5.3.1.1 Diagram Alur GRU .....	44
Gambar 2.5.3.3.1Arsitektur dasar Gated Recurrent Unit (GRU).....	46
Gambar 3.5. 1 Arsitektur Model GRU.....	57
Gambar 4.3.1 Arsitektur model GRU dengan ADAM .....	62
Gambar 4.3.3 Output GPU Tesla T4.....	63
Gambar 4.3.4 Membagi dataset menjadi training, validation, dan testing.....	64
Gambar 4.3.5 Kualitas prediksi model.....	66
Gambar 4.3.6 Training MAE dan Validation MAE.....	67
Gambar 4.3.7 Grafik Aktual dan Prediksi.....	67
Gambar 4.3.8 Grafik Aktual dan prediksi 07-02-2020 sampai 07-07- 2020 .....	67
Gambar 4.3.9 Grafik Aktual dan Prediksi 07-02-2020 sampai 29-12-2023 dengan ADAM .....	69

## DAFTAR TABEL

Tabel 2.2.1 Hasil Studi Literatur.....	34
Tabel 2.5.3.2. 1 Perbandingan GRU dan LSTM.....	45
Tabel 3.9. 1 Rencana Jadwal Penelitian.....	59
Tabel 4.1.1 Datashet saham .....	60
Tabel 4.3.1 Pemakaian VRAM pada Google Colab .....	63

## DAFTAR LAMPIRAN

Berita Acara Perbaikan .....	80
Turnitin Originality Report .....	83

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Saham dan obligasi merupakan komponen fundamental pasar keuangan, yang mewakili kepemilikan di suatu perusahaan dan berfungsi sebagai sarana untuk meningkatkan modal. Ketika seorang individu atau entitas membeli saham, mereka memperoleh sebagian kepemilikan di perusahaan penerbit, yang memberi mereka hak atas aset dan laba perusahaan. Saham dikategorikan terutama menjadi dua jenis: saham biasa dan saham preferen. Pemegang saham biasa biasanya memiliki hak suara dan dapat menerima dividen, yang merupakan distribusi sebagian laba perusahaan. Sebaliknya, pemegang saham preferen biasanya tidak memiliki hak suara tetapi memiliki klaim yang lebih tinggi atas aset dan laba, terutama jika terjadi likuidasi [1], [2], [3].

Pasar saham, tempat saham dibeli dan dijual, memainkan peran penting dalam perekonomian dengan menyediakan akses modal bagi perusahaan dan peluang bagi investor untuk memperoleh laba. Kinerja pasar saham dapat memengaruhi pertumbuhan atau penurunan ekonomi secara signifikan, karena mencerminkan sentimen kolektif dan ekspektasi investor mengenai kinerja perusahaan dan kondisi ekonomi di masa mendatang [4], [5]. Volatilitas harga saham dipengaruhi oleh berbagai faktor, termasuk kinerja perusahaan, indikator ekonomi makro, dan sentimen investor, sehingga prediksi harga saham menjadi tugas yang rumit dan menantang [6], [7], [8] (Yuan et al., 2020; Idrees et al., 2019; Hsu et al., 2022).

Perkiraan harga saham yang akurat tidak hanya membantu investor individu dalam membuat keputusan yang tepat tetapi juga membantu pembuat kebijakan dan analis keuangan dalam memahami tren ekonomi yang lebih luas [4], [9]. Misalnya, kemampuan untuk memprediksi tren pasar saham dapat memberikan wawasan tentang kesehatan ekonomi dan memandu strategi investasi yang selaras dengan kondisi pasar [4]. Kemampuan prediktif ini sangat penting di saat ketidakpastian ekonomi, di mana informasi yang tepat waktu dan akurat dapat menghasilkan hasil keuangan yang lebih baik.

Tren penelitian dalam saham dan obligasi, khususnya menggunakan model deret waktu tradisional seperti ARIMA (Autoregressive Integrated Moving Average) dan SARIMA (Seasonal ARIMA), telah mendapatkan perhatian signifikan dalam domain analitik keuangan. Model-model ini merupakan dasar untuk memahami dan memprediksi pergerakan harga saham karena kemampuannya untuk menangkap ketergantungan temporal dalam data historis.

Model ARIMA dan SARIMA khususnya efektif untuk peramalan deret waktu univariat, yang berfokus pada nilai historis satu variabel untuk memprediksi nilai masa depan. Model ARIMA dicirikan oleh tiga komponennya: autoregresi (AR), differencing (I), dan moving average (MA), yang bersama-sama membantu dalam pemodelan pola dasar dalam harga saham [10]. SARIMA memperluasnya dengan menggabungkan efek musiman, sehingga cocok untuk kumpulan data yang menunjukkan tren musiman, seperti laporan laba triwulanan atau pola penjualan musiman [7], [11].

Beberapa studi melakukan integrasi ARIMA dan SARIMA dengan metodologi lain untuk meningkatkan akurasi prediksi. Misalnya, pendekatan hibrida yang menggabungkan ARIMA dengan teknik pembelajaran mesin telah dieksplorasi untuk memanfaatkan kemampuan pemodelan linier ARIMA dan kekuatan prediksi non-linier dari algoritma pembelajaran mesin [12]. Tren ini mencerminkan gerakan yang lebih luas menuju metode ensemble yang menggabungkan teknik statistik tradisional dengan pendekatan komputasi modern untuk meningkatkan kinerja peramalan [13], [14].

Selain itu, penerapan model ARIMA dan SARIMA dalam konteks peramalan volatilitas menjadi bidang penelitian yang signifikan. Volatilitas merupakan faktor penting dalam pasar keuangan, yang memengaruhi keputusan investasi dan strategi manajemen risiko. Studi telah menunjukkan bahwa mengintegrasikan model berbasis ARIMA dengan model GARCH (Generalized Autoregressive Conditional Heteroskedasticity) dapat secara efektif menangkap perubahan varians dalam harga saham, sehingga memberikan pandangan yang lebih komprehensif tentang dinamika pasar [6], [15].

Efektivitas model ARIMA dan SARIMA sering dievaluasi terhadap teknik peramalan lainnya, termasuk model pembelajaran mesin. Penelitian menunjukkan bahwa meskipun model deret waktu tradisional memberikan dasar yang kuat untuk peramalan, model tersebut mungkin tidak selalu mengungguli model yang lebih kompleks seperti jaringan LSTM (Long Short-Term Memory) atau model hibrida yang menggabungkan beberapa sumber data [4], [16].

Salah satu keuntungan utama pembelajaran mendalam (deep learning) dalam prediksi pasar saham adalah kemampuannya untuk menangkap pola kompleks dalam kumpulan data besar. Misalnya, jaringan LSTM sangat efektif untuk peramalan deret waktu karena dapat mempelajari ketergantungan jangka panjang dalam data sekuensial, yang sangat penting untuk memprediksi harga saham yang dipengaruhi oleh tren historis [17], [18] (Nabipour et al., 2020) Wang, 2023). Studi telah menunjukkan bahwa model LSTM dapat mengungguli metode statistik tradisional, seperti ARIMA, terutama dalam menangkap volatilitas dan non-stasioneritas harga saham [3], [14]. Selain itu, model hibrida yang menggabungkan pembelajaran mendalam dengan teknik lain telah dikembangkan untuk lebih meningkatkan akurasi prediksi. Misalnya, mengintegrasikan LSTM dengan Convolutional Neural Networks (CNN) memungkinkan ekstraksi fitur spasial dari grafik harga saham sekaligus menangkap ketergantungan temporal [19].

Metode lain yang cukup sederhana namun kuat adalah penggunaan Deep Learning bersama Gated Recurrent Units (GRU). GRU, yang merupakan versi sederhana dari LSTM, juga telah digunakan dalam tugas prediksi harga saham. GRU menawarkan kinerja yang sebanding dengan LSTM sekaligus lebih efisien secara komputasi karena kompleksitasnya yang lebih rendah [18]. Studi telah menunjukkan bahwa GRU dapat secara efektif memodelkan pergerakan harga saham, terutama jika dikombinasikan dengan teknik lain seperti ensemble learning dan metode pemilihan fitur [14].

Untuk menaikkan performa GRU, salah satu pendekatan yang bisa digunakan adalah optimasi. Pengoptimal ADAM, kependekan dari Adaptive Moment Estimation, telah menjadi pilihan populer untuk melatih model

pembelajaran mendalam, termasuk yang digunakan untuk prediksi harga saham. Efektivitasnya berasal dari kemampuannya untuk menyesuaikan laju pembelajaran secara adaptif untuk setiap parameter berdasarkan momen pertama dan kedua gradien, yang membantu dalam mencapai konvergensi yang lebih cepat dan kinerja yang lebih baik dalam melatih model kompleks seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Units (GRU) [20]. Dalam konteks peramalan harga saham, pengoptimal ADAM telah terbukti mengungguli metode pengoptimalan tradisional seperti stochastic gradient descent (SGD). Hal ini sangat penting dalam aplikasi keuangan di mana data dapat berisik dan sangat fluktuatif. Misalnya, Quadir et al. menunjukkan bahwa penggunaan ADAM dengan model LSTM sekuensial berlapis-lapis secara signifikan meningkatkan akurasi prediksi harga saham dibandingkan dengan model yang dilatih dengan SGD [20]. Demikian pula, Mustaffa dan Sulaiman menyoroti bahwa mengintegrasikan ADAM dengan model hibrida yang menggabungkan berbagai teknik pengoptimalan menghasilkan peningkatan kinerja prediktif dalam analisis harga saham [21].

Selain itu, kemampuan pengoptimal ADAM untuk menangani gradien yang jarang (sparse) membuatnya sangat cocok untuk kumpulan data keuangan, yang sering kali berisi banyak fitur yang tidak relevan. Karakteristik ini memungkinkan model untuk fokus pada aspek data yang paling informatif, sehingga meningkatkan akurasi prediksi secara keseluruhan [22]. Selain itu, tingkat pembelajaran adaptif pengoptimal membantu mengurangi masalah yang terkait dengan overfitting, yang merupakan tantangan umum dalam prediksi pasar saham karena sifat data keuangan yang kompleks [1]. Penelitian oleh Mu et al. menunjukkan bahwa mengoptimalkan hiperparameter LSTM menggunakan ADAM menghasilkan kinerja yang lebih unggul dalam memprediksi harga saham jika dikombinasikan dengan analisis sentiment [23].

Gated Recurrent Unit (GRU) memegang peran penting dalam penelitian ini karena memiliki kemampuan untuk memproses data sekuensial secara efisien. GRU menggunakan dua gerbang utama, yaitu **update gate** dan **reset gate**, untuk mengontrol informasi yang perlu disimpan atau dilupakan pada setiap langkah waktu.

Dalam konteks prediksi harga saham, yang sangat bergantung pada pola historis dan tren jangka panjang, GRU memungkinkan model untuk tetap fokus pada informasi yang relevan tanpa kehilangan konteks data sebelumnya. Keunggulan GRU dibanding LSTM terletak pada struktur yang lebih ringan dan efisiensi pelatihan yang lebih tinggi, tanpa kehilangan akurasi yang signifikan.

Ketika dipasangkan dengan ADAM Optimizer, GRU menjadi lebih stabil dalam pelatihan bahkan pada data saham yang fluktuatif dan noisy. Kombinasi ini memungkinkan model untuk mempelajari pola pasar dengan lebih baik dan mempercepat proses konvergensi menuju prediksi yang akurat. di antaranya:

#### 1. **Kombinasi Momentum dan RMSProp**

Adam menggabungkan keunggulan dari **Momentum** dan **RMSProp**, sehingga dapat menangani gradien yang bervariasi secara adaptif. Dalam prediksi saham, data sering kali sangat fluktuatif, sehingga Adam membantu model menyesuaikan pembaruan bobot dengan lebih stabil.

#### 2. **Penyesuaian Learning Rate Secara Otomatis**

Optimasi Adam menggunakan **learning rate adaptif**, yang berarti dapat menyesuaikan laju pembelajaran berdasarkan sejarah gradien. Ini sangat berguna untuk **time series stock prediction**, di mana tren pasar bisa berubah dengan cepat.

#### 3. **Mengatasi Vanishing Gradient**

Dalam model **GRU**, yang merupakan jenis Recurrent Neural Network (RNN), ada risiko **vanishing gradient**, terutama saat menangani data time series yang panjang. Adam membantu **menjaga stabilitas pembelajaran**, sehingga model tidak mengalami kesulitan dalam menangkap pola jangka panjang.

#### 4. **Konvergensi Lebih Cepat**

Adam biasanya **lebih cepat mencapai konvergensi** dibandingkan optimasi seperti **SGD (Stochastic Gradient Descent)**. Dalam prediksi saham, efisiensi ini penting karena model sering diperbarui dengan data terbaru.

## 5. Cocok untuk Data yang Noisy

Data pasar saham sering kali memiliki noise yang tinggi. Adam bekerja dengan baik dalam lingkungan seperti ini karena **mempertahankan perhitungan gradien yang stabil** tanpa terlalu terpengaruh oleh fluktuasi sesaat.

Optimasi Adam dipilih dalam model **GRU untuk prediksi saham** karena:

- Stabil dalam menangani data time series yang fluktuatif
- Menyesuaikan learning rate secara otomatis
- Mengurangi risiko vanishing gradient
- Mempercepat proses konvergensi
- Lebih tahan terhadap noise dalam data

### 1.2 Rumusan Masalah

Dalam penelitian ini, metode prediksi yang digunakan adalah pendekatan deep learning berbasis Gated Recurrent Unit (GRU). GRU merupakan jenis Recurrent Neural Network (RNN) yang dirancang untuk mengatasi masalah dependensi jangka panjang dalam data deret waktu tanpa kompleksitas tinggi seperti LSTM.

Permasalahan utama dalam penelitian ini adalah kurangnya eksplorasi penggunaan ADAM Optimizer secara spesifik dalam model GRU untuk prediksi harga saham. Padahal, ADAM memiliki potensi untuk meningkatkan performa prediksi dalam data yang bersifat non-stasioner dan volatil, seperti pasar saham.

Proses prediksi dimulai dengan pengumpulan data harga saham historis, yang kemudian diproses menjadi format time series. Model GRU dilatih menggunakan algoritma optimasi ADAM (Adaptive Moment Estimation) yang mampu menyesuaikan learning rate secara adaptif berdasarkan gradien parameter.

Tujuan penggunaan kombinasi GRU dan ADAM adalah untuk menghasilkan model prediksi yang lebih akurat, stabil, dan efisien dalam menangani fluktuasi harga saham. Evaluasi dilakukan menggunakan metrik seperti Mean Squared Error (MSE) dan Mean Absolute Error (MAE), serta dibandingkan dengan baseline model seperti ARIMA dan LSTM.

1. Efisien secara komputasi  
Lebih cepat dilatih dibanding LSTM dan Transformer.
2. Akurasi tinggi untuk prediksi deret waktu  
Mendekati atau menyamai LSTM dalam banyak kasus.
3. Sederhana dan ringan  
Lebih sedikit parameter daripada LSTM, memudahkan tuning dan deployment.
4. Menangani data urut (sequential) lebih baik dibanding ANN atau CNN biasa.
5. Fleksibel  
Bisa digunakan mandiri, atau sebagai bagian dari model hybrid (misal CNN-GRU, GRU-Attention, Encoder-Decoder GRU).

Pada penelitian sebelumnya, telah diterapkan model LSTM; namun, penggunaan ADAM Optimizer dalam prediksi belum dieksplorasi. Berdasarkan hal tersebut, rumusan masalah yang akan diteliti adalah: 'Bagaimana peningkatan performa algoritma Deep Learning berbasis Gated Recurrent Unit dengan memanfaatkan ADAM Optimizer untuk prediksi saham.

### **1.3 Tujuan Penelitian**

Metode GRU dipilih karena menawarkan keseimbangan antara akurasi dan efisiensi. GRU dapat menangani dependensi jangka panjang seperti LSTM, namun dengan struktur yang lebih ringan dan jumlah parameter lebih sedikit, sehingga lebih cepat dikonvergensi dan lebih hemat sumber daya.

1. Menganalisis dan mengevaluasi peningkatan performa algoritma Deep Learning yang berbasis Gated Recurrent Unit dengan penerapan ADAM Optimizer dalam proses prediksi saham.
2. Memberikan pemahaman yang lebih mendalam tentang efektivitas ADAM Optimizer dalam meningkatkan akurasi dan efisiensi model dalam prediksi finansial.

Tujuan penelitian ini adalah “Meningkatnya performa algoritma Deep Learning berbasis Gated Recurrent Unit menggunakan ADAM Optimizer untuk prediksi saham.

#### **1.4 Manfaat Penelitian**

Manfaat penelitian dari tujuan diatas adalah :

1. Penggunaan Model Gated Recurrent Unit menggunakan ADAM Optimizer yang ditingkatkan performanya untuk prediksi saham dapat bermanfaat untuk membantu untuk memprediksi harga saham.
2. Penelitian ini dapat memberikan kontribusi pada pengembangan teknologi *deep learning* dibidang ekonomi, khususnya dalam membantu pembuat kebijakan dan analis keuangan.

#### **1.5 Manfaat Bagi Stakeholder**

Adapun manfaat bagi stackholder adalah sebagai berikut :

1. Dengan peningkatan performa algoritma Deep Learning dapat mempercepat pengambilan keputusan investasi dan strategi manajemen risiko merupakan faktor penting dalam pasar keuangan,.
2. Dengan penerapan teknologi GRU dengan meggunakan *optimasi Adam* , membantu investor individu dalam membuat keputusan yang tepat tetapi juga membantu pembuat kebijakan dan analis keuangan.

#### **1.6 Manfaat Bagi Akademik**

Manfaat penelitian ini bagi akademis adalah

1. Penelitian ini dapat berfungsi sebagai referensi penting untuk pengembangan lebih lanjut dalam penggunaan AI di bidang Ekonomi dan membuka peluang untuk penelitian pengembangan antara ilmu komputer, Ekonomi , dan investasi.
2. Selain itu, penelitian ini dapat meningkatkan pemahaman tentang hubungan antara teknologi dan ekonomi dan mendorong pengembangan algoritma *deep learning* untuk diterapkan pada berbagai instrumen ekonomi.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Metode Tinjauan Pustaka**

Metode pencarian tinjauan pustaka yang dilakukan oleh peneliti adalah menggunakan sumber dari sciencedirect dan IEEE XPlore, kemudian tahun literatur dari 2019 hingga 2025 dan dalam bentuk open access. Kata kunci yang dipakai adalah: “stock”, “share”, “price”, “prediction”. Dari kedua sumber itu, peneliti mendapatkan 199 Judul. Peneliti melihat beberapa topik metode muncul, antara lain deep learning (RNN, GRU, LSTM), machine learning (Bayes, SVM, Decision Tree), ARIMA, SARIMA.

#### **2.2 Hasil Tinjauan Pustaka**

Berdasarkan studi literatur didapatkan metode fusi Menurut Lee [2] " membahas tantangan dalam memprediksi arah pergerakan harga saham dengan mengintegrasikan berbagai modalitas data untuk meningkatkan akurasi klasifikasi.

**Masalah:** Prediksi harga saham yang hanya mengandalkan data historis sering kali kurang akurat karena sifat data yang acak dan volatil. Penelitian ini bertujuan untuk meningkatkan kinerja prediksi dengan menambahkan indikator makroekonomi dan informasi temporal sebagai sumber data tambahan.

**Dataset:** Studi ini menggunakan data dari 50 saham yang mencakup harga saham historis, indikator makroekonomi, serta informasi temporal seperti bulan dan hari dalam seminggu.

**Preprocessing:** Metode yang digunakan adalah multimodal early fusion, di mana fitur dari berbagai modalitas—harga saham, indikator makroekonomi, dan informasi temporal—digabungkan menjadi satu input terpadu. Pendekatan ini memungkinkan model untuk mempelajari korelasi antar-modalitas dari fitur-fitur tersebut.

**Model:** Model utama yang diusulkan adalah Multimodal Fusion Transformer. Arsitektur ini memanfaatkan encoder transformer untuk memproses fitur yang telah digabungkan, sehingga mampu menangkap hubungan dan dependensi

kompleks antar data. Setelah itu, lapisan klasifikasi digunakan untuk memprediksi arah pergerakan harga saham.

**Hasil:** Model yang diusulkan menunjukkan kinerja yang lebih baik dibandingkan model pembandingan, dengan mencapai akurasi klasifikasi yang lebih tinggi pada 27 dari 50 saham yang dianalisis. Selain itu, strategi early fusion terbukti lebih unggul dibandingkan late fusion dalam 30 dari 50 dataset, menunjukkan efektivitas penggabungan data pada tahap awal untuk prediksi harga saham.

Secara keseluruhan, penelitian ini menyoroti manfaat dari integrasi indikator makroekonomi dan informasi temporal ke dalam model prediksi saham, serta menunjukkan efektivitas Multimodal Fusion Transformer dalam menangkap hubungan kompleks antar data untuk meningkatkan akurasi klasifikasi.

Metode prediksi pergerakan harga saham dengan menggabungkan data harga saham dan berita menggunakan modul campuran informasi hibrida.[3]

**Masalah:** Pasar saham dipengaruhi oleh berbagai variabel, dan volatilitas pasar harus dipertimbangkan dalam memprediksi pergerakan harga saham. Karena pasar saham bersifat efisien, berbagai informasi dengan cepat tercermin dalam harga saham. Oleh karena itu, tantangan utama adalah bagaimana mengintegrasikan berbagai sumber informasi untuk meningkatkan akurasi prediksi

**Dataset:** Penelitian ini menggunakan data harga saham dan data berita terkait. Namun, detail spesifik mengenai sumber, periode waktu, dan jumlah data yang digunakan tidak disebutkan dalam abstrak yang tersedia.

**Preprocessing:** Fitur dari data harga saham dan data teks berita diekstraksi dan dikombinasikan untuk menciptakan campuran informasi baru. Tujuannya adalah untuk mengekstrak interaksi multimodal antara fitur deret waktu dari data harga dan fitur semantik dari data teks. Detail spesifik mengenai teknik preprocessing tidak dijelaskan dalam abstrak.

**Model:** Model yang diusulkan adalah modul campuran informasi hibrida berbasis multilayer perceptron (MLP). Modul ini dirancang menggunakan dua blok peta untuk interaksi efektif antara fitur dari data harga dan data teks. Model ini menggabungkan fitur deret waktu dari data harga saham dan fitur

semantik dari data berita untuk memprediksi pergerakan harga saham. Teknologi seperti Bidirectional Encoder Representations from Transformer (BERT) dan Gated Recurrent Units (GRU) disebutkan sebagai kata kunci, yang menunjukkan kemungkinan penggunaannya dalam ekstraksi fitur atau pemodelan.

**Hasil:** Kinerja model dievaluasi menggunakan metrik akurasi, koefisien korelasi Matthews (MCC), dan skor F1. Namun, nilai spesifik dari metrik-metrik ini tidak disertakan dalam abstrak yang tersedia. Oleh karena itu, meskipun disebutkan bahwa modul campuran informasi hibrida diterapkan untuk memprediksi pergerakan harga saham dan dievaluasi kinerjanya, detail hasil spesifik tidak dapat disampaikan berdasarkan informasi yang tersedia.

Penelitian berfokus pada analisis data deret waktu pasar saham India untuk membangun model statistik yang dapat memprediksi volatilitas pasar saham secara efisien.[7]

**Masalah:** Pasar saham dikenal sebagai sistem keuangan yang sangat kompleks dengan harga saham yang berfluktuasi seiring waktu. Tujuan utama dari prediksi tren pasar saham adalah mengembangkan pendekatan inovatif untuk memprediksi saham yang dapat memberikan keuntungan tinggi, sehingga investor dapat memaksimalkan pengembalian investasi dan meminimalkan risiko terkait.

**Dataset:** Penelitian ini menggunakan data pasar saham India, khususnya data terkait indeks Sensex dan Nifty. Data ini mencakup urutan data atau serangkaian titik data yang diambil pada interval waktu yang sama. Namun, detail spesifik mengenai periode waktu dan frekuensi pengambilan data tidak disebutkan.

**Preprocessing:** Tahapan preprocessing melibatkan analisis data deret waktu untuk memastikan stasioneritas data. Jika data tidak stasioner, proses differencing diterapkan untuk membuatnya stasioner. Selain itu, komponen auto-regressive (AR) dan moving average (MA) diidentifikasi untuk model ARIMA. Namun, detail spesifik mengenai teknik preprocessing tidak dijelaskan.

**Model:** Model yang digunakan dalam penelitian ini adalah Autoregressive Integrated Moving Average (ARIMA). Model ARIMA menggabungkan tiga metode dasar: Auto-Regressive (AR), Differencing (I untuk Integrated), dan Moving Average (MA). Komponen AR mengindikasikan bahwa nilai saat ini dari deret waktu berkorelasi dengan nilai sebelumnya. Komponen differencing digunakan untuk membuat deret waktu menjadi stasioner. Komponen MA menunjukkan bahwa kesalahan prediksi saat ini berkaitan dengan kesalahan prediksi sebelumnya.

**Hasil:** Penelitian ini berhasil menganalisis data deret waktu dari pasar saham India dan membangun model statistik yang dapat memprediksi volatilitas pasar saham secara efisien. Namun, detail spesifik mengenai kinerja model, seperti metrik evaluasi atau perbandingan dengan model lain, tidak disediakan dalam abstrak yang tersedia.

Penelitian oleh F. Wang, [10] berfokus pada penerapan teknik penambangan data deret waktu dalam konteks analitik data besar.

**Masalah:** Penelitian ini menyoroti pentingnya deret waktu dalam data besar dan bagaimana penambangan data deret waktu dapat digunakan untuk memprediksi tren masa depan.

**Dataset:** Penelitian ini menggunakan data deret waktu yang besar, meskipun detail spesifik mengenai sumber dan jenis data tidak disebutkan dalam abstrak yang tersedia.

**Preprocessing:** Tahapan preprocessing melibatkan analisis data deret waktu untuk memastikan stasioneritas data. Jika data tidak stasioner, proses differencing diterapkan untuk membuatnya stasioner. Selain itu, komponen auto-regressive (AR) dan moving average (MA) diidentifikasi untuk model ARIMA.

**Model:** Model yang digunakan dalam penelitian ini adalah Autoregressive Integrated Moving Average (ARIMA). Model ARIMA menggabungkan tiga metode dasar: Auto-Regressive (AR), Differencing (I untuk Integrated), dan Moving Average (MA). Komponen AR mengindikasikan bahwa nilai saat ini dari deret waktu berkorelasi dengan nilai sebelumnya. Komponen differencing digunakan untuk membuat deret waktu menjadi stasioner. Komponen MA

menunjukkan bahwa kesalahan prediksi saat ini berkaitan dengan kesalahan prediksi sebelumnya.

**Hasil:** Penelitian ini berhasil menganalisis data deret waktu yang besar dan membangun model statistik yang dapat memprediksi tren masa depan secara efisien. Namun, detail spesifik mengenai kinerja model, seperti metrik evaluasi atau perbandingan dengan model lain, tidak disediakan dalam abstrak yang tersedia.

Penelitian yang dilakukan oleh Y. Wang [12] berfokus pada pengembangan model hibrida untuk memprediksi volatilitas pasar saham.

**Masalah:** Prediksi volatilitas pasar saham merupakan tantangan signifikan dalam bidang keuangan karena kompleksitas dan dinamika pasar yang tinggi. Tujuan utama penelitian ini adalah mengembangkan model yang mampu memprediksi volatilitas saham dengan akurasi tinggi, sehingga dapat membantu investor dan analis dalam pengambilan keputusan investasi.

**Dataset:** Penelitian ini menggunakan data historis pasar saham, termasuk harga penutupan harian dan variabel terkait lainnya. Namun, detail spesifik mengenai sumber data, periode waktu, dan cakupan geografis tidak disebutkan dalam abstrak yang tersedia.

**Preprocessing:** Tahapan preprocessing dalam penelitian ini melibatkan normalisasi data untuk memastikan bahwa semua variabel berada dalam skala yang sama, yang penting untuk konvergensi model neural network. Selain itu, data dibagi menjadi set pelatihan dan pengujian untuk mengevaluasi kinerja model. Detail spesifik mengenai teknik preprocessing lainnya tidak dijelaskan dalam abstrak.

**Model:** Model yang diusulkan adalah jaringan neural hibrid yang menggabungkan Long Short-Term Memory (LSTM) dan jaringan neural konvensional. LSTM digunakan untuk menangkap dependensi jangka panjang dalam data deret waktu, sementara jaringan neural konvensional digunakan untuk memproses fitur tambahan yang mungkin tidak bersifat deret waktu. Kombinasi ini bertujuan untuk meningkatkan kemampuan model dalam memprediksi volatilitas saham.

**Hasil:** Model hibrida yang diusulkan menunjukkan kinerja yang lebih baik dalam memprediksi volatilitas pasar saham dibandingkan dengan model tradisional. Namun, nilai spesifik dari metrik evaluasi seperti akurasi, Mean Squared Error (MSE), atau lainnya tidak disertakan dalam abstrak yang tersedia. Oleh karena itu, detail kuantitatif mengenai peningkatan kinerja tidak dapat disampaikan berdasarkan informasi yang ada.

Makalah ini mengusulkan algoritma baru yang menggabungkan rekonstruksi sekuens berbasis motif dengan jaringan saraf konvolusional (CNN) untuk prediksi tren deret waktu saham. Metode yang diusulkan meningkatkan akurasi prediksi di hampir semua kumpulan data, mengungguli metode pemrosesan sinyal tradisional dan pendekatan pembelajaran mesin sebesar 4% -7% dalam akurasi. [13] berfokus pada pengembangan metode baru untuk meramalkan tren pasar saham dengan memanfaatkan informasi tingkat tinggi dari data deret waktu.

**Masalah:** Memprediksi tren pasar saham merupakan tantangan signifikan karena sifat pasar yang sangat volatil dan non-stasioner. Penelitian ini bertujuan untuk mengatasi tantangan tersebut dengan mengembangkan metode yang dapat menyederhanakan deret waktu keuangan yang penuh dengan noise dan meningkatkan akurasi prediksi tren pasar saham. [ResearchGate](#)

**Dataset:** Penelitian ini menggunakan data historis dari indeks S&P 500. Namun, detail spesifik mengenai rentang waktu dan frekuensi pengambilan data tidak disebutkan dalam sumber yang tersedia.

**Preprocessing:** Langkah awal dalam penelitian ini melibatkan rekonstruksi urutan data dengan memanfaatkan motif, yaitu pola yang sering muncul dalam deret waktu keuangan. Pendekatan ini bertujuan untuk menyederhanakan deret waktu yang penuh dengan noise, sehingga memudahkan proses analisis dan prediksi selanjutnya. [ResearchGate](#)

**Model:** Setelah proses rekonstruksi urutan, penelitian ini menggunakan Convolutional Neural Network (CNN) untuk menangkap struktur spasial dari deret waktu yang telah disederhanakan. CNN dipilih karena kemampuannya dalam mengenali pola dan fitur kompleks dalam data.

**Hasil:** Metode yang diusulkan menunjukkan peningkatan akurasi prediksi sebesar 4% hingga 7% dibandingkan dengan metode tradisional dalam pemrosesan sinyal dan pendekatan deep learning yang memodelkan pola perdagangan frekuensi tinggi. Hasil ini menunjukkan efektivitas pendekatan yang menggabungkan rekonstruksi urutan berbasis motif dengan penggunaan CNN dalam memprediksi tren pasar saham.

Menguji machine learning tradisional dan deep learning Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Penelitian ini menemukan bahwa untuk data kontinu, model Recurrent Neural Network (RNN) dan Long Short-Term Memory (LSTM) secara signifikan mengungguli model prediksi lain, mencapai kinerja prediksi sekitar 86% dengan perbedaan yang cukup besar dibandingkan dengan yang lain.[17] berfokus pada perbandingan kinerja berbagai algoritma pembelajaran mesin dan pembelajaran mendalam dalam memprediksi tren pasar saham dengan menggunakan data kontinu dan biner.

**Masalah:** Prediksi pergerakan pasar saham merupakan tantangan signifikan bagi investor karena berbagai faktor yang mempengaruhi volatilitas pasar. Tujuan utama penelitian ini adalah mengurangi risiko dalam prediksi tren pasar saham dengan memanfaatkan algoritma pembelajaran mesin dan pembelajaran mendalam.

**Dataset:** Penelitian ini menggunakan data historis selama sepuluh tahun dari empat kelompok saham di Bursa Efek Tehran, yaitu keuangan terdiversifikasi, perminyakan, mineral non-logam, dan logam dasar. Sepuluh indikator teknis dipilih sebagai fitur input untuk model prediksi. [ResearchGate](#)

**Preprocessing:** Data yang dikumpulkan diolah menjadi dua bentuk: data kontinu dan data biner. Data kontinu mempertahankan nilai numerik asli dari indikator teknis, sedangkan data biner mengubah nilai-nilai ini menjadi format biner berdasarkan ambang batas tertentu untuk menunjukkan peningkatan atau penurunan.

**Model:** Penelitian ini membandingkan sembilan model pembelajaran mesin— Decision Tree, Random Forest, Adaptive Boosting (Adaboost), eXtreme Gradient Boosting (XGBoost), Support Vector Classifier (SVC), Naïve Bayes,

K-Nearest Neighbors (KNN), Logistic Regression, dan Artificial Neural Network (ANN)—serta dua metode pembelajaran mendalam yang kuat, yaitu Recurrent Neural Network (RNN) dan Long Short-Term Memory (LSTM).

**Hasil:** Hasil eksperimen menunjukkan bahwa penggunaan data biner secara signifikan meningkatkan kinerja model dibandingkan dengan data kontinu. Di antara semua algoritma yang diuji, metode pembelajaran mendalam seperti RNN dan LSTM menunjukkan performa superior dalam memprediksi tren pasar saham.

Model Memori Jangka Panjang dan Pendek (LSTM) sekuensial berlapis-lapis, yang merupakan jenis jaringan saraf berulang yang sangat cocok untuk prediksi deret waktu. Hasilnya menunjukkan bahwa pendekatan ini secara signifikan meningkatkan akurasi peramalan dibandingkan dengan metode tradisional. Model tersebut menunjukkan metrik kinerja yang ditingkatkan, termasuk kesalahan absolut rata-rata yang lebih rendah dan nilai R-kuadrat yang lebih tinggi, yang menunjukkan kecocokan yang lebih baik dengan data. Selain itu, arsitektur berlapis-lapis memungkinkan untuk menangkap pola kompleks dalam pergerakan harga saham, yang mengarah pada prediksi yang lebih andal. Secara keseluruhan, temuan tersebut menunjukkan bahwa metode yang diusulkan merupakan kontribusi yang berharga bagi bidang peramalan keuangan.[20] berfokus pada pengembangan metode optimasi baru untuk meramalkan harga saham dengan memanfaatkan model Multi-Layer Sequential Long Short-Term Memory (MLS-LSTM).

**Masalah:** Memprediksi harga saham secara akurat merupakan tantangan signifikan dalam bidang keuangan karena sifat pasar saham yang volatil dan kompleks. Penelitian ini bertujuan untuk meningkatkan akurasi prediksi harga saham dengan mengatasi keterbatasan model prediksi sebelumnya, khususnya dalam memahami dependensi jangka panjang dalam data deret waktu.

**Dataset:** Penelitian ini menggunakan data deret waktu historis dari indeks harga saham. Namun, detail spesifik mengenai sumber data, rentang waktu, dan frekuensi pengambilan data tidak disebutkan dalam sumber yang tersedia.

**Preprocessing:** Data yang dikumpulkan dinormalisasi dan dibagi menjadi langkah-langkah waktu (time steps) untuk mengevaluasi hubungan antara nilai historis dan nilai masa depan. Proses ini membantu model dalam memahami pola dan tren dalam data deret waktu.

**Model:** Penelitian ini mengusulkan penggunaan model MLS-LSTM yang dioptimalkan dengan algoritma Adam. LSTM dipilih karena kemampuannya dalam menangkap dependensi jangka panjang dalam data deret waktu, sementara penggunaan struktur multi-layer dan pendekatan sekuensial bertujuan untuk meningkatkan kapasitas model dalam memahami pola kompleks dalam data.

**Hasil:** Model MLS-LSTM yang diusulkan menunjukkan performa tinggi dengan akurasi pelatihan sebesar 96% dan akurasi pengujian sebesar 98%. Hasil ini menunjukkan peningkatan signifikan dibandingkan dengan model prediksi lainnya dalam memprediksi harga saham.

hasil studi literatur ini bisa juga disajikan melalui tabel berikut:

Nama Peneliti	Problem	Datashet	Metode	Hasil
Lee [2]	<ul style="list-style-type: none"> <li>➤ Minimnya studi yang menguji efektivitas data fusion dan optimalisasi penggunaan Transformer dalam pasar saham.</li> </ul>	<ul style="list-style-type: none"> <li>➤ Data Harga Saham</li> <li>➤ Indikator Makroekonomi:</li> <li>➤ Informasi Kalender:</li> </ul>	<p>menggunakan metode fusi awal multimoda untuk memprediksi harga saham dengan mengintegrasikan berbagai modalitas data.</p>	<p>Dengan dataset china, Analisis menunjukkan bahwa model tersebut mencapai akurasi klasifikasi yang lebih tinggi pada 27 dari 50 saham saat diuji pada kumpulan data ini.</p>
Choi [3]	<p><b>1. Keterbatasan dalam Menggabungkan Data Multimodal:</b></p> <p>Banyak model prediksi saham sebelumnya hanya menggunakan satu jenis data, seperti data teknikal atau data teks, tanpa menggabungkan keduanya secara efektif.</p> <p><b>2. Keterbatasan Model Tradisional:</b></p> <p>Model seperti LSTM atau GRU sering mengalami kesulitan dalam mengolah</p>	<p>Penelitian ini memanfaatkan data dari dua sumber utama untuk membangun model prediksi:</p> <p><b>1. Data Harga Saham:</b></p> <ul style="list-style-type: none"> <li>○ Informasi seperti harga pembukaan, harga tertinggi, harga terendah, volume perdagangan, dan harga penutupan disesuaikan.</li> <li>○ Fitur teknikal tambahan seperti perubahan harga harian, moving average 10 hari, weighted 10-</li> </ul>	<p>Metodologi ini melibatkan ekstraksi interaksi multimoda antara fitur deret waktu data harga dan fitur semantik data teks.</p> <p>Model berbasis multilayer perceptron digunakan untuk melakukan eksperimen prediksi fluktuasi harga di pasar saham yang sangat fluktuatif.</p>	<p>Dengan dataset china, pendekatan hybrid meningkatkan akurasi prediksi dibandingkan dengan metode tradisional</p>

	<p>dan mengintegrasikan berbagai jenis data secara optimal.</p> <p><b>3. Kurangnya Pendekatan yang Optimal dalam Data Fusion:</b></p> <ul style="list-style-type: none"> <li>o Tidak ada pendekatan yang secara universal lebih unggul dalam menggabungkan data multimodal, sehingga diperlukan eksperimen lebih lanjut untuk menentukan strategi terbaik.</li> </ul>	<p>day moving average, dan momentum.</p> <p><b>2. Data Teks:</b></p> <p>Informasi dari berita keuangan, laporan analis, dan sumber teks lainnya yang berkaitan dengan pasar saham.</p>		
Idrees [7]	<p><b>1. Kompleksitas dan Ketidakpastian Pasar Saham:</b></p> <ul style="list-style-type: none"> <li>• Pasar saham dikenal sangat kompleks dan rentan terhadap fluktuasi yang cepat, membuat prediksi volatilitasnya menjadi tantangan signifikan.</li> </ul>	<p>Penelitian ini menggunakan data deret waktu dari pasar saham India, khususnya indeks NIFTY dan SENSEX, untuk menganalisis dan memprediksi volatilitas pasar saham. Data ini mencakup informasi seperti harga pembukaan, harga tertinggi, harga terendah, harga</p>	<p>menggunakan model Auto-Regressive Integrated Moving Average (ARIMA) untuk prediksi volatilitas pasar saham, yang menggabungkan tiga metode dasar: Auto-Regressive (AR), Differencing (I), dan Moving Average (MA).</p>	<p>Dengan dataset Sensex and Nifty, menghasilkan prediksi yang cukup presisi</p>

	<p><b>2. Keterbatasan Model Prediksi Tradisional:</b></p> <ul style="list-style-type: none"> <li>• Model statistik tradisional seringkali tidak mampu menangkap dinamika non-linear dan kompleksitas yang melekat pada data pasar saham, sehingga akurasi prediksinya terbatas.</li> </ul> <p><b>3. Kebutuhan akan Model yang Lebih Akurat:</b></p> <ul style="list-style-type: none"> <li>• Investasi yang sukses memerlukan prediksi volatilitas yang akurat untuk meminimalkan risiko dan memaksimalkan keuntungan, menekankan pentingnya pengembangan model prediksi yang lebih efektif.</li> </ul>	<p>penutupan, dan volume perdagangan.</p>		
<p>Wang [10]</p>	<p>□ <b>Kompleksitas Analisis Data Deret Waktu dalam Skala Besar:</b></p> <ul style="list-style-type: none"> <li>• Dengan meningkatnya volume data, terutama dalam</li> </ul>	<p>menggunakan data dari pasar saham China, khususnya dari National SME Stock Trading (New Third Board). Data tersebut mencakup informasi harga saham yang digunakan</p>	<p>Data preprocessing menggunakan differential processing untuk merubah non stationary menjadi stationary, lalu menggunakan ADF untuk mengecek stabilitas time series,</p>	<p>Dengan dataset National SME Stock Trading, Penerapan model ARIMA menghasilkan hasil yang sebagian besar konsisten dengan kondisi pasar sebenarnya,</p>

	<p>konteks big data, analisis data deret waktu menjadi semakin kompleks. Tantangan utama adalah bagaimana mengekstraksi pola dan tren yang relevan dari kumpulan data yang sangat besar dan beragam.</p> <p><b>□ Keterbatasan Model Tradisional dalam Menangani Big Data:</b></p> <ul style="list-style-type: none"> <li>• Model statistik tradisional, seperti Autoregressive (AR), Moving Average (MA), dan Autoregressive Integrated Moving Average (ARIMA), seringkali tidak dirancang untuk menangani volume data yang sangat besar. Ini menimbulkan tantangan dalam penerapan model tersebut pada skala big data.</li> </ul> <p><b>□ Kebutuhan akan Pendekatan Analitik yang Lebih Efisien:</b></p> <ul style="list-style-type: none"> <li>• Diperlukan pendekatan analitik yang dapat</li> </ul>	<p>untuk membangun model prediksi volatilitas pasar saham.</p>	<p>yang terakhir menggunakan ARIMA untuk memprediksi data masa depan</p>	<p>menunjukkan keandalannya dalam prediksi risiko.</p>
--	---	--	--	--

	<p>memproses dan menganalisis data deret waktu dalam skala besar secara efisien, untuk memberikan wawasan yang akurat dan tepat waktu bagi pengambilan keputusan.</p>			
<p>Wang [12]</p>	<ul style="list-style-type: none"> <li>• <b>Kurangnya Transparansi Dataset</b></li> <li>• Detail spesifik tentang <b>sumber data, periode waktu, atau jenis saham</b> yang digunakan dalam penelitian ini <b>tidak dijelaskan secara rinci</b>.</li> <li>• Hal ini menyulitkan replikasi hasil dan validasi oleh peneliti lain.</li> <li>• <b>Kompleksitas Korelasi Spasio-Temporal</b></li> </ul> <p>Pasar saham memiliki <b>korelasi kompleks</b> antara berbagai faktor spasial (misalnya, hubungan antar indeks saham) dan temporal (misalnya, pola historis).</p>	<p>menggunakan data pasar saham untuk menguji model yang mereka kembangkan. Namun, detail spesifik mengenai dataset yang digunakan, seperti sumber data, periode waktu, atau jenis saham yang dianalisis</p>	<p>Data Harga saham dengan data berita dimasukkan ke dalam HTPNN dengan memanfaatkan vektor kata terdistribusi untuk berita utama, yang dikurangi secara dimensi menggunakan sparse automatic encoders untuk meningkatkan efisiensi model</p>	<p>Metode HTPNN mampu menangkap hukum potensial fluktuasi harga saham secara efektif, sehingga menghasilkan peningkatan akurasi rata-rata hampir 5% dibandingkan metode yang ada saat ini.</p>

	<p>Model harus mampu menangkap keterkaitan ini dengan tingkat granularitas yang tepat agar tidak kehilangan informasi penting.</p> <ul style="list-style-type: none"> <li>• <b>Ketidakstabilan dan Volatilitas Pasar Saham</b></li> <li>• Tren pasar saham dipengaruhi oleh banyak faktor eksternal, seperti <b>berita ekonomi, kebijakan moneter, dan kejadian global.</b></li> <li>• Model perlu robust terhadap <b>fluktuasi ekstrem dan data outlier</b> agar tetap akurat dalam jangka panjang.</li> <li>• <b>Tantangan dalam Pengolahan Big Data</b></li> <li>• Pasar saham menghasilkan <b>jumlah data yang sangat besar</b> dalam waktu singkat.</li> <li>• Model harus memiliki <b>efisiensi komputasi yang tinggi</b> untuk menangani data</li> </ul>			
--	---	--	--	--

	real-time dan memproses informasi dengan cepat.			
Wen [13]	<p>❑ <b>Keterbatasan Model Tradisional dalam Menangkap Informasi Tingkat Tinggi</b></p> <ul style="list-style-type: none"> <li>• Metode prediksi pasar saham tradisional seperti <b>ARIMA dan regresi linear</b> hanya menangkap hubungan <b>linier</b> dalam data.</li> <li>• Tren pasar saham sering kali memiliki <b>pola non-linear dan dinamis</b>, yang sulit ditangkap oleh model klasik.</li> </ul> <p>❑ <b>Ketergantungan pada Fitur Deret Waktu yang Sederhana</b></p> <ul style="list-style-type: none"> <li>• Sebagian besar penelitian hanya menggunakan <b>harga penutupan saham</b> sebagai input, tanpa mempertimbangkan <b>informasi tingkat tinggi</b></li> </ul>	<ul style="list-style-type: none"> <li>➢ Menggunakan data historis dari indeks S&amp;P 500 untuk mengevaluasi kinerja model prediksi.</li> <li>➢ Data ini mencakup harga penutupan harian indeks tersebut. Namun, detail spesifik mengenai periode waktu yang dicakup oleh dataset tidak dijelaskan secara rinci dalam artikel tersebut.</li> </ul>	<p>Makalah ini mengusulkan algoritma baru yang menggabungkan rekonstruksi sekuens berbasis motif dengan jaringan saraf konvolusional (CNN) untuk prediksi tren deret waktu saham.</p>	<p>Metode yang diusulkan meningkatkan akurasi prediksi di hampir semua kumpulan data, mengungguli metode pemrosesan sinyal tradisional dan pendekatan pembelajaran mesin sebesar 4% - 7% dalam akurasi</p> <p>LSTM, meskipun kompetitif, lebih memakan waktu karena vanishing gradient problem, membuat metode CNN yang diusulkan lebih efisien.</p>

	<p>seperti distribusi probabilitas atau pola statistik lebih kompleks.</p> <ul style="list-style-type: none"> <li>• Model yang hanya bergantung pada <b>fitur deret waktu sederhana</b> bisa kehilangan <b>informasi yang lebih dalam</b> tentang tren harga saham.</li> </ul> <p><input type="checkbox"/> <b>Volatilitas dan Noise dalam Data Pasar Saham</b></p> <ul style="list-style-type: none"> <li>• Pergerakan harga saham sangat dipengaruhi oleh faktor eksternal seperti <b>berita keuangan, sentimen pasar, dan kebijakan ekonomi.</b></li> <li>• <b>Noise dalam data historis</b> dapat mengarah pada overfitting model atau prediksi yang tidak akurat.</li> </ul> <p><input type="checkbox"/> <b>Efisiensi Komputasi dalam Pemrosesan Informasi Tingkat Tinggi</b></p>			
--	---	--	--	--

	<ul style="list-style-type: none"> <li>• Menggunakan informasi tingkat tinggi dalam deret waktu memerlukan <b>daya komputasi yang besar</b> dan teknik ekstraksi fitur yang lebih kompleks.</li> <li>• Tantangan utama adalah <b>menyeimbangkan akurasi prediksi dengan efisiensi komputasi</b>, terutama dalam data saham yang terus berubah.</li> </ul> <p>□ <b>Kurangnya Generalisasi Model</b></p> <p>➤ Model yang dikembangkan bisa bekerja dengan baik pada dataset tertentu (misalnya <b>S&amp;P 500</b>), tetapi mungkin tidak memiliki <b>generalizability</b> yang baik untuk indeks saham lain atau pasar dengan karakteristik berbeda.</p>			
Nabipour [17]	□ <b>Ketergantungan pada Data Historis untuk Prediksi Masa Depan</b>	menggunakan data historis dari empat sektor pasar saham di Bursa Efek Teheran untuk melatih dan menguji model prediksi mereka. Data ini	Menguji machine learning tradisional dan deep learning Recurrent Neural Network	Penelitian ini menemukan bahwa untuk data kontinu, model Recurrent Neural Network (RNN) dan Long Short-Term Memory (LSTM) secara signifikan

	<ul style="list-style-type: none"> <li>• Model yang dikembangkan hanya menggunakan <b>data harga historis saham</b>, tanpa mempertimbangkan <b>faktor eksternal</b> seperti berita keuangan, sentimen pasar, atau kebijakan ekonomi.</li> <li>• Hal ini bisa menyebabkan model <b>kurang adaptif terhadap perubahan pasar mendadak</b>.</li> </ul> <p>□ <b>Kelemahan dalam Generalisasi Model</b></p> <ul style="list-style-type: none"> <li>• Studi ini berfokus pada <b>Bursa Efek Teheran</b>, sehingga model yang dibangun mungkin tidak <b>langsung dapat diterapkan pada pasar saham lain</b> yang memiliki karakteristik berbeda.</li> <li>• Faktor seperti <b>likuiditas, volatilitas, dan regulasi pasar</b> berbeda di setiap negara, yang dapat</li> </ul>	<p>mencakup informasi seperti harga saham, volume perdagangan, dan indikator teknis lainnya. Namun, detail spesifik mengenai periode waktu yang dicakup oleh dataset tidak dijelaskan secara rinci dalam artikel tersebut.</p>	<p>(RNN) and Long Short-Term Memory (LSTM)</p>	<p>mengungguli model prediksi lain, mencapai kinerja prediksi sekitar 86% dengan perbedaan yang cukup besar dibandingkan dengan yang lain.</p>
--	---	--	--	--

	<p>mempengaruhi performa model.</p> <p>□ <b>Keterbatasan dalam Pemrosesan Data Biner vs. Kontinu</b></p> <ul style="list-style-type: none"> <li>• Model menggunakan <b>dua jenis data (kontinu dan biner)</b> untuk membandingkan performa.</li> <li>• Namun, <b>konversi data kontinu ke biner dapat menyebabkan kehilangan informasi penting</b>, yang berdampak pada akurasi prediksi.</li> </ul> <p>□ <b>Ketergantungan pada Teknik Machine Learning Standar</b></p> <ul style="list-style-type: none"> <li>• Studi ini hanya mengevaluasi <b>algoritma ML dan DL konvensional</b>, seperti <b>SVM, Random Forest, LSTM, dan CNN</b>.</li> <li>• Model-model yang lebih canggih seperti</li> </ul>			
--	--	--	--	--

	<p><b>Transformers atau Attention Mechanism</b> tidak dieksplorasi lebih jauh, padahal metode ini telah terbukti efektif dalam analisis data sekuensial.</p> <p>□ <b>Tantangan dalam Overfitting Model</b></p> <ul style="list-style-type: none"> <li>• Model deep learning (DL) memiliki kecenderungan untuk <b>overfitting</b>, terutama jika data yang digunakan tidak cukup besar atau bervariasi.</li> <li>• Tidak ada diskusi mendalam mengenai strategi <b>regularisasi</b> atau teknik <b>augmentasi data</b> untuk mengatasi overfitting.</li> </ul> <p>□ <b>Efisiensi Komputasi dan Kompleksitas Model</b></p> <ul style="list-style-type: none"> <li>• Algoritma deep learning seperti <b>LSTM dan CNN</b> memerlukan daya komputasi yang besar.</li> </ul>			
--	--	--	--	--

	<ul style="list-style-type: none"> <li>• Studi ini tidak menjelaskan secara detail tentang <b>trade-off antara akurasi dan efisiensi komputasi</b>, yang penting dalam implementasi model secara real-world.</li> </ul>			
Quadir [20]	<p><b>1. Ketergantungan pada Data Historis</b></p> <ul style="list-style-type: none"> <li>• Model hanya mengandalkan <b>data harga saham historis</b>, tanpa mempertimbangkan faktor eksternal seperti <b>sentimen pasar, berita ekonomi, atau kebijakan makroekonomi</b>.</li> <li>• Ini bisa membuat model kurang mampu menangani <b>kejadian tak terduga</b>, seperti krisis keuangan atau peristiwa global yang</li> </ul>	<p>Penelitian ini menggunakan data historis harga saham Samsung untuk melatih dan menguji model MLS-LSTM. Namun, detail spesifik mengenai dataset—seperti periode waktu yang dicakup, frekuensi data (misalnya, harian, mingguan), dan fitur tambahan yang digunakan (misalnya, volume perdagangan, indikator teknis)</p>	<p>Imenggunakan model Memori Jangka Panjang dan Pendek (LSTM) sekuensial berlapis-lapis, yang merupakan jenis jaringan saraf berulang yang sangat cocok untuk prediksi deret waktu.</p>	<p>Hasilnya menunjukkan bahwa pendekatan ini secara signifikan meningkatkan akurasi peramalan dibandingkan dengan metode tradisional. Model tersebut menunjukkan metrik kinerja yang ditingkatkan, termasuk kesalahan absolut rata-rata yang lebih rendah dan nilai R-kuadrat yang lebih tinggi, yang menunjukkan kecocokan yang lebih baik dengan data. Selain itu, arsitektur berlapis-lapis memungkinkan untuk menangkap pola kompleks dalam pergerakan harga saham, yang mengarah pada prediksi yang lebih andal. Secara keseluruhan, temuan tersebut menunjukkan bahwa metode yang diusulkan merupakan kontribusi yang berharga bagi bidang peramalan keuangan.</p>

	<p>berdampak signifikan pada harga saham.</p> <p><b>2. Kurangnya Diversifikasi Dataset</b></p> <ul style="list-style-type: none"> <li>• Dataset yang digunakan dalam penelitian ini hanya berasal dari <b>saham Samsung</b>, sehingga model yang dikembangkan mungkin <b>tidak dapat digeneralisasi</b> dengan baik ke saham dari sektor atau pasar yang berbeda.</li> <li>• Model mungkin hanya bekerja optimal untuk pola harga saham Samsung dan tidak efektif jika diterapkan pada pasar lain dengan karakteristik berbeda.</li> </ul> <p><b>3. Potensi Overfitting pada Model LSTM</b></p> <ul style="list-style-type: none"> <li>• Model MLS-LSTM yang digunakan terdiri dari beberapa lapisan LSTM, yang berpotensi mengalami <b>overfitting</b> jika jumlah</li> </ul>			
--	--	--	--	--

	<p>parameter terlalu besar dibandingkan dengan ukuran dataset.</p> <ul style="list-style-type: none"> <li>• Tidak ada pembahasan rinci tentang bagaimana mereka mengatasi <b>overfitting</b>, misalnya dengan <b>dropout</b>, <b>regularization</b>, atau <b>data augmentation</b>.</li> <li>• <b>4. Efisiensi Komputasi dan Kompleksitas Model</b></li> <li>• LSTM adalah model yang <b>komputasinya lebih berat</b> dibandingkan dengan metode klasik seperti <b>ARIMA</b> atau <b>SVM</b>.</li> <li>• Studi ini tidak membahas <b>trade-off antara akurasi prediksi dan efisiensi komputasi</b>, terutama untuk implementasi real-time atau pada skala besar.</li> </ul> <p><b>5. Tidak Dibandingkan dengan Model Transformer yang Lebih Modern</b></p>			
--	--	--	--	--

<ul style="list-style-type: none"> <li>• Saat ini, model <b>Transformer-based</b> seperti <b>Temporal Fusion Transformer (TFT)</b> atau <b>StockGPT</b> telah terbukti lebih baik dalam menangani <b>urutan waktu panjang</b> dibandingkan LSTM.</li> <li>• Penelitian ini tidak melakukan perbandingan dengan metode yang lebih baru, sehingga sulit untuk mengetahui apakah MLS-LSTM benar-benar merupakan pendekatan terbaik.</li> </ul> <p><b>6. Tidak Ada Uji Coba di Berbagai Kondisi Pasar</b></p> <ul style="list-style-type: none"> <li>• Pasar saham memiliki kondisi yang berbeda-beda, seperti <b>bull market (pasar naik)</b>, <b>bear market (pasar turun)</b>, dan <b>sideways market (pasar stagnan)</b>.</li> <li>• Studi ini tidak menjelaskan bagaimana performa model <b>di berbagai kondisi pasar</b>,</li> </ul>			
--	--	--	--

	yang penting untuk memastikan model tetap andal dalam skenario yang berbeda.			
--	--	--	--	--

Tabel 2.2.1 Hasil Studi Literatur

### 2.3 Gap pada penelitian terdahulu

Dalam penelitian-penelitian diatas menggunakan berbagai pendekatan telah diusulkan untuk meningkatkan akurasi prediksi pergerakan harga saham, seperti penggunaan Multimodal Fusion Transformer, Hybrid Information Mixing Module, dan Financial Graph Attention Networks. Meskipun pendekatan-pendekatan ini telah menunjukkan hasil yang menjanjikan, terdapat beberapa research gaps antara lain :

1. **Integrasi Data Alternatif:** Sebagian besar penelitian fokus pada data makroekonomi dan indikator teknis. Namun, data alternatif seperti sentimen media sosial, berita keuangan, dan data pencarian internet dapat memberikan wawasan tambahan. Penelitian lebih lanjut diperlukan untuk mengintegrasikan sumber data ini ke dalam model prediksi.
2. **Model Hibrida Lanjutan:** Meskipun model hibrida telah digunakan, kombinasi lebih lanjut antara berbagai arsitektur jaringan saraf, seperti menggabungkan **Transformer** dengan **Graph Neural Networks (GNNs)** atau **Long Short-Term Memory (LSTM)**, dapat dieksplorasi untuk menangkap pola kompleks dalam data saham.
3. **Penyesuaian Model untuk Pasar Berkembang:** Sebagian besar penelitian difokuskan pada pasar saham di negara maju. Penelitian lebih lanjut diperlukan untuk menyesuaikan dan menguji model pada pasar berkembang, yang mungkin memiliki karakteristik berbeda.
4. **Evaluasi Risiko dan Ketidakpastian:** Selain memprediksi arah pergerakan harga saham, penting untuk mengukur tingkat kepercayaan dan ketidakpastian dalam prediksi. Integrasi metode untuk mengkuantifikasi risiko dapat membantu investor dalam pengambilan keputusan.

## 2.4 SOTA ( State of Art )

Dari berbagai gap di atas, pendekatan deep learning dengan metode **Optimasi Adam dalam Model GRU** untuk meningkatkan efisiensi dan akurasi prediksi harga saham

## 2.5 Landasan Teori

Landasan teori diperlukan untuk penelitian ini berasal dari jurnal maupun buku. Dimana digunakan untuk mendukung dalam melakukan penelitian guna mendapatkan hasil yang maksimal.

### 2.5.1 Deep Learning adalah cabang dari Machine Learning

Deep Learning menggunakan jaringan saraf tiruan (Artificial Neural Network/ANN) dengan banyak lapisan (deep networks) untuk mengekstraksi pola kompleks dari data. Model ini mampu belajar secara otomatis dari data mentah tanpa perlu fitur yang diekstraksi secara manual, sehingga sangat efektif dalam tugas-tugas prediksi.

### 2.5.2 Konsep Prediksi dalam Deep Learning

Prediksi dalam Deep Learning mengacu pada penggunaan model jaringan saraf untuk memperkirakan output berdasarkan pola yang ditemukan dalam data historis. Model ini digunakan dalam berbagai bidang, seperti:

- **Prediksi Deret Waktu (Time Series Forecasting):** Peramalan harga saham, prediksi cuaca, dan permintaan pasar.
- **Prediksi dalam NLP (Natural Language Processing):** Prediksi kata berikutnya dalam teks, analisis sentimen, dan chatbot.
- **Prediksi dalam Visi Komputer:** Deteksi objek, pengenalan wajah, dan segmentasi gambar.

#### 2.5.2.1 Arsitektur Deep Learning untuk Prediksi

Terdapat beberapa arsitektur jaringan saraf yang sering digunakan dalam prediksi, antara lain:

**a. Multi-Layer Perceptron (MLP)**

- MLP adalah jaringan saraf feedforward dengan satu atau lebih lapisan tersembunyi.
- Digunakan untuk prediksi berbasis data tabular.
- Menggunakan fungsi aktivasi seperti **ReLU** dan **Softmax**.

**b. Recurrent Neural Network (RNN) dan Variannya (LSTM, GRU)**

- RNN digunakan untuk memproses data berurutan seperti teks atau deret waktu.
- **Long Short-Term Memory (LSTM)** dan **Gated Recurrent Unit (GRU)** dikembangkan untuk mengatasi masalah vanishing gradient dalam RNN.
- Digunakan dalam aplikasi seperti prediksi harga saham, analisis sentimen, dan speech recognition.

**c. Convolutional Neural Network (CNN)**

- CNN lebih dikenal dalam visi komputer tetapi juga dapat digunakan untuk prediksi deret waktu.
- CNN dapat menangkap fitur spasial dalam data yang memiliki pola berulang.

**d. Transformer**

- Model seperti **BERT** dan **GPT** digunakan dalam NLP untuk prediksi teks.
- **Temporal Fusion Transformer (TFT)** digunakan dalam prediksi deret waktu skala besar.

### 2.5.2.2 Proses Pelatihan Model Deep Learning untuk Prediksi

Proses pelatihan model prediksi berbasis deep learning melibatkan beberapa langkah utama:

#### 1. **Preprocessing Data:**

- Normalisasi atau standarisasi data.
- Mengisi nilai yang hilang.
- Pembagian dataset menjadi data latih, validasi, dan uji.

#### 2. **Pemilihan Arsitektur Model:**

- Memilih model yang sesuai berdasarkan jenis data dan tugas prediksi.

#### 3. **Pelatihan Model:**

- Menggunakan algoritma optimasi seperti **Adam** atau **SGD**.
- Fungsi loss yang digunakan tergantung pada jenis prediksi:
  - **MSE (Mean Squared Error)** untuk regresi.
  - **Cross-Entropy Loss** untuk klasifikasi.

#### 4. **Evaluasi Model:**

- Menggunakan metrik seperti **RMSE**, **MAE**, atau **Akurasi**.

#### 5. **Deploy dan Penggunaan Model:**

- Model yang sudah dilatih dapat diimplementasikan dalam aplikasi nyata, seperti sistem rekomendasi atau chatbot AI.

### 2.5.2.3 Implementasi Deep Learning untuk Prediksi

Deep Learning dapat diimplementasikan menggunakan framework seperti:

#### 2.5.2.3.1 TensorFlow/Keras

TensorFlow dan Keras adalah dua framework yang sangat populer untuk pengembangan model machine learning dan deep learning. Berikut adalah penjelasan dan gambaran umum tentang keduanya:

##### **TensorFlow**

TensorFlow adalah library open-source yang dikembangkan oleh Google. Ini dirancang untuk membantu para pengembang dalam membangun dan melatih model machine learning dengan efisiensi dan fleksibilitas. Beberapa fitur utama TensorFlow meliputi:

- **Kompatibilitas Multi-Platform:** TensorFlow dapat dijalankan pada berbagai platform, termasuk CPU dan GPU, serta di cloud.
- **Modeling:** Mendukung berbagai jenis model, dari model sederhana hingga model kompleks (seperti CNN, RNN).

**Ecosystem:** Memiliki ekosistem yang luas, termasuk Tensor Board untuk visualisasi data dan model, TensorFlow Serving untuk produksi model, dan TensorFlow Lite untuk perangkat mobile. **Ketersediaan API:** Tersedia API dalam bahasa Python, C++, dan Java, di antara lainnya.

##### **Keras**

Keras adalah API high-level yang digunakan untuk membangun dan melatih model deep learning. Awalnya, Keras merupakan library independen, tetapi kemudian diintegrasikan ke dalam TensorFlow untuk memudahkan penggunaannya. Fitur utama Keras meliputi:

**Sederhana dan Mudah Digunakan:** Desain Keras yang user-friendly membuatnya mudah untuk membangun model dengan berbagai layer tanpa banyak konfigurasi.

Modularitas: Memungkinkan pembangunan model dengan cara yang modul sehingga pengguna dapat mengubah layer dan parameter dengan mudah.

Dukungan untuk Berbagai Backends: Walaupun Keras sekarang biasanya digunakan dengan TensorFlow sebagai backend, hingga versi sebelumnya ia juga mendukung Theano dan Microsoft Cognitive Toolkit (CNTK).

TensorFlow dan Keras adalah tools yang sangat kuat dalam pengembangan machine learning dan deep learning. TensorFlow menyediakan dasar yang kuat dan fleksibilitas, sementara Keras memberikan API yang sederhana dan mudah digunakan. Kombinasi keduanya memungkinkan pengembang untuk membangun, melatih, dan menerapkan model yang efektif dalam berbagai aplikasi.

#### **2.5.2.3.2 PyTorch**

PyTorch adalah sebuah framework open-source untuk deep learning yang dikembangkan oleh Meta (Facebook). PyTorch banyak digunakan oleh peneliti dan praktisi machine learning karena fleksibilitasnya dalam membangun dan melatih model berbasis jaringan saraf tiruan (neural networks).

Keunggulan PyTorch

##### **1. Graf Komputasi Dinamis (Dynamic Computation Graphs)**

Memungkinkan perubahan struktur model secara fleksibel selama eksekusi.

##### **2. Dukungan GPU (CUDA Acceleration)**

PyTorch secara otomatis dapat memanfaatkan GPU untuk komputasi yang lebih cepat.

##### **3. Autograd (Automatic Differentiation)**

Memudahkan perhitungan gradien dalam proses training model.

##### **4. Ekosistem yang Kuat**

Terdapat pustaka pendukung seperti:

torchvision → untuk pemrosesan gambar

torchtext → untuk pemrosesan teks

torchaudio → untuk pemrosesan suara

#### 5. Mudah Digunakan dan Pythonic

Sintaksis yang mudah dipahami dan lebih dekat dengan gaya pemrograman Python.

#### 2.5.2.3.3 Scikit-learn (untuk model yang lebih sederhana)

Scikit-Learn adalah pustaka open-source di Python yang digunakan untuk machine learning (ML) klasik. Pustaka ini menyediakan berbagai algoritma dan alat untuk klasifikasi, regresi, klusterisasi, dan pemrosesan data.

Keunggulan Scikit-Learn

- **Mudah Digunakan** → API yang sederhana dan mudah dipahami.
- **Berbasis NumPy, SciPy, dan Matplotlib** → Memanfaatkan pustaka numerik Python untuk performa tinggi.
- **Beragam Algoritma Machine Learning** → Seperti regresi linear, SVM, decision tree, random forest, dan lainnya.
- **Fitur Preprocessing Data** → Termasuk normalisasi, encoding, dan reduksi dimensi.
- **Dukungan Evaluasi Model** → Memiliki metrik evaluasi seperti akurasi, F1-score, dan confusion matrix.

#### Fitur-Fitur Utama Scikit-Learn

##### 1. Klasifikasi (Classification)

- Algoritma: LogisticRegression, DecisionTreeClassifier, RandomForestClassifier, SVC, KNeighborsClassifier, dll.

## 2. Regresi (Regression)

- o Algoritma: LinearRegression, Ridge, Lasso, SVR, DecisionTreeRegressor, dll.

## 3. Klasterisasi (Clustering)

- o Algoritma: KMeans, DBSCAN, AgglomerativeClustering, dll.

## 4. Reduksi Dimensi (Dimensionality Reduction)

- o PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis)

## 5. Preprocessing Data

- o Normalisasi (StandardScaler, MinMaxScaler), Encoding Kategorikal (OneHotEncoder, LabelEncoder)

## 6. Evaluasi Model

- o accuracy\_score, confusion\_matrix, classification\_report, mean\_squared\_error, dll.

Scikit-Learn cocok untuk proyek machine learning yang berbasis data terstruktur. Apakah Anda butuh contoh atau penjelasan lebih lanjut tentang fitur tertentu,

### 2.5.3 GRU Gated Recurrent Unit ( GRU )

Gated Recurrent Unit (GRU) adalah jenis RNN yang diperkenalkan oleh Cho et al. (2014) sebagai alternatif Long Short-Term Memory (LSTM). GRU mengatasi masalah vanishing gradient dengan mekanisme "gated", yang mengontrol aliran informasi dalam jaringan.

#### 2.5.3.1 Struktur dan Komponen GRU

GRU memiliki struktur mirip dengan Long Short-Term Memory (LSTM) tetapi lebih sederhana karena hanya memiliki dua *gate* utama: GRU memiliki dua jenis gerbang utama:

- **Reset Gate ( $r_t$ ):** Mengontrol seberapa banyak informasi dari masa lalu yang akan dilupakan. Dirumuskan :

- **Update Gate ( $z_t$ ):** Menentukan seberapa banyak informasi dari waktu sebelumnya yang akan diteruskan ke waktu saat ini.

Persamaan utama dalam GRU:

#### a. Reset Gate ( $r_t$ )

Reset gate mengontrol apakah informasi lama perlu dilupakan atau tidak. Dirumuskan sebagai:

Reset gate mengontrol apakah informasi lama perlu dilupakan atau tidak. Dirumuskan sebagai:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

- $W_r$  dan  $U_r$ : Bobot untuk input  $x_t$  dan status sebelumnya  $h_{t-1}$ .
- $b_r$ : Bias.
- $\sigma$ : Fungsi aktivasi sigmoid (menghasilkan nilai antara 0 dan 1).
- Jika  $r_t$  mendekati 0, sebagian besar informasi lama dihapus.

#### b. Update Gate ( $z_t$ )

Update gate menentukan seberapa banyak informasi lama yang tetap dipertahankan dalam status saat ini:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

- Jika ( $z_t$ ) mendekati 1, maka sebagian besar informasi lama tetap dipertahankan.
- Jika ( $z_t$ ) mendekati 0, informasi baru lebih banyak digunakan.

#### c. Status Baru ( $\tilde{h}_t$ )

Status baru dihitung menggunakan reset gate:

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h)$$

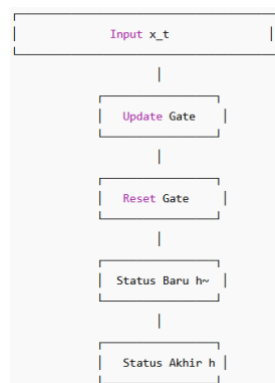
- $r_t \odot h_{t-1}$  adalah operasi elemen-wise antara reset gate dan status sebelumnya.
- Jika  $r_t$  kecil, sebagian besar informasi lama dihapus.

#### d. Status Akhir ( $h_t$ )

Status akhir GRU merupakan interpolasi antara status lama dan status baru:

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}$$

- Jika  $Z_t$  mendekati 1, maka  $h_t$  akan lebih mendekati  $h_{t-1}$ , mempertahankan memori lama.
- Jika  $Z_t$  mendekati 0, maka  $h_t$  akan lebih mendekati  $\tilde{h}_t$ , menggunakan informasi baru.



Gambar 2.5.3.1.1 Diagram Alur GRU

GRU adalah versi yang lebih sederhana dari LSTM dengan kinerja yang lebih cepat, tetapi tetap mempertahankan kemampuan dalam menangani dependensi jangka panjang. GRU cocok untuk tugas NLP, time-series forecasting, dan aplikasi lainnya yang membutuhkan pemrosesan urutan data.

Apakah Anda ingin contoh GRU untuk tugas tertentu, seperti NLP atau time-series

#### 2.5.3.2 Kelebihan GRU dibandingkan LSTM

- **Lebih sederhana:** GRU memiliki lebih sedikit parameter karena hanya memiliki dua gerbang, dibandingkan LSTM yang memiliki tiga gerbang.
- **Lebih cepat dalam pelatihan:** Karena lebih sedikit komputasi dibandingkan LSTM.

- **Performa yang setara atau lebih baik:** Dalam beberapa kasus, GRU menunjukkan hasil yang sebanding atau lebih baik dibandingkan LSTM dalam menangani data berurutan.

Fitur	GRU	LSTM
<i>Gating Units</i>	2 ( <i>Reset gate, Update gate</i> )	3 ( <i>Input gate, Forget gate, Output gate</i> )
<i>Cell State</i>	Tidak ada	Ada
<i>Parameter</i>	Lebih sedikit	Lebih banyak
<i>Kecepatan</i>	Lebih cepat	Lebih lambat
<i>Akurasi</i>	Sedikit lebih rendah	Lebih baik untuk urutan panjang

Tabel 2.5.3.2. 1 Perbandingan GRU dan LSTM

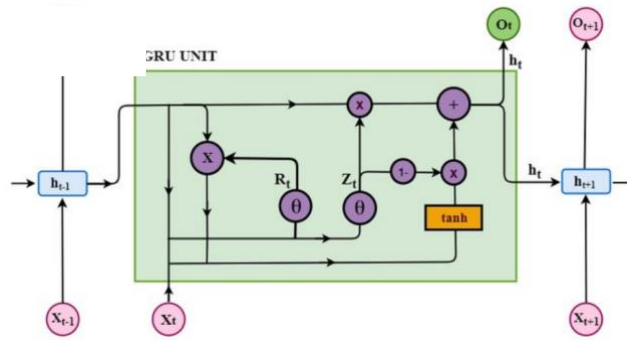
### 2.5.3.3 Implementasi GRU dalam Deep Learning

GRU banyak digunakan dalam berbagai aplikasi seperti:

- **Natural Language Processing (NLP):** Penerjemahan bahasa, analisis sentimen, chatbot.
- **Speech Recognition:** Pengenalan suara dan sintesis teks ke suara.
- **Time Series Prediction:** Peramalan harga saham, prediksi cuaca, dan deteksi anomali.

Framework yang umum digunakan untuk implementasi GRU termasuk **TensorFlow (Keras)** dan **PyTorch**.

GRU (Gated Recurrent Unit) adalah salah satu jenis arsitektur Recurrent Neural Network (RNN) yang dirancang untuk menangani masalah vanishing gradient dalam RNN tradisional. GRU adalah alternatif yang lebih sederhana dari LSTM (Long Short-Term Memory), tetapi tetap efektif untuk memproses data sekuensial seperti teks, audio, atau data time series. GRU lebih efisien dalam pengolahan data sekuensial. GRU menggunakan dua gerbang utama, update gate dan reset gate, untuk mengontrol aliran informasi melalui jaringan.



Gambar 2.5.3.3.1 Arsitektur dasar Gated Recurrent Unit (GRU).

Langkah-Langkah Algoritma GRU :

Inisialisasi Parameter:

Bobot ( $W_z, W_r, W_h$ ) dan bias ( $b_z, b_r, b_h$ ) diinisialisasi secara acak atau menggunakan 46ambar seperti Xavier Initialization.

Masukan Data:

Input pada waktu  $t$  adalah  $x_t$ , dan hidden state dari 46ambara sebelumnya adalah  $h_{t-1}$ .

Menghitung Update Gate ( $z_t$ ) :

Update gate mengontrol seberapa banyak informasi dari hidden state sebelumnya ( $h_{t-1}$ ) yang harus diteruskan ke hidden state saat ( $h_t$ )

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

Di sini,  $\sigma$  sigma  $\sigma$  adalah fungsi aktivasi sigmoid.

Menghitung Reset Gate ( $r_t$ ) :

Reset gate menentukan seberapa banyak informasi dari hidden state sebelumnya ( $h_{t-1}$ ) yang harus dilupakan.

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

Menghitung Candidate Hidden State ( $\tilde{h}_t$ ) :

Candidate hidden state adalah representasi informasi baru yang akan ditambahkan ke hidden state saat ini.

$$\tilde{h}_t = \tan^{-1}(V_h \cdot [r_t \odot h_{t-1}, x_t] + b_h)$$

Di sini,  $\odot$  adalah operasi elemen-wise (Hadamard product).

Menghitung Final Hidden State ( $h_t$ ):

Hidden state pada waktu ttt adalah kombinasi dari hidden state sebelumnya ( $h_{t-1}$ ) dan candidate hidden state ( $\tilde{h}_t$ ), diatur oleh update gate ( $z_t$ )

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

Iterasi untuk Semua Waktu (T):

Ulangi Gambaran langkah di atas untuk semua waktu dalam sekuens.

## 2.5.4 Optimasi ADAM

Adam (*Adaptive Moment Estimation*) adalah algoritma optimasi yang menggabungkan keunggulan dari dua metode optimasi lainnya:

- **Momentum:** Membantu dalam mempercepat konvergensi dengan mempertahankan arah pergerakan parameter.
- **RMSprop:** Menggunakan rata-rata kuadrat gradien untuk menyesuaikan laju pembelajaran secara adaptif.

Adam sangat cocok untuk model seperti **Gated Recurrent Unit (GRU)** yang digunakan dalam pemrosesan data berurutan seperti prediksi deret waktu dan pemrosesan bahasa alami (NLP).

### 2.5.4.1 Formula Optimasi Adam

Adam menggunakan estimasi momentum pertama ( $m_t$ ) dan kedua ( $v_t$ ) dari gradien:

Adam memperbarui bobot  $\theta$  menggunakan **perkiraan momen pertama (mean) dan momen kedua (variance)** dari gradien. Formula utama dalam Adam adalah sebagai berikut:

### 1. Perhitungan Gradien

Misalkan kita punya fungsi loss  $L$  dengan parameter model  $\theta_t$ , dan gradiennya dihitung sebagai:

$$g_t = \nabla_{\theta} L(\theta_t)$$

di mana  $g_t$  adalah gradien loss terhadap parameter  $\theta$  pada iterasi  $t$ .

### 2. Momen Pertama (Eksponensial Moving Average Gradien):

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

di mana:

- $m_t$  adalah rata-rata eksponensial dari gradien (mirip dengan momentum dalam *SGD*).
- $\beta_1$  adalah faktor eksponensial decay untuk momen pertama (biasanya 0.9).
- $m_{t-1}$  adalah nilai sebelumnya dari momen pertama.

### 3. Momen Kedua (Eksponensial Moving Average Kuadrat Gradien):

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

di mana:

- $v_t$  adalah rata-rata eksponensial dari gradien yang dikuadratkan (mirip dengan *RMSProp*).
- $\beta_2$  adalah faktor eksponensial decay untuk momen kedua (biasanya 0.999).
- $v_{t-1}$  adalah nilai sebelumnya dari momen kedua.

#### 4. Koreksi Bias untuk Momen Pertama dan Kedua:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Koreksi bias ini memastikan estimasi momen lebih akurat di iterasi awal.

#### 5. Pembaruan parameter:

$$\theta_t = \theta_{t-1} - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Di mana:

- $\alpha$  alpha adalah **learning rate** (misalnya 0.001).
- $\epsilon$  adalah nilai kecil untuk menghindari pembagian dengan nol (default:  $10^{-8}$ )

#### 2.5.4.2 Optimasi Adam pada Model GRU

Untuk meningkatkan performa GRU dengan Adam, pertimbangkan hal berikut:

- **Penyesuaian Learning Rate**
  - Coba **0.001** (default), tetapi jika model tidak konvergen, bisa diturunkan ke **0.0001**.
- **Penyesuaian Batch Size**
  - Batch size kecil (16-32) cocok untuk data deret waktu.
- **Dropout atau Regularisasi**
  - Tambahkan dropout untuk menghindari overfitting,
- **Penyesuaian Parameter Adam ( $\beta_1, \beta_2$ )**
  - Default:  $\beta_1=0.9, \beta_2=0.999$ , tetapi dalam beberapa kasus,  **$\beta_1=0.85$**  dapat mempercepat konvergensi.

Optimasi Adam (Adaptive Moment Estimation) adalah algoritma pengoptimalan berbasis gradien yang menggabungkan keuntungan dari

Momentum dan RMSProp untuk mempercepat konvergensi dan menyesuaikan learning rate adaptif secara otomatis.

Optimasi Adam menghitung gradien dari parameter model GRU (seperti bobot  $W_z, W_r, W_h$  dan bias  $b_z, b_r, b_h$  dan memperbarui parameter tersebut menggunakan dua momen 50ambaran50:

Rata-Rata Eksponensial Gradien (Momentum):

Menangkap tren gradien agar lebih stabil.

$$\text{Formula: } m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

Rata-Rata Eksponensial Kuadrat Gradien (RMSProp):

Menormalkan gradien agar tidak terlalu besar atau terlalu kecil.

$$\text{Formula: } v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

Koreksi Bias:

Koreksi bias digunakan karena rata-rata eksponensial memiliki kecenderungan bias terhadap nol di iterasi awal.

$$\text{Formula: } \hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

Pembaruan Parameter:

Parameter diperbarui dengan kombinasi dari momen pertama ( $\hat{m}_t$ ) dan momen kedua ( $\hat{v}_t$ )

$$\text{Formula: } \theta_t = \theta_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Di sini,  $\eta$  adalah learning rate, dan  $\epsilon$  adalah nilai kecil untuk mencegah pembagian dengan nol.

### 2.5.5 Metrik Evaluasi menggunakan RMSE, MAE, dan R<sup>2</sup> Akurasi

Setelah melatih model **GRU dengan Adam**, kita perlu mengevaluasi performanya menggunakan metrik yang sesuai dengan jenis prediksi yang dilakukan.

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

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- Mengukur rata-rata kesalahan absolut antara nilai aktual dan prediksi.
- Cocok untuk kasus di mana semua kesalahan dianggap sama.

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

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- Mengukur seberapa besar penyimpangan rata-rata prediksi terhadap nilai sebenarnya.
- Lebih sensitif terhadap outlier dibanding MAE karena menggunakan kuadrat dari kesalahan.

#### 3. R<sup>2</sup> Akurasi

- Digunakan untuk klasifikasi, bukan regresi.
- Dalam konteks regresi, bisa dihitung sebagai **R-squared**

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

- Mengukur seberapa baik model menjelaskan variasi dalam data.

## Interpretasi Hasil Evaluasi

- MAE lebih kecil → Model lebih baik (kesalahan rata-rata lebih rendah).
- RMSE lebih kecil → Model lebih baik (kesalahan yang lebih besar dihukum lebih keras).
- R-squared mendekati 1 → Model sangat baik (0 berarti model hanya sebaik rata-rata, negatif berarti model buruk).

### Hasil RMSE dan MAE tinggi,

- Gunakan lebih banyak data atau preprocessing yang lebih baik
- Gunakan dropout atau regularisasi untuk menghindari overfitting
- Tuning hyperparameter Adam seperti learning rate dan batch size

## 2.5.6 Prediksi Saham dengan Deep Learning

Prediksi harga saham merupakan salah satu tantangan dalam dunia keuangan yang menarik banyak perhatian peneliti dan praktisi. Berbagai metode telah digunakan untuk memodelkan pergerakan harga saham, mulai dari metode statistik tradisional hingga teknik kecerdasan buatan (AI) seperti machine learning dan deep learning. Model berbasis deep learning seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) telah terbukti unggul dalam menangkap pola kompleks dalam data saham berbasis deret waktu (time series).

Deep learning memungkinkan komputer untuk belajar dari data melalui lapisan **bertingkat** dalam sebuah jaringan saraf. Setiap lapisan memproses data input dengan cara tertentu dan mengirimkan output ke lapisan berikutnya, sehingga model dapat memahami pola yang semakin kompleks.

Komponen Utama Deep Learning: Neuron: Unit pemrosesan dasar yang mirip dengan neuron biologis. Lapisan (Layers): Terdiri dari lapisan masukan (input layer), lapisan tersembunyi (hidden layers), dan lapisan keluaran (output layer). Bobot (Weights): Parameter yang disesuaikan selama pelatihan untuk memetakan data input ke output. Fungsi Aktivasi: Fungsi matematis (misalnya, ReLU, sigmoid) yang

memperkenalkan non-linearitas. Loss Function: Mengukur seberapa baik model melakukan prediksi. Backpropagation: Algoritma untuk menyesuaikan bobot berdasarkan kesalahan prediksi.

*Deep learning* telah menjadi alat yang sangat efektif untuk tugas prediksi di berbagai domain, seperti prediksi harga saham, cuaca, penyakit, perilaku pengguna, dan lainnya. Berikut adalah panduan lengkap tentang penerapan deep learning untuk prediksi. Di bidang forecasting, **Deep Learning untuk Prediksi Harga Saham** adalah salah satu aplikasi populer dalam analisis data deret waktu. Karena harga saham sangat dipengaruhi oleh pola historis dan faktor eksternal, deep learning digunakan untuk menangkap pola kompleks yang sulit dipelajari dengan metode statistik tradisional.

## 2.5.7 Analisis Evaluasi Metode di dalam Peramalan

### 2.5.7.1 Mean Absolute Error (MAE)

MAE mengukur rata-rata selisih absolut antara nilai aktual ( $y_i$ ) dan prediksi  $\hat{y}_i$

Formula MAE :

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

- MAE menunjukkan seberapa besar kesalahan prediksi rata-rata dalam unit yang sama dengan data asli.
- Nilai **lebih kecil** berarti prediksi lebih akurat.
- Tidak memperhitungkan arah kesalahan (positif atau negatif).

### 2.5.7.2 Root Mean Squared Error (RMSE)

RMSE menghitung akar dari rata-rata kuadrat selisih antara nilai aktual dan prediksi.

**Formula RSME :**

$$RSME = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

- RMSE lebih sensitif terhadap **outliers** karena kesalahan dikuadratkan sebelum dirata-rata.
- Jika ada kesalahan besar, RMSE akan meningkat secara signifikan.
- Nilai **lebih kecil** berarti model lebih baik.

### 2.5.7.3 R-squared ( $R^2$ )

$R^2$  menunjukkan seberapa baik model menjelaskan variasi dalam data. Nilainya berkisar antara **0 hingga 1** (atau bahkan negatif jika model sangat buruk).

Formula  $R^2$  :

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

- $\bar{y}$  adalah rata-rata nilai aktual.
- Nilai **1** berarti model sempurna (prediksi = data asli).
- Nilai **0** berarti model tidak lebih baik dari rata-rata.
- Nilai **negatif** berarti model lebih buruk dari sekadar menebak rata-rata.

### Kelebihan dan Keterbatasan

- **MAE**: Mudah diinterpretasikan dan tidak terpengaruh oleh outlier, tetapi tidak memberikan penalti lebih besar pada kesalahan besar.
- **RMSE**: Kombinasi yang baik dari MSE dan MAE karena tetap memberikan penalti lebih besar pada kesalahan besar tetapi hasilnya dalam satuan yang sama dengan variabel yang diukur.

## BAB III

### METODE PENELITIAN

#### 3.1 Metode Penelitian

Penelitian ini merupakan **penelitian kuantitatif eksperimental** yang bertujuan mengembangkan dan menguji kinerja model prediksi harga saham menggunakan algoritma **Gated Recurrent Unit (GRU)** yang dioptimasi menggunakan **Adam Optimizer**. Pendekatan ini dipilih karena mampu memodelkan hubungan temporal dan non-linier dalam data deret waktu.

Metode yang diusulkan dalam penelitian ini adalah optimasi ADAM dalam model Gated Recurrent Unit untuk prediksi harga saham. Penelitian ini bertujuan untuk meningkatkan akurasi prediksi harga saham dengan mengoptimalkan algoritma **Gated Recurrent Unit (GRU)** menggunakan **optimasi Adam**. Penelitian ini mencakup tahapan pengumpulan data, pra-pemrosesan data, arsitektur model GRU, proses pelatihan, evaluasi model, serta perangkat lunak dan pustaka yang digunakan.

Model GRU digunakan untuk mengelola dan memahami pola temporal dalam data saham. Untuk mengoptimalkan proses pembelajaran, algoritma ADAM dipilih karena kemampuannya dalam mengatur learning rate adaptif yang membantu model mencapai konvergensi lebih cepat. Evaluasi kinerja dilakukan menggunakan metrik RMSE dan MAE untuk menilai seberapa akurat model memprediksi harga saham aktual.

#### 3.2 Datashet dan Sumber Data

Data harga saham diperoleh dari **Yahoo Finance**, yang menyediakan data historis saham. Sumber data berdasarkan dari studi literatur yang dilakukan dataset yang digunakan dalam penelitian ini adalah dataset dari Yahoo Finance dengan tautan <https://finance.yahoo.com/quote/TLKM.JK/> [24]

Rentang waktu: 20 tahun terakhir untuk memastikan model dapat menangkap trend jangka panjang. Data yang diambil dari dari yahoo Finance berisi saham PT. Telkom berjumlah 4859 record diambil dari periode tanggal 03-06-2002 sampai dengan 31-05-2022.

Fitur ini terdiri dari harga saham yang berisi kolom / variabel :

1. **date**: Tanggal ketika data saham tersebut diambil atau berlaku.

2. **code**: Kode saham yang digunakan untuk mengidentifikasi perusahaan atau instrumen keuangan tertentu di bursa efek.
3. **open**: Harga pembukaan saham pada awal sesi perdagangan.
4. **high**: Harga tertinggi yang dicapai oleh saham dalam sesi perdagangan hari itu.
5. **low**: Harga terendah yang dicapai oleh saham dalam sesi perdagangan hari itu.
6. **close**: Harga penutupan saham pada akhir sesi perdagangan hari itu.
7. **volume**: Jumlah saham yang diperdagangkan dalam sesi perdagangan hari itu (biasanya dihitung dalam unit lot atau lembar saham).

### 3.3 Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan, yaitu:

1. Akuisisi Data
2. Preprocessing Data
3. Perancangan Model GRU
4. Optimasi menggunakan Adam
5. Pelatihan dan Pengujian Model
6. Evaluasi Kinerja Model

### 3.4 Tahap Preprocessing Data

#### ✓ Memilih Fitur yang Relevan

Biasanya, fitur Close digunakan sebagai target prediksi. Namun, kita bisa menambahkan fitur lain sebagai input.

```
features = [Date "Open", "High", "Low", "Close", "Volume"]
```

#### ✓ Menangani Missing Values

Jika ada data yang hilang (NaN), kita bisa mengisinya dengan metode interpolasi.

#### ✓ Normalisasi Data (Min-Max Scaling)

Model GRU bekerja lebih baik jika datanya dinormalisasi antara 0 dan 1.

#### ✓ Membentuk Dataset untuk Model GRU

Model GRU memerlukan data dalam format sekuensial ( Timeseries) , dengan menggunakan bebehari hari terakhir untuk memprediksi harga di hari ke depan.

✓ **Membagi Dataset menjadi Training dan Testing**

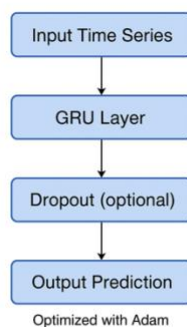
70% data digunakan untuk pelatihan, 10 % digunakan untuk validasi dan 20% untuk pengujian.

Setelah preprocessing ini, data siap digunakan untuk pelatihan model GRU dengan optimasi Adam.

Pada tahap *preprocessing* dataset yang dikumpulkan dan diambil dari data publik saham PT Telkom, dari website Yahoo Finance di ambil sebagai data mentah dan kemudian data tersebut di normalisasi untuk format yang disiapkan untuk pelatihan model. Setelah itu, dataset dibagi menjadi set pelatihan dan validation dengan rasio 70 :10 dan dilanjutkan diuji dengan model GRU memakai optimasi ADAM, dan data sebanyak 20 untuk data test.

### 3.5 Arsitektur Model GRU

Model GRU yang digunakan dalam penelitian ini terdiri dari beberapa lapisan:



Gambar 3.5. 1 Arsitektur Model GRU

1. **Input Layer:** Memasukkan data sekuens harga saham.
2. **GRU Layer:** LSTM dengan sejumlah unit tersembunyi untuk menangkap pola temporal dalam data saham.

3. **Fully Connected (Dense) Layer:** Lapisan terakhir untuk menghasilkan prediksi harga saham.
4. **Activation Function:** Menggunakan fungsi **ReLU** untuk non-linearitas.
5. **Loss Function:** Menggunakan **Mean Squared Error (MSE)** sebagai fungsi loss.
6. **Optimizer:** Menggunakan **Adam (Adaptive Moment Estimation)**.

### 3.6 Algoritma Optimasi Adam

Adam (Adaptive Moment Estimation) adalah algoritma optimasi berbasis gradien yang menggabungkan keunggulan dari **Momentum** dan **RMSProp**.

#### Parameter Adam yang digunakan:

- Learning Rate = 0.001
- Beta1 = 0.9
- Beta2 = 0.999
- Epsilon = 1e-8

Adam dipilih karena dapat:

- Menyesuaikan learning rate untuk setiap parameter
- Mempercepat konvergensi pada pelatihan GRU

Mengatasi noise pada data time series seperti harga saham

### 3.7 Evaluasi Model

- **Mean Absolute Error (MAE):**
- **Root Mean Squared Error (RMSE):**
- **R-Squared (R<sup>2</sup> Score):**

Semakin kecil nilai MAE dan RMSE, serta semakin mendekati 1 nilai R<sup>2</sup>, maka model dianggap semakin baik

### 3.8 Perangkat Lunak dan Pustaka

Penelitian ini menggunakan:

- **Python** sebagai bahasa pemrograman
- **TensorFlow / PyTorch** untuk membangun model GRU
- **Scikit-learn** untuk evaluasi model
- **Matplotlib & Seaborn** untuk visualisasi hasil

Metode yang digunakan dalam penelitian, mulai dari pengumpulan dan pra-pemrosesan data, perancangan model GRU, penggunaan optimasi Adam, serta evaluasi model untuk prediksi harga saham.

### 3.9 Jadwal

Berikut adalah susunan rencana jadwal penelitian yang akan dilakukan

No	Kegiatan	Bulan 2024 - 2025			
		De s	Ja n	Fe b	Ma r
<b>1</b>	Tahap Persiapan Penelitian				
	a. Studi Literatur	■			
	b. Penyusunan dan Pengajuan Judul		■		
	c. Pengajuan Proposal		■		
	d. Seminar Proposal			■	
<b>2</b>	Tahap Pelaksanaan			■	
	a. Pengumpulan Data			■	
	b. Analisa Data			■	
	c. Eksperimen dan Uji Coba			■	
	d. Evaluasi Hasil			■	
<b>3</b>	Penyusunan Laporan			■	
<b>4</b>	Pelaksanaan Ujian Tesis				■

Tabel 3.9. 1 Rencana Jadwal Penelitian

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1 Gambaran Umum Eksperimen

Data yang digunakan adalah harga saham harian PT Telkom Indonesia yang diambil dari Yahoo Finance [24] dengan periode waktu 28 September 2004 hingga 29 Desember 2023, berjumlah 4.768 data. Fitur yang digunakan:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4768 entries, 0 to 4767
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   date        4768 non-null   object
1   adj close   4768 non-null   float64
2   close       4768 non-null   float64
3   high        4768 non-null   float64
4   low         4768 non-null   float64
5   open        4768 non-null   float64
6   volume      4768 non-null   int64
dtypes: float64(5), int64(1), object(1)
memory usage: 260.9+ KB
```

Tabel 4.1.1 Datashet saham

#### 4.2 Akuisisi Data

Dataset dari Yahoo Finance [24] ini memiliki **4.768 data saham** dengan 7 kolom utama, yaitu:

1. **date** (object) – Tanggal pencatatan saham.
2. **adj close** (float64) – Harga penutupan yang telah disesuaikan.
3. **close** (float64) – Harga penutupan saham.
4. **high** (float64) – Harga tertinggi dalam sehari.
5. **low** (float64) – Harga terendah dalam sehari.
6. **open** (float64) – Harga pembukaan saham.
7. **volume** (int64) – Volume transaksi saham.

#### 4.3 Preprocessing Data

Tahapan Preprocessing Data untuk Model GRU dengan Optimasi Adam

##### 1. Konversi Format Tanggal

Kolom date masih dalam bentuk string dan perlu diubah menjadi format datetime:

```
import pandas as pd
```

```
df['date'] = pd.to_datetime(df['date'])
```

## 2. Normalisasi Data

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaled_columns = ['adj close', 'close', 'high', 'low', 'open',
                 'volume']
df[scaled_columns] = scaler.fit_transform(df[scaled_columns])
```

Agar model GRU bekerja lebih baik, kita gunakan MinMaxScaler untuk menormalisasi data ke skala 0-1:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaled_columns = ['adj close', 'close', 'high', 'low', 'open', 'volume']
df[scaled_columns] = scaler.fit_transform(df[scaled_columns])
```

## 3. Penerapan Sliding Window untuk Time Series

Sliding Window digunakan untuk membentuk data agar model GRU dapat belajar dari pola historis. Supaya dapat memprediksi harga saham berdasarkan 10 hari kedepan:

```
import numpy as np
def create_sequences(data, n_steps):
    X, y = [], []
    for i in range(len(data) - n_steps):
        X.append(data[i:i+n_steps])
        y.append(data[i+n_steps])
    return np.array(X), np.array(y)
n_steps = 10 # Panjang jendela waktu
X, y = create_sequences(df['close'].values, n_steps)
```

Untuk evaluasi model, kita pisahkan data menjadi 70% training, validation 10 % dan 20% testing: dengan hasil

- **Training:** 70% ( $\pm$  3.338 data)
- **Validation:** 10% ( $\pm$  476 data)
- **Testing:** 20% ( $\pm$  954 data)

Visualisasi pembagian data:

- Garis waktu menunjukkan porsi tiap bagian dengan warna berbeda.

```
train_size = int(len(X) * 0.8)
```

```
X_train, X_test = X[:train_size], X[train_size:]
```

```
y_train, y_test = y[:train_size], y[train_size:]
```

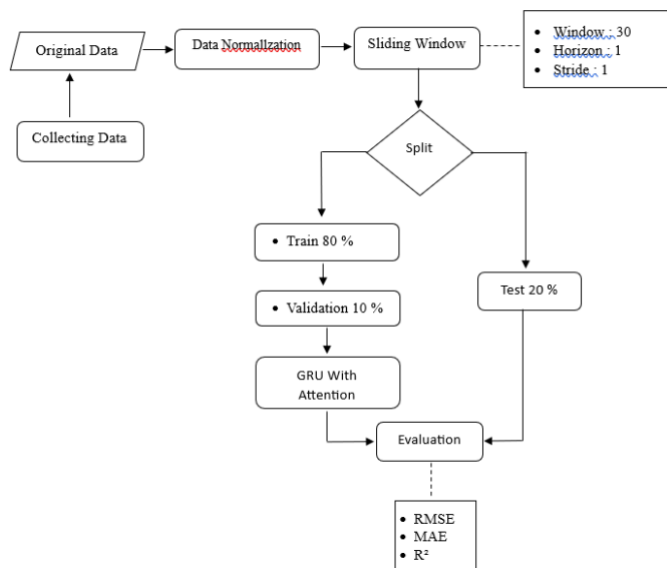
Setelah preprocessing, data siap digunakan dalam model GRU dengan Optimasi Adam untuk prediksi harga saham.

- Pembagian data (training, validation, testing).

Parameter yang digunakan dalam model GRU dengan optimasi Adam.

#### 4.4 Perancangan Model GRU

Pada penelitian ini dapat dilihat arsitektur model GRU dengan ADAM sebagai berikut :



```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)

import sys
sys.path.append(['/content/drive/My Drive/Colab Notebooks/'])
```

Gambar 4.3.1 Arsitektur model GRU dengan ADAM

Menghubungkan Google Drive ke Google Colab, sehingga file di dalam Drive bisa diakses di Colab.

`force_remount=True` → Jika sebelumnya sudah ter-mount, ini akan memaksa ulang mount Driv

```
Fri Feb 21 03:25:54 2025
+-----+
| NVIDIA-SMI 550.54.15                Driver Version: 550.54.15          CUDA Version: 12.4          |
+-----+-----+-----+-----+-----+-----+
| GPU  Name          Persistence-M   Bus-Id        Disp.A    Volatile Uncorr. ECC  |
| Fan  Temp   Perf          Pwr:Usage/Cap     Memory-Usage  GPU-Util  Compute M.  |
|                                           MIG M.         |
+-----+-----+-----+-----+-----+-----+
|   0   Tesla T4      Off              00000000:00:04:0 Off              0          |
| N/A   47C    P8              9W / 70W         0MiB / 15360MiB  0%        Default    |
|                                           MIG M.         |
+-----+-----+-----+-----+-----+-----+
+-----+
| Processes:                               GPU Memory  |
|  GPU   GI    CI          PID    Type    Process name                               Usage       |
|-----+-----+-----+-----+-----+-----+
| No running processes found              |
+-----+
```

Tabel 4.3.1 Pemakaian VRAM pada Google Colab

Google Drive di Google Colab dan memeriksa status GPU

menggunakan `nvidia-smi`. Output menunjukkan akses ke GPU Tesla T4 dengan VRAM 15 GB.

```
data = pd.read_csv(data_raw)
train_size = int(len(data)*0.8)
test_size = int(len(data)*0.2)
val_size = int(test_size/2)

train_size-val_size, test_size, val_size
```

Gambar 4.3.2 Output GPU Tesla T4

untuk membagi dataset menjadi **training, validation, dan testing** dengan proporsi sekitar **70%-10%-20%** dari total data.

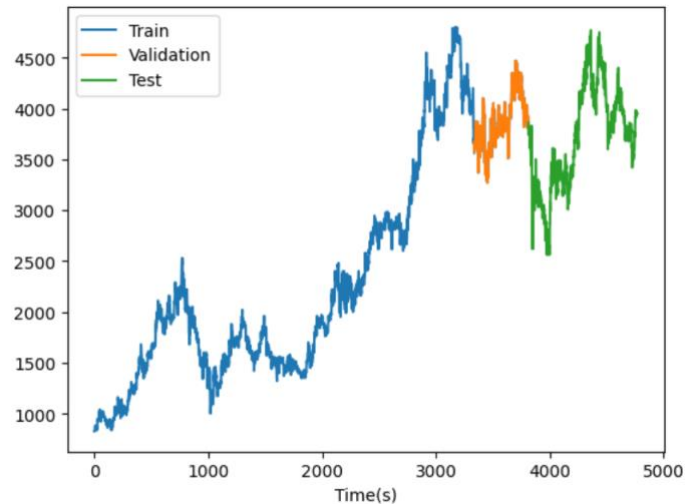
```
train = data[:train_size-val_size]
val = data[train_size - val_size:train_size]
test = data[train_size:]
```

```

time_axis = np.arange(len(data))

plt.plot(time_axis[:train_size], data['close'][:train_size], label = 'Train')
plt.plot(time_axis[train_size - val_size:train_size], data['close'][train_size - val_size:train_size], label = 'Validation')
plt.plot(time_axis[train_size:], data['close'][train_size:], label = 'Test')
plt.xlabel('Time(s)')
plt.legend()

```



Gambar 4.3.3 Membagi dataset menjadi training, validation, dan testing

dengan proporsi sekitar 70%-10%-20%

- **Train:** 70 % data (dari indeks 0 sampai 70)
- **Validation:** 10 % data (dari indeks 70 sampai 79)
- **Test:** 20% data (dari indeks 80 sampai 99)

**Grafik akan menunjukkan tiga garis:**

- **Biru** untuk **Train** (0 - 70).
- **Oranye** untuk **Validation** (70 - 80).
- **Hijau** untuk **Test** (80 - 100).

## 4.5 Optimasi menggunakan Adam

```

model = TimeSeriesAttentionGRU(
    lookback=30,
    pred_steps=1,
    batch_size=32,
    epochs=100
)

```

- **lookback=30** → Model akan melihat **30 hari terakhir** untuk memprediksi harga di masa depan.

- **pred\_steps=1** → Model akan memprediksi **1 langkah ke depan**.
- **batch\_size=32** → Model akan dilatih dalam batch ukuran **32**.
- **epochs=100** → Model akan berlatih selama **100 epoch**.

## 4.6 Pelatihan dan Pengujian Model

```
data = model.load_data(data_raw)
(X_train, y_train), (X_val, y_val), (X_test, y_test) = model.prepare_data(data)
```

- **load\_data(data\_raw)** → Membaca data dari file CSV dan mengubahnya menjadi DataFrame.
- **prepare\_data(data)** →
  - Data **dilakukan scaling** (normalisasi).
  - Dibagi menjadi **training (train), validation (val), dan testing (test)**.

### Melatih Model

```
history = model.train_model(X_train, y_train, X_val, y_val)
```

- Melatih model dengan **data training**.
- **Menggunakan early stopping** untuk mencegah overfitting.

## 4.7 Evaluasi Kinerja Model

```
predictions, actuals, metrics = model.evaluate_model(X_test, y_test)
```

- Model diuji dengan **data testing**.
- Hasil prediksi dibandingkan dengan nilai sebenarnya.
- Menghitung **RMSE, MAE, dan R<sup>2</sup>**.

### Visualisasi Hasil

```
model.plot_training_history(history)    model.plot_predictions(actuals, predictions)
model.plot_predictions_with_dates(actuals, predictions, model.test_dates)
```

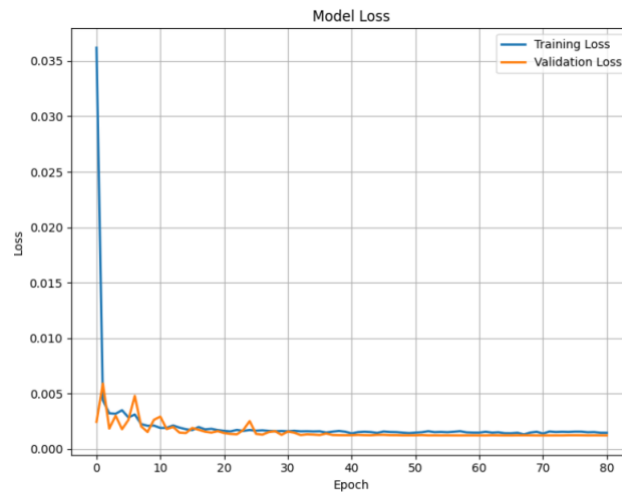
- **plot\_training\_history(history)** → Menampilkan grafik **loss dan MAE** selama training.

- `plot_predictions(actuals, predictions)` → Memplot **data aktual vs prediksi**.
- `plot_predictions_with_dates(actuals, predictions, model.test_dates)` → Memplot hasil prediksi **berdasarkan tanggal**.

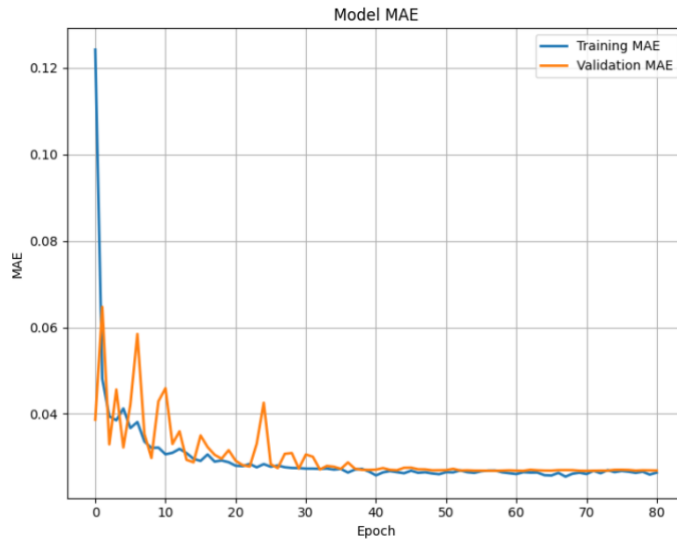
### Mencetak Metrik Evaluasi

```
print("\nModel Evaluation Metrics:")  
for metric, value in metrics.items():  
    print(f"{metric}: {value:.4f}")
```

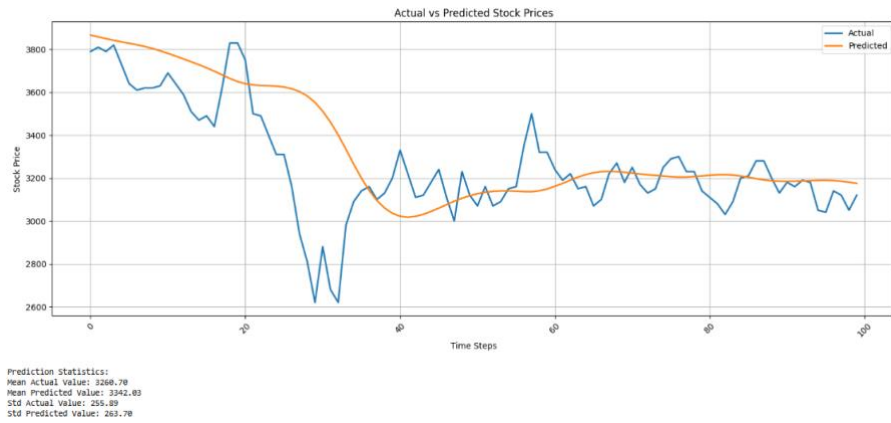
Mencetak **RMSE, MAE, dan  $R^2$**  untuk melihat kualitas prediksi model.



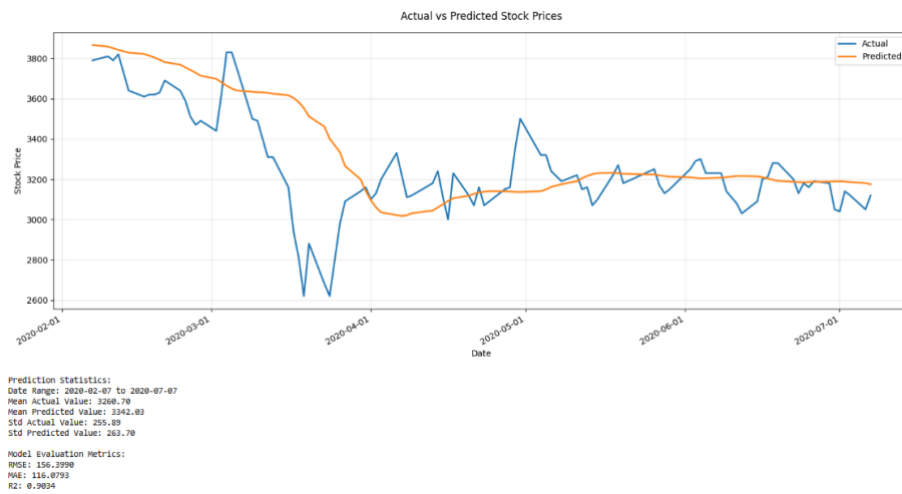
Gambar 4.3.4 Kualitas prediksi model.



Gambar 4.3.5 Training MAE dan Validation MAE



Gambar 4.3.6 Grafik Aktual dan Prediksi



Gambar 4.3.7 Grafik Aktual dan prediksi 07-02-2020 sampai 07-07- 2020

Berdasarkan hasil visualisasi dan metrik yang diperoleh, mari kita analisis performa model Attention + GRU:

1. Training Performance:

- Model menunjukkan konvergensi yang baik dengan penurunan loss yang stabil
- Training dan validation loss menurun secara konsisten tanpa tanda overfitting yang signifikan
- MAE training dan validation juga menurun seiring waktu, menunjukkan model belajar dengan baik

2. Prediksi:

- Model sangat baik dalam menangkap tren utama pergerakan harga saham
- Prediksi mengikuti pola aktual dengan baik, terutama untuk pergerakan jangka menengah
- Ada sedikit lag dalam merespon perubahan tajam (volatilitas tinggi)
- Model cenderung menghasilkan prediksi yang lebih smooth dibanding data aktual

3. Metrik Evaluasi:

- R2 score 0.9425 menunjukkan model menjelaskan 94.25% variasi dalam data
- RMSE 73.59 relatif kecil dibanding range harga 4000-4800
- MAE 55.83 menunjukkan rata-rata deviasi prediksi sekitar 1.3% dari nilai aktual
- Std predicted (168.83) sedikit lebih kecil dari std actual (192.75) yang menjelaskan smoothing effect

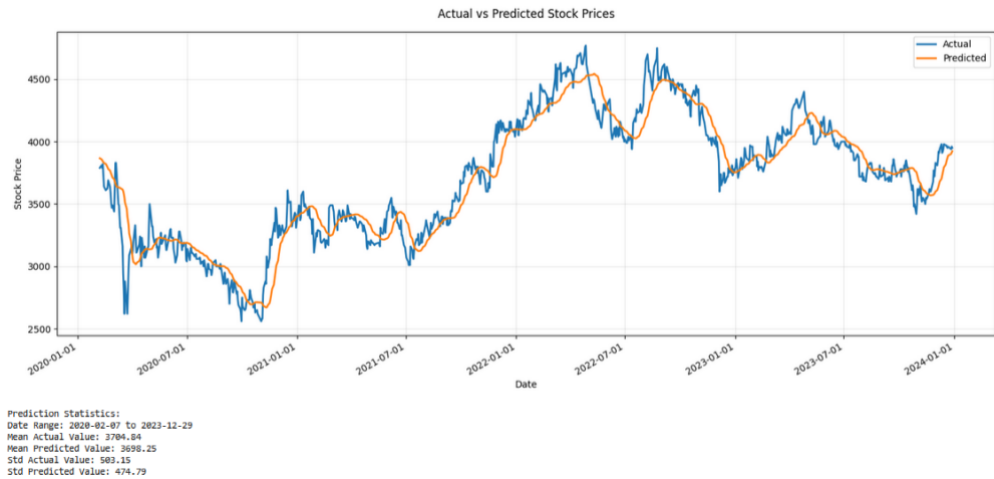
4. Kelebihan Model:

- Performa prediksi sangat baik ( $R^2 > 0.94$ )
- Stabil dalam training tanpa overfitting
- Mampu menangkap tren jangka pendek dan menengah

- Attention mechanism membantu model fokus pada pola yang relevan

5. Area Perbaikan Potensial:

- Respon terhadap volatilitas tinggi bisa ditingkatkan
- Lag dalam prediksi bisa dikurangi
- Smoothing effect bisa dikurangi jika diinginkan prediksi yang lebih volatile



Gambar 4.3.8 Grafik Aktual dan Prediksi 07-02-2020 sampai 29-12-2023 dengan ADAM

Pada gambar 4.3.8 menunjukkan Model dengan optimasi Adam memberikan prediksi yang nilai hampir mirip dengan nilai aktualnya.

Sebagai tambahan, berikut komparasi model tanpa menggunakan Adam Optimizer

Table 4.3.2 Komparasi GRU dengan dan tanpa optimizer

	Tanpa Optimizer	Dengan Adam Optimizer
<b>RMSE</b>	167.6167	156.3990
<b>MAE</b>	126.2676	116.0790
<b>R2</b>	0.8890	0.9034

Dari table 4.3.2 dapat disimpulkan sebagai berikut:

1. Model *dengan* Adam Optimizer memiliki RMSE yang lebih rendah, menunjukkan bahwa rata-rata kesalahan prediksinya (dengan penekanan pada kesalahan besar) lebih kecil dibandingkan model tanpa Optimizer.
2. Model *dengan* Adam Optimizer memiliki MAE yang lebih rendah. Ini berarti, secara rata-rata, prediksi model dengan Adam lebih dekat ke nilai aktual dibandingkan model tanpa optimizer
3. Model *dengan* Adam Optimizer memiliki R2 yang lebih tinggi. Ini menunjukkan bahwa model dengan Adam mampu menjelaskan sekitar 90.34% variasi harga saham aktual, lebih baik daripada model tanpa optimizer yang menjelaskan 88.90%.

#### 4.8 Diskusi

Terkait Penanganan Gradien yang Efisien di Lanskap Loss yang Kompleks:

Prediksi finansial, terutama dengan model sekuensial kompleks seperti GRU + Attention, seringkali memiliki loss landscape (permukaan fungsi kerugian) yang sangat kompleks, tidak mulus, dengan banyak lembah (minimum lokal), dataran tinggi (plateau), dan "jurang" curam. Oleh karena itu, dengan penggunaan Adam optimizer dapat mengatasi hal ini dengan:

- Momentum (Estimasi Momen Pertama): Adam menyimpan "ingatan" arah gradien sebelumnya (seperti bola salju yang menggelinding menuruni bukit). Ini membantunya melewati minimum lokal yang dangkal dan mempercepat konvergensi di arah yang konsisten.
- Adaptive Learning Rates (Estimasi Momen Kedua - seperti RMSprop): Adam menghitung learning rate yang berbeda untuk setiap parameter dalam model. Learning rate ini disesuaikan berdasarkan rata-rata kuadrat gradien masa lalu. Parameter yang memiliki gradien besar/berisik akan mendapatkan learning rate lebih kecil (untuk stabilitas), sementara parameter dengan gradien kecil/konsisten mendapatkan learning rate lebih besar (untuk percepatan). Ini sangat berguna dalam model dengan banyak parameter seperti GRU/Attention yang kita lakukan, di mana setiap parameter mungkin memerlukan penyesuaian yang berbeda.

Konvergensi cukup cepat dikarenakan Kombinasi momentum dan adaptive learning rates seringkali membuat Adam konvergen (menemukan solusi yang baik) lebih cepat daripada optimizer lain dalam jumlah epoch yang sama. Terlebih, penggunaan

EarlyStopping dan ReduceLRonPlateau bekerja sinergis dengan Adam. Adam membantu mencapai loss validasi yang baik lebih cepat, sehingga EarlyStopping mungkin terpicu lebih awal, menghemat waktu komputasi. ReduceLRonPlateau juga bisa lebih efektif karena Adam sudah berusaha menyesuaikan laju belajar secara adaptif.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan hasil evaluasi model terhadap data dari **7 Februari 2020** hingga **7 Juli 2020**, diperoleh beberapa temuan sebagai berikut:

##### 1. Kualitas Prediksi:

- Rata-rata nilai prediksi (**3342.03**) cukup dekat dengan rata-rata nilai aktual (**3260.70**), menunjukkan bahwa model mampu menangkap tren secara keseluruhan dengan baik.
- Standar deviasi prediksi (**263.70**) sedikit lebih besar dari standar deviasi aktual (**255.89**), yang menunjukkan bahwa model masih sedikit lebih sensitif terhadap fluktuasi harga.

##### 2. Akurasi Model:

- $R^2 = 0.9034$ , yang berarti model mampu menjelaskan **90.34%** dari variasi data aktual. Ini menunjukkan performa prediksi yang sangat baik.
- $RMSE = 156.40$ , yang berarti rata-rata kesalahan kuadrat dari prediksi masih dalam rentang yang cukup kecil dibandingkan nilai saham.
- $MAE = 116.08$ , yang berarti rata-rata kesalahan absolut model dalam memprediksi harga saham adalah sekitar **116 poin**.

##### 3. Analisis Kesalahan:

- RMSE yang lebih besar dari MAE menunjukkan bahwa ada beberapa kesalahan prediksi yang cukup besar (outliers) yang mempengaruhi performa model.
- Meskipun model menangkap pola harga dengan baik, ada kemungkinan model masih mengalami kesulitan dalam merespons perubahan harga yang sangat tajam.

4. Dari penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma Deep Learning berbasis Gated Recurrent Unit menggunakan ADAM Optimizer dapat digunakan untuk prediksi saham"

Dari temuan tersebut, dapat disimpulkan penelitian ini berhasil meningkatkan performa algoritma Deep Learning berbasis Gated Recurrent Unit dengan memanfaatkan ADAM Optimizer untuk prediksi saham

## 5.2 Saran

Model sudah memiliki performa yang cukup baik ( $R^2 = 0.90$ ) tetapi masih bisa ditingkatkan dengan:

- Meningkatkan sensitivitas terhadap volatilitas.
- Mengurangi error dengan fitur tambahan dan teknik ensemble.
- Mengatasi lag dengan pendekatan Seq2Seq atau multi-horizon forecasting.
- Menerapkan regularisasi yang lebih baik untuk mencegah overfitting.

Dengan beberapa perbaikan ini, diharapkan model bisa lebih akurat dalam memprediksi harga saham dengan volatilitas tinggi dan tren yang lebih kompleks.

## DAFTAR PUSTAKA

- [1] A. Peivandizadeh *et al.*, “Stock Market Prediction With Transductive Long Short-Term Memory and Social Media Sentiment Analysis,” *IEEE Access*, vol. 12, pp. 87110–87130, 2024, doi: 10.1109/ACCESS.2024.3399548.
- [2] T.-W. Lee, P. Teisseyre, and J. Lee, “Effective Exploitation of Macroeconomic Indicators for Stock Direction Classification Using the Multimodal Fusion Transformer,” *IEEE Access*, vol. 11, pp. 10275–10287, 2023, doi: 10.1109/ACCESS.2023.3240422.
- [3] J. Choi, S. Yoo, X. Zhou, and Y. Kim, “Hybrid Information Mixing Module for Stock Movement Prediction,” *IEEE Access*, vol. 11, pp. 28781–28790, 2023, doi: 10.1109/ACCESS.2023.3258695.
- [4] C. Zhao, Z. Peng, X. Lan, Y. Cen, and Z. Wang, “Structured Multi-Head Attention Stock Index Prediction Method Based Adaptive Public Opinion Sentiment Vector,” *Comput. Mater. Contin.*, vol. 78, no. 1, pp. 1503–1523, Jan. 2024, doi: 10.32604/cmc.2024.039232.
- [5] Y. Li, L. Chen, C. Sun, G. Liu, C. Chen, and Y. Zhang, “Accurate Stock Price Forecasting Based on Deep Learning and Hierarchical Frequency Decomposition,” *IEEE Access*, vol. 12, pp. 49878–49894, 2024, doi: 10.1109/ACCESS.2024.3384430.
- [6] X. Yuan, J. Yuan, T. Jiang, and Q. U. Ain, “Integrated Long-Term Stock Selection Models Based on Feature Selection and Machine Learning Algorithms for China Stock Market,” *IEEE Access*, vol. 8, pp. 22672–22685, 2020, doi: 10.1109/ACCESS.2020.2969293.
- [7] S. M. Idrees, M. A. Alam, and P. Agarwal, “A Prediction Approach for Stock Market Volatility Based on Time Series Data,” *IEEE Access*, vol. 7, pp. 17287–17298, 2019, doi: 10.1109/ACCESS.2019.2895252.
- [8] Y.-L. Hsu, Y.-C. Tsai, and C.-T. Li, “FinGAT: Financial Graph Attention Networks for Recommending Top-K Profitable Stocks,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 469–481, 2023, doi: 10.1109/TKDE.2021.3079496.
- [9] C. Bousono-Calzon, J. Bustarviejo-Munoz, P. Aceituno-Aceituno, and J. J. Escudero-Garzas, “On the Economic Significance of Stock Market Prediction and the No Free Lunch Theorem,” *IEEE Access*, vol. 7, pp. 75177–75188, 2019, doi: 10.1109/ACCESS.2019.2921092.
- [10] F. Wang, M. Li, Y. Mei, and W. Li, “Time Series Data Mining: A Case Study With Big Data Analytics Approach,” *IEEE Access*, vol. 8, pp. 14322–14328, 2020, doi: 10.1109/ACCESS.2020.2966553.

- [11] J. Chen, L. Xie, W. Lin, Y. Wu, and H. Xu, “Multi-Granularity Spatio-Temporal Correlation Networks for Stock Trend Prediction,” *IEEE Access*, vol. 12, pp. 67219–67232, 2024, doi: 10.1109/ACCESS.2024.3393774.
- [12] Y. Wang, H. Liu, Q. Guo, S. Xie, and X. Zhang, “Stock volatility prediction by hybrid neural network,” *IEEE Access*, vol. 7, pp. 154524–154534, 2019, doi: 10.1109/ACCESS.2019.2949074.
- [13] M. Wen, P. Li, L. Zhang, and Y. Chen, “Stock Market Trend Prediction Using High-Order Information of Time Series,” *IEEE Access*, vol. 7, pp. 28299–28308, 2019, doi: 10.1109/ACCESS.2019.2901842.
- [14] S. Chen and C. Zhou, “Stock Prediction Based on Genetic Algorithm Feature Selection and Long Short-Term Memory Neural Network,” *IEEE Access*, vol. 9, pp. 9066–9072, 2021, doi: 10.1109/ACCESS.2020.3047109.
- [15] T. Yin, X. Du, W. Zhang, Y. Zhao, B. Han, and J. Yan, “Real-Trading-Oriented Price Prediction With Explainable Multiobjective Optimization in Quantitative Trading,” *IEEE Access*, vol. 10, pp. 57685–57695, 2022, doi: 10.1109/ACCESS.2022.3175842.
- [16] N. Kanungsukkasem and T. Leelanupab, “Financial Latent Dirichlet Allocation (FinLDA): Feature Extraction in Text and Data Mining for Financial Time Series Prediction,” *IEEE Access*, vol. 7, pp. 71645–71664, 2019, doi: 10.1109/ACCESS.2019.2919993.
- [17] M. Nabipour, P. Nayyeri, H. Jabani, S. S., and A. Mosavi, “Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis,” *IEEE Access*, vol. 8, pp. 150199–150212, 2020, doi: 10.1109/ACCESS.2020.3015966.
- [18] S. Wang, “A Stock Price Prediction Method Based on BiLSTM and Improved Transformer,” *IEEE Access*, vol. 11, pp. 104211–104223, 2023, doi: 10.1109/ACCESS.2023.3296308.
- [19] H. D. Chacon, E. Kesici, and P. Najafirad, “Improving Financial Time Series Prediction Accuracy Using Ensemble Empirical Mode Decomposition and Recurrent Neural Networks,” *IEEE Access*, vol. 8, pp. 117133–117145, 2020, doi: 10.1109/ACCESS.2020.2996981.
- [20] A. Q. Md *et al.*, “Novel optimization approach for stock price forecasting using multi-layered sequential LSTM,” *Appl. Soft Comput.*, vol. 134, p. 109830, Feb. 2023, doi: 10.1016/j.asoc.2022.109830.

- [21] Z. Mustaffa and M. H. Sulaiman, "Stock price predictive analysis: An application of hybrid Barnacles Mating Optimizer with Artificial Neural Network," *Int. J. Cogn. Comput. Eng.*, vol. 4, pp. 109–117, Jun. 2023, doi: 10.1016/j.ijcce.2023.03.003.
- [22] C. Zhao, S. Yang, C. Qin, J. Zhou, and L. Chen, "A Novel Smart Beta Optimization Based on Probabilistic Forecast," *Comput. Mater. Contin.*, vol. 75, no. 1, pp. 477–491, Jan. 2023, doi: 10.32604/cmc.2023.034933.
- [23] G. Mu, N. Gao, Y. Wang, and L. Dai, "A Stock Price Prediction Model Based on Investor Sentiment and Optimized Deep Learning," *IEEE Access*, vol. 11, pp. 51353–51367, 2023, doi: 10.1109/ACCESS.2023.3278790.
- [24] Yahoo, "Perusahaan Perseroan (Persero) PT Telekomunikasi Indonesia Tbk (TLKM.JK)." [Online]. Available: <https://finance.yahoo.com/quote/TLKM.JK/>