

**PREDIKSI HARGA SAHAM BERDASARKAN SENTIMEN PUBLIK ATAS
LAYANAN TELEKOMUNIKASI MENGGUNAKAN PENDEKATAN
*GATED RECURRENT UNIT***

SKRIPSI

ERIC SAMUEL SIMBOLON

181402083



**PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024**

**PREDIKSI HARGA SAHAM BERDASARKAN SENTIMEN PUBLIK ATAS
LAYANAN TELEKOMUNIKASI MENGGUNAKAN PENDEKATAN
*GATED RECURRENT UNIT***

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana
Teknologi Informasi

ERIC SAMUEL SIMBOLON

181402083



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA MEDAN**

2024

PERSETUJUAN

iii

PERSETUJUAN

Judul : **Prediksi Harga Saham berdasarkan sentimen publik atas layanan telekomunikasi menggunakan pendekatan *Gated Recurrent unit***

Kategori : Skripsi

Nama Mahasiswa : **Eric Samuel Simbolon**

Nomor Induk Mahasiswa : **181402083**

Program Studi : **Sarjana (S1) Teknologi Informasi**

Fakultas : **Fakultas Ilmu Komputer Dan Teknologi Informasi Universitas Sumatera Utara**

Medan, 10 Januari 2024

Komisi Pembimbing:

Pembimbing 1,



Dedy Arisandi S.T., M.Kom
NIP. 197908312009121002


Pembimbing 2,



Rossy Nurhasanah S.Kom., M.Kom
NIP. 198707012019032016

Diketahui/disetujui oleh
Program Studi S1-Teknologi Informasi

Ketua,


Dedy Arisandi, ST., M.Kom
NIP. 197908312009121002

PERNYATAAN

PREDIKSI HARGA SAHAM BERDASARKAN *SENTIMENT* PUBLIC ATAS
LAYANAN TELEKOMUNIKASI MENGGUNAKAN PENDEKATAN
GATED RECURRENT UNIT

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 10 Januari 2024

Eric Samuel Simbolon

181402083

UCAPAN TERIMA KASIH

Dengan penuh rasa syukur, penulis menyampaikan ucapan terima kasih kepada Tuhan Yang Maha Esa atas segala karunia-Nya yang telah membimbing penulis dalam menyelesaikan skripsi ini, yang merupakan salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi S1 Teknologi Informasi Universitas Sumatera Utara. Penulis tidak dapat menyelesaikan skripsi ini tanpa bantuan dari semua pihak, baik dalam bentuk bimbingan, doa, maupun dukungan. Pada kesempatan ini, dengan segala kerendahan hati, penulis menyampaikan ucapan terima kasih kepada:

1. Keluarga tercinta, orangtua Ayah Fendi Harli Simbolon dan Ibu Mega Hutagalung dan juga abang saya Kevin Valentino Simbolon.
2. Bapak Dr. Muryanto Amin, S.Sos., M.Si selaku Rektor Universitas Sumatera Utara.
3. Ibu Dr. Maya Silvi Lydia, M.Sc selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera.
4. Bapak Dedy Arisandi, ST., M.Kom. selaku Dosen Pembimbing I saya dan juga Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
5. Ibu Rossy Nurhasanah S.Kom., M.Kom selaku Dosen Pembimbing II yang telah membimbing penulis sampai penyelesaian skripsi ini.
6. Bapak Indra Aulia S.TI., M.Kom selaku Mantan Dosen Pembimbing I satu saya dahulu yang membantu penulis dalam penyelesaian skripsi ini.
7. Terima kasih kepada Palis, Parhan, Felix, Kapi, Hari, Jaki, Abi, Eldwin, Hemlut, M. Luthfi, Ridho, Yedija, M. Daifulah, Raihan, dan Hafizha Azhar yang telah menjadi bagian tak terpisahkan dari setiap langkah penulis. Kebersamaan, semangat, dan kerja sama yang luar biasa telah membuat proses ini menjadi perjalanan yang berharga.
8. Rekan-rekan sesama mahasiswa angkatan 2018 di Program Studi Teknologi Informasi Universitas Sumatera Utara.

Tuhan yang Maha Esa senantiasa melimpahkan karunia-Nya kepada semua pihak yang telah memberikan doa dan dukungan kepada penulis dalam menyelesaikan skripsi ini.

Medan, 10 Januari 2024

Eric Samuel Simbolon

ABSTRAK

Pasar saham merupakan instrumen investasi yang sangat populer di Indonesia, yang dipengaruhi oleh sejumlah faktor termasuk sentimen publik terhadap layanan telekomunikasi. Penelitian ini bertujuan untuk menganalisis dan memprediksi pergerakan harga saham Telkom berdasarkan sentimen publik di platform *Twitter*, dengan menerapkan pendekatan *deep learning* menggunakan *Gated Recurrent Unit* (GRU). Data *Twitter* yang digunakan khususnya mencakup *tweet* yang merujuk pada layanan Telkom (\$TLKM.JK), sementara data historis saham dari *Yahoo Finance* digunakan sebagai pendukung. Analisis sentimen dilakukan dengan memanfaatkan *VADER* untuk mengklasifikasikan sentimen menjadi positif, negatif, atau netral. Data sentimen dibagi menjadi 80% untuk proses pelatihan dan 20% untuk pengujian model. Berbeda dengan studi sebelumnya yang menggunakan model LSTM dan melaporkan RMSE sebesar 1120,6517, hasil penelitian ini menunjukkan bahwa model GRU dapat memprediksi harga saham Telkom dengan tingkat akurasi mencapai 90%. Hasil evaluasi model ini menunjukkan MSE sebesar 102,43 dan RMSE sebesar 10,120770.

Kata kunci: Pasar saham, harga saham, sentimen publik, *Twitter*, *deep learning*, *Gated recurrent Unit*, *Vader*

**PREDICTION OF STOCK PRICES BASED ON PUBLIC SENTIMENT ON
TELECOMMUNICATIONS SERVICES USING
GATED RECURRENT UNIT**

ABSTRACT

Stock market serves as a highly popular investment instrument in Indonesia, influenced by various factors, including public sentiment towards telecommunication services. This research aims to analyze and predict the movement of Telkom's stock prices based on public sentiment on the Twitter platform, employing a deep learning approach utilizing the Gated Recurrent Unit (GRU). The Twitter data used specifically includes tweets referring to Telkom's services (\$TLKM.JK), while historical stock data from Yahoo Finance is utilized as a supporting dataset. Sentiment Analysis is conducted using VADER to classify sentiments into positive, negative, or neutral categories. The sentiment data is split with 80% for the Training process and 20% for model testing. In contrast to previous studies using LSTM models and Reporting an RMSE of 1120.6517, the findings of this research indicate that the GRU model can predict Telkom's stock prices with an accuracy level reaching 90%. The evaluation results of this model show an MSE of 102.43 and an RMSE of 10.120770.

Keywords : Stock market, stock price, public sentiment, Twitter, deep learning, Gated recurrent unit, Vader

DAFTAR ISI

PERSETUJUAN	iii
PERNYATAAN.....	iv
UCAPAN TERIMAKASIH	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian	5
1.7 Sistematika Penulisan.....	6
BAB II LANDASAN TEORI.....	7
2.1 <i>Aspect-based Sentiment Analysis</i> (ABSA).....	7
2.2 Pasar Modal	8
2.3 <i>Deep learning</i>	8
2.3.1 <i>Gated Recurrent Unit</i> (GRU)	9
2.4 <i>Term Frequency / Inverse Document</i> (TF-IDF)	12
2.5 <i>Valence Aware Dictionary dan Sentiment Reasoner</i> (Vader)	13
2.6 <i>Data Scraping</i>	13
2.6.1 <i>Twitter API</i>	15
2.6.2 <i>Yahoo Finance API</i>	15
2.7 <i>Text mining</i>	16
2.8 Metode Evaluasi	17
2.9 Penelitian Terdahulu.....	19

BAB III ANALISIS DAN PERANCANGAN SISTEM	26
3.1 Arsitektur Umum.....	26
3.2 Pengumpulan Data	26
3.2.1 <i>Twitter</i>	26
3.2.2 Data Saham.....	33
3.3 <i>Preprocessing</i> Data	38
3.3.1 <i>Normalization</i>	38
3.3.2 <i>Case folding</i>	39
3.3.3 <i>Stemming</i>	40
3.3.4 <i>Punctuation Removal</i>	41
3.3.5 <i>Stopword Removal</i>	42
3.3.6 <i>Tokenization</i>	43
3.3.7 Menghilangkan <i>Missing value</i>	44
3.4 <i>Word Embedding</i> - TFIDF	45
3.5 <i>Vader</i>	47
3.6 Perancangan Model algoritma <i>Gated Recurrent Unit</i>	49
3.6.1 Pembentukan <i>Input</i> dan <i>Output</i>	49
3.6.2 Inisialisasi Model.....	50
3.6.3 Penentuan <i>Hyperparameter</i>	50
3.6.4 Pelatihan Model.....	50
3.6.5 Evaluasi Model	50
3.6.6 Penyimpanan Model	50
3.7 <i>Output</i>	50
3.8 <i>Flowchart</i> Diagram	50
3.9 Rancangan Sistem	52
3.9.1 Antarmuka Pengguna	52
BAB IV IMPLEMENTASI DAN PENGUJIAN	57
4.1 Perangkat Keras.....	57
4.2 Perangkat Lunak.....	57
4.3 Implementasi Perancangan Tampilan Antarmuka	58
4.3.1 Halaman Beranda	58
4.3.2 Halaman <i>Training</i>	58
4.3.3 Halaman <i>Testing</i>	60

4.4	Pengujian Sistem	63
4.4.1	<i>Hyperparameter Model Gated Recurrent Unit</i>	63
4.4.2	Hasil Pengujian Sistem	64
4.4.3	Evaluasi Model	68
BAB V	PENUTUP	71
5.1	Kesimpulan.....	71
5.2	Saran.....	71
DAFTAR PUSTAKA	73

DAFTAR GAMBAR

Gambar 2. 1	Arsitektur GRU	10
Gambar 2. 2	Alur proses pada data <i>Scraping</i>	14
Gambar 3. 1	Arsitektur Umum.....	27
Gambar 3. 2	Cuitan Pengguna Terhadap Produk Telkom di <i>Twitter</i>	27
Gambar 3. 3	Data Saham TLKM di <i>Yahoo Finances</i>	33
Gambar 3. 4	<i>Flowchart</i> Diagram <i>Website</i> rancangan	52
Gambar 3. 5	Halaman Utama Beranda.....	53
Gambar 3. 6	Halaman <i>Training</i>	54
Gambar 3. 7	Halaman <i>Testing</i>	55
Gambar 4. 1	Tampilan antarmuka halaman Beranda	58
Gambar 4. 2	Antarmuka halaman <i>Training</i> Data <i>Twitter</i>	59
Gambar 4. 3	Antarmuka halaman <i>Training</i> Data Saham	59
Gambar 4. 4	Halaman <i>testing</i> dengan hasil prediksi <i>sentiment</i>	60
Gambar 4. 5	Halaman <i>testing</i> dengan hasil prediksi saham.....	61
Gambar 4. 6	Halaman <i>testing</i> dengan hasil Evaluasi MSE, RMSE dan <i>Chart Comparison</i>	62
Gambar 4. 7	Halaman <i>testing</i> dengan hasil Evaluasi <i>Classification Report</i> dan <i>Confusion Matrix</i>	63
Gambar 4. 8	<i>Confusion Matrix</i>	68
Gambar 4. 9	Perbandingan Harga Saham Prediksi Dengan Harga Asli	70

DAFTAR TABEL

Tabel 2. 1	Nilai <i>Confusion Matrix</i>	18
Tabel 2. 2	Penelitian Terdahulu	22
Tabel 3. 1	Contoh hasil <i>crawling</i> Data <i>Twitter</i>	31
Tabel 3. 2	Contoh hasil <i>crawling</i> Data Historis Saham.....	37
Tabel 3. 3	Contoh Hasil Normalisasi.....	39
Tabel 3. 4	Contoh Kamus Normalisasi Kata	39
Tabel 3. 5	Tabel Hasil <i>Case folding</i>	40
Tabel 3. 6	Tabel Hasil <i>Stemming</i>	41
Tabel 3. 7	Tabel Hasil <i>Punctuation Removal</i>	42
Tabel 3. 8	Tabel Hasil <i>Stopword Removal</i>	43
Tabel 3. 9	<i>Stopword corpus</i>	43
Tabel 3. 10	Tabel Hasil <i>Tokenization</i>	44
Tabel 3. 11	Contoh Tabel Hasil Pembobotan TF-IDF.....	46
Tabel 3. 12	Hasil analisis <i>sentiment</i> menggunakan <i>Vader</i> yang telah dibantu TF-IDF.....	49
Tabel 4. 1	Tabel Hasil <i>Hyperparameter</i>	63
Tabel 4. 2	Hasil Prediksi Model <i>Gated Recurrent Unit</i>	65
Tabel 4. 3	<i>Classification Report</i> dan <i>Confusion Matrix</i>	68
Tabel 4. 4	Keterangan Prediksi dan <i>Actual Confusion Matrix</i>	69

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pasar saham merupakan tempat di mana terjadi kegiatan jual-beli saham, melibatkan penjual, pembeli, serta lembaga dan individu dengan kepentingan dalam saham. Pihak utama yang terlibat, seperti investor, spekulasi, dan pemerintah, memiliki tujuan yang sama, yaitu mencapai laba maksimal melalui analisis fundamental dan teknikal. Tren yang sangat fluktuatif di pasar saham, yang dikenal sebagai volatilitas, menciptakan tantangan dalam memprediksi pergerakan saham. Peneliti tertarik untuk mengembangkan teknik canggih guna meramal harga saham dengan tingkat akurasi yang tinggi, karena prediksi yang tepat dapat menghasilkan keuntungan yang signifikan (Usmani & Shamsi, 2021).

Dalam upaya memprediksi tren saham, ada dua pendekatan dasar yang sering digunakan, yaitu analisis teknis dan analisis fundamental. Analisis teknis mencermati data historis dan volume harga saham, sementara analisis fundamental tidak hanya mempertimbangkan statistik saham tetapi juga mengevaluasi kinerja industri, peristiwa politik, dan keadaan ekonomi (Patel *et al.*, 2015). Analisis fundamental dianggap lebih realistis karena mengevaluasi pasar dalam cakupan yang lebih luas.

Sumber data untuk analisis fundamental dapat diperoleh dari berita, *tweet*, laporan tahunan, dan sumber lainnya. Data tekstual tersebut dapat dianalisis lebih mendalam untuk menggali informasi yang relevan. Data tekstual, terutama berita, dianggap sebagai sumber informasi yang lebih baik daripada data numerik karena memungkinkan prediksi tren finansial dengan landasan yang jelas (seperti pembenaran atau alasan) (Chan & Franklin, 2011). Artikel berita yang mencakup kata kunci atau frasa seperti "pengunduran diri" atau "risiko gagal bayar" dapat membantu investor memprediksi penurunan harga saham. Berita yang tidak pasti, seperti perang, terorisme, bencana alam, dan peristiwa politik, juga dapat memengaruhi tren pasar (Nassirtoussi *et al.*, 2015). Oleh karena itu, perlu dilakukan penelitian lebih

mendalam untuk mengekstrak informasi dari data tekstual dengan hasil yang lebih baik.

Data tekstual untuk penelitian ini akan diperoleh dari situs microblogging *Twitter*. *Twitter*, dengan lebih dari 500 juta pengguna dan 400 juta *tweet* per hari menyediakan wadah untuk berbagi pendapat dan sentimen masyarakat. *Twitter* dapat digunakan sebagai sumber data untuk analisis sentimen, yaitu studi komputasional terhadap opini, sentimen, dan emosi melalui teks dalam kalimat atau dokumen (Pang & Lee, 2008).

Studi sentimen terhadap respon masyarakat di *Twitter* telah menjadi fokus penelitian di berbagai negara. Beberapa penelitian bahkan menghubungkan respon *Twitter* dengan harga saham atau nilai tukar. Wardhani *et al.*, (2020) menunjukkan bahwa ada hubungan yang signifikan antara respon *Twitter* dan harga saham, meskipun nilainya cukup kecil. Namun, penelitian oleh Nisar & Yeung (2018) menunjukkan bahwa hubungan tersebut tidak signifikan. Penelitian lain oleh Sul, Dennis, & Yuan (2016) menyimpulkan bahwa sentimen dalam media sosial dapat memprediksi *return* saham di masa depan. Studi ini sejalan dengan penelitian oleh Shi (2022), yang menemukan bahwa sentimen investor memiliki efek positif dalam jangka panjang dan sebaliknya pada jangka menengah.

Penelitian ini akan fokus pada analisis sentimen terhadap *tweet* berbahasa Indonesia yang membicarakan merek atau *brand provider* seluler yang memiliki popularitas tinggi. Pertumbuhan pengguna telekomunikasi di Indonesia yang terus meningkat menimbulkan persaingan antar *provider* untuk menarik dan mempertahankan pelanggan. Opini dan sentimen pelanggan mengenai *provider* dapat tercermin dalam media sosial seperti *Twitter*. Oleh karena itu, analisis sentimen di media sosial dapat menjadi indikator kualitas produk dan layanan yang diberikan oleh *provider* kepada pelanggan.

Pemilihan GRU sebagai metode pengolahan data teks, terutama dalam menganalisis *tweet* berbahasa Indonesia yang membahas merek atau *brand provider* seluler, didasarkan pada kemampuannya dalam menangani data berbasis urutan. GRU memiliki keunggulan dalam memahami konteks jangka panjang dan mengatasi masalah *vanishing gradient* yang mungkin terjadi pada model *Recurrent Neural Network* (RNN) tradisional. Oleh karena itu, melalui pendekatan GRU, diharapkan penelitian ini dapat memberikan kontribusi positif dalam meningkatkan akurasi prediksi harga saham berdasarkan sentimen publik di media sosial.

Berdasarkan latar belakang yang telah dipaparkan diatas, penulis ingin

mengajukan pengembangan algoritma yang dapat membantu para investor agar mendapat informasi yang lebih baik dalam melakukan investasi Saham kedepannya dengan judul “**Prediksi Harga Saham Berdasarkan Sentimen Publik Atas Layanan Telekomunikasi Menggunakan Pendekatan *Gated Recurrent Unit***”.

1.2 Rumusan Masalah

Harga saham di pasar saham memiliki tren fluktuatif yang bergantung kepada suatu kondisi fundamental. Kondisi fundamental saat ini banyak diberikan masyarakat melalui media sosial sebagai wujud ekspresi masyarakat terhadap situasi industri, peristiwa politik dan/atau keadaan ekonomi. Meneliti pasar dengan Teknik fundamental secara konvensional membutuhkan waktu yang cukup lama. Masyarakat perlu mengumpulkan banyak berita dari media sosial lalu menentukan sendiri jika berita tersebut itu baik atau bukan terhadap Saham suatu perusahaan. Hal ini tentu membutuhkan waktu yang cukup lama dan juga dibutuhkannya pakar agar bisa memberikan penilaian maksimal dalam sebuah berita. Oleh karena itu dibutuhkan sebuah sistem yang bisa meminimalisir waktu dalam mengumpulkan berita dan penilaian pakar agar bisa lebih efektif dan efisien kedepannya.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk memprediksi harga saham berdasarkan sentimen publik di media sosial dengan mengimplementasikan algoritma *Gated Recurrent Unit* (GRU)

1.4 Batasan Masalah

Untuk bisa memberikan hasil yang akurat dan presisi serta mencapai tujuan penelitian, penulis perlu memberikan Batasan masalah pada penelitian ini. Adapun Batas masalah yang dibuat pada penelitian ini adalah sebagai berikut:

1. Data yang digunakan pada penelitian ini hanya kumpulan *tweets* berbahasa Indonesia hasil *Scraping* dari platform media sosial *Twitter* yang disimpan dalam file dokumen dengan format .csv.
2. Kalimat *tweet* yang digunakan pada penelitian terdiri dari beberapa kata, tanpa angka, gambar ataupun emoji.
3. Penelitian ini hanya akan memprediksi harga saham perusahaan telekomunikasi

yaitu Telkom (TLKM.JK).

4. Penelitian ini hanya akan mencakup periode waktu 1 Januari 2018 – 31 Desember 2022
5. Hasil dari penelitian ini akan memprediksi setiap 7 hari atau per minggu
6. Data Saham akan diambil dari *Yahoo Finances*.
7. Proses memprediksi harga hanya berdasarkan sentimen positif dan negatif pada sebuah *tweet*
8. Penelitian ini tidak akan mengambil faktor-faktor lain yang mungkin mempengaruhi harga saham selain sentimen publik, seperti kondisi ekonomi, politik, dan faktor lainnya.

1.5 Manfaat Penelitian

Adapun manfaat dari penelitian yang akan dicapai sebagai berikut:

1. Penelitian ini dapat menghasilkan model prediksi harga saham yang dapat menggunakan data sentimen publik sebagai *input* untuk memprediksi pergerakan harga saham perusahaan telekomunikasi.
2. Penelitian ini dapat memberikan informasi tentang pola dan tren sentimen publik terkait layanan telekomunikasi yang dapat digunakan sebagai bahan pertimbangan bagi perusahaan telekomunikasi dalam mengambil keputusan bisnis.
3. Penelitian ini dapat memberikan solusi berbasis teknologi yang dapat membantu investor dalam membuat keputusan investasi yang lebih baik.
4. Penelitian ini dapat menjadi salah satu contoh aplikasi *deep learning* dalam bidang keuangan untuk memprediksi pergerakan harga saham berdasarkan sentimen publik.
5. Penelitian ini dapat menambah wawasan dan pengetahuan tentang bagaimana *deep learning* dapat digunakan untuk memprediksi harga saham berdasarkan sentimen publik.
6. Menjadi referensi di bagian *Text processing* dengan menggunakan algoritma *Gated Recurrent Unit* di penelitian masa depan.

1.6 Metodologi Penelitian

Untuk melakukan penelitian akan dilakukan tahap tahap sebagai berikut:

1. Studi Literatur

Pada tahap studi literatur, penulis mengumpulkan beberapa data referensi berupa jurnal, buku, artikel, dan sumber bacaan lainnya yang berkaitan pada penelitian, dan referensi mengenai text *Preprocessing*, *Word Embedding*, dan metode *Gated Recurrent Unit*

2. Analisis Permasalahan

Setelah studi literatur, selanjutnya adalah tahap analisis permasalahan. Analisis Permasalahan merupakan langkah yang dilakukan penulis guna memahami konsep *Gated Recurrent Unit* yang digunakan pada penelitian untuk menganalisis kalimat ulasan pengguna *Twitter* serta melihat efek *sentiment* terhadap harga saham yang ditentukan di batasan masalah.

3. Pengumpulan Data

Dalam mendapatkan hasil penelitian yang akurat perlu dilakukannya pengumpulan data yang diambil melalui *crawling* dari *Twitter* dan *Yahoo Finance*. Data akan diukur validasinya berdasarkan kaidah Bahasa Indonesia yang tepat serta terpusat berdasarkan aspek yang sudah ditentukan.

4. Perancangan dan Implementasi Sistem

Tahap ini akan merencanakan struktur umum sistem sebelum dilakukannya tahap penerapan. Proses perencanaan akan menggunakan tiga tahapan, yaitu pra-pengolahan, ekstraksi fitur, pelatihan data, dan pengujian data. Setelah dilakukan perencanaan, maka pelaksanaan akan dilakukan berdasarkan struktur umum tersebut. Proses pelaksanaan akan dilakukan untuk menyelesaikan masalah yang telah ditentukan.

5. Pengujian Sistem

Tahap ini dilakukan untuk melakukan tolak ukur uji berdasarkan implementasi yang dilakukan, hasil aktual akan diukur secara sistematis dengan menggunakan metode evaluasi sesuai untuk menghasilkan nilai akurasi yang akurat dan tepat.

6. Dokumentasi dan Penyusunan Laporan

Hasil dalam rancangan, proses dan pengujian berdasarkan implementasi sistem akan di dokumentasi dan hasil analisis akan dilampirkan dalam laporan. Dokumentasi akan dilakukan dengan sedemikian detail dan akurat untuk

menyampaikan hasil yang telah dilaksanakan.

1.7 Sistematika Penulisan

Struktur penyusunan laporan skripsi ini dibagi menjadi lima bagian, yakni:

Bab 1 : Pendahuluan

Bab I ini akan menguraikan latar belakang dan tujuan umum penelitian. Subbab-subbab yang akan dijabarkan meliputi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metodologi penelitian, dan tata penyusunan laporan.

Bab 2 : Landasan Teori

Bagian landasan teori merupakan penyajian terstruktur mengenai konsep-konsep yang digunakan. Pembahasan kerangka teori akan menampilkan informasi mengenai algoritma dan proses yang berdasarkan arsitektur umum yang diterapkan dalam penelitian ini.

Bab 3 : Analisis dan Perancangan

Penjelasan mengenai perancangan arsitektur umum menjadi fokus di bab ini, yang melibatkan detail proses dan tahapan arsitektur umum. Penetapan langkah-langkah arsitektur didasarkan pada penelitian sebelumnya yang relevan, dengan penyesuaian yang sesuai dengan konteks penelitian penulis.

Bab 4 : Implementasi dan Pengujian

Bagian ini mencakup penerapan hasil perancangan yang telah diuraikan sebelumnya. Proses uji coba akan dilakukan secara teliti, menampilkan perbandingan antara hasil prediksi dan hasil aktual dengan menggunakan metode evaluasi yang telah ditetapkan.

Bab 5 : Kesimpulan dan Saran

Pada bab akhir ini, disampaikan kesimpulan berdasarkan hasil penelitian, serta rekomendasi untuk penelitian selanjutnya guna perbaikan dan peningkatan hasil maksimal.

BAB II

LANDASAN TEORI

2.1 *Aspect-based Sentiment Analysis (ABSA)*

Aspect Based *Sentiment Analysis* (ABSA) adalah proses penentuan dari sebuah kalimat untuk menyatakan *sentiment* terhadap aspek yang ditentukan. ABSA merupakan turunan dari *Sentiment Analysis* yang merujuk pada sebuah sifat emosi yang diungkapkan dalam teks, implementasi adalah ABSA umumnya pada kalimat *review*, ungkapan perasaan dari media sosial ataupun pernyataan dari sebuah tanggapan. ABSA berfokus pada aspek yang sudah diekstrak berdasarkan kata kunci yang ditentukan baik itu dalam kata sifat, benda dan akan dilakukan Analisa berdasarkan *sentiment* terkait. dapat berupa positif, negatif, maupun netral pada tiap-tiap aspek (Pavlopoulos & Androutsopoulos, 2014).

Tujuan utama dalam penentuan ABSA umumnya digunakan untuk mempermudah Analisa secara detail dan mendalam terhadap teks ulasan secara otomatis, hal ini dapat mempermudah membuat kesimpulan serta memakan waktu lebih singkat dalam mengambil keputusan. Secara sederhananya ABSA dapat melakukan klasifikasi teks terhadap ulasan dan opini tertentu.

Proses penentuan ABSA dapat digunakan berdasarkan label atau kluster. Proses label akan diimplementasi berdasarkan validasi dari pihak ahli sastra terkait yang akan membantu penentuan label kata yang didasari kata kunci yang sudah ditentukan sebelumnya, sehingga memiliki data latih yang akurat, namun adanya ruang lingkup yang terbatas karena adanya pemilihan aspek yang terbatas. Lain hal dengan kluster proses penentuan aspek didasari langsung melalui algoritma yang sudah ditentukan dan menggunakan *unsupervised* Teknik, proses penentuan lebih mudah dan cepat namun tidak seakurat berdasarkan label berdasarkan verifikasi oleh ahli Bahasa.

2.2 Pasar Modal

Pasar modal menyediakan platform untuk berbagai transaksi efek seperti saham, obligasi, dan reksadana yang diterbitkan oleh pemerintah atau swasta, sesuai dengan Undang-Undang Republik Indonesia Nomor 8 Tahun 1995. Pasar modal dianggap sebagai opsi pendanaan alternatif bagi pemerintah dan swasta, dengan pemerintah dapat menerbitkan obligasi dan swasta dapat menerbitkan saham serta obligasi (Nasution, 2015).

Dalam klasifikasi pasar modal, terdapat empat jenis pasar, yaitu pasar perdana, pasar sekunder, pasar ketiga, dan pasar keempat. Pasar modal memainkan peran vital dalam perekonomian dengan meningkatkan efisiensi alokasi sumber daya, memacu pertumbuhan ekonomi, dan menciptakan lapangan kerja. Saham, sebagai salah satu bentuk efek di pasar modal, memberikan hak kepemilikan pada perusahaan dan keuntungan bagi pemegang saham, termasuk hak untuk memperoleh dividen dan berpartisipasi dalam Rapat Umum Pemegang Saham (RUPS) (Putri, 2015).

Saham juga menjadi instrumen investasi yang diminati, terdiri dari berbagai jenis seperti *blue-chip stock*, *income stocks*, *growth stocks*, *speculative stocks*, dan *counter cyclical stocks*. Keuntungan umum dari investasi saham adalah Capital Gain, tetapi juga membawa risiko seperti *Capital Loss*, Risiko Likuiditas, dan Saham *Delisting* dari Bursa. Prediksi pergerakan harga saham tetap menjadi tantangan, dan penelitian terus dilakukan untuk mengembangkan teknik-teknik canggih (Usmani & Shamsi, 2021).

2.3 Deep learning

Geoffrey Hinton pada tahun 2006 mengenalkan konsep *deep learning* melalui struktur jaringan saraf yang dikenal sebagai *deep belief nets*. Terjadi kemajuan besar dalam perkembangan *deep learning* setelah ditemukannya metode implementasi yang lebih efisien menggunakan unit *rendering* (GPU) di tahun 2009.

Deep learning, yang bersifat universal, dapat diaplikasikan pada berbagai ranah aplikasi dan memiliki kemampuan pembelajaran yang luas. Kekuatan *robust deep learning* terletak pada ketahanannya terhadap variasi data yang beragam tanpa memerlukan fitur yang dirancang khusus. *Deep learning* bahkan dapat secara otomatis mempelajari fitur-fitur optimal. *Deep learning* mempunyai kemampuan untuk diadaptasi ke kebutuhan yang kompleks, contohnya dapat dilihat pada jaringan ResNet buatan Microsoft yang terdiri dari 1.202 Layer dan diterapkan pada *supercomputer*

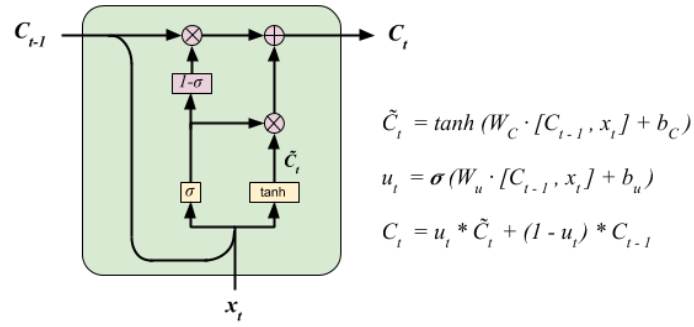
(Kumar *et al.*, 2016), dapat diintegrasikan pada sistem paralel dengan ribuan komputer (Van Essen *et al.*, 2015). Dari aspek metode pembelajarannya, *deep learning* ini bisa dibagi menjadi empat pembelajaran (Suyanto *et al.*, 2019) :

- a. Pembelajaran yang diawasi: *Deep learning* menciptakan fungsi untuk memetakan *input* ke *output* berdasarkan data berlabel yang diberikan. Fungsi ini dapat dipakai memberi solusi pada masalah klasifikasi atau regresi.
- b. Pembelajaran yang tidak diawasi : *Deep learning* secara otomatis memodelkan kumpulan *input* tanpa bantuan *output* yang diinginkan. Model ini dapat digunakan untuk menyelesaikan masalah klasterisasi.
- c. Pembelajaran semi-diawasi: *Deep learning* memakai sampel *input* yang beberapa memiliki label dan beberapa tidak. Model ini dapat digunakan untuk meningkatkan kinerja pembelajaran yang diawasi dengan memanfaatkan data yang tidak berlabel.
- d. Pembelajaran penguatan: *Deep learning* mempelajari kebijakan untuk melakukan tindakan berdasarkan pengamatan terhadap lingkungan. Setiap tindakan menghasilkan konsekuensi, dan umpan balik dari lingkungan digunakan untuk membimbing *deep reinforcement learning*.

2.3.1 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) merupakan salah satu varian unit *Recurrent Neural Network* (RNN) yang dimanfaatkan untuk memodelkan data berurutan. Pengenalan GRU pertama kali dilakukan oleh Cho, Merrienboer, Gulcehre, & Bougares (2014) melalui artikel berjudul "*Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*" pada tahun 2014.

GRU memiliki dua *gate*, yakni *forget gate* dan *update gate*. *Forget gate* berfungsi untuk menentukan informasi yang perlu diabaikan dari keadaan sebelumnya, sementara *update gate* bertugas untuk menentukan informasi yang perlu diperbarui dari keadaan sebelumnya. Gambar berikut menunjukkan arsitektur GRU:



Gambar 2. 1 Arsitektur GRU

Forget gate

Forget gate menentukan informasi apa yang harus dilupakan dari keadaan sebelumnya.

Forget gate dihitung dengan menggunakan persamaan 2.1 rumus berikut:

$$ft = \sigma(Wf \cdot h(t-1) + Uf \cdot xt) \quad (2.1)$$

Dimana:

ft : output dari *forget gate* pada waktu t

σ : fungsi aktivasi *sigmoid*

Wf dan Uf : bobot yang menghubungkan *input* dan *output forget out*

$h(t-1)$: keadaan RNN pada waktu $t-1$

xt : *input* RNN pada waktu t

Update gate

Update gate menentukan informasi apa yang harus diperbarui dari keadaan sebelumnya. *Update gate* dihitung dengan menggunakan persamaan 2.2 rumus berikut:

$$ut = \sigma(Wu \cdot h(t-1) + Uu \cdot xt) \quad (2.2)$$

Dimana :

ut : output dari *update gate* pada waktu t

σ : fungsi aktivasi *sigmoid*

Wu dan Uu : bobot yang menghubungkan *input* dan *output update gate*

Output Gate

Output Gate menentukan *output* dari RNN pada waktu t . *Output Gate* dihitung dengan menggunakan rumus persamaan 2.3 berikut:

$$ot = \sigma(Wo \cdot h(t-1) + Uo \cdot xt) \quad (2.3)$$

Dimana :

ot : *output* dari *Output Gate* pada waktu t

σ : fungsi aktivitas *sigmoid*

Wo dan Uo : bobot yang menghubungkan *input* dan *output Output Gate*

Output dari GRU adalah perkalian antara *output* dari *update gate* dan *output* dari keadaan RNN pada waktu t . Berikut adalah persamaan rumus 2.4 *outputnya*:

$$yt = ot \cdot ht \quad (2.4)$$

Untuk meningkatkan kinerja model GRU, perlu dilakukan optimasi pada beberapa parameter kunci, antara lain:

- a. Bobot pada Setiap *Gate*: Bobot pada setiap *gate* sangat memengaruhi kemampuan model dalam menangkap dan memahami pola data. Pengaturan bobot yang optimal dapat membantu model GRU lebih efektif dalam mengekstrak fitur dari data berurutan.
- b. *Learning Rate*: *Learning Rate* mengontrol seberapa besar langkah yang diambil selama proses pembelajaran. Jika *Learning Rate* terlalu kecil, model mungkin memerlukan waktu yang lama untuk konvergen. Sebaliknya, *Learning Rate* yang terlalu besar dapat menyebabkan model melewati titik optimum. Oleh karena itu, pemilihan *Learning Rate* yang sesuai perlu dioptimalkan.
- c. Jumlah *Epochs*: Jumlah *epochs* menentukan seberapa banyak iterasi dilakukan selama proses pelatihan. Terlalu sedikit *epochs* mungkin membuat model belum sempat memahami pola data secara menyeluruh, sementara terlalu banyak *epochs* dapat menyebabkan *overfitting*. Jumlah *epochs* perlu diatur sedemikian rupa untuk mencapai keseimbangan yang baik antara *underfitting* dan *overfitting*.

Optimasi parameter-parameter ini dapat dilakukan menggunakan metode uji coba dan evaluasi iteratif. Pengujian dilakukan dengan memvariasikan nilai-nilai parameter dan mengamati dampaknya pada kinerja model, yang dapat diukur melalui matrik evaluasi seperti *mean squared error* (MSE) atau akurasi prediksi. Sebagai contoh, penerapan teknik *grid search* atau *random search* dapat membantu menemukan kombinasi parameter terbaik untuk model GRU yang digunakan dalam prediksi harga saham berdasarkan sentimen publik.

2.4 Term Frequency / Inverse Document (TF-IDF)

Term Frequency – Inverse Document Frequency (TF-IDF) adalah salah satu proses *Word Embedding* yang berfungsi sebagai konversi kata menjadi sebuah nilai. Nilai yang diberikan tentu memiliki bobot yang berbeda, dan semua tiap unik kata akan memiliki bobot berdasarkan parameter yang sudah disesuaikan berdasarkan jumlah keseluruhan kata dalam dokumen. TF-IDF penting digunakan karena mesin dapat mengenali entitas sebuah kata dalam bentuk nilai yang sudah dibobot sehingga membantu dalam proses konversi tiap kata menjadi format data yang lebih terstruktur (Sharef *et al.*, 2016). TF-IDF merupakan penggabungan berdasarkan dua metode yakni *Term Frequency* dan *Inverse Frequency*.

TF digunakan dengan mengukur nilai bobot dalam sebuah kata dengan berdasarkan frekuensi kemunculan tiap kata dalam sebuah dokumen, umumnya nilai bobot akan semakin besar Ketika frekuensi kemunculan kata dalam sebuah dokumen banyak, sehingga mesin dapat menyimpulkan sebagai ciri dalam sebuah kelas. Kita dapat contohkan jika terdapat 500 kata “bagus” dalam 1000 data maka dapat ditentukan bahwa nilai TF pada kata “bagus” adalah $500/1000$ atau 0.5.

IDF merupakan hal terbalik dengan TF nilai bobot yang akan diukur dalam *Inverse Document Frequency* berdasarkan kelangkaan kata dalam seluruh dokumen. Nilai semakin mendekati bobot 0 maka dianggap sebagai kata umum yang digunakan dalam sebuah dokumen, hal ini menjadi peran penting dikarenakan beberapa kata unik namun menyangkut pada ciri kata dalam sebuah kelas sering saja muncul namun mesin tidak melihat hal itu menjadi bobot yang besar akibat frekuensi kemunculan data. IDF menjadi faktor penting untuk menjadikan bobot nilai semakin besar dikarenakan kelangkaan kata dalam dokumen.

Berdasarkan dua metode diatas maka kita dapat simpulkan semakin besar tingkat kemunculan data maka nilai TF semakin besar namun berbanding terbalik dengan IDF. Maka dari itu untuk menyimpulkan nilai bobot dengan TF-IDF maka diperoleh persamaan 2.5 sebagai berikut.

$$tfidf(d, t) = tf(d, t) \cdot \log \left(\frac{n}{df(t)} \right) \quad (2.5)$$

Dimana:

N : jumlah total dokumen dalam kumpulan.

df(t) : jumlah dokumen yang berisi term t.

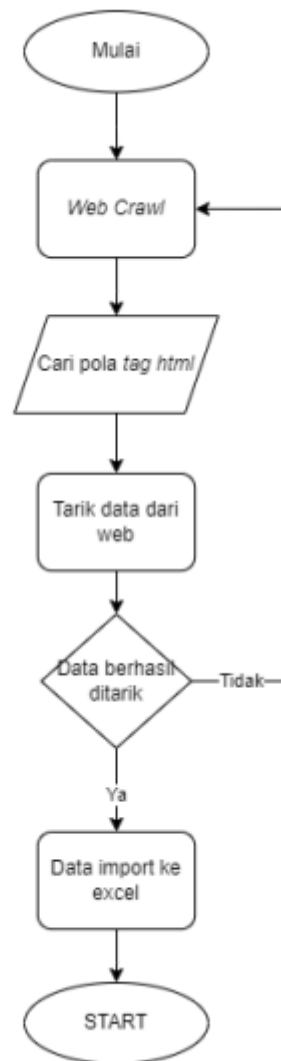
2.5 *Valence Aware Dictionary dan Sentiment Reasoner (Vader)*

Penelitian ini menggunakan *Valence Aware Dictionary and Sentiment Reasoner (Vader)* sebagai salah satu alat analisis sentimen. *Vader* adalah perpustakaan berlandaskan leksikon yang efisien dalam mengevaluasi sentimen pada teks tanpa memerlukan label pada teks tersebut (Reshma *et al.*, 2016). Berbeda dengan pendekatan *machine learning*, C.J. Hutton dan Eric memperkenalkan *Vader* pada tahun 2014 dengan metode pengembangan yang berfokus pada pendekatan berpusat pada manusia. Pengembangan ini mengintegrasikan Pendekatan campuran, yaitu validasi empiris dan analisis kualitatif, dengan menggunakan akal sehat dan pertimbangan manusia (Hutto & Gilbert, 2014). Daftar leksikon yang digunakan oleh *Vader* mencakup kata-kata yang dapat dikategorikan secara umum menjadi tiga orientasi semantik: positif, negatif, dan netral. Pendekatan berbasis leksikon digunakan untuk menentukan orientasi semantik kata atau frasa, menjadi salah satu metode analisis sentimen yang memanfaatkan daftar kata-kata yang menunjukkan pendapat (Nafan & Amalia, 2019). Skor emosi diberikan untuk setiap kata dalam daftar, berkisar dari negatif hingga positif. Keunggulan deteksi polaritas *Vader* terletak pada ketersediaan kamus yang memberikan nilai untuk setiap kata. Proses penentuan polaritas kalimat didapatkan melalui penyatuan atribut "*compound*" dari setiap kata (Ghiassi & Lee, 2018). Kalkulasi sentimen menggunakan *Vader* dibagi menjadi empat kelompok: positif, negatif, netral, dan skor *compound*. Skor *compound* merupakan jumlah semua skor positif, negatif, dan netral yang telah dinormalisasi antara -1 dan +1. Ketika nilai skor *compound* ≥ 0.05 , sentimen dikategorikan positif (diwakili oleh angka 1). Apabila nilai skor *compound* > -0.05 dan < 0.05 , sentimen dikategorikan sebagai netral. Sedangkan, jika nilai skor *compound* ≤ -0.05 , sentimen dikategorikan sebagai negatif (diwakili oleh angka -1) (Karim & Das, 2018).

2.6 *Data Scraping*

Data Scraping ada sebuah metode yang umum digunakan untuk mengekstrak informasi berdasarkan manipulasi HTML dalam sebuah HTML. Umumnya, proses ini dilakukan untuk mengambil informasi spesifik berdasarkan *tag* HTML sehingga mendapatkan informasi yang lebih bersih dan tidak memakan waktu yang lama. Secara praktis proses *Scraping* lebih fleksibel dalam mengambil informasi yang diinginkan sehingga dapat melakukan proses ekstraksi lebih cepat. Alur proses pada data dapat

dilihat pada gambar 2.2.



Gambar 2. 2 Alur proses pada data *Scraping*

Alur proses mendapatkan data pada gambar 2.2 dapat kita lihat proses untuk mengambil data akan dilakukan *web crawler* terlebih dahulu, dimana kita akan menentukan target *website* yang akan diekstrak informasinya, lalu proses lebih lanjut dengan mencari pola meta dan *tag* HTML untuk mengambil informasi secara spesifik, hal itu dilakukan secara berulang berdasarkan jumlah data yang kita inginkan atau jumlah data yang ada dalam *website*. Setelah data sudah diekstrak selanjutnya data yang berhasil diambil akan disimpan dalam bentuk *csv* (*Comma Separated Value*) yang nantinya akan diproses menjadi sejauh data latih dan data uji. Target *Website* yang akan kiat *Scraping* adalah *Twitter* dan *Yahoo Finance*.

2.6.1 *Twitter API*

Twitter adalah suatu layanan jejaring sosial yang berbasis microblogging. Layanan ini dikelola oleh *Twitter, Inc.* Platform ini memberikan kemampuan kepada penggunanya untuk mengirim dan membaca cuitan *tweet* yang mencakup gambar, teks, atau video. Ciri khasnya terletak pada batasan panjang karakter pesan hingga 280, yang digunakan untuk status atau cuitan. Pesan *Twitter*, yang terdiri dari tulisan hingga 140 karakter, dapat ditemukan di halaman profil pengguna. Meskipun bersifat publik, pengguna *Twitter* memiliki kendali terhadap visibilitas pesannya, dapat membatasi akses hanya kepada teman atau pengikut. *Application Programming Interface* (API) merupakan seperangkat aturan dan spesifikasi yang diterapkan oleh program perangkat lunak untuk berkomunikasi satu sama lain (Gu *et al.*, 2014). Fungsi yang dikembangkan dengan menggunakan API kemudian melakukan panggilan sistem sesuai dengan operasi sistemnya (Trupthi *et al.*, 2017).

Awalnya, Summize memberikan layanan pencarian data di *Twitter*. Setelah diakuisisi, Summize di rebranded menjadi *Twitter Search*, Membuat *search API* menjadi objek tersendiri. Komponen API *Twitter* terbagi menjadi dua kelompok (Campan *et al.*, 2018). Ada *Search API* yang dibuat agar memudahkan pelanggan dalam menjalankan kueri pencarian di konten *Twitter*. Pengguna dapat memanfaatkannya untuk mencari *tweet* berdasarkan kata kunci khusus atau menemukan *tweet* lebih spesifik dengan memperhitungkan nama pengguna *Twitter*. *Search API* juga memberikan akses pada data *trending topic*. *Streaming API* digunakan oleh pengembang untuk kebutuhan yang lebih intensif, seperti melakukan penelitian dan analisis data. *Streaming API* memungkinkan pembuatan aplikasi yang memonitor statistik pembaruan status, pengikut, dan sebagainya.

2.6.2 *Yahoo Finance API*

Yahoo Finance atau *Yfinance* merupakan API real-time yang menyediakan data dari *crypto* dan juga *stock market*. *Yahoo Finance* merupakan sumber informasi keuangan paling populer di Amerika (Lawrence *et al.*, 2017). *Yahoo Finance* memiliki banyak varian dalam Pricingnya untuk pengguna Basic atau gratis dibatasi dengan 100 *calls*/harinya. *Yahoo Finance API* merupakan API yang juga menyediakan data *chart* yang dapat digunakan bagi pengguna gratis, *chart* berisikan data secara real-time dan juga *history* dari data *crypto* dan juga *stock market*.

2.7 Text mining

Text mining digunakan untuk mengatasi masalah klasifikasi, pengelompokan, ekstraksi informasi, dan pencarian informasi (Berry & Kogan, 2010). Teknik ini mengintegrasikan pendekatan penambangan data, pembelajaran mesin, pemrosesan bahasa alami, serta informasi dan manajemen informasi (Yang *et al.*, 2018).

Fungsinya mencakup pengambilan informasi dari teks yang bersifat tidak terstruktur, memudahkan transfer Pengetahuan lintas domain, dan sering digunakan untuk mendukung pengambilan keputusan dalam konteks intelijen bisnis (Westergaard *et al.*, 2018).

Preprocessing data melibatkan langkah-langkah untuk menyiapkan dan mentransformasi data agar menghasilkan *output* yang lebih baik dan efisien (Alasadi & Bhaya, 2017). Tahap ini melibatkan beberapa langkah, antara lain:

a. *Case folding*.

Case folding adalah sebuah tahapan dalam *Preprocessing* yang bertujuan untuk mengubah setiap huruf kapital menjadi huruf kecil. Proses ini merupakan hal penting karena mesin komputer membedakan karakter huruf kapital dan kecil, sehingga perlu adanya penyesuaian karakter untuk menghilangkan variasi yang tidak relevan terhadap proses *Word Embedding*.

b. *Tokenisasi*.

Tokenization adalah proses atau tahap akhir dalam sebuah *Preprocessing*. Proses ini bertujuan untuk memisahkan tiap kata atau *token* menjadi satu bagian. Sehingga mesin dapat mengenali *token* dan akan dilakukan proses *Word Embedding* berdasarkan *token* yang sudah dipisah. Proses *Tokenization* dipisah berdasarkan spasi sehingga dipecah menjadi beberapa *token* dalam suatu kalimat.

c. *Normalisasi*.

Normalisasi merupakan salah satu tahapan *Preprocessing* yang bertujuan untuk mengurangi *noise* pada data. Proses ini mencakup untuk mengurangi *redundant* kata, serta menyesuaikan makna kata tanpa harus memiliki bentuk kata yang berbeda. Umumnya semua akta sinonim yang memiliki makna yang sama akan disatukan menjadi satu kata umum dan kata tidak baku dan *typo* akan diubah menjadi kata baku sesuai dengan KBBI.

d. *Filtering Stopword Removal*.

Stopword merupakan proses penghapusan kata yang tidak memiliki makna. Proses penghapusan ini merupakan hal penting karena merupakan salah satu *noise* dalam data, ada atau tidak adanya kata stopwords ini tidak akan berpengaruh makna dari sebuah kalimat. Salah satu kata stopwords dalam Bahasa Indonesia adalah kata sambung seperti “dan”, “sama”, “ke” lalu ada kata subjek seperti “aku”, “kamu”, “kita”. Jadi hasil *preprocess* dalam sebuah kalimat hanya terdiri dari kata kerja dan benda ataupun kalimat sifat yang mengacu terhadap sebuah kalimat, sehingga kita mendapatkan ciri kata yang ada dalam seluruh data. Jumlah kemunculan kata stopwords umumnya sangat banyak dalam sebuah kalimat karena merupakan kata dukungan agar dipahami oleh manusia sehingga dianggap sebagai kendala bagi komputer dalam memahami isi data.

e. *Punctuation Removal*

Punctuation Removal adalah proses mengurangi *noise* pada data dengan melakukan proses penghapusan tanda baca, karakter spesial, angka dan juga emoji yang tidak memiliki keterkaitan dan makna dalam sebuah dokumen, proses *Scraping* umumnya menarik emoji dengan simbol dan beberapa *tag* yang tidak dapat difilter dalam proses *scrapping* maka dari itu perlu adanya penghapusan lebih detail seperti diantaranya tanda baca adalah titik “.”, koma “,”, titik dua “:”, titik koma.

2.8 Metode Evaluasi

Tahap paling akhir dari langkah-langkah penelitian ini mencakup proses evaluasi. Prediksi yang telah dihasilkan dikenakan uji coba untuk memverifikasi tingkat keakuratannya. Hasil yang diperoleh dievaluasi menggunakan metode *mean squared error* (MSE), *root mean squared error* (RMSE). Penekanan pada penggunaan MSE, RMSE, dan pengecualian MAPE sebagai metode evaluasi didasarkan pada praktik umum dalam permasalahan prediksi harga saham dengan berbagai metode *deep learning*, sebagaimana dijelaskan oleh Hu, Zhao, & Khushi (2021).

Berikut adalah Penjelasan masing masing metode Evaluasi yang akan digunakan

1. *Mean squared error* (MSE)

MSE merupakan nilai rata-rata dari kesalahan kuadrat antara nilai aktual dan hasil prediksi. Dalam perhitungan MSE, dilakukan pengurangan antara nilai

aktual dan nilai prediksi. Selanjutnya, nilai tersebut dikuadratkan, dijumlahkan secara keseluruhan, dan dibagi dengan total jumlah data. Persamaan rumus MSE dapat dijelaskan dalam persamaan 2.6 berikut:

$$MSE = \frac{\sum(Aktual - Prediksi)^2}{n} \quad (2.6)$$

Di mana MSE dinyatakan sebagai rata-rata dari kesalahan kuadrat, dengan "aktual" merujuk pada data yang sebenarnya, "prediksi" mengindikasikan nilai prediksi dari variabel aktual, dan "n" mencakup jumlah observasi.

2. *Root mean squared error (RMSE).*

RMSE merupakan perhitungan yang melibatkan pengkuadratan kesalahan (data aktual - data prediksi), kemudian hasilnya dibagi oleh jumlah data dan diambil akar kuadrat. Persamaan untuk RMSE dapat dijelaskan dalam persamaan 2.7 berikut:

$$RMSE = \sqrt{\frac{\sum(Aktual - Prediksi)^2}{n}} \quad (2.7)$$

(RMSE = *root mean square error*, aktual = data sebenarnya, prediksi = nilai prediksi dari variabel aktual dan n = banyaknya observasi.)

3. *Confusion Matrix*

Confusion Matrix merupakan sebuah visualisasi dalam bentuk heatmap yang menampilkan secara keseluruhan hasil prediksi dengan hasil aktual berdasarkan tiap aspek. Dan penentuan ini disebut dengan *true positive*, *true negative*, *false positive* dan *false negative*. perhitungan diperoleh berdasarkan *Confusion Matrix* yang digambarkan pada Tabel 2.1.

Tabel 2. 1 Nilai *Confusion Matrix*

	<i>Aktual Positive (1)</i>	<i>Aktual Negatif (0)</i>
<i>Prediksi Positive (1)</i>	<i>TP</i> (<i>True Positive</i>)	<i>FP</i> (<i>False Positive</i>)
<i>Prediksi Negatif (0)</i>	<i>TN</i> (<i>True Negative</i>)	<i>FN</i> (<i>False Negative</i>)

4. *Classification Report*

Selanjutnya setelah mendapatkan hasil tiap klasifikasi maka akan dilakukan proses pengujian dengan *Classification Report*, hasil tabel tersebut berisi dengan *accuracy*, *precision*, *recall* dan *F1-Score* semua ini akan ditampilkan berdasarkan sentimen dan aspek dan diukur dengan menggunakan persamaan. Adapun persamaan yang digunakan pada evaluasi sebagai berikut:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \times 100\% \quad (2.8)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \times 100\% \quad (2.9)$$

$$F1 - score = \frac{2TP}{(2TP + FP + FN)} \times 100\% \quad (2.10)$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{Total\ Data} \times 100\% \quad (2.11)$$

Keterangan: TP = Aktual Positif; Prediksi Positif Aspek

FP = Aktual Negatif; Prediksi Positif Aspek

TN = Aktual Positif; Prediksi Negatif Aspek

FN = Aktual Negatif; Prediksi Negatif Aspek

2.9 Penelitian Terdahulu

Penambahan opini atau sentimen didasarkan pada pandangan spesifik orang-orang. Sentimen analisis berperan dalam penggalan informasi untuk banyak aplikasi, seperti ulasan produk, perawatan kesehatan, politik, dan pengawasan. Memprediksi pergerakan saham adalah bidang studi yang penting dan aktif dan melibatkan prediksi yang akurat. Ada banyak kemajuan dalam beberapa tahun terakhir untuk mengembangkan model prediksi SM global. Banyak pendekatan dan metode standar yang digunakan dalam tinjauan literatur, termasuk analisis prediksi deret waktu. Banyak teknik pemodelan pembelajaran mesin juga digunakan.

Penelitian dalam membangun sebuah model untuk mengamati pergerakan harga saham menggunakan metode regresi linear (Bhuriya *et al.*, 2017). *Dataset* (kumpulan data) yang digunakan adalah saham *Tata Consultancy Historical* (TCS) di India. Variabel *independent* yang digunakan dalam penelitian ini adalah harga pembuka, harga tertinggi, harga terendah, dan jumlah tren. Sementara itu, variabel *dependent* target yang digunakan adalah harga penutup. Penelitian ini menggunakan tiga metode regresi linear yang mendapatkan nilai *confidence* sebesar 0,97, metode *polynomial* yang

mendapatkan nilai *confidence* sebesar 0,468., dan Metode RBF yang mendapatkan nilai *confidence* sebesar 0,5652.

Dalam penelitiannya, Izzah, Sari, Widyastuti, & Cinderatama (2017) menggunakan metode regresi linear untuk memprediksi pergerakan harga saham perusahaan. *Dataset* yang digunakan adalah saham *Jakarta Composite Index (JCI)*. Metode regresi linear berganda (*multiple linear Regression*) menunjukkan nilai *mean square error* sebesar 15087,465, nilai *root mean square error* sebesar 122,831, serta nilai *mean absolute percentage error* sebesar 3,255. Namun, data saham yang bersifat deret waktu mengakibatkan penggunaan model linear masih belum cukup, sehingga model *deep learning* lebih unggul daripada model linear (M *et al.*, 2018).

Salah satu metode *deep learning* yaitu *Recurrent Neural Network (RNN)* dapat digunakan dalam memprediksi saham (Jahan & Sajal, 2018). Penelitian ini menggunakan data harga penutup saham *Advanced Micro Device (AMD)*. Data diambil selama 168 hari kerja. Data *Training* yang digunakan sebanyak 168 hari, sedangkan data *testing* yang digunakan sebanyak 12 hari. Model dibangun menggunakan *neuron* 500 dan iterasi sebanyak 5000. Model tersebut berhasil mendapatkan hasil *mean squared error (MSE)* dibawah 5%.

Meskipun demikian, arsitektur metode RNN juga memiliki kekurangan, yaitu tidak dapat memproses *sequence* yang panjang. Model RNN juga memiliki permasalahan terkait *vanishing gradient* dan *exploding gradient*. Nilai *gradient* didapatkan dari hasil aktivasi fungsi *tanh* dengan *range* [-1,1]. *vanishing gradient* adalah keadaan ketika hasil perkalian beberapa *gradient* bernilai nol. Sementara itu, *exploding gradient* adalah keadaan ketika bilangan bobot pada $W > 1$ sehingga, nilai *gradient* akan terus membesar (Suyanto *et al.*, 2019).

Metode *Long Short-Term Memory (LSTM)* dapat mempelajari pola dari data deret waktu. Arsitektur LSTM juga dapat mengatasi permasalahan *vanishing gradient* dan *exploding gradient*.

Dalam penelitiannya, (Ghosh *et al.*, 2019) menggunakan metode LSTM telah dilakukan menggunakan *dataset Bombay Stock Exchange (BSE)*. Harga penutupan diprediksi dari 5 perusahaan yang bergerak di berbagai sektor. Model dilatih menggunakan data selama 3 bulan, 6 bulan, 1 tahun dan 3 tahun. Nilai *error* pada setiap perusahaan menunjukkan penurunan selama 3 bulan hingga 3 tahun. Nilai *error* terbaik yang didapatkan yaitu sebesar 0,874805.

Dalam sebuah penelitian, (Mathur *et al.*, 2019) menggunakan beberapa arsitektur LSTM, antara lain model *deep long short-term memory* (DLSTM), model *long short-term memory projected* (LSTMP), dan model *deep long short-term memory Projected* (DLSTMP). *Dataset* saham yang digunakan adalah saham *Apple Inc.* (AAPL), saham *Google* (GOOG) dan saham *Tesla, Inc* (TSLA). Percobaan dilakukan sebanyak 5 kali. Nilai *epoch* yang dipakai sebanyak 10 *epoch*. Arsitektur LSTMP dan DLSTMP menghasilkan nilai *loss function* MSE masing-masing sebesar 0,5770 dan 0,00031. Model LSTM dan RNN menunjukkan hasil lebih baik dibandingkan dengan *machine learning* pada umumnya.

Metode *autoregressive integrated moving average* (ARIMA) dan LSTM juga diimplementasikan pada perusahaan luar negeri (Joosery & Deepa, 2019). *Dataset* yang digunakan adalah saham GOOGL (*alphabet.inc*), saham NKE (*nike.inc*), saham NOK (*nokia oyy*) dan saham SNE (*sony corp*). Model dilatih dengan menggunakan data selama 1 bulan, 3 bulan, 6 bulan, 1 tahun, 5 tahun dan 10 tahun. Evaluasi pada penelitian menggunakan *mean squared error* (MSE). Model ARIMA mendapatkan akurasi sebesar 96,766%. Model LSTM mendapatkan akurasi sebesar 97,549%. Model *attention* LSTM mendapatkan akurasi sebesar 98,070%.

Tabel 2. 2 Penelitian Terdahulu

No.	Peneliti	Judul	Metode	Keterangan
1.	(Bhuriya <i>et al.</i> , 2017)	<i>Stock market Prediction Using a Linear Regression</i>	Regresi Linear (3 variasi), <i>Polynomial</i> , RBF (Radial Basis Function)	Penelitian ini menggunakan tiga metode regresi linear (biasa, <i>polynomial</i> , dan RBF) untuk menganalisis pergerakan harga saham dengan dataset TCS di India. Variabel independen melibatkan harga pembuka, tertinggi, terendah, dan jumlah tren, dengan variabel dependen harga penutup. Hasil dinilai dengan <i>confidence level</i> masing-masing: 0,97, 0,468, dan 0,5652.
2.	(Izzah <i>et al.</i> , 2017)	<i>Mobile App for Stock Prediction Using Improved Multiple Linear Regression</i>	Regresi Linear Berganda (Multiple Linear Regression)	Penelitian ini menggunakan regresi linear berganda pada data saham Jakarta Composite Index (JCI). Evaluasi model menghasilkan <i>mean square error</i> : 15087,465, <i>root mean square error</i> : 122,831, dan <i>mean absolute percentage error</i> : 3,255. Namun, model linear kurang efektif pada data deret waktu, disarankan model <i>deep learning</i> (M <i>et al.</i> , 2018)
3.	(Jahan & Sajal, 2018)	<i>Stock Price Prediction using Recurrent Neural Network</i>	<i>Recurrent Neural Network</i> (RNN)	Penelitian menggunakan RNN untuk memprediksi harga penutup saham AMD dengan data 168 hari. Model dengan 500 neuron dan 5000 iterasi

No.	Peneliti	Judul	Metode	Keterangan
		<i>(RNN) Algorithm on Time-Series Data</i>		menghasilkan MSE di bawah 5%. Namun, RNN memiliki keterbatasan dalam memproses <i>sequence</i> panjang dan masalah <i>vanishing/exploding gradient</i> . LSTM digunakan untuk mengatasi masalah tersebut.
4.	(Khedr <i>et al.</i> , 2017)	Predicting Stock market Behavior using Data Mining Technique and News Sentiment Analysis	Pengklasifikasi Naïve Bayes untuk sentimen berita	Penelitian ini menggunakan analisis dan kerangka kerja dioptimalkan untuk meningkatkan akurasi prediksi harga saham, dengan fokus pada rasio kesalahan minimum. Metode melibatkan <i>Sentiment Analysis</i> (SA) pada berita keuangan, pemanfaatan nilai Social Media (SM) historis, dan pengklasifikasi Naïve Bayes untuk menilai polaritas teks. Dengan menggabungkan informasi harga saham masa lalu dan sentimen berita, model prediksi mencapai akurasi hampir 89,80%, hasil yang lebih baik dibandingkan penelitian sebelumnya.
5.	(Ghosh <i>et al.</i> , 2019)	Stock Price Prediction Using LSTM on Indian Share Market	Long Short-Term Memory (LSTM)	Penelitian menggunakan LSTM untuk memprediksi harga penutupan saham dari 5 perusahaan di berbagai sektor di BSE. Model dilatih dengan data 3 bulan, 6 bulan, 1 tahun, dan 3 tahun. Terdapat

No.	Peneliti	Judul	Metode	Keterangan
				penurunan nilai <i>error</i> selama periode 3 bulan hingga 3 tahun, dengan nilai <i>error</i> terbaik mencapai 0,874805.
6.	(Mathur <i>et al.</i> , 2019)	<i>Stock market Price Prediction Using LSTM RNN</i>	Arsitektur LSTM, termasuk DLSTM, LSTMP, dan DLSTMP	Penelitian menggunakan beberapa arsitektur LSTM, seperti DLSTM, LSTMP, dan DLSTMP, untuk menganalisis saham <i>Apple Inc.</i> (AAPL), <i>Google</i> (GOOG), dan <i>Tesla, Inc.</i> (TSLA). Eksperimen diulang sebanyak 5 kali dengan 10 <i>epoch</i> . Hasil menunjukkan bahwa LSTMP dan DLSTMP memiliki nilai <i>loss function</i> MSE masing-masing sebesar 0,5770 dan 0,00031. Model LSTM dan RNN menunjukkan hasil yang lebih baik dibandingkan dengan <i>machine learning</i> pada umumnya.
7.	(Joosery & Deepa, 2019)	<i>Comparative analysis of time-series forecasting algorithms for stock price prediction</i>	ARIMA (Autoregressive Integrated Moving Average), LSTM (Long Short-Term Memory), <i>Attention LSTM</i>	.Penelitian mengimplementasikan metode ARIMA dan LSTM, termasuk <i>Attention LSTM</i> , pada saham perusahaan luar negeri seperti GOOGL (Alphabet Inc.), NKE (Nike Inc.), NOK (Nokia Oyj), dan SNE (Sony Corp). Model dilatih menggunakan data dengan rentang waktu 1 bulan, 3 bulan, 6 bulan, 1 tahun, 5 tahun, dan 10 tahun. Evaluasi kinerja dilakukan dengan <i>mean</i>

No.	Peneliti	Judul	Metode	Keterangan
				<i>squared error</i> (MSE), yang menunjukkan bahwa model ARIMA mencapai akurasi sebesar 96,766%, model LSTM mencapai 97,549%, dan model <i>Attention</i> LSTM mencapai akurasi tertinggi, yaitu 98,070%.

Membedakan penelitian ini dari pendahulunya, penelitian ini memanfaatkan algoritma *Gated Recurrent Unit* (GRU) dalam memprediksi harga saham Telkom berdasarkan sentimen publik di *Twitter*. Berbeda dengan penggunaan *Linear Regression*, LSTM, atau SVM pada penelitian sebelumnya, GRU unggul dalam kesederhanaan, kemudahan pelatihan, dan efisiensi memori. Sentimen publik juga akan diekstrak langsung dari *Twitter*, berbeda dari studi terdahulu yang mengandalkan media sosial lain atau survei. Hal ini memungkinkan penggambaran sentimen yang lebih akurat tentang layanan telekomunikasi Indonesia. Terakhir, penelitian ini tidak terpaku pada metrik seperti RMSE atau MAE. Sebaliknya, penelitian ini akan mengevaluasi akurasi model dalam memprediksi arah pergerakan harga saham Telkom per minggu, memberikan kejelasan yang lebih baik mengenai kinerjanya. Dengan pendekatan inovatif ini, penelitian ini berambisi memberikan kontribusi baru bagi bidang prediksi harga saham.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Arsitektur Umum

Untuk mendapatkan *system* yang sesuai perlu adanya rancangan yang dilakukan untuk mendapat hasil terbaik, arsitektur umum digambarkan pada 4 bagian utama antara lain *Preprocessing*, *Word Embedding*, *Vader* dan data modeling. Dalam proses *Preprocessing* merupakan tahap pengolahan data awal dengan melakukan beberapa metode seperti *Normalization*, *Case folding*, *Stemming*, *Stopword Removal*, *puntuational removal*, dan *Tokenization*. Lalu setelah data sudah minim oleh *noise* proses konversi kata menjadi bobot akan dilakukan dengan TF-IDF dan *Vader*. Pada tahap ini akan ditentukan pada tiap nilai bobot kata dan berakhir pada *Gated Recurrent Unit* untuk mendapatkan hasil klasifikasi dan dijadikan sebagai dasar acuan. Adapun tahapan tahapan tersebut dilakukan secara sistematis yang digambarkan pada gambar 3.1.

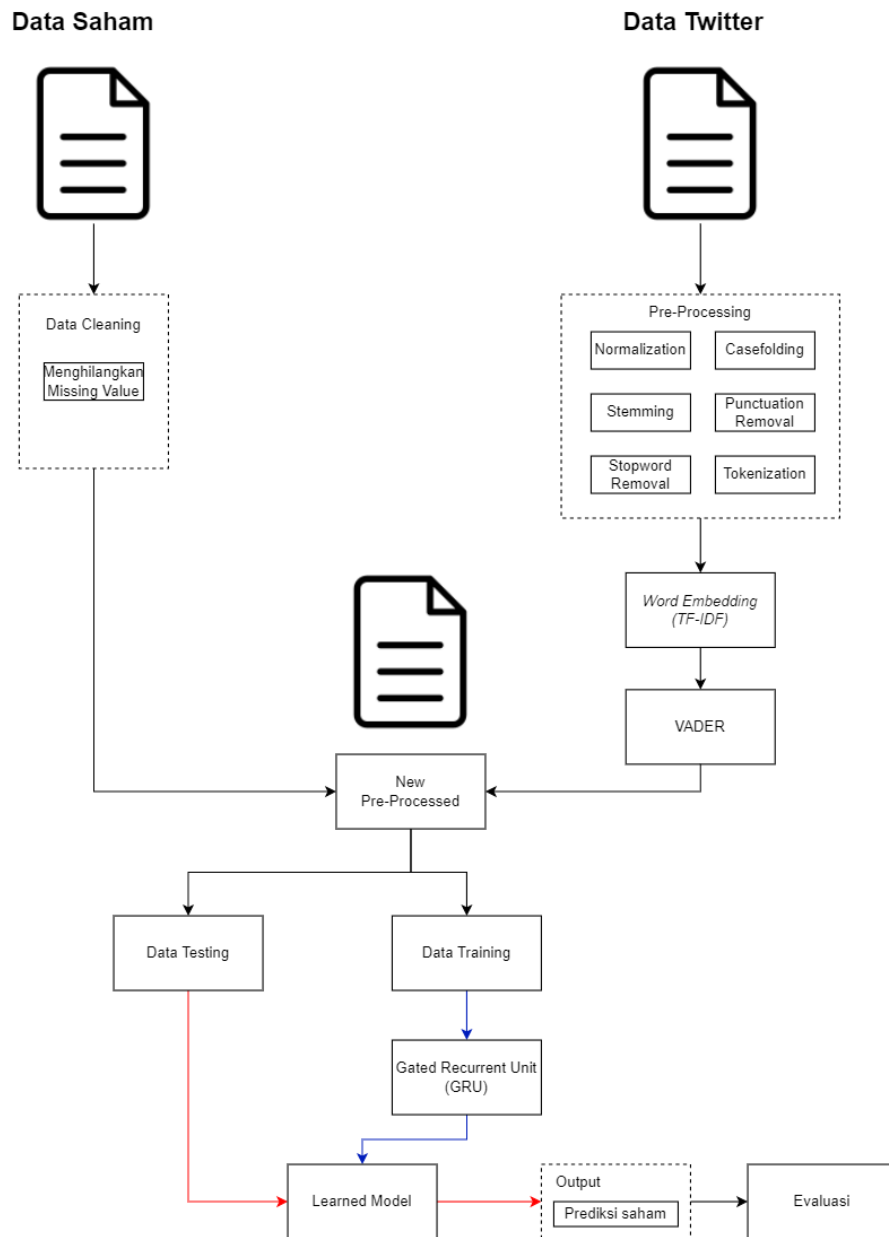
3.2 Pengumpulan Data

Dalam fase pengumpulan data, dipilih jenis data yang akan digunakan dalam penelitian. Tahap pengumpulan data memiliki peran krusial dalam suatu studi, dan keakuratan serta kejelasan sumber data yang digunakan menjadi faktor utama. Penelitian ini akan mengumpulkan 2 jenis data yaitu Data *Twitter* dan data Saham

3.2.1 Twitter

Twitter adalah salah satu media sosial tempat orang mengungkapkan opini mereka. Proses pengambilan data digunakan dengan *crawling* data dalam *twitter* dan *scrapping* untuk produk Telkom dengan mengambil kata kunci “#TELKOM” untuk mendapatkan opini terhadap produk telkom. Data dikumpulkan selama 4 tahun dari tanggal 01-01-2018 sampai dengan 31-12-2022. Total data yang diperoleh dan digunakan sebanyak 18.960 data, dengan perbandingan data latih dan data uji 8:2, dengan total *actual* 15.158 data latih dan 3802 data uji. Data yang sudah didapat akan digunakan dan disimpan dalam format csv file. Contoh pada data ulasan pada *twitter*

dapat dilihat dalam gambar 3.2. Beberapa modul yang digunakan untuk mengakses data *Twitter* mencakup *Twitter API*, *library twitterscraper*, dan *library GetOldTweets3*.



Gambar 3. 1 Arsitektur Umum



Gambar 3. 2 Cuitan Pengguna Terhadap Produk Telkom di *Twitter*

Terdapat beberapa langkah yang perlu diambil saat melakukan *crawling* data *Twitter* menggunakan *Google Colab*. Berikut adalah pseudocode dari *google colab* tersebut :

```
import os
import datetime

# Batasi jumlah hasil yang diambil
max_results = 10000

# Gunakan Twitter search untuk mencari tweet yang di-favoritkan minimal 10000
kali dan berbahasa Indonesia
twitter_search = "telkom lang:id until:2018-09-31 since:2018-09-01"

# Tentukan nama file dengan format "<kueri pencarian>_<tanggal saat ini>.json"
filename = f"{twitter_search.replace(' ', '_').replace(':', '-').replace('#',
')}_{datetime.date.today().strftime('%Y-%m-%d')}.json"

USING_TOP_SEARCH = False

snsrape_params = '--jsonl --max-results'
twitter_search_params = ""

if USING_TOP_SEARCH:
    twitter_search_params += "--top"

snsrape_search_query = f"snsrape {snsrape_params} {max_results} twitter-
search {twitter_search_params} '{twitter_search}' > {filename}"

print(snsrape_search_query)

os.system(snsrape_search_query)
```

Setelah mendapatkan data, kita perlu memasukkan data tersebut dalam format .csv serta mengambil data data yang diperlukan saja. Berikut adalah pseudocode nya:

```
import pandas as pd
import ast
import json

# Membaca file JSON hasil dari perintah CLI sebelumnya dan membuat dataframe
pandas
tweets_df = pd.read_json(filename, lines=True)

NAMA_FILE_CSV = 'telkom9.csv'

# Membuat kamus untuk mengganti nama kolom
new_columns = {
    'conversationId': 'Conv. ID',
    'url': 'URL',
    'date': 'Date',
    'rawContent': 'Tweet',
    'id': 'ID',
    'replyCount': 'Replies',
    'retweetCount': 'Retweets',
    'likeCount': 'Likes',
    'quoteCount': 'Quotes',
    'bookmarkCount': 'Bookmarks',
    'lang': 'Language',
    'links': 'Links',
    'media': 'Media',
    'retweetedTweet': 'Retweeted Tweet',
    'username': 'Username'
}

if len(tweets_df) == 0:
    print('Pencarian tidak ditemukan coba ganti keyword lain, keywordmu: ',
```

```

twitter_search)
    exit()
else:
    # Memilih kolom yang akan digunakan dan mengganti nama kolom menggunakan
    # kamus yang telah dibuat
    tweets_df = tweets_df.loc[:, ['url', 'date', 'rawContent', 'id',
                                  'replyCount', 'retweetCount', 'likeCount', 'quoteCount',
                                  'conversationId', 'lang', 'links',
                                  'media', 'retweetedTweet', 'bookmarkCount', 'username']]
    tweets_df = tweets_df.rename(columns=new_columns)

    # Ekstrak fullUrl dari kolom media dan url dari kolom links
    tweets_df['Media'] = tweets_df['Media'].apply(lambda x: x[0]['fullUrl'] if
    isinstance(x, list) and x and isinstance(x[0], dict) and 'fullUrl' in x[0] else None)
    tweets_df['Links'] = tweets_df['Links'].apply(lambda x: x[0]['url'] if isinstance(x,
    list) and x and isinstance(x[0], dict) and 'url' in x[0] else None)

    # Menampilkan dataframe tweets_df
    display(tweets_df)

    # Simpan ke csv
    tweets_df.to_csv(NAMA_FILE_CSV, index=False)

```

Setelah semua proses selesai, berikut adalah contoh hasil dari *crawling* data *Twitter*:

Tabel 3. 1 Contoh hasil *crawling* Data *Twitter*

Date	<i>Tweet</i>	ID	Replies	Retweets	Likes	Quotes	Conv. ID	Language	Links	Media	Retweeted <i>Tweet</i>	Bookmarks	Username
2018-01-30 23:59:44+ 00:00	Terima kasih@TelkomCare respons cepat sekali kemarin siang saya lapor gangguan, sore selesai. Bravo telkom tingkatkan terus pelayanannya.	958489940391706624	0	0	0	0	958489940391706624	in				0	partono_dwi
2018-01-30 23:54:15+ 00:00	Demi mencari sebangkah berlian satu malam tidur di mobil di depan telkom poso Kerja kerja kerja Semangat sendiri 🤔🤔👉👉	958488561430315008	0	0	0	0	958488561430315008	in				0	Rommy_WilsonJR
2018-01-30 23:53:59+ 00:00	@racoona292 Kak, yang meluncurkan aplikasi My IndiHome adalah PT. Telkom Indonesia. -Afifah	958488495516925952	0	0	0	0	958488169648857088	in				0	TelkomCare
2018-01-30 23:52:05+ 00:00	@racoona292 Bisa Kak, aplikasi my IndiHome untuk pelanggan Telkom. Trims -Arim	958488017894760449	0	0	0	0	958486908077719552	in				0	TelkomCare
2018-01-	@racoona292 Kak,	95848788	0	0	0	0	95848743158	in				0	TelkomCar

30 23:51:34+ 00:00	saat ini pembayaran <i>tagihan</i> Telkom hanya bisa melalui loket <i>online</i> , ATM, banking, Plasa Telkom serta melalui tokopedia. - Rima	57532129 28					9675008						e
--------------------------	--	----------------	--	--	--	--	---------	--	--	--	--	--	---

3.2.2 Data Saham

Setelah mendapatkan *sentiment* publik dari *twitter*, tentu kita membutuhkan data saham untuk dipelajari. Data saham yang dipilih adalah data saham TELKOM(TLKM.JK). Data saham Telkom (TLKM.JK) dikumpulkan menggunakan metode pengumpulan data dokumen. Dokumen yang digunakan adalah data historis harga saham Telkom (TLKM.JK) yang tersedia di situs *web Yahoo Finance*. Data tersebut mencakup harga pembuka, harga penutupan, harga tertinggi, dan harga terendah. Jangka waktu nya adalah dari tanggal 01-01-2018 sampai dengan 31-12-2022. Data dikumpulkan selama 4 tahun dari tanggal 01-01-2018 sampai dengan 31-12-2022. Total data yang diperoleh dan digunakan sebanyak 260 data, dengan perbandingan data latih dan data uji 8:2, dengan total *actual* 208 data latih dan 52 data uji. Data yang sudah didapat akan digunakan dan disimpan dalam format csv file. Contoh pada data ulasan pada *twitter* dapat dilihat dalam gambar 3.3



Gambar 3. 3 Data Saham TLKM di *Yahoo Finances*

Terdapat beberapa Langkah juga untuk *mengcrawling* data saham dari *Yahoo finance* menggunakan *google colab*. Berikut adalah pseudocode nya:

```
import yfinance as yf
import pandas as pd
from datetime import datetime, timedelta

# Ticker symbol for TELKOM on the IDX
ticker_symbol = "TLKM.JK"
```

```

# Create a Ticker object
telkom = yf.Ticker(ticker_symbol)

# Calculate the date range for the desired period
start_date = "2018-01-01"
end_date = "2023-02-02"

# Fetch historical stock price data on a weekly basis
stock_history = telkom.history(period="1wk", start=start_date, end=end_date)

# Reset the index to make Date a column
stock_history.reset_index(inplace=True)

# Convert 'Date' column to datetime type
stock_history['Date'] = pd.to_datetime(stock_history['Date'])

# Extract Date, Open, High, Low, and Close Price columns
stock_data = stock_history[["Date", "Open", "High", "Low", "Close"]]

# Add Year and ISO Week columns
stock_data['Year'] = stock_data['Date'].dt.year
stock_data['ISO_Week'] = stock_data['Date'].dt.strftime("%G-%V") # ISO year and
ISO week format

# Calculate the first date of each week based on Year and ISO Week
stock_data['FirstDateOfWeek'] = stock_data.apply(lambda row:
datetime.strptime(row['ISO_Week'] + '-1', "%G-%V-%u"), axis=1)

# Group data by Year and ISO Week
grouped_data = stock_data.groupby(['Year',
'ISO_Week', 'FirstDateOfWeek']).mean()

# Save data to a CSV file

```

```

csv_filename = "telkom_weekly_stock_prices_by_year_with_date.csv"
grouped_data.to_csv(csv_filename)

print("Weekly TELKOM stock price data separated by year and saved to",
      csv_filename)

```

Data yang telah didapat akan diekspor ke .csv. Berikut adalah pseudocode nya:

```

import pandas as pd

def add_comparison_week(df):
    # Initialize an empty list to store the comparison values
    comparison_list = []

    # Initialize a variable to store the previous week's price prediction
    prev_price = None

    # Iterate through the rows of the DataFrame
    for index, row in df.iterrows():
        # If it's the first row, there's no previous week to compare
        if prev_price is None:
            comparison_list.append("")
        else:
            # Compare the current week's price prediction with the previous week
            if row['price prediction'] > prev_price:
                comparison_list.append('Higher')
            elif row['price prediction'] < prev_price:
                comparison_list.append('Lower')
            else:
                comparison_list.append('Equal')

        # Update the previous week's price prediction
        prev_price = row['price prediction']

```



```
# Add the comparison list as a new column in the DataFrame
df['Comparison Week'] = comparison_list

# Create a sample DataFrame
data = {'week': [1, 2, 3, 4],
        'price prediction': [2000, 3000, 1500, 2500]}
df = pd.DataFrame(data)

# Call the Function to add the "Comparison Week" column
add_comparison_week(df)

# Print the resulting DataFrame
print(df)
```

Berikut adalah hasil dari *crawling* data saham tersebut :

Tabel 3. 2 Contoh hasil *crawling* Data Historis Saham

Year	ISO_Week	FirstDateOfWeek	Open	High	Low	Close
2018	2018-01	2018-01-01	3558.473213843815	3576.511557598739	3510.917578883331	3538.79501953125
2018	2018-02	2018-01-08	3451.8828679888675	3471.561060323391	3425.645279122652	3435.484375
2018	2018-03	2018-01-15	3430.564877254511	3450.243070050419	3391.208492140061	3420.72578125
2018	2018-04	2018-01-22	3335.453676318067	3376.449909407341	3284.6183439847146	3332.173974609375
2018	2018-05	2018-01-29	3314.1356603380787	3342.013100399659	3266.580027395588	3281.338671875
2018	2018-06	2018-02-05	3253.4613426196056	3292.8177298164765	3227.2237510905984	3263.300439453125
2018	2018-07	2018-02-12	3287.898187734108	3312.495929693131	3268.219994355624	3289.538037109375
2018	2018-08	2018-02-19	3323.9747300919757	3337.093524978817	3299.376988737119	3317.41533203125
2018	2018-09	2018-02-26	3305.9364634502112	3327.254505851848	3278.0590227262683	3309.216162109375
2018	2018-10	2018-03-05	3328.8942704108536	3355.1318607703315	3297.7371320970087	3333.813818359375

3.3 *Preprocessing Data*

Preprocessing merupakan tahap awal dalam pemrosesan data agar menghasilkan data yang lebih bersih dan terstruktur sehingga mesin dapat lebih mudah mengenali dan mengolah berdasarkan ciri data yang sudah ada melalui data *Training* dan data *testing*. Beberapa metode dalam *Preprocessing* digunakan agar menghasilkan data yang baik antara lain *Normalization*, *Stemming*, *Stopword Removal*, *Case folding*, *Punctuation Removal*, dan *Tokenization*. Untuk Data historis saham hanya perlu menghilangkan *Missing value*. Langkah tersebut akan dijelaskan lebih detail.

3.3.1 *Normalization*

Proses *Normalization* dilakukan untuk menghilangkan duplikasi dalam dokumen dan juga menghilangkan *noise* dalam data dengan mengubah kata sinonim menjadi satu makna, mengubah kata *typo*, bahasa tidak baku dan Bahasa asing. Dilakukan secara perulangan yang dilakukan proses identifikasi tiap kata, jika kata tidak sesuai dengan KBBI maka akan diubah. Contoh implementasi kamus normalisasi pada tabel 3.3 Adapun pseudocode nya adalah sebagai berikut:

```
def normalize_text(text, stdword_, nonstdword_):
    text = text.split(" ")
    for i in range(len(text)):
        if text[i] in nonstdword_:
            index = nonstdword_.index(text[i])
            text[i] = stdword_[index]
    return ' '.join(map(str, text))
```

Berikut adalah pengertian dari kode tersebut:

1. *def normalize_text(text, stdword_, nonstdword_):*: Fungsi ini menerima tiga parameter, yaitu *text* (teks yang akan dinormalisasi), *stdword_* (kamus kata-kata standar), dan *nonstdword_* (kamus kata-kata non-standar).
2. *text = text.split(" ")*: Memecah teks menjadi daftar kata-kata menggunakan spasi sebagai pemisah. Ini akan menghasilkan daftar kata-kata dari teks.
3. *for i in range(len(text))*: Melakukan iterasi untuk setiap kata dalam daftar kata-kata.
4. *if text[i] in nonstdword_*: Memeriksa apakah kata saat ini dalam iterasi terdapat dalam kamus kata-kata non-standar (*nonstdword_*).
5. *index = nonstdword_.index(text[i])*: Jika kata tersebut non-standar, mendapatkan indeksnya dalam kamus *nonstdword_*.
6. *text[i] = stdword_[index]*: Menggantikan kata non-standar dengan bentuk standar

yang sesuai dari kamus `stdword_`.

7. `return ''.join(map(str, text))`: Menggabungkan kata-kata yang telah dinormalisasi menjadi sebuah teks baru, dengan kata-kata dipisahkan oleh spasi.

Tabel 3. 3 Contoh Hasil Normalisasi

Contoh Kalimat	Hasil Normalisasi
Tks buat pulsa, tsel lagi <i>promo</i> kuota murah bgt, mohon segera dapatkan!	Telkomsel buat kredit, Telkomsel lagi <i>promosi</i> paket data murah banget, mohon segera dapatkan!
Gue lg di bandara, internet nya kenceng banget. Telkomsel emang the best!	Saya sedang berada di bandara, internetnya sangat kencang. Telkomsel memang yang terbaik!
Telkomsel paket internetnya murah banget, cocok buat pelajar kayak gue.	Paket internet Telkomsel sangat murah, cocok untuk pelajar seperti saya.

Dalam penelitian ini, praproses teks pada tahap normalisasi menggunakan data referensi berisi berbagai kemungkinan kesalahan penulisan kata. Contoh Kamus Normalisasi kata bisa dilihat pada tabel 3.4

Tabel 3. 4 Contoh Kamus Normalisasi Kata

Kata	Normalisasi
tk	Telkomsel
tsel	Telkomsel
pulsa	Kredit
gb	Gigabyte
kuota	paket data
brg	Barang
<i>promo</i>	<i>Promosi</i>
mhn	Mohon
bgt	Banget

3.3.2 Case folding

Case folding adalah proses penyederhanaan kata dalam sebuah dokumen dengan melakukan proses penyeragaman huruf menjadi huruf kecil. Umumnya penulisan kaidah yang sesuai KBBI perlu adanya huruf kapital di awal kalimat atau setelah tanda baca titik, ataupun huruf kapital untuk singkatan, namun permasalahannya ialah bobot nilai huruf besar dan huruf kecil memiliki nilai yang dianggap berbeda sehingga diidentifikasi dengan entitas kata yang memiliki banyak variasi namun mempunyai

makna yang sama terhadap yang lain, maka dari itu perlu adanya penyeragaman tiap kata dalam sebuah kalimat. Contoh hasil *Case folding* dapat dilihat pada Tabel 3.5. Adapun pseudocode fungsi *Case folding* yang dapat digunakan sebagai berikut;

```
# Case folding
data['Preprocess'] = data['Preprocess'].str.lower()
```

Berikut adalah pengertian kodingan tersebut:

1. *data['Preprocess']*: Ini memilih kolom yang berlabel '*Preprocess*' pada Data Frame data.
2. *.str.lower()*: Ini adalah metode *string* dalam pandas yang mengubah semua karakter dalam sebuah *string* menjadi huruf kecil.
3. *str*: Ini digunakan untuk menunjukkan bahwa operasi selanjutnya terkait dengan manipulasi *string*.
4. *lower()*: Metode ini mengubah semua karakter dalam *string* menjadi huruf kecil.
5. *data['Preprocess'] = ...*: Ini menetapkan hasil konversi huruf kecil kembali ke kolom '*Preprocess*' dalam Data Frame data. Jadi, teks asli dalam kolom '*Preprocess*' digantikan dengan versi huruf kecil.

Tabel 3. 5 Tabel Hasil *Case folding*

Contoh Kalimat	Hasil <i>Case folding</i>
Telkomsel sedang mempromosikan kuota internet yang murah!	telkomsel sedang mempromosikan kuota internet yang murah!
Internet Telkomsel cepat dan stabil!	internet telkomsel cepat dan stabil!
Paket internet Telkomsel terbaik di Indonesia!	paket internet telkomsel terbaik di indonesia!

3.3.3 Stemming

Stemming adalah salah satu *preprocess* yang bertujuan untuk mengubah *token* kata imbuhan menjadi kata dasar. Proses penentuan imbuhan berasal dari deteksi *corpus* yang sudah dilatih menggunakan *library* sastrawi merupakan *corpus* untuk kata imbuhan Bahasa Indonesia, sehingga langsung menghilangkan kata Suffixes (“-nya”, “ku”, “mu”, “-kah”). Imbuhan di awal dan di akhir akan tetap dihapus karena mesin hanya membutuhkan kata dasar yang mempengaruhi kata, mesin tidak perlu imbuhan yang mempengaruhi kata kerja baik berdasarkan subjek atau pun benda dikarenakan

mesin tidak mengenali hal tersebut, dan dua kata seperti mencintai dan cinta memiliki makna yang sama namun akan dikenali oleh mesin sebagai dua entitas, maka dari itu eliminasi imbuhan pada kata dapat membantu efisiensi mesin dalam melatih data

sehingga mengurangi variasi kata dan mendapatkan hasil akurat tanpa harus menghilangkan makna pada kata tersebut, seperti contoh pada tabel 3.6.

Tabel 3. 6 Tabel Hasil *Stemming*

Contoh Kalimat	Hasil <i>Stemming</i>
Telkomsel menyediakan berbagai paket data dengan kuota yang melimpah.	Telkomsel sediakan bagai paket data dengan kuota yang limpah.
Pengalaman menggunakan Telkomsel sangat memuaskan! Jaringan cepat, paket data melimpah, puas banget deh!	Pengalaman guna Telkomsel sangat puas! Jaringan cepat, paket data limpah, puas banget deh!
Telkomsel kenapa ya sering banget lelet, padahal paket data sudah banyak. Harap diperbaiki, dong!	Telkomsel kenapa ya sering banget lelet, padahal paket data sudah banyak. Harap perbaiki, dong!

3.3.4 *Punctuation Removal*

Punctuation Removal merupakan tahap selanjutnya yang menjadi faktor penting dalam *Preprocessing*. Dalam penelitian ini, data seringkali memiliki tanda petik, koma, dan titik yang tidak memiliki makna langsung terhadap tiap kalimat. Maka dari itu, perlu adanya penghapusan tanda baca, karakter, angka dan emoji agar mendapatkan data yang memiliki bobot yang seimbang. Emoji dan angka tidak dimasukkan kedalam kasus karena umumnya ulasan berbentuk kata kerja yang berisi pujian terhadap stasiun oleh karena itu perlu adanya dibatasi berdasarkan kata. Proses penghapusan karakter spesial dilakukan melalui *library* dari *nlk* yang dapat kita modifikasi berdasarkan list karakter yang akan dihapus, dalam kasus ini semua karakter akan dihapus baik itu tanda baca, angka, emoji ataupun karakter yang tidak memiliki makna langsung terhadap sebuah kalimat dalam dokumen. Adapun pseudocode dalam proses eliminasi stopword sebagai berikut:

#Punctuation Removal

```
data['Preprocess'] = data['Preprocess'].str.replace('[^a-zA-Z0-9]+',' ',regex=True)
```

Berikut adalah pengertian kode tersebut:

1. `data['Preprocess']`: Ini merujuk pada kolom 'Preprocess' di DataFrame data.
2. `.str.replace('[^a-zA-Z0-9]+',' ',regex=True)`: Metode ini digunakan untuk

menggantikan setiap karakter yang tidak termasuk dalam kumpulan karakter [a-zA-Z0-9] (huruf dan angka) dengan spasi. Dengan kata lain, semua karakter yang bukan huruf atau angka akan digantikan dengan spasi.

3. `[^a-zA-Z0-9]`: Pada ekspresi reguler ini, `^` di dalam kurung siku (`[]`) menunjukkan negasi, dan `a-zA-Z0-9` merupakan kumpulan karakter huruf dan angka.
4. `' '`: Ini adalah *string* pengganti, dalam hal ini, spasi.
5. `regex=True`: Parameter ini menandakan bahwa ekspresi reguler digunakan dalam proses pencarian dan penggantian.

Tabel 3. 7 Tabel Hasil *Punctuation Removal*

Contoh Kalimat	Hasil <i>Punctuation Removal</i>
Pelanggan Telkomsel merasa senang dengan <i>promo</i> terbarunya, diskon 50%!	Pelanggan Telkomsel merasa senang dengan <i>promo</i> terbarunya diskon 50
Wah, Telkomsel lagi ngasih <i>promo</i> besar-besaran nih! Diskon 30% buat paket internet. Seru banget!	Wah Telkomsel lagi ngasih <i>promo</i> besar-besaran nih Diskon 30 buat paket internet Seru banget
Paket data Telkomsel murah banget, bisa buat nonton <i>streaming</i> seharian. Makin betah langganan!	Paket data Telkomsel murah banget bisa buat nonton <i>streaming</i> seharian Makin betah langganan

3.3.5 Stopword Removal

Stopword Removal digunakan untuk menghapus kata hubung yang tidak dihapus tidak memiliki makna dalam sebuah kalimat, hal yang tidak mempengaruhi dan tidak memiliki ciri langsung terhadap *sentiment* dan aspek dalam kalimat akan dihapus dikarenakan akan membuat *noise* pada data sehingga kata – kata hubung tersebut mempengaruhi nilai bobot terhadap *sentiment* dan aspek dalam data latih. Dalam studi kasus penelitian ini contoh dalam stopwords adalah “di”, “ketika”, “siapa” yang sebenarnya sebagai keterangan dalam sebuah objek. Proses eliminasi kata hubung tersebut dilakukan dengan *library* yang sudah disediakan melalui internet, namun perlu adanya modifikasi tambahan untuk *corpus* stopwords dikarenakan tidak semua stopwords dalam sebuah *corpus online* mendeteksi berdasarkan opini publik terhadap Telkomsel.

Tabel 3. 8 Tabel Hasil *Stopword Removal*

Contoh Kalimat	Hasil <i>Stopword Removal</i>
Telkomsel paket datanya murah dan kuotanya banyak.	Telkomsel paket data murah kuota banyak
Telkomsel lagi depan banget nih, paket data hemat buat pelanggan setia. Top deh!	Telkomsel depan banget, paket data hemat pelanggan setia. Top deh!
Hari ini jaringan Telkomsel lagi kencang banget! Browsing lancar, <i>download</i> cepat. Mantap!	Hari jaringan Telkomsel kencang! Browsing lancar, <i>download</i> cepat. Mantap!

Berikut beberapa contoh *stopword corpus* yang dimodifikasi sebagai tambahan dalam studi kasus penelitian yang diperoleh pada data penelitian.

Tabel 3. 9 *Stopword corpus*

Kata	Jenis Kata
telkomsel	Nama merek
jaringan	Kata benda
kuota	Kata benda
harga	Kata benda
paket	Kata benda

3.3.6 Tokenization

Tokenization merupakan tahap akhir dalam *Preprocessing* dimana bertujuan untuk pemenggalan tiap kata yang disebut *token* dalam sebuah kalimat. Proses ini bertujuan agar proses *Word Embedding* akan lebih mudah dan terstruktur. Proses ini dilakukan dengan pemenggalan berdasarkan *white space*. Adapun pseudocode yang digunakan pada tahapan ini sebagai berikut;

```
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
# Kalimat contoh
text = "Ini adalah contoh kalimat. Tokenisasi membantu memecah kalimat
menjadi kata-kata."
# Tokenisasi kata
tokens_word = word_tokenize(text)
print("Token Kata:", tokens_word)
# Tokenisasi kalimat
tokens_sentence = sent_tokenize(text)
print("Token Kalimat:", tokens_sentence)
```


Berikut adalah pengertian kode tersebut:

1. `import nltk`: Mengimpor modul NLTK.
2. `from nltk.tokenize import word_tokenize, sent_tokenize`: Mengimpor fungsi `word_tokenize` dan `sent_tokenize` dari modul `nltk.tokenize`.
3. `text = "Ini adalah contoh kalimat. Tokenisasi membantu memecah kalimat menjadi kata-kata."`: Menginisialisasi sebuah kalimat contoh.
4. `tokens_word = word_tokenize(text)`: Menggunakan `word_tokenize` untuk memecah kalimat menjadi *token* kata. Hasilnya adalah daftar kata-kata.
5. `tokens_sentence = sent_tokenize(text)`: Menggunakan `sent_tokenize` untuk memecah teks menjadi *token* kalimat. Hasilnya adalah daftar kalimat.
6. `print("Token Kata:", tokens_word)`: Mencetak *token* kata ke layar.
7. `print("Token Kalimat:", tokens_sentence)`: Mencetak *token* kalimat ke layar.

Tabel 3. 10 Tabel Hasil *Tokenization*

Kalimat Sebelum <i>Tokenisasi</i>	Hasil <i>Tokenisasi</i>
Telkomsel jaringannya lemot banget, nih. Mana lagi lagi gangguan. Gak bisa apa apa.	['Telkomsel', 'jaringannya', 'lemot', 'banget', ',', 'nih', ',', 'Mana', 'lagi', 'lagi', 'gangguan', ',', 'Gak', 'bisa', 'apa', 'apa', '.']
Pelayanan pelanggan Telkomsel selalu ramah dan membantu.	['Pelayanan', 'pelanggan', 'Telkomsel', 'selalu', 'ramah', 'dan', 'membantu', '.']
Telkomsel sering kali memberikan <i>promo</i> menarik untuk paket data.	['Telkomsel', 'sering', 'kali', 'memberikan', 'promo', 'menarik', 'untuk', 'paket', 'data', '.']

3.3.7 Menghilangkan *Missing value*

Bagian ini khusus untuk *preprocess* data historis saham. Di data saham terkadang terdapat data yang kosong. Untuk memperbaiki masalah ini, dapat dilakukan dengan cara mengosongkan data tersebut, mengisi data tersebut sendiri, menggunakan global konstanta atau bisa memakai nilai median / nilai rata rata. Dalam penelitian ini, untuk mengatasi masalah tersebut penulis menghapus data yang memiliki *Missing value* yang terdapat di data historis saham TLKM. Adapun pseudocode nya adalah sebagai berikut:

```
data_saham_cleaned = data_saham.dropna()
```

Berikut adalah pengertian kodingan diatas:

1. `data_saham` adalah DataFrame yang berisi data saham.

2. *Metode dropna()* digunakan untuk menghapus baris (axis=0) yang mengandung setidaknya satu nilai yang hilang.
3. Hasil operasi ini ditetapkan ke variabel baru *data_saham_cleaned*, yang sekarang berisi DataFrame yang telah dibersihkan dari baris yang memiliki nilai yang hilang.

3.4 Word Embedding - TFIDF

TF-IDF memproses setiap kata pada data menjadi vektor yang selanjutnya dikonversi (dilakukan perubahan) ke dalam bentuk nilai numerik agar dapat diproses menjadi model yang akan digunakan pada proses klasifikasi. Nilai kata akan diukur oleh TF- IDF Vectorizer berdasarkan bobot pada seluruh data yang ada. Proses ini digunakan agar mempermudah perhitungan n kata yang keluar secara frekuen untuk klasifikasi pada tiap rating.

Term Frequency (TF) menentukan seberapa sering suatu kata muncul dalam sebuah dokumen yang dimaksudkan. Nilai bobot dikalkulasi dengan menghitung jumlah total akta yang muncul dalam seluruh dokumen dibagi dengan keseluruhan kata. Jadi apabila semakin banyak frekuensi semakin besar pula nilai bobot yang diberikan dan akan mempengaruhi nilai probabilitas terhadap *sentiment* yang sudah dilatih sehingga dapat memberikan presisi dan akurasi terhadap data uji.

Inverse Document Frequency (IDF) Langkah selanjutnya setelah menghitung nilai bobot berdasarkan jumlah kemunculan kata. Perhitungan dilakukan berbanding terbalik dengan *Term Frequency* dimana nilai bobot akan lebih besar jika adanya kelangkaan atau kata yang jarang digunakan dalam dokumen. Hal ini bertujuan agar melihat variasi nilai yang memiliki hubungan dokumen namun tidak sering dimunculkan sehingga ada nilai yang diberikan agar lebih seimbang. Penerapan dan penentuan nilai bobot dapat dilihat dalam tabel 3.11. Adapun pseudocodenya sebagai berikut:

```
from sklearn.feature_extraction.text import TfidfVectorizer
def calculate_tfidf(data):
    """
    Fungsi ini menghitung TF-IDF dari data teks.
    Parameters:
    - data: List dari teks yang akan dihitung TF-IDF-nya.
    Returns:
    - tfidf_matrix: Matriks TF-IDF hasil perhitungan.
    """
```

```

# Inisialisasi objek TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()
# Menghitung TF-IDF dari data
tfidf_matrix = tfidf_vectorizer.fit_transform(data)
# Mengembalikan matriks TF-IDF
return tfidf_matrix
# Contoh penggunaan
data_teks = [
    "Ini adalah contoh dokumen pertama.",
    "Contoh dokumen kedua berisi beberapa kata.",
    "Dokumen ketiga adalah dokumen terakhir."
]
# Menghitung TF-IDF dari data teks
tfidf_result = calculate_tfidf(data_teks)
# Menampilkan hasil TF-IDF
print("Hasil TF-IDF:")
print(tfidf_result.toarray())

```

Berikut adalah penjelasan kodingan diatas:

1. *Import library*: Dalam kode di atas, *library* TfidfVectorizer dari scikit-learn digunakan untuk menghitung TF-IDF.
2. *Fungsi calculate_tfidf*: Fungsi ini menerima *input* berupa list teks dan mengembalikan matriks TF-IDF hasil perhitungan.
3. *Inisialisasi TfidfVectorizer*: Objek *tfidf_vectorizer* diinisialisasi menggunakan TfidfVectorizer().
4. *Menghitung TF-IDF*: *tfidf_vectorizer.fit_transform(data)* digunakan untuk menghitung TF-IDF dari data teks.
5. *Hasil Perhitungan*: Matriks hasil perhitungan TF-IDF disimpan dalam variabel *tfidf_matrix*.
6. *Contoh Penggunaan*: Sebuah contoh penggunaan diberikan dengan menggunakan beberapa dokumen teks.
7. *Menampilkan Hasil*: Hasil TF-IDF dalam bentuk matriks array ditampilkan.

Tabel 3. 11 Contoh Tabel Hasil Pembobotan TF-IDF

Kata	Frekuensi	Jumlah Cuitan	TF	IDF	TF-IDF
bagus	20	100	0.2	1	0.2
keren	15	100	0.15	1	0.15
oke	10	100	0.1	1	0.1

suka	8	100	0.08	1	0.08
mantap	7	100	0.07	1	0.07
<i>recommended</i>	6	100	0.06	1	0.06
terbaik	5	100	0.05	1	0.05
puas	4	100	0.04	1	0.04
<i>recommended banget</i>	3	100	0.03	1	0.03

3.5 Vader

Setelah melewati TF-IDF, kata-kata untuk *Vader* akan memiliki representasi vektor. Representasi vektor ini akan mencerminkan frekuensi kemunculan kata dalam dokumen dan frekuensi kemunculan kata dalam kumpulan dokumen. *Vader* harus melewati TF-IDF karena TF-IDF dapat meningkatkan akurasi model *Vader*. Hal ini karena TF-IDF dapat menangkap konteks kata dalam dokumen.

Misalnya, kata "*good*" dapat memiliki sentimen positif atau negatif, tergantung pada konteksnya. Kata "*good*" dapat memiliki sentimen positif jika digunakan dalam kalimat "*The movie was good.*" Namun, kata "*good*" dapat memiliki sentimen negatif jika digunakan dalam kalimat "*The food was not good.*"

TF-IDF dapat membantu *Vader* untuk membedakan antara kedua konteks tersebut. Hal ini karena TF-IDF akan menghitung frekuensi kemunculan kata "*good*" dalam dokumen. Jika kata "*good*" lebih sering muncul dalam dokumen yang memiliki sentimen positif, maka kata "*good*" akan memiliki representasi vektor yang lebih positif. Adapun pseudocode yang digunakan pada tahapan ini sebagai berikut;

```
import vaderSentiment
def analyze_sentiment(text):
    """
    Menganalisis sentimen teks menggunakan Vader.
    Args:
        text: Teks yang akan dianalisis.
    Returns:
        Skor sentimen teks.
    """
    # Mengkonversi teks menjadi kata-kata.
    tokens = nltk.word_tokenize(text)
    # Menghitung TF-IDF untuk setiap kata.
    tf_idf = tf_idf(tokens)
    # Menghitung skor sentimen.
    sentiment = vaderSentiment.SentimentIntensityAnalyzer().polarity_scores(text)["compound"]
    return sentiment
```

```
text = "The movie was good."
sentiment = analyze_sentiment(text)
print(sentiment)
```

Berikut adalah penjelasan kodingan tersebut:

1. Import *library*:
 - a. *import vader Sentiment*: Mengimpor *library VADER Sentiment* untuk analisis sentimen.
2. Definisi Fungsi *analyze_sentiment*:
 - a. *def analyze_sentiment(text)::* Mendefinisikan fungsi *analyze_sentiment* yang menerima teks sebagai *input* dan mengembalikan skor sentimen.
 - b. *text*: Parameter *input* berupa teks yang akan dianalisis sentimennya.
 - c. *Returns*: Fungsi ini mengembalikan skor sentimen teks.
3. Tokenisasi Teks:
 - a. *tokens = nltk.word_tokenize(text)*: Menggunakan *library NLTK* untuk melakukan *tokenisasi* teks menjadi kata-kata. *Tokenisasi* adalah proses memecah teks menjadi unit kata atau *token*.
4. Menghitung TF-IDF:
 - a. *tf_idf = tf_idf(tokens)*: pemanggilan fungsi *tf_idf(tokens)* yang seharusnya menghitung TF-IDF untuk setiap kata. TF-IDF adalah metode yang umum digunakan untuk mengevaluasi pentingnya sebuah kata dalam suatu dokumen.
5. Menghitung Skor Sentimen menggunakan *VADER*:
 - a. *vaderSentiment.SentimentIntensityAnalyzer().polarity_Scores(text)["compound"]*: Menggunakan objek *SentimentIntensityAnalyzer* dari *VADER Sentiment* untuk menghitung skor sentimen. Dalam hal ini, kita menggunakan nilai "*compound*" yang memberikan skor sentimen keseluruhan.
6. Contoh Penggunaan dan Menampilkan Hasil:
 - a. *text = "The movie was good."*: Sebuah contoh teks yang akan dianalisis sentimennya.
 - b. *sentiment = analyze_sentiment(text)*: Memanggil fungsi *analyze_sentiment* untuk menganalisis sentimen dari teks.
 - c. *print(sentiment)*: Menampilkan skor sentimen hasil analisis.

Tabel 3. 12 Hasil analisis *sentiment* menggunakan *Vader* yang telah dibantu TF-IDF

Teks	Skor Sentimen
"Sinyal Telkomsel ngadat lagi di daerah sini! Sering banget kejadian kayak gini."	-0.4232
"Kaget banget tiba-tiba tagihan internet IndiHome melonjak drastis! Ada apa nih Telkom?"	-0.7432
" <i>Customer service</i> Telkomsel lama banget dihubungnya! Kesal nungguinnya."	-0.5832
"Telkomsel Orbit penyelamat banget buat mahasiswa kos kayak aku! Internetan kenceng tanpa kabel, top deh!"	0.9522
"Mau tanya dong, paket internet IndiHome yang mana yang paling cocok buat keluarga?"	0.0332
"Seneng banget ada layanan MyTelkomsel! Bikin isi pulsa & beli paket data jadi gampang dan cepet."	0.9212
"Kecepatan internet IndiHome ku akhir-akhir ini stabil banget! Puas deh langganan sama Telkom!"	0.8732
"Telkomsel lagi ada promo paket data nih, tertarik sih tapi baca dulu deh detailnya."	0.0832

Setelah mendapat kan Skor Sentimen setiap teks maka akan dilanjutkan dengan membuat rata-rata skor sentiment minggu tersebut. Artinya jumlah skor sentimen setiap teks pada minggu tersebut akan ditambahkan lalu dibagi dengan total jumlah teks nya. Hal ini dilakukan karena jumlah tweet setiap minggu berbeda-beda jumlahnya.

3.6 Perancangan Model algoritma *Gated Recurrent Unit*

Dalam penelitian ini, *GRU Classifier* digunakan menggunakan *library* Python yaitu *TensorFlow*. *TensorFlow* merupakan salah satu *library* yang memiliki proses GRU tanpa melakukan perhitungan manual secara langsung terhadap algoritma yang diterapkan. Setelah setiap kata sudah menjadi angka, maka nilai tersebut akan dilakukan proses data latih. Untuk mendapatkan model yang terbaik, maka akan dilakukan *Hyperparameter*. Penjelasan secara detail langkah untuk mendapatkan model dari algoritma GRU:

3.6.1 Pembentukan *Input* dan *Output*

Untuk mempersiapkan data latih bagi model GRU, langkah awal adalah menentukan *input* dan *output* yang sesuai. Data yang telah melalui tahap *preprocessing* akan diatur sebagai *input*, sedangkan label atau targetnya merupakan pergerakan harga

saham Telkom yang akan diprediksi.

3.6.2 Inisialisasi Model

Model GRU diinisialisasi menggunakan *TensorFlow*. Pemilihan jumlah unit GRU, *dropout rates*, dan fungsi aktivasi dilakukan secara cermat untuk mencapai keseimbangan antara kompleksitas model dan kemampuan generalisasi.

3.6.3 Penentuan Hyperparameter

Untuk meningkatkan performa model, dilakukan penentuan *hyperparameter* melalui proses *hyperparameter tuning*. Beberapa *hyperparameter* yang dioptimalkan meliputi *learning rate*, *batch size*, jumlah *epoch*, dan lainnya.

3.6.4 Pelatihan Model

Model yang telah diinisialisasi dan diberi *hyperparameter* akan dilatih menggunakan data latih. Proses pelatihan ini bertujuan untuk menyesuaikan parameter model agar dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya.

3.6.5 Evaluasi Model

Setelah pelatihan selesai, model dievaluasi menggunakan data validasi untuk mengukur performa dan mengidentifikasi potensi *overfitting* atau *underfitting*.

3.6.6 Penyimpanan Model

Model yang telah berhasil dilatih akan disimpan untuk digunakan pada tahap prediksi harga saham.

3.7 Output

Setelah mendapatkan hasil dari proses pengujian, akan dihasilkan *output* yang memberikan wawasan mendalam mengenai prediksi pergerakan harga saham. Prediksi ini merupakan hasil analisis model *Gated Recurrent Unit* (GRU) terhadap data uji, yang mengungkapkan proyeksi potensial pergerakan harga saham di masa mendatang.

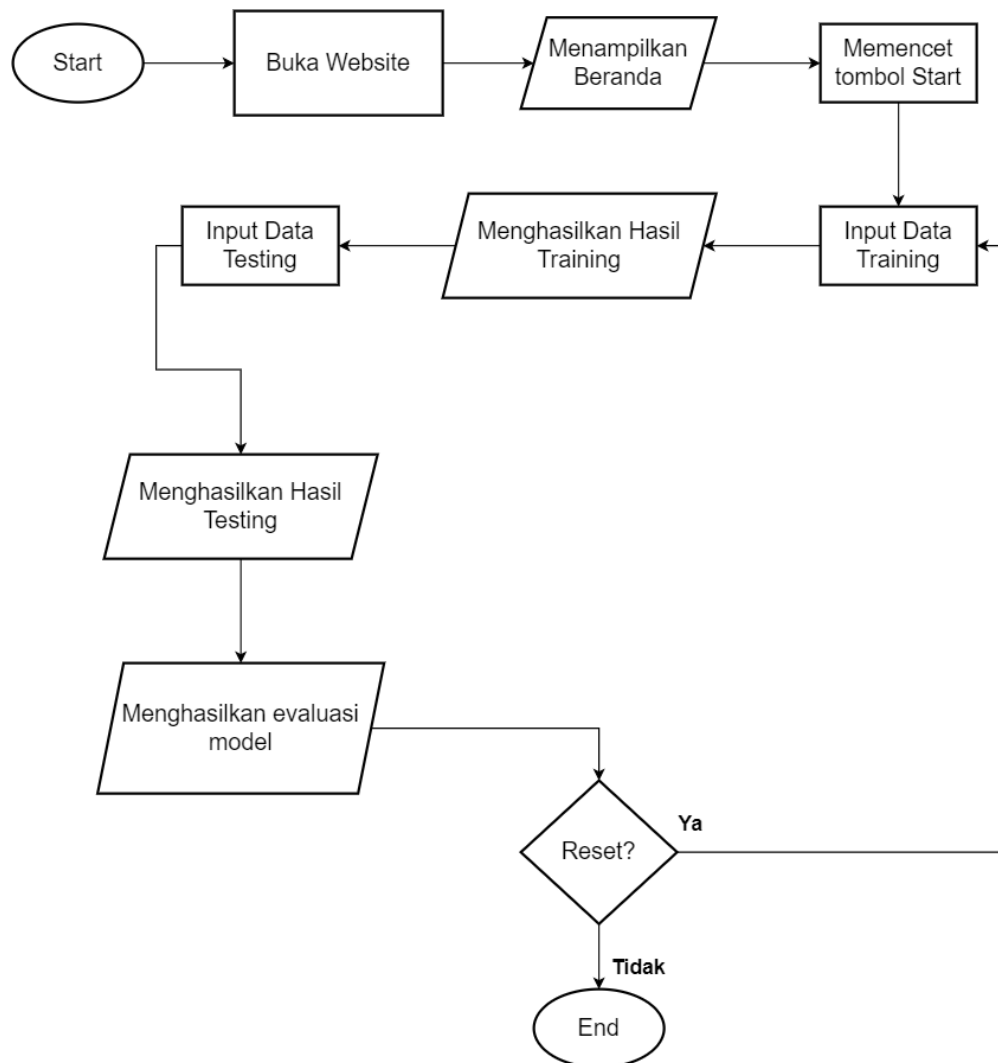
3.8 Flowchart Diagram

Flowchart diagram atau yang biasa dikenal juga dengan istilah diagram alur merupakan diagram yang menggambarkan alur dari proses pengerjaan pada sebuah sistem. Diagram ini dibuat untuk memberikan penjelasan tentang tahapan- tahapan yang akan dilakukan ketika pengguna menggunakan *website*. Untuk proses alur kerja ketika *website* berjalan dapat dilihat pada gambar 3.4.

Proses pada *website* dimulai dengan membuka halaman beranda setelah pengguna mengaksesnya. Pada beranda, pengguna dapat memulai proses dengan mengklik tombol "*Start*". Setelah itu, mereka akan diarahkan ke halaman *Training*, di mana diminta untuk memasukkan data pelatihan yang diperlukan. *Website* kemudian mengolah data tersebut untuk menghasilkan model pembelajaran.

Setelah proses *Training* selesai, pengguna diarahkan ke halaman *testing*, di mana mereka diminta untuk memasukkan data yang akan diuji. *Website* akan mengolah data *testing* tersebut dan mengevaluasi model, serta menampilkan hasilnya. Selanjutnya, pengguna diberikan opsi untuk mengulang (*reset*) proses dari awal dengan kembali ke halaman *Training* untuk memasukkan data baru atau memilih untuk mengakhiri proses.

Seluruh proses tersebut diimplementasikan dalam *website* yang menggunakan berbagai *library* dan modul seperti *Flask*, *googletrans*, *keras*, *matplotlib*, *nlTK*, *numpy*, *pandas*, *scikit_learn*, *scipy*, *seaborn*, *tensorflow*, *tqdm*, dan *Werkzeug*. Pastikan bahwa semua requirement-program ini terpenuhi dalam lingkungan pengembangan *website* untuk memastikan kelancaran dan keberhasilan eksekusi program tersebut.



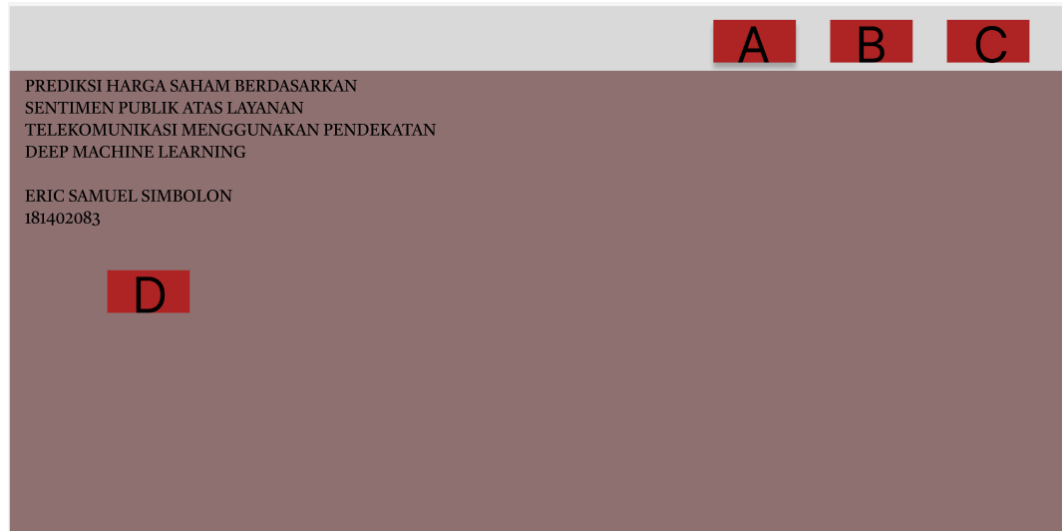
Gambar 3. 4 Flowchart Diagram Website rancangan

3.9 Rancangan Sistem

3.9.1 Antarmuka Pengguna

Untuk dapat melihat sistem secara keseluruhan perlu adanya media yang dapat melakukan eksekusi dengan antarmuka yang memadai, antarmuka akan dibuat dengan berbasis *web* dan akan dibagi berdasarkan tiga halaman antara lain halaman beranda, halaman *Training*, dan halaman *testing*. Berikut penjelasan rancangan berdasarkan halaman.

1. Halaman Utama Beranda

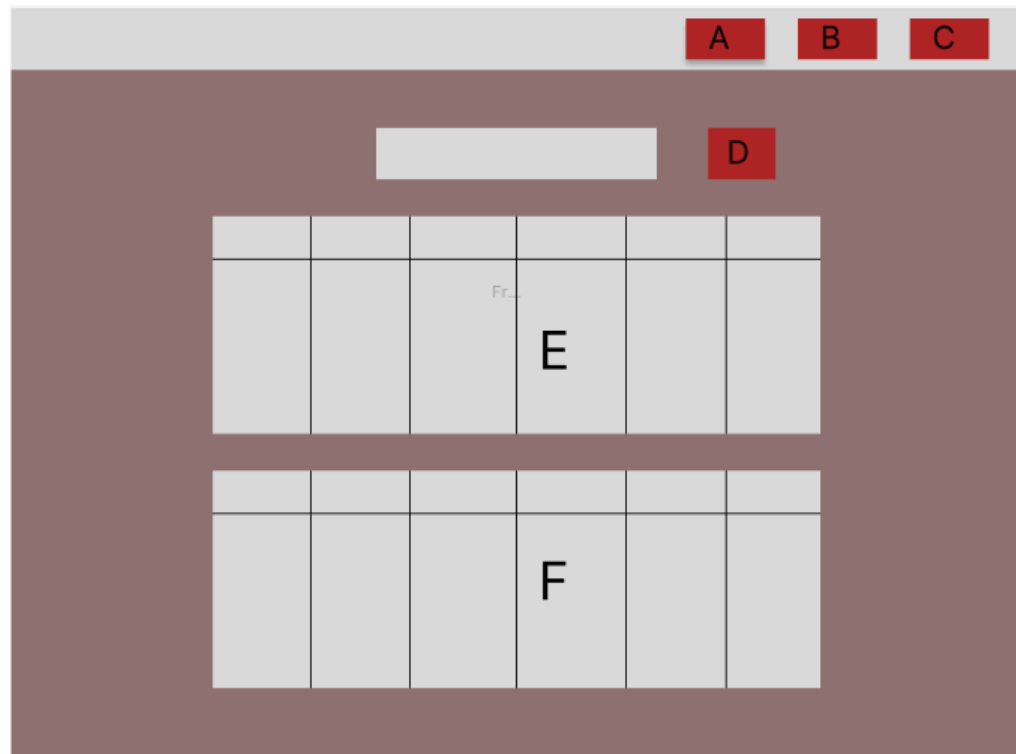


Gambar 3. 5 Halaman Utama Beranda

Halaman utama beranda merupakan halaman awal yang akan ditampilkan ketika *web* diakses. Tujuan dibangun halaman utama sebagai media informasi untuk melihat judul serta nama penulis, disertai beberapa *Button* pendukung untuk navigasi. Tiap huruf dalam sebuah rancangan antarmuka dijelaskan sebagai berikut :

- a. *Button* Halaman *Home*
- b. *Button* Halaman *Training*
- c. *Button* Halaman *Testing*
- d. *Button* *Start Website*

2. Halaman *Training*

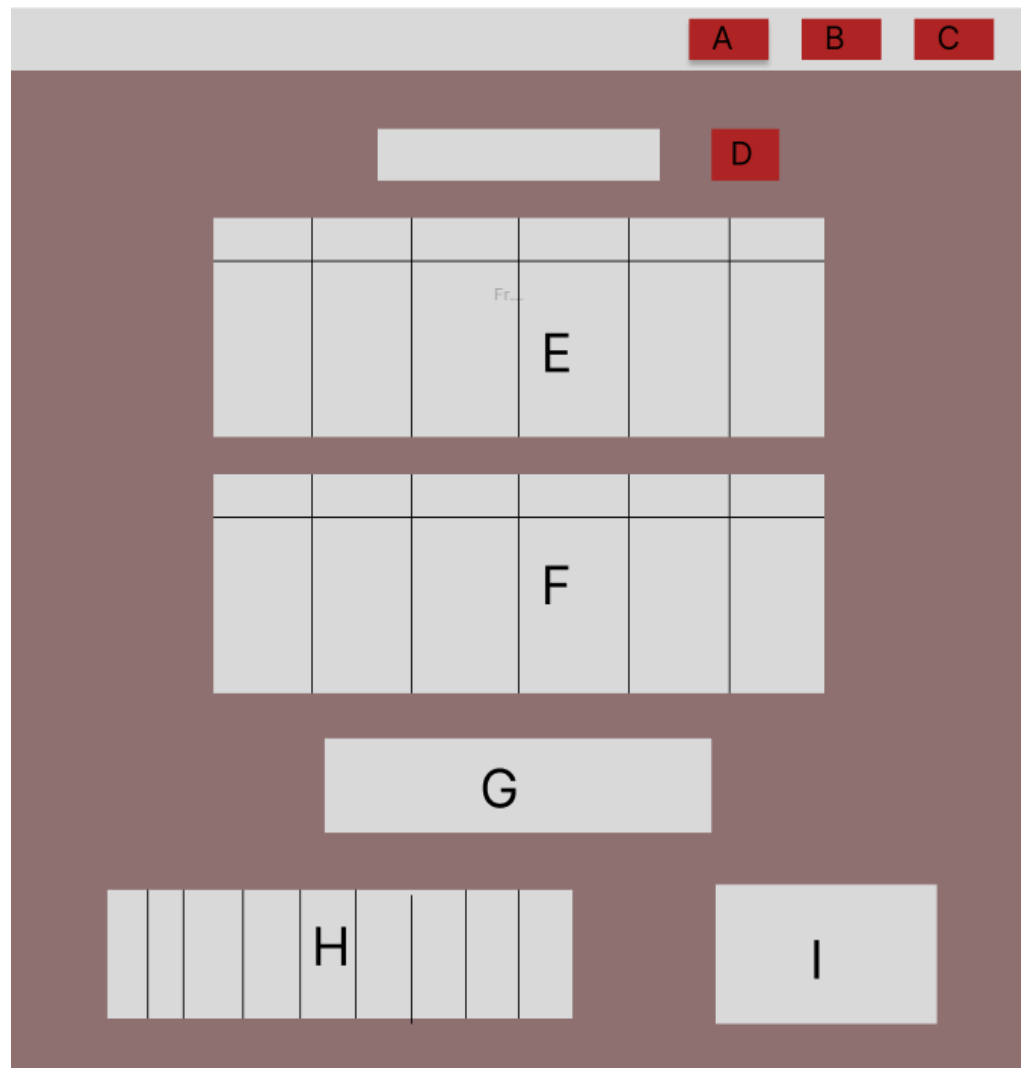


Gambar 3. 6 Halaman *Training*

Selanjutnya navigasi akan diarahkan sesuai dengan proses alur utama yang itu melakukan *Training*, proses *Training* akan dilakukan secara langsung dengan menginput data berformat csv ke dalam *website*, kemudian hasil *Training* akan disimpan dalam model, Kolom E dan F adalah hasil dari *preprocess* tersebut. Berikut penjelasan untuk tiap label :

- a. *Button* Halaman Utama
- b. *Button* Halaman *Training*
- c. *Button* Halaman *Testing*
- d. *Button Browser Data Training File*
- e. Tabel hasil *Preprocessing Sentiment*
- f. Tabel hasil *Preprocessing data saha*

3. Halaman *Testing*



Gambar 3. 7 Halaman *Testing*

Selanjutnya setelah model disimpan, maka kita akan melakukan proses uji data pada halaman *testing* dimana *flow* untuk awal dilakukan *input* data *testing* seperti proses sebelumnya, hasil model yang disimpan akan di load Kembali dan melakukan prediksi berdasarkan data *testing* yang disediakan. Hasil dalam proses ini ada 2 yaitu data *tabular* untuk *sentiment* dan data *tabular* untuk data historis saham. Lalu setelah itu diberikan informasi evaluasi dalam bentuk Visual atau angka. Hasil Evaluasi terdiri dari MSE, RMSE, *Confusion Matrix* dan *Classification Report*. Untuk gambaran umumnya juga dibuat *Chart Comparison* sebagai pembanding. Untuk detail rancangan dapat dijelaskan sebagai berikut:

1. *Button* Halaman Utama
2. *Button* Halaman *Training*
3. *Button* Halaman *Testing*

4. *Button Browse Data Testing File*
5. *Tabel Hasil Sentimen Score*
6. *Tabel Hasil Prediksi Harga Saham*
7. *Informasi Comparison Chart, MSE, RMSE*
8. *Classification Report*
9. *Confusion Matrix*

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Implementasi merupakan fase penerapan dan uji coba sistem berdasarkan hasil analisis dan perancangan sebelumnya. Melalui implementasi, kita dapat memperoleh pemahaman yang lebih detail mengenai konteks penelitian yang sedang dilakukan. Pembuatan sistem ini membutuhkan perangkat keras (*hardware*) maupun perangkat lunak (*software*) pendukung, antara lain :

4.1 Perangkat Keras

Adapun spesifikasi perangkat keras yang digunakan dalam penelitian ini adalah :

1. Prosesor Intel core i5 5th gen 2.20 GHz
2. Kapasitas RAM sebesar 8 GB
3. Kapasitas Harddisk sebesar 1 TB

4.2 Perangkat Lunak

Adapun perangkat lunak yang dibutuhkan dalam pembangunan sistem adalah :

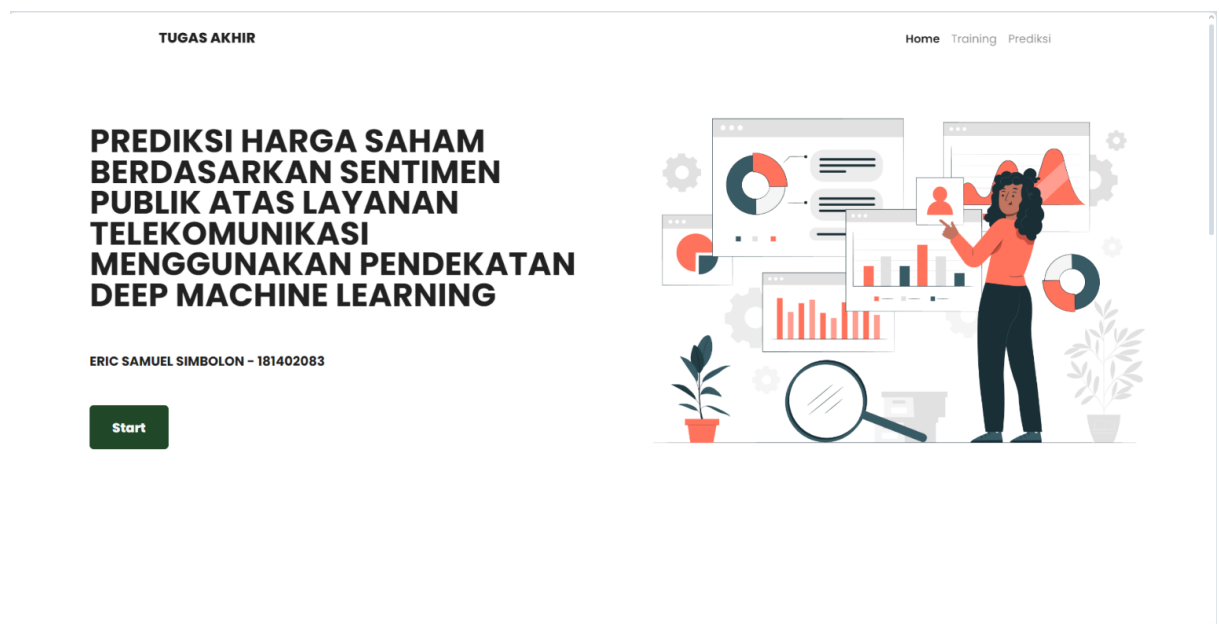
1. Sistem Operasi Windows 10 *Home* 64-bit
2. Visual Studio Code 2019 64-bit
3. Python 3.8.7
4. Library bahasa pemrograman Python, antara lain :
 - a. Flask v.1.1.2
 - b. Pandas v.1.1.5
 - c. Scikit-*learn* v.0.24.2
 - d. Matplotlib v.3.3.1
 - e. Nltk v.3.6.2
 - f. Numpy v.1.19.5
 - g. Sastrawi v.1.0.1
 - h. Seaborn v.0.10.1

4.3 Implementasi Perancangan Tampilan Antarmuka

Implementasi tampilan antarmuka dikerjakan berdasarkan rancangan yang telah tertera pada bab selanjutnya. Tampilan antarmuka dibuat sederhana sesuai dengan ruang lingkup kebutuhan penelitian

4.3.1 Halaman Beranda

Halaman beranda merupakan halaman pertama atau *landing page* yang muncul saat program dijalankan. Program dibagi menjadi komponen menu dan komponen konten. Komponen menu merupakan komponen tetap yang akan selalu muncul pada setiap halaman. Komponen konten pada halaman beranda terdapat informasi judul penelitian dan identitas peneliti. Halaman beranda dapat dilihat pada gambar 4.1.



Gambar 4. 1 Tampilan antarmuka halaman Beranda

4.3.2 Halaman *Training*

Halaman *Training* dibagi menjadi 2 komponen seperti halaman yang lain yaitu menu komponen menu yang digunakan untuk mengakses halaman lain dan komponen konten untuk *Training*. Pada komponen konten *Training*, *user* akan dihadapkan pada form untuk mengupload file dengan format .csv yang memuat data untuk dilakukan *Training*. Proses *Training* akan dijalankan setelah *user submit* data, setelah proses *Training* selesai maka *learn* model akan disimpan dan akan muncul data hasil *Preprocessing* pada konten halaman *Training*. Halaman *Training* dimuat pada gambar 4.2 dan gambar 4.3. Pada gambar 4.2, kita bisa melihat hasil *Training* untuk bagian data

Twitter. Data *twitter* di rangkum dalam 1 *tabular* dengan judul “Data hasil *Preprocessing*” Disini bisa dilihat tahun, minggu, *Tweet*, Hasil *Preprocessing*, *Translated Tweet*, dan *Sentiment Score* nya. Terdapat juga kolom *search* jika ingin mencari kata kata spesifik dari *tweet* yang telah dilatih

Year	Week	Tweet	Hasil Preprocessing	Translated Tweet	Sentiment Score
2018	Week 01	@pedersen145 Boleh Kak, silakan diinfokan yg ingin ditanyakan perihal produk Telkom nya via DM. Terimakasih - Pia	boleh kakak silakan diinfokan ingin ditanyakan perihal produk telkom via direct message terima kasih pia	can you please inform me about telkom products via direct message, thank you pia	0.5859
2018	Week 01	Watu wa Telkom , tweets zahappy New Year zitafika kesho. Stay strong	watu wa telkom twit zahappy new tahun zitafika kesho stay strong	telkom people twit zahappy new year will arrive tomorrow stay strong	0.5106
2018	Week 01	@TelkomCare selamat pagi, saya pelanggan telkom indihome, begini	selamat pagi saya pelanggan telkom indihome begini jaringan	good morning, i am a Telkom Indihome customer. This is how	0.6908

Gambar 4. 2 Antarmuka halaman *Training Data Twitter*

Pada gambar 4.3 Kita bisa melihat hasil dari *Training* data nya Untuk bagian Data Saham nya. Data Saham dirangkum dalam satu data *tabular* dengan judul “Rekapitulasi Sentimen *Score* Perminggu” Disini bisa dilihat Tahun, Minggu, Rata-rata *sentiment Score* minggu tersebut, *Open*, *High*, *Low*, dan *Close*. Disini juga ada bagian *search* untuk mencari sesuatu yang spesifik seperti minggu atau harga tertentu

Year	Week	Sentiment Score	Open	High	Low	Close
2018	Week 01	0.30573500000000015	3558.47	3578.51	3510.92	3538.8
2018	Week 02	0.2605405063281396	3451.88	3471.56	3425.65	3435.48
2018	Week 03	0.17765499999999995	3430.56	3450.24	3391.21	3420.73
2018	Week 04	0.24441875000000005	3335.45	3376.45	3284.62	3332.17
2018	Week 05	0.2364675	3314.14	3342.01	3266.58	3281.34
2018	Week 06	0.3165937500000001	3253.46	3292.82	3227.22	3263.3
2018	Week 07	0.16560124999999998	3287.9	3312.5	3268.22	3289.54
2018	Week 08	0.117295	3323.97	3337.09	3299.38	3317.42
2018	Week 09	0.2337837500000001	3305.94	3327.25	3278.06	3309.22
2018	Week 10	0.19421749999999993	3328.89	3355.13	3297.74	3333.81

Gambar 4. 3 Antarmuka halaman *Training Data Saham*

4.3.3 Halaman *Testing*

Halaman *testing* digunakan untuk menguji *learn* model yang telah di *Training*. Pada halaman *testing*, *user* akan diminta untuk mengunggah data *testing* untuk selanjutnya dilakukan proses *testing* yang menghasilkan data yang telah dilakukan prediksi menggunakan *learn* model yang sebelumnya disimpan. Pada halaman *testing* juga terdapat hasil evaluasi atau pengujian berupa MSE, RMSE, *Confusion Matrix* serta *Classification Report* yang memuat hasil pengujian berupa *recall*, *precision*, *f1-Score*, *support*, serta *accuracy*. Perbandingan harga *actual* dan harga prediksi juga disajikan dalam bentuk grafik disini. Halaman *testing* ditampilkan pada gambar 4.4 dan gambar 4.5.

Pada gambar 4.4 kita bisa melihat hasil data saham yang telah di *testing*. Bisa dilihat di bagian table nya ada Tahun, minggu, *tweet*, hasil *Preprocessing*, *Translated tweet*, dan *sentiment Score*. Data dari *twitter* ini akan dihitung rata rata per minggu dan melalui pembelajaran dari model sebelumnya akan melakukan komputasi untuk memprediksi harga saham

TUGAS AKHIR						
Prediksi Data						
Upload dataset untuk proses Prediksi						
<input type="button" value="Choose File"/> translated_tweet_test.csv <input type="button" value="Reset"/>						
Hasil Prediksi						
Data Hasil Sentimen						
Year	Week	Tweet	Hasil Preprocessing	Translated Tweet	Sentiment Score	
2022	Week 01	@ariefrasyad Salah satu contoh hoax dari buzzer ganjaris ... itu lobang proyek Telkom, bukan sumur resapan. Cebong buzzer bayaran jahat ini sangat memusuhi Anies ...	salah satu contoh hoaks dari buzzer ganjaris itu lubang proyek telkom bukan sumur resapan kecebong buzzer bayaran jahat ini sangat memusuhi anies	an example of a hoax from the buzzer reward is that the telkom project hole is not a tadpole infiltration well this evil paying buzzer is very hostile to anies	-0.8104	
2022	Week 01	Salah satu contoh hoax dari buzzer ganjaris ... itu lobang proyek Telkom, bukan sumur resapan. Cebong buzzer bayaran jahat ini sangat memusuhi Anies ...	salah satu contoh hoaks dari buzzer ganjaris itu lubang proyek telkom bukan sumur resapan kecebong buzzer bayaran jahat ini sangat memusuhi anies	an example of a hoax from the buzzer reward is that the telkom project hole is not a tadpole infiltration well this evil paying buzzer is very hostile to anies	-0.8104	
2022	Week 01	@akbartaufig Baik, selanjutnya Kurnia bantu via DM bapak ya. -Kurnia	baik selanjutnya kurnia bantu via direct message bapak kurnia	OK, then Kurnia will help via direct message from Mr. Kurnia	0.6841	
2022	Week 01	@akbartaufig Selamat pagi, Bapak Akbar. Terkait pelaporannya telah Kurnia terima dan bantu respon via DM	selamat pagi bapak akbar terkait pelaporannya telah kurnia terima dan bantu respon via direct	Good morning, Mr. Akbar, regarding the report, I have received it and helbed respond via	0.6597	

Gambar 4. 4 Halaman *testing* dengan hasil prediksi *sentiment*

Pada gambar 4.5 bisa dilihat hasil inti dari program penelitian ini. Terdapat Tahun, Minggu, Rata-rata *Sentiment Score* Minggu tersebut, *Open*, *High*, *Low*, *Close Actual*, *Close Prediction*, *Weekly Comparison Actual*, *Weekly Comparison Prediction*. Terdapat juga *search* bar jika ingin mencari sesuatu seperti *sentiment Score* tertentu.

TUGAS AKHIR

Showing 1 to 10 of 3,802 entries

Previous 1 2 3 4 5 ... 381 Next

Stock Price Prediction

Show 10 entries

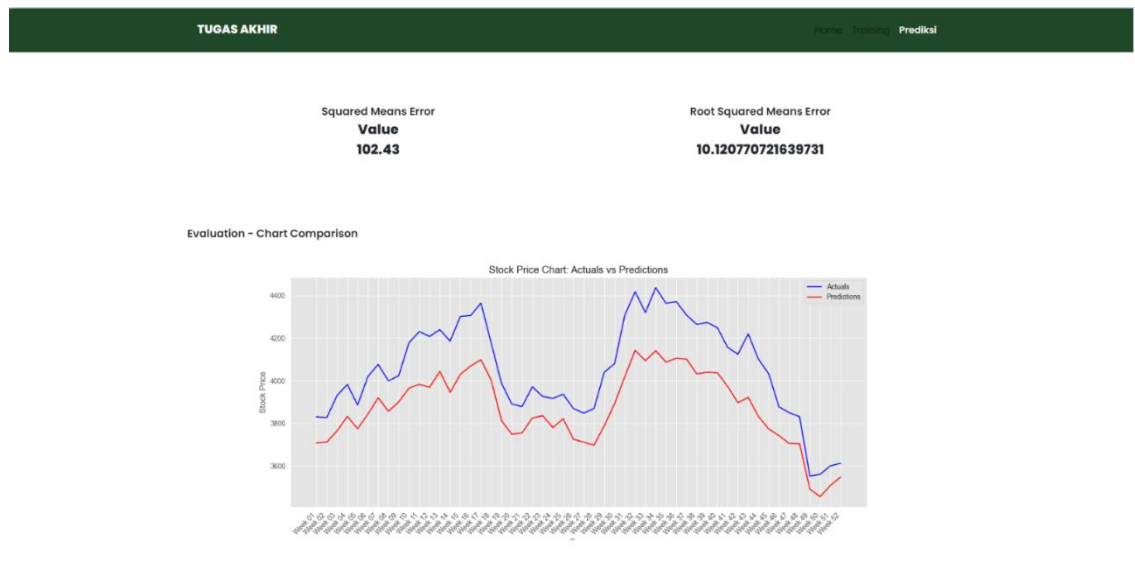
Year	Week	Sentiment Score	Open	High	Low	Close Actual	Close Prediction	Weekly Comparison Actual	Weekly Comparison Prediction
2022	Week 01	0.2928222222222223	3814.11	3865.95	3787.82	3830.77	3708.1943	Higher	
2022	Week 02	0.25899589041095903	3836.32	3862.24	3795.59	3827.07	3711.853	Lower	Higher
2022	Week 03	0.19522328767123281	3925.2	3951.12	3871.5	3930.75	3764.3433	Higher	Higher
2022	Week 04	0.26180000000000003	4001.11	4021.47	3940.01	3982.59	3832.3206	Higher	Higher
2022	Week 05	0.34108767123287687	3897.42	3915.94	3880.39	3885.85	3774.2441	Lower	Lower
2022	Week 06	0.23537945205479452	4015.92	4049.25	3962.23	4019.62	3843.2935	Higher	Higher
2022	Week 07	0.38060684931506855	4077.02	4106.64	4032.58	4077.02	3919.9185	Higher	Higher
2022	Week 08	0.373109589041096	3978.89	4039.99	3943.71	3999.26	3856.6938	Lower	Lower
2022	Week 09	0.1755739728027397	4159.72	4162.81	3988.91	4023.94	3900.4265	Higher	Higher
2022	Week 10	0.2564630136986302	4158.49	4247.36	4093.68	4178.85	3965.2454	Higher	Higher

Showing 1 to 10 of 52 entries

Previous 1 2 3 4 5 6 Next

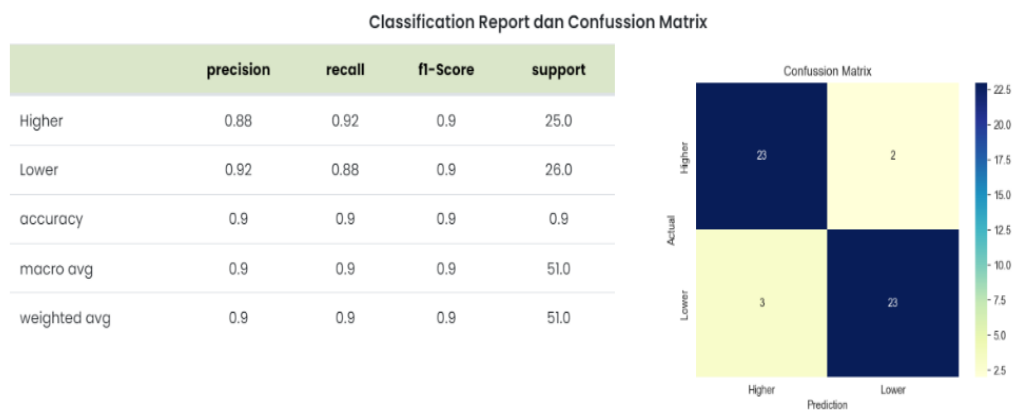
Gambar 4.5 Halaman *testing* dengan hasil prediksi saham

Pada gambar 4.6, kita dapat melihat tampilan halaman *testing* yang menyajikan hasil evaluasi kinerja model. Evaluasi ini mencakup pengukuran *Mean squared error* (MSE) dan *Root mean squared error* (RMSE), yang memberikan gambaran tentang sejauh mana perbedaan antara nilai prediksi dan nilai aktual pada data *testing*. *Mean squared error* (MSE) merupakan metrik yang mengukur rata-rata dari kuadrat perbedaan antara nilai prediksi dan nilai aktual. Semakin rendah nilai MSE, semakin baik performa model. *Root mean squared error* (RMSE) adalah akar kuadrat dari MSE, memberikan nilai yang lebih mudah diinterpretasikan. RMSE memberikan gambaran tentang seberapa dekat prediksi model dengan nilai aktual, dan seperti MSE, semakin rendah nilainya, semakin baik. Selain evaluasi numerik, gambar 4.6 juga menampilkan *Chart Comparison* yang memberikan visualisasi perbandingan antara harga saham aktual dan prediksi model. Grafik ini membantu kita untuk lebih memahami sejauh mana model dapat merepresentasikan perilaku harga saham pada data *testing*. Kombinasi hasil evaluasi numerik dan visualisasi grafik pada halaman *testing* ini memberikan informasi yang komprehensif mengenai performa model dalam menghasilkan prediksi pada dataset yang belum pernah dilihat sebelumnya."



Gambar 4. 6 Halaman *testing* dengan hasil Evaluasi MSE, RMSE dan *Chart Comparison*

Pada gambar 4.7, kita dapat melihat halaman *testing* yang menampilkan hasil evaluasi dengan menggunakan *Classification Report* dan *Confusion Matrix*. Evaluasi ini bersifat khusus untuk model yang melakukan klasifikasi, seperti yang mungkin terjadi pada kasus pengenalan sentimen atau kategorisasi. *Classification Report* memberikan *insight* mendalam tentang performa model dalam mengklasifikasikan data. *Report* ini mencakup *Precision*, *Recall*, dan *F1-Score* untuk setiap kelas yang dihasilkan oleh model. *Precision* mengukur seberapa akurat model dalam mengidentifikasi *instance* positif, *Recall* mengukur sejauh mana model dapat mendeteksi *instance* positif, dan *F1-Score* adalah harmonic mean dari *Precision* dan *Recall*. *Confusion Matrix*, di sisi lain, adalah tabel yang menyajikan perbandingan antara klasifikasi aktual dan klasifikasi yang diprediksi oleh model. Matriks ini berguna untuk mengevaluasi sejauh mana model mampu membedakan antara kelas positif dan negatif. Pada gambar 4.7, kita dapat memeriksa bagaimana model mengatasi setiap kelas dan mengidentifikasi pola kesalahan klasifikasi yang mungkin terjadi. Dengan kombinasi *Classification Report* dan *Confusion Matrix*, kita dapat memahami performa model secara lebih mendalam dalam konteks klasifikasi pada data *testing*.



Gambar 4. 7 Halaman *testing* dengan hasil Evaluasi *Classification Report* dan *Confusion Matrix*

4.4 Pengujian Sistem

4.4.1 Hyperparameter Model Gated Recurrent Unit

Dalam membangun model GRU, model akan ditentukan parameter yang terbaik untuk mendapatkan hasil akurasi yang tinggi. Proses itu akan dilakukan dengan metode *Hyperparameter*. *Hyperparameter* bertujuan melakukan proses latih yang secara berulang berdasarkan semua kemungkinan parameter yang ditentukan, nantinya kita akan melihat hasil akurasi tertinggi dan akan digunakan sebagai parameter utama kita dalam penelitian. Parameter yang ditentukan dalam proses ini yaitu *Epoch*, GRU Layer, *Number of units per GRU layer*, *Dropout rate*, *optimizer*, dan MSE. Hasil *Hyperparameter* dapat dilihat dalam tabel 4.1.

Tabel 4. 1 Tabel Hasil *Hyperparameter*

<i>Epochs</i>	<i>GRU Layer</i>	<i>Number of units per GRU Layer</i>	<i>Dropout rate</i>	<i>Optimizer</i>	<i>MSE</i>
1	4	50	0.2	rmsprop	138.68
5	4	50	0.2	rmsprop	110.23
10	4	50	0.2	rmsprop	98.43
15	4	50	0.2	rmsprop	97.32
20	4	50	0.2	rmsprop	96.67
25	4	50	0.2	rmsprop	95.82

Berdasarkan tabel di atas, terlihat bahwa nilai MSE menurun seiring dengan bertambahnya jumlah *epoch*. Nilai MSE terendah adalah 95.82, yang dicapai untuk

epoch 25. Namun, perlu diingat bahwa nilai *epoch* yang terlalu tinggi dapat menyebabkan *overfitting*. Oleh karena itu, nilai *epoch* 15 mungkin merupakan pilihan yang lebih baik, karena memiliki MSE yang mendekati nilai terendah, tetapi tidak terlalu tinggi. Kita perlu melihat lebih dekat pada nilai MSE untuk *epoch* 10 dan 15. Nilai MSE untuk *epoch* 10 adalah 98.43, sedangkan nilai MSE untuk *epoch* 15 adalah 97.32. Perbedaan nilai MSE ini sangat kecil, yaitu hanya 1.11. Hal ini menunjukkan bahwa model telah belajar pola data dengan cukup baik untuk *epoch* 10, dan tidak ada peningkatan kinerja yang signifikan untuk *epoch* yang lebih tinggi. Selain itu, *epoch* 10 memiliki waktu pelatihan yang lebih singkat daripada *epoch* 15. Oleh karena itu, *epoch* 10 dapat dianggap sebagai pilihan yang lebih baik, karena memberikan kinerja yang baik tanpa menyebabkan *overfitting* dan memiliki waktu pelatihan yang lebih singkat. Berdasarkan analisis di atas, dapat disimpulkan bahwa nilai *Hyperparameter* yang optimal untuk model GRU untuk prediksi harga saham adalah sebagai berikut:

1. GRU Layer = 4
2. Number of units per GRU Layer = 50
3. Dropout rate = 0.2
4. Optimizer = rmsprop
5. *Epochs* = 10

4.4.2 Hasil Pengujian Sistem

Model hasil implementasi algoritma *Gated Recurrent Unit* yang telah disimpan perlu dilakukan pengujian sistem menggunakan data *testing*. Data *testing* yang berisi kalimat opini *twitter* dan label akan dilakukan pengujian dengan cara memasukkan kalimat opini pada model untuk diprediksi. Data yang akan dilatih adalah data dari tanggal 01-01-2022 sampai dengan 31-12-2022

Hasil prediksi yang dihasilkan dari model akan dibandingkan dengan minggu sebelumnya seperti pada Tabel 4.2.

Tabel 4. 2 Hasil Prediksi Model *Gated Recurrent Unit*

Tahun	Minggu	<i>Sentiment Score</i>	<i>Open</i>	<i>High</i>	<i>Low</i>	<i>Close Actual</i>	<i>Close Prediction</i>	<i>Weekly Comparison Actual</i>	<i>Weekly Comparison Prediction</i>
2022	1	0.2928222222222223	3814.11	3865.95	3767.82	3830.77	3708.1943	<i>N/A</i>	<i>Higher</i>
2022	2	0.25869589041095903	3836.32	3862.24	3795.59	3827.07	3711.853	<i>Lower</i>	<i>Higher</i>
2022	3	0.19522328767123281	3925.2	3951.12	3871.5	3930.75	3764.3433	<i>Higher</i>	<i>Higher</i>
2022	4	0.26180000000000003	4001.11	4021.47	3940.01	3982.59	3832.3206	<i>Higher</i>	<i>Higher</i>
2022	5	0.34108767123287687	3897.42	3915.94	3860.39	3885.85	3774.2441	<i>Lower</i>	<i>Lower</i>
2022	6	0.23537945205479452	4015.92	4049.25	3962.23	4019.62	3843.2935	<i>Higher</i>	<i>Higher</i>
2022	7	0.38060684931506855	4077.02	4106.64	4032.58	4077.02	3919.9185	<i>Higher</i>	<i>Higher</i>
2022	8	0.373109589041096	3978.89	4039.99	3943.71	3999.26	3856.6938	<i>Lower</i>	<i>Lower</i>
2022	9	0.1755739726027397	4159.72	4162.81	3986.91	4023.94	3900.4265	<i>Higher</i>	<i>Higher</i>
2022	10	0.2564630136986302	4158.49	4247.36	4093.68	4178.85	3965.2454	<i>Higher</i>	<i>Higher</i>
2022	11	0.13845753424657534	4254.76	4273.28	4180.7	4230.69	3983.2178	<i>Higher</i>	<i>Higher</i>
2022	12	0.1945780821917809	4195.52	4226.99	4173.3	4208.48	3969.4717	<i>Lower</i>	<i>Lower</i>
2022	13	0.4047150684931508	4239.95	4269.58	4204.77	4239.95	4044.1658	<i>Higher</i>	<i>Higher</i>
2022	14	0.13126712328767123	4189.96	4215.88	4149.23	4186.26	3945.4995	<i>Lower</i>	<i>Lower</i>
2022	15	0.15540958904109592	4311.7	4325.58	4267.72	4302.44	4030.5679	<i>Higher</i>	<i>Higher</i>
2022	16	0.25265068493150694	4328.82	4354.75	4289.94	4306.61	4069.4775	<i>Higher</i>	<i>Higher</i>
2022	17	0.25443013698630135	4364.93	4425.1	4302.44	4364.93	4098.7705	<i>Higher</i>	<i>Higher</i>
2022	18	0.35356301369863036	4188.57	4264.02	4116.6	4176.54	4002.597	<i>Lower</i>	<i>Lower</i>

2022	19	0.06050958904109593	4012.22	4102.94	3930.75	3988.15	3812.6912	<i>Lower</i>	<i>Lower</i>
2022	20	0.1604794520547945	3906.68	3943.71	3862.71	3890.48	3748.5125	<i>Lower</i>	<i>Lower</i>
2022	21	0.31580684931506847	3874.28	3909.0	3830.31	3878.91	3754.861	<i>Lower</i>	<i>Higher</i>
2022	22	0.20387123287671224	4006.2	4038.6	3934.45	3971.48	3825.0554	<i>Higher</i>	<i>Higher</i>
2022	23	0.4255164383561646	3956.45	3983.11	3905.51	3926.15	3835.6655	<i>Lower</i>	<i>Higher</i>
2022	24	0.3848260273972602	3874.92	3942.1	3836.54	3917.15	3780.3748	<i>Lower</i>	<i>Lower</i>
2022	25	0.4493465753424658	3926.74	3965.13	3876.84	3936.34	3821.178	<i>Higher</i>	<i>Higher</i>
2022	26	0.13154109589041096	3884.52	3915.23	3846.14	3869.17	3724.761	<i>Lower</i>	<i>Lower</i>
2022	27	0.17533835616438354	3871.09	3890.28	3794.32	3848.06	3711.418	<i>Lower</i>	<i>Lower</i>
2022	28	0.05518904109589043	3867.25	3919.07	3809.67	3869.17	3697.313	<i>Higher</i>	<i>Lower</i>
2022	29	-0.01285616438356163	4009.27	4049.57	3974.72	4039.98	3788.6875	<i>Higher</i>	<i>Higher</i>
2022	30	0.22693698630136988	4087.96	4112.91	4034.22	4080.28	3891.8237	<i>Higher</i>	<i>Higher</i>
2022	31	0.23495616438356162	4235.74	4325.94	4197.35	4308.67	4019.9104	<i>Higher</i>	<i>Higher</i>
2022	32	0.29914246575342457	4423.82	4452.61	4379.68	4418.07	4143.1685	<i>Higher</i>	<i>Higher</i>
2022	33	0.3727301369863013	4315.87	4392.64	4236.7	4320.67	4094.1604	<i>Lower</i>	<i>Lower</i>
2022	34	0.25344931506849305	4429.58	4487.16	4372.0	4437.26	4141.1733	<i>Higher</i>	<i>Higher</i>
2022	35	0.2700013698630137	4312.51	4421.9	4285.64	4364.33	4087.5696	<i>Lower</i>	<i>Lower</i>
2022	36	0.26566438356164385	4360.49	4423.82	4331.7	4372.0	4105.733	<i>Higher</i>	<i>Higher</i>
2022	37	0.3389986301369863	4350.89	4373.92	4302.91	4308.67	4101.653	<i>Lower</i>	<i>Lower</i>
2022	38	0.24228767123287664	4268.37	4308.67	4239.58	4264.53	4031.9973	<i>Lower</i>	<i>Lower</i>
2022	39	0.3177369863013698	4247.25	4320.19	4201.19	4274.12	4040.3362	<i>Higher</i>	<i>Higher</i>

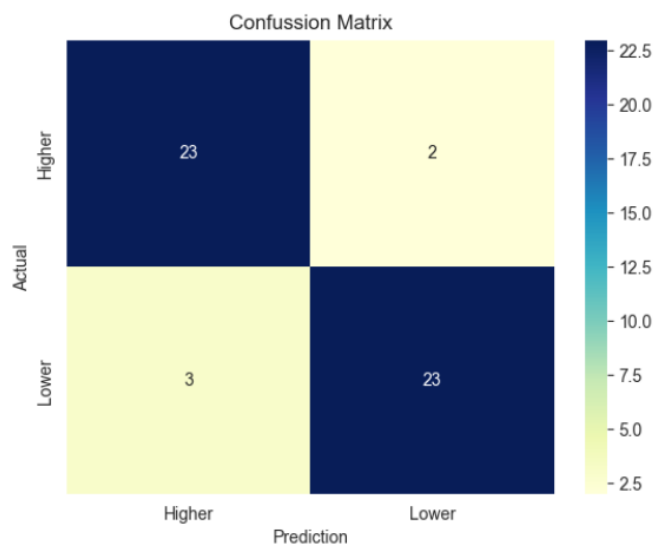
2022	40	0.274231506849315	4274.12	4304.83	4229.98	4249.17	4037.5886	<i>Lower</i>	<i>Lower</i>
2022	41	0.3262164383561643	4162.81	4195.44	4126.34	4157.05	3973.3037	<i>Lower</i>	<i>Lower</i>
2022	42	0.16899589041095897	4074.52	4166.65	4059.17	4124.42	3896.9277	<i>Lower</i>	<i>Lower</i>
2022	43	-0.0832273972602739	4229.98	4262.61	4187.76	4220.39	3921.804	<i>Higher</i>	<i>Higher</i>
2022	44	-0.1256698630136986	4124.42	4160.89	4041.9	4101.39	3831.8987	<i>Lower</i>	<i>Lower</i>
2022	45	-0.1266054794520548	4028.46	4064.93	3990.08	4032.3	3773.3643	<i>Lower</i>	<i>Lower</i>
2022	46	0.2074520547945205	3882.6	3919.07	3848.06	3876.84	3741.995	<i>Lower</i>	<i>Lower</i>
2022	47	0.15942602739726025	3842.3	3878.76	3826.94	3849.97	3705.331	<i>Lower</i>	<i>Lower</i>
2022	48	0.20557260273972597	3828.86	3871.09	3803.91	3830.78	3704.125	<i>Lower</i>	<i>Lower</i>
2022	49	0.2113191780821918	3569.77	3629.26	3496.84	3552.49	3491.0466	<i>Lower</i>	<i>Lower</i>
2022	50	0.0665123287671233	3544.82	3604.31	3519.87	3560.17	3455.2336	<i>Higher</i>	<i>Lower</i>
2022	51	0.21346027397260267	3583.2	3631.18	3548.66	3600.47	3508.232	<i>Higher</i>	<i>Higher</i>
2022	52	0.3031600000000001	3613.91	3654.21	3579.36	3611.99	3547.3132	<i>Higher</i>	<i>Higher</i>

4.4.3 Evaluasi Model

Model yang telah dihasilkan melalui proses *Training* algoritma *Gated Recurrent Unit* selanjutnya dievaluasi menggunakan *Confusion Matrix*. Hasil prediksi yang telah dilakukan menggunakan data *testing* akan dibandingkan dengan harga aktual dari data tersebut. Hal tersebut dilakukan guna mengukur performa dari model yang dihasilkan seberapa baik model dalam menentukan naik atau turunnya berdasarkan data yang diberikan. Dibantu dengan *library sklearn* berikut hasil *Classification Report* dan heatmap *Confusion Matrix* yang dihasilkan.

Tabel 4. 3 *Classification Report* dan *Confusion Matrix*

Index	<i>Precision</i>	<i>Recall</i>	<i>F1 -Score</i>	<i>Support</i>
<i>Higher</i>	0.88	0.92	0.9	25.0
<i>Lower</i>	0.92	0.88	0.9	26.0
<i>Accuracy</i>	0.9	0.9	0.9	0.9
<i>macro avg</i>	0.9	0.9	0.9	51.0
<i>weighted avg</i>	0.9	0.9	0.9	51.0



Gambar 4. 8 *Confusion Matrix*

Berdasarkan informasi dari heatmap pada gambar 4.8, evaluasi dapat ditentukan dengan mengambil tiap value TPHP (*True Positive Hasil Prediksi*), FNHP (*False*

Negative Hasil Prediksi), FPHP (*False Positive* Hasil Prediksi) dan TNHP (*True Negative* Hasil Prediksi) pada tabel 4.4 berikut.

Tabel 4. 4 Keterangan Prediksi dan *Actual Confusion Matrix*

No	Hasil Prediksi	Total
1	TPHP(<i>True Positive</i> Hasil Prediksi)	23
2	FPHP(<i>False Positive</i> Hasil Prediksi)	3
3	TNHP(<i>True Negative</i> Hasil Prediksi)	23
4	FNHP(<i>False Negative</i> Hasil Prediksi)	2

Setelah mendapatkan hasil prediksi model *Gated Recurrent* unit dan juga data dari *Confusion Matrix*, maka dapat dilakukan perhitungan untuk mencari *Precision*, *Recall*, *F1-Score*, Akurasi, MSE, dan RMSE. Rumus Persamaan dibuat sesuai dengan persamaan di bagian metode evaluasi pada Bab 2. Berikut adalah perhitungannya :

$$\begin{aligned}
 \text{Precision aspek Hasil Prediksi} &= \frac{TPHP}{TPHP+FPHP} \times 100 \\
 &= \frac{23}{23+3} \times 100 = 0.88
 \end{aligned}$$

$$\begin{aligned}
 \text{Recall aspek Hasil Prediksi} &= \frac{TPHP}{TPHP+FNHP} \times 100 \\
 &= \frac{23}{23+2} \times 100 = 0.92
 \end{aligned}$$

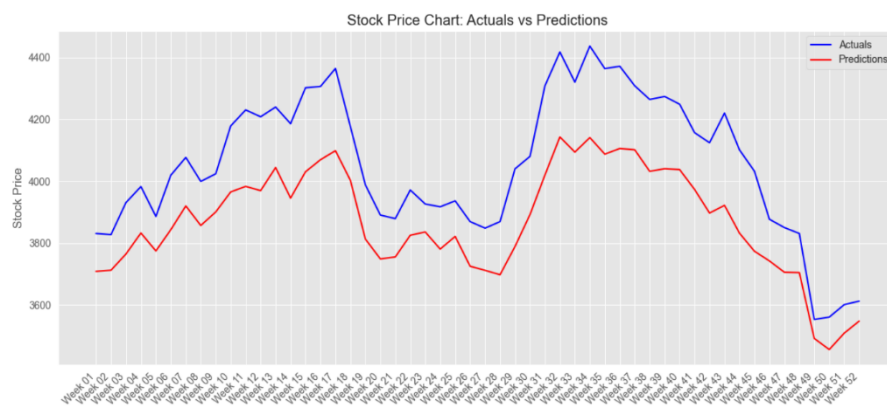
$$\begin{aligned}
 \text{F1-Score aspek Hasil Prediksi} &= \frac{2TPHP}{(2TPHP+FPHP+FNHP)} \times 100 \\
 &= \frac{2 \times 23}{2 \times 23 + 3 + 2} \times 100 = 0.9020
 \end{aligned}$$

$$\begin{aligned}
 \text{Akurasi Hasil Prediksi} &= \frac{TPHP+TNHP}{\text{Total data}} = \frac{23+23}{51} \\
 &= \frac{46}{51} = 0.9
 \end{aligned}$$

$$\begin{aligned}
 \text{MSE}(\text{mean square error}) &= \frac{\sum (\text{Aktual} - \text{Prediksi})^2}{n} \\
 &= \frac{(3830.77 - 3708.19)^2 + (3827.07 - 3711.85)^2 + \dots + (3611.99 - 3547.31)^2}{52} \\
 &= \frac{5326.36}{52} = 102.43
 \end{aligned}$$

$$\begin{aligned}
 \text{RMSE}(\text{root mean square error}) &= \sqrt{\frac{\sum (\text{Aktual} - \text{Prediksi})^2}{n}} \\
 &= \sqrt{\frac{(3830.77 - 3708.19)^2 + (3827.07 - 3711.85)^2 + \dots + (3611.99 - 3547.31)^2}{52}} \\
 &= \sqrt{\frac{5326.36}{52}} = \sqrt{102.43} \\
 &= 10.120770
 \end{aligned}$$

Evaluation - Chart Comparison

**Gambar 4. 9** Perbandingan Harga Saham Prediksi Dengan Harga Asli

Dalam Proses Pengujian data, dapat dilihat Algoritma dapat memprediksi prediksi naik atau turunnya saham tersebut setiap minggu dengan akurasi 90%. Dari evaluasi *Confusion Matrix* tersebut, dapat dilihat Model *Gated Recurrent* unit hanya melakukan kesalahan prediksi sebanyak 5 kali. Pelatihan data dari tanggal 01-01-2022 sampai 31-12-2022 ini menghasilkan Nilai MSE sebesar 102.43 dan nilai RMSE sebesar 10.120770. Hasil MSE dan RMSE tersebut tergolong cukup rendah mengingat objektif yang diprediksi adalah data harga saham. Data harga saham mempunyai variansi yang tinggi dikarenakan harga saham dipengaruhi oleh berbagai faktor seperti ekonomi, berita, politik dan lain-lainnya. Dan juga jangka waktu yang digunakan adalah setiap minggu dalam satu tahun dimana banyak peristiwa yang bisa terjadi. Grafik hasil prediksi saham dapat dilihat pada gambar 4.9. Garis berwarna biru adalah harga Aktual saham di dan garis berwarna merah adalah hasil prediksi

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan, maka dapat disimpulkan bahwa:

1. Pemilihan *Epoch* yang tepat dalam pelatihan model mempengaruhi kinerja akhir, dengan peningkatan MSE pada percobaan *epoch* yang lebih lanjut
2. Kelemahan dari program *website* aplikasi prediksi harga saham dengan pendekatan GRU ini terletak pada saat membuat harga prediksi ketika rata rata sentiment score pada minggu tersebut negative. Hasil aktual dan prediksi akan mempunyai jarak sekitar 230-350 rupiah.
3. Model GRU mampu memprediksi pergerakan harga saham dengan tingkat akurasi 90%, dan evaluasi *Confusion Matrix* mengungkapkan hanya terjadi 5 kesalahan prediksi.
4. Evaluasi statistik model, seperti *Mean squared error* (MSE) sebesar 102.43 dan *Root mean squared error* (RMSE) sebesar 10.120770, menunjukkan kinerja yang memuaskan.

5.2 Saran

Penelitian yang dilakukan memiliki kekurangan yang dapat ditingkatkan, oleh karena itu pengembangan penelitian yang dapat dilakukan ke depannya yaitu:

1. Meningkatkan jumlah dan kualitas data pelatihan. Semakin banyak dan beragam data pelatihan yang digunakan, maka model akan semakin mampu memahami pola sentimen publik dan meresponsnya dengan lebih akurat.
2. Tuning parameter model. Penyesuaian parameter model, seperti ukuran batch, jumlah epoch, atau unit dalam lapisan GRU, dapat membantu menemukan konfigurasi teroptimal yang meningkatkan kinerja model secara keseluruhan.
3. Evaluasi dengan metrik tambahan. Penggunaan metrik tambahan, seperti Mean absolute percentage error (MAPE), dapat memberikan perspektif tambahan

terkait tingkat akurasi dan relevansi model dalam meramalkan pergerakan harga saham.

4. Monitoring data eksternal. Integrasi data eksternal, terutama yang berkaitan dengan peristiwa ekonomi atau berita industri, dapat memperkaya pemahaman model terhadap konteks yang memengaruhi sentimen pasar.
5. Validasi dengan data real-time. Uji coba model dengan data harga saham real-time dapat memberikan gambaran langsung tentang sejauh mana model dapat beradaptasi dengan perubahan pasar secara cepat dan efektif.

DAFTAR PUSTAKA

- Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12(16), 4102–4107. <https://doi.org/https://doi.org/10.3923/jeasci.2017.4102.4107>
- Berry, M. W., & Kogan, J. (2010). *Text Mining* (First edit). John Wiley & Sons, Ltd.
- Bhuriya, D., Kaushal, G., Sharma, A., & Singh, U. (2017). Stock market predication using a linear regression. *Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017*, 510–513. <https://doi.org/https://doi.org/10.1109/ICECA.2017.8212716>
- Campan, A., Atnafu, T., Truta, T. M., & Nolan, J. (2018). Is Data Collection through Twitter Streaming API Useful for Academic Research? *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, 3638–3643. <https://doi.org/10.1109/BigData.2018.8621898>
- Chan, S., & Franklin, J. (2011). A text-based decision support system for financial sequence prediction. *Decision Support Systems*, 52(1), 189–198. <https://doi.org/http://dx.doi.org/10.1016/j.dss.2011.07.003>
- Cho, K., Merrienboer, B. van, Gulcehre, C., & Bougares, F. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *ArXiv*. <https://doi.org/http://dx.doi.org/10.3115/v1/D14-1179>
- Ghiassi, M., & Lee, S. (2018). A domain transferable lexicon set for Twitter sentiment analysis using a supervised machine learning approach. *Expert Systems with Applications*, 106, 197–216. <https://doi.org/https://doi.org/10.1016/j.eswa.2018.04.006>
- Ghosh, A., Bose, S., Maji, G., Debnath, N. C., & Sen, S. (2019). Stock price prediction using lstm on indian share market. *EPiC Series in Computing*, 63, 101–110. <https://doi.org/https://doi.org/10.29007/qgcz>
- Gu, X., Zhang, H., Zhang, D., & Kim, S. (2014). DeepAM : Migrate APIs with Multi-modal Sequence to Sequence Learning. *IJCAI International Joint Conference on Artificial Intelligence*, 3675–3681.
- Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, 4(1), 1–30. <https://doi.org/https://doi.org/10.3390/ASI4010009>
- Hutto, C. J., & Gilbert, E. (2014). VADER : A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media*, 8(1), 216–225. <https://doi.org/https://doi.org/10.1609/icwsm.v8i1.14550>
- Izzah, A., Sari, Y. A., Widyastuti, R., & Cinderatama, T. A. (2017). Mobile app for

- stock prediction using Improved Multiple Linear Regression. *Proceedings - 2017 International Conference on Sustainable Information Engineering and Technology, SIET* 2017, 150–154. <https://doi.org/https://doi.org/10.1109/SIET.2017.8304126>
- Jahan, I., & Sajal, S. (2018). *Stock Price Prediction using Recurrent Neural Network (RNN) Algorithm on Time-Series Data*. June.
- Joosery, B., & Deepa, G. (2019). Comparative analysis of time-series forecasting algorithms for stock price prediction. *ACM International Conference Proceeding Series*. <https://doi.org/https://doi.org/10.1145/3373477.3373699>
- Karim, M., & Das, S. (2018). Sentiment Analysis on Textual Reviews. *IOP Conference Series: Materials Science and Engineering*. <https://doi.org/10.1088/1757-899X/396/1/012020>
- Khedr, A. E., Salama, S. E., & Yaseen, N. (2017). Predicting stock market behavior using data mining technique and news sentiment analysis. *International Journal of Intelligent Systems and Applications*, 9(7), 22–30. <https://doi.org/https://doi.org/10.5815/ijisa.2017.07.03>
- Kumar, Y. J., Goh, O. S., Basiron, H., Choon, N. H., & Suppiah, P. C. (2016). A review on automatic text summarization approaches. *Journal of Computer Science*, 12(4), 178–190. <https://doi.org/https://doi.org/10.3844/jcssp.2016.178.190>
- Lawrence, A., Ryans, J. P., Sun, E., & Laptev, N. (2017). Earnings Announcement Promotions: A Yahoo Finance Field Experiment. *SSRN Electronic Journal*. <https://doi.org/https://dx.doi.org/10.2139/ssrn.2940223>
- M, H., E.Ab, G., Menon, V. K., & P, S. K. (2018). NSE Stock Market Prediction Using Deep-Learning Models. *Procedia Computer Science*, 132, 1351–1362. <https://doi.org/10.1016/j.procs.2018.05.050>
- Mathur, R., Pathak, V., & Bandil, D. (2019). Emerging Trends in Expert Applications and Security. *Proceedings of ICETEAS 2018*, 841. <https://doi.org/https://doi.org/10.1007/978-981-13-2285-3>
- Nafan, M. Z., & Amalia, A. E. (2019). Kecenderungan Tanggapan Masyarakat terhadap Ekonomi Indonesia berbasis Lexicon Based Sentiment Analysis. *Jurnal Media Informatika Budidarma*, 3(4), 268. <https://doi.org/https://doi.org/10.30865/mib.v3i4.1283>
- Nassirtoussi, A. K., Aghabozorgi, S., Wah, T. Y., & Ngo, D. (2015). Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment. *Expert Systems with Applications*, 42(1), 306–324. <https://doi.org/http://dx.doi.org/10.1016/j.eswa.2014.08.004>
- Nasution, Y. S. J. (2015). Peranan Pasar Modal Dalam Perekonomian Negara. *HUMAN FALAH: Jurnal Ekonomi Dan Bisnis Islam*, 2(1), 95–112. <https://doi.org/http://dx.doi.org/10.30829/hf.v2i1.180>

- Nisar, T. M., & Yeung, M. (2018). Twitter as a Tool for Forecasting Stock Market Movements: A Short-window Event Study. *The Journal of Finance and Data Science*, 4(2). <https://doi.org/http://dx.doi.org/10.1016/j.jfds.2017.11.002>
- Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2), 1–135. https://www.researchgate.net/publication/215470760_Opinion_Mining_and_Sentiment_Analysis
- Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications*, 42(4), 2162–2172. <https://doi.org/http://doi.org/10.1016/j.eswa.2014.10.031>
- Pavlopoulos, J., & Androutsopoulos, I. (2014). Aspect Term Extraction for Sentiment Analysis: New Datasets, New Evaluation Measures and an Improved Unsupervised Method. *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*. <https://doi.org/http://dx.doi.org/10.3115/v1/W14-1306>
- Putri, L. P. (2015). PENGARUH PROFITABILITAS TERHADAP HARGA SAHAM PADA PERUSAHAAN PERTAMBANGAN BATU BARA DI INDONESIA. *Jurnal Ilmiah Manajemen Dan Bisnis*, 16(2), 1–239. <https://doi.org/https://doi.org/10.30596/jimb.v16i2.955>
- Reshma, U., B, B. G. H., Kale, M., Mankame, P., & Kulkarni, G. (2016). Deep Learning for Digital Text Analytics : Sentiment Analysis. *Arnekt Solutions*, 1–8.
- Sharef, N. M., Zin, H. M., & Nadali, S. (2016). Overview and Future Opportunities of Sentiment Analysis Approaches for Big Data. *Journal of Computer Science*, 12(3), 153–168. <https://doi.org/http://dx.doi.org/10.3844/jcssp.2016.153.168>
- Shi, Y. (2022). *Sentiment Analysis*. *Advances in Big Data Analytics*. https://doi.org/http://dx.doi.org/10.1007/978-981-16-3607-3_7
- Sul, H. K., Dennis, A. R., & Yuan, L. (2016). Trading on Twitter: Using Social Media Sentiment to Predict Stock Returns: Trading on Twitter. *Decision Sciences*, 48(3). <https://doi.org/http://dx.doi.org/10.1111/dec.12229>
- Suyanto, Ramadhani, K. N., & Satria Mandala. (2019). *DEEP LEARNING MODERNISASI MACHINE LEARNING UNTUK BIG DATA*. Informatika Bandung.
- Trupthi, M., Pabboju, S., & Narasimha, G. (2017). Sentiment analysis on twitter using streaming API. *Proceedings - 7th IEEE International Advanced Computing Conference, IACC 2017*, 915–919. <https://doi.org/https://doi.org/10.1109/IACC.2017.0186>
- Usmani, S., & Shamsi, J. A. (2021). News sensitive stock market prediction: Literature review and suggestions. *PeerJ Computer Science*, 7, 1–36. <https://doi.org/10.7717/PEERJ-CS.490>
- Van Essen, B., Kim, H., Pearce, R., Boakye, K., & Chen, B. (2015). LBANN: Livermore big artificial neural network HPC toolkit. *Proceedings of MLHPC*

2015: *Machine Learning in High-Performance Computing Environments - Held in Conjunction with SC 2015: The International Conference for High Performance Computing, Networking, Storage and Analysis*.
<https://doi.org/10.1145/2834892.2834897>

- Wardhani, E. D., Areka, S. K., Nugroho, A. W., Zakaria, A. R., Prakasa, A. D., & Nooraeni, R. (2020). Sentiment Analysis Using Twitter Data Regarding BPJS Cost Increase and Its Effect on Health Sector Stock Prices. *Indonesian Journal of Artificial Intelligence and Data Mining*, 3(1), 1–8.
<https://doi.org/10.24014/ijaidm.v3i1.8245>
- Westergaard, D., Stærfeldt, H., Christian, T., Jensen, L. J., & Brunak, S. (2018). A comprehensive and quantitative comparison of text-mining in 15 million full- text articles versus their corresponding abstracts. *PLoS Computational Biology*, 14(2), 1–16. <https://doi.org/https://doi.org/10.1371/journal.pcbi.1005962>
- Yang, D., Kleissl, J., Gueymard, C. A., Pedro, H. T. C., & Coimbra, C. F. M. (2018). History and trends in solar irradiance and PV power forecasting: A preliminary assessment and review using text mining. *Solar Energy*, 168, 60–101.
<https://doi.org/https://doi.org/10.1016/j.solener.2017.11.023>