IMPLEMENTASI METODE VADER-LSTM DALAM PENGUJIAN PENGARUH SENTIMEN INVESTOR TERHADAP PREDIKSI HARGA SAHAM

Diajukan untuk Memenuhi Persyaratan Memperoleh

Gelar Sarjana Komputer

Disusun oleh

Ravi Edho Nugraha

11190910000038

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH

JAKARTA

2023 M/1444 H

IMPLEMENTASI METODE VADER-LSTM DALAM PENGUJIAN PENGARUH SENTIMEN INVESTOR TERHADAP PREDIKSI HARGA SAHAM

Diajukan untuk Memenuhi Persyaratan Memperoleh

Gelar Sarjana Komputer

Disusun oleh

Ravi Edho Nugraha

11190910000038

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH

JAKARTA

2023 M/1444 H

PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa:

1. Skripsi ini merupakan hasil karya asli saya yang diajukan untuk memenuhi salah satu persyaratan memperoleh gelar Strata 1 di UIN Syarif Hidayatullah Jakarta.
2. Semua sumber yang saya gunakan dalam penulisan ini telah saya cantumkan sesuai dengan ketentuan yang berlaku di UIN Syarif Hidayatullah Jakarta.
3. Apabila di kemudian hari terbukti karya ini bukan hasil karya saya sendiri atau merupakan hasil jiplakan karya orang lain, maka saya bersedia menerima sanksi yang berlaku di UIN Syarif Hidayatullah Jakarta.

Jakarta, 20 Juli 2023

Ravi Edho Nugraha

NIM. 111909100000038

LEMBAR PERSETUJUAN PUBLIKASI SKRIPSI

Sebagai civitas akademika UIN Syarif Hidayatullah Jakarta. Saya yang bertanda tangan di bawah ini:

Nama	: Ravi Edho Nugraha
NIM	: 11190910000038
Program Studi	: Teknik Informatika
Fakultas	: Sains dan Teknologi
Jenis Karya	: Skripsi

Demi pembuatan ilmu pengetahuan saya menyetujui untuk memberikan kepada UIN Syarif Hidayatullah Jakarta Hak Bebas Royalti Non Eksklusif (*Non Exclusive Royalty Free Right*) atas karya ilmiah yang berjudul:

IMPLEMENTASI METODE VADER-LSTM DALAM PENGUJIAN PENGARUH SENTIMEN INVESTOR TERHADAP PREDIKSI HARGA SAHAM

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non Eksklusif ini UIN Syarif Hidayatullah Jakarta berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data, merawat dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/ pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Tangerang, 6 September 2023

Ravi Edho Nugraha

ABSTRAK

Nama : Ravi Edho Nugraha

Program Studi : Teknik Informatika

Judul : Implementasi Metode VADER-LSTM dalam Pengujian Pengaruh Sentimen Investor terhadap Prediksi Harga Saham

Pasar saham merupakan pasar uang yang volatile dan mudah berubah seiring berjalannya waktu. Sentimen dan emosi pelaku pengguna pasar saham dikenal dapat memengaruhi harga saham sebenarnya. Perusahaan Tesla, Inc. merupakan perusahaan yang saat ini tengah naik daun dan hangat dibicarakan oleh pengguna media sosial Twitter. Oleh karena itu, penulis berniat membangun sistem pendeteksi harga saham perusahaan Tesla, Inc dan mencari hubungan antara sentimen investor dengan harga saham Tesla. Pada penelitian ini, dibangun sebuah sistem prediksi harga saham dengan bantuan sentimen analisis. Algoritma VADER digunakan untuk menganalisis serta melabelkan sentimen pengguna twitter, sedangkan algoritma deep learning LSTM digunakan untuk memprediksi harga saham dengan bantuan dataset sentimen investor perusahaan. Model terbaik yang dihasilkan penelitian ini membuktikan bahwa dengan menambahkan sentimen pengguna dalam model, dapat meningkatkan akurasi prediksi harga saham, dengan nilai metrik MAE sebesar 15.75, MSE sebesar 387.15, RMSE sebesar 19.54, dan nilai MAPE sebesar 5.48%.

Kata kunci : Saham, Analisis Sentimen, VADER, *Deep Learning*, LSTM, Prediksi Harga Saham

Jumlah pustaka : 64 (tahun 1943-2023)

ABSTRACT

Name : Ravi Edho Nugraha

Study Program : Teknik Informatika

Title : *Implementation of VADER-LSTM Method for Testing the Influence of Investor Sentiment on Stock Price Predictions*

The stock market is a volatile money market and can easily change over time. It is known that the sentiments and emotions of stock market users can influence actual stock prices. Tesla Company, Inc. is a company that is currently on the rise and is hotly discussed by Twitter social media users. Therefore, the author intends to build a stock price detection system for the company Tesla, Inc and look for the relationship between investor sentiment and Tesla's stock price. In this research, a stock price prediction system was built with the help of sentiment analysis. The VADER algorithm is used to analyze and label Twitter user sentiment, while the LSTM deep learning algorithm is used to predict stock prices with the help of company investor sentiment datasets. The best model produced by this research proves that by adding user sentiment to the model, it can increase the accuracy of stock price predictions, with an MAE metric value of 15.75, MSE of 387.15, RMSE of 19.54, and a MAPE value of 5.48%.

Kata kunci : Stocks, Sentiment Analysis, VADER, Deep Learning, LSTM, Stock Price Prediction

Jumlah pustaka : 64 (in 1943-2023)

KATA PENGANTAR

Puji dan syukur senantiasa penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat, hidayah, serta karunia-Nya sehingga penulis dapat menyelesaikan penelitian berjudul "IMPLEMENTASI METODE VADER-LSTM DALAM PENGUJIAN PENGARUH SENTIMEN INVESTOR TERHADAP PREDIKSI HARGA SAHAM". Penyusunan penelitian skripsi ini dilakukan sebagai syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) pada Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.

Dalam penyusunan skripsi ini penulis menyadari bahwa banyak pihak yang telah membantu penyelesaian penelitian ini, oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih banyak kepada:

1. Allah Subhanahu Wa Ta'ala yang telah memberikan karunia dan nikmat-Nya sehingga penulis diberikan kemudahan, kekuatan, kesehatan serta kelancaran dalam proses penyusunan skripsi ini.
2. Bapak Husni Teja Sukmana, S.T., M.Sc, Ph.D. selaku Dekan Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.
3. Ibu Dewi Khairani, M.Sc. selaku Ketua Program Studi Teknik Informatika dan Bapak Saepul Aripiyanto, M.Kom selaku Sekretaris Program Studi Teknik Informatika.
4. Kedua orang tua penulis yang selalu memberikan banyak motivasi, arahan, serta doa dan dukungan penuh kepada penulis untuk menyelesaikan skripsi.

5. Ibu Khodijah Hulliyah, M.Si., Ph.D serta Ibu Siti Ummi Masruroh, M.Sc sebagai dosen pembimbing skripsi penulis yang selalu menuangkan waktu dan tenaga untuk memberikan bimbingan, arahan, saran dan motivasi kepada penulis dalam penyusunan skripsi ini.
6. Teman-teman terdekat saya dari *Discord Server Among Us* yakni Firdan, Afif, Daffa, Hugo, Bintang, Fariz, dan Ical yang selalu ada di setiap saat untuk memberikan ide, pendapat, masukan serta motivasi dalam pengerjaan skripsi ini.
7. Seluruh teman-teman jurusan Teknik Informatika angkatan 2019 UIN Syarif Hidayatullah Jakarta.
8. Seluruh pihak yang ikut terlibat yang tidak dapat disebutkan satu persatu yang telah membantu penyelesaian skripsi ini.

Penulis menyadari skripsi ini tidak luput dari berbagai kekurangan. Penulis mengharapkan saran dan kritik demi kesempurnaan dan perbaikannya sehingga akhirnya skripsi ini dapat memberikan manfaat bagi bidang pendidikan, terutama di bidang sentimen analisis dan deep learning, serta dapat diterapkan di lapangan dan dikembangkan lagi lebih lanjut.

Tangerang, 21 Juli 2023

Ravi Edho Nugraha

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Salah satu hal yang sering diperdagangkan di pasar modal adalah saham atau stock. Saham dapat diartikan sebagai salah satu tanda keikutsertaan dan kepemilikan seseorang dalam suatu perusahaan. Saham berbentuk secarik kertas atas nama perseorangan atau kelompok yang memegang perusahaan yang mengeluarkan surat berharga tersebut. Porsi kepemilikan saham juga ditentukan oleh sebagaimana besar investasi yang telah ditaruh pada perusahaan tersebut.

Bagi perusahaan, investasi yang didapatkan melalui perdagangan saham dapat berguna menjadi dana yang mendukung berkembangnya perusahaan. Dan dengan banyaknya perusahaan berkembang di suatu negara, pasar saham akan memberi dampak positif bagi perekonomian negara tersebut. Namun, di sisi lain pasar saham merupakan pasar yang sangat volatile, di mana pergerakan harga saham dapat berubah drastis setiap saat. Sehingga banyak yang mengatakan bahwa dengan yang berkecimpung dalam pasar saham atau stock market mengambil risiko yang besar, dan dapat menghasilkan return yang besar pula (high risk, high rewards). Sangat banyak faktor yang memengaruhi pergerakan harga pasar saham. Salah satu faktor di antaranya adalah faktor investor.

Beberapa studi menunjukkan bahwa mood dan emosi investor berpengaruh dengan harga penutupan suatu saham. Seperti yang ditemukan

oleh (Cohen-Charash et al., 2013), suasana hati investor yang baik dan menyenangkan dapat meningkatkan harga pembukaan saham di NASDAQ pada hari berikutnya, dan sebaliknya suasana yang buruk cenderung membuat harga pembukaan mengalami penurunan. Studi lain yang dilakukan oleh (Tarczyński et al., 2021) menemukan bahwa perubahan mood investor yang disebabkan faktor cuaca dapat memengaruhi harga saham perusahaan energi di pasar saham Warsaw.

Investor juga biasanya menggunakan sosial media untuk berkomunikasi antara satu sama lain, atau hanya untuk sekedar menuangkan pemikirannya kepada khalayak banyak. Dengan ini semua orang dapat "mengakses" mood pengguna tersebut. Hal ini bisa diambil keuntungannya terlebih lagi apabila banyak pengguna yang melakukan hal yang sama seperti berbondong-bondong membeli saham yang sedang naik harga.

Keseluruhan hal ini dapat kita otomatisasikan dengan menggunakan sentiment analysis. Sentiment analysis adalah bagian dari natural language processing yang memproses teks dan menentukan atau mengklasifikasikan opini atau sentiment berdasarkan teks yang ditulis. Aplikasi sentiment analysis dalam pembelian saham yakni dengan memprediksi mood dari pengguna sosial media dan pada akhirnya dapat merekomendasikan saham apa atau kapan untuk dibeli.

Terdapat banyak metode machine learning yang dapat digunakan untuk implementasi sentiment analysis. Penelitian yang dilakukan oleh (Jing et al., 2021) membangun implementasi sentiment analysis untuk prediksi harga

saham menggunakan metode konvensional seperti *Support Vector Regression, Convolutional Neural Networks,* dan *Long Short Term Memory* (LSTM), serta juga digunakan metode hybrid seperti *Genetic Algorithm-Support Vector Regression*, *Genetic Algorithm-Convolutional Neural Networks*, dan CNN-LSTM. Dari studi yang dilakukan didapatkan metode LSTM dan hybrid CNN-LSTM mendapatkan tingkat akurasi tertinggi dengan nilai *Mean Absolute Percentage Error* (MAPE) masing-masing 0.0675 dan 0.0449.

Pada studi lain yang dilakukan oleh (Gondaliya et al., 2021) yang mengimplementasikan berbagai metode machine learning konvensional dalam sentiment analysis seperti *Decision Tree, K-Nearest Neighbor, Logistic Regression, Naïve Bayes, Random Forest, Support vector machine,* mendapatkan hasil bahwa metode *Support Vector Machine* memiliki tingkat akurasi tertinggi, yakni 78%.

Metode sentiment analysis menggunakan LSTM dan VADER merupakan dua metode dari dua cabang kecerdasan buatan (deep learning dan machine learning konvensional) yang memiliki tingkat akurasi yang tinggi (Ribeiro et al., 2016). LSTM memiliki keunggulan dimana sangat cocok untuk digunakan dataset berupa *time series*, sedangkan VADER mampu mengklasifikasi data yang belum diberi label dengan keakuratan tinggi. Oleh karena itu, penulis akan menggunakan kombinasi kedua metode diatas untuk menjalankan model prediksi di mana metode VADER digunakan untuk

sentiment analysis sosial media, dan LSTM digunakan untuk menganalisis riwayat harga saham dengan hasil analisis sentimen.

Elon Musk, yang dikenal karena kekayaannya yang luar biasa, menjadi orang terkaya di dunia pada tahun 2020. Elon Musk mendirikan x.com pada tahun 1999, yang kemudian menjadi eBay, Space X pada tahun 2002, dan Tesla pada tahun 2003. Terlepas dari semua penghargaan Elon Musk, usahanya yang paling terkenal adalah kesuksesannya dengan Tesla. Tesla, yang terutama memproduksi mobil listrik dan solusi penyimpanan energi, adalah merek mobil dengan pertumbuhan tercepat di dunia. Musk menjadi ketua perusahaan Tesla Inc. pada tahun 2004 setelah menginvestasikan lebih dari $30 juta di perusahaan tersebut (Seedhouse, 2013). Sejak saat itu, Tesla dianggap sebagai korporasi dengan potensi tak terbatas dalam perkembangan teknologi, dengan sahamnya yang meroket di bawah kepemimpinan Elon Musk.

Prestasi Elon Musk memang mencengangkan, namun ia mendapat kecaman karena seringnya menggunakan Twitter, yang dianggap tidak sopan dan tidak profesional. Namun, penelitian baru menunjukkan bahwa jika digunakan secara efektif, Twitter dapat menjadi alat pemasaran yang efektif. Dengan munculnya Twitter sebagai alat media sosial yang populer dalam beberapa tahun terakhir, korelasi antara tweet Twitter dan return saham menjadi jelas. (Pyeong Kang Kim et al., 2021) menerbitkan penelitian pada tahun 2019 yang menemukan korelasi antara return saham dan tweet, meskipun ia tidak dapat menemukan hubungan apa pun antara jumlah tweet

dan volume perdagangan saham, yang menunjukkan bahwa tweet dapat memengaruhi harga saham. Selain itu, ScienceDirect, sebuah jurnal sains terkemuka, menyajikan beberapa penelitian yang mencapai hasil serupa, yang menyiratkan bahwa sentimen baik dalam tweet dan liputan Twitter pasti akan meningkatkan nilai saham (Nisar & Yeung, 2018) (Teti et al., 2019).

Dari masalah yang dijelaskan di atas, penulis berniat untuk melakukan penelitian mengenai implementasi metode VADER dan *Long Short Term Memory* untuk melakukan prediksi nilai harga terhadap saham Tesla Inc. menggunakan bantuan sentiment analysis investor pada Twitter.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan di atas, dapat diidentifikasikan beberapa masalah, yakni:

1. Implementasi prediksi harga saham dengan bantuan analisis sentimen masyarakat di media sosial masih belum lengkap dan optimal.
2. Masih belum jelas apakah sentimen masyarakat mengenai pasar saham berpengaruh terhadap perubahan harga saham suatu perusahaan.

1.3 Rumusan Masalah

Dari masalah-masalah yang telah diidentifikasikan di atas, dapat dituliskan beberapa rumusan masalah untuk penelitian ini, antara lain:

1. Bagaimana implementasi terbaik untuk model prediksi harga saham dengan bantuan analisis sentimen media sosial menggunakan metode VADER-LSTM?

2. Bagaimana akurasi prediksi harga saham apabila mengimplementasikan analisis sentimen media sosial menggunakan metode VADER-LSTM?
3. Bagaimana pengaruh penggunaan data sentimen investor terhadap performa model prediksi harga saham menggunakan metode VADER-LSTM?

1.4 Batasan Masalah

Untuk membatasi ruang lingkup penelitian dalam penyelesaian masalah di atas, dirumuskan beberapa batasan-batasan, yaitu:

1.4.1 Metode

1. Algoritma VADER digunakan untuk menganalisis sentimen dataset tweet yang belum diberi label
2. Model LSTM digunakan untuk memprediksi harga saham berdasarkan dataset saham dan juga sentimen tweet
3. Dilakukan fine-tuning hyperparameter untuk mencari konfigurasi terbaik untuk melatih dataset.

1.4.2 Tools

1. Menggunakan bahasa pemrograman Python
2. Menggunakan beberapa library seperti *numpy, tensorflow, matplotlib, nltk, sklearn,* dan lain-lainnya.
3. Menggunakan Visual Studio Code dengan ekstensi Jupyter Notebook untuk dijalankan di local.

1.4.3 Proses

1. Ruang lingkup penelitian ini membahas prediksi harga saham dengan bantuan sentiment analysis masyarakat dari media sosial.
2. Dua dataset yang digunakan diperoleh dari Kaggle berupa data saham dan data tweet yang telah dicrawl dalam jangka waktu 30-09-2021 hingga 30-09-2022.
3. Praproses data tweet menggunakan *library* nltk, dan praproses data saham menggunakan library sklearn.
4. Analisis serta pelabelan sentimen pada dataset tweet menggunakan algoritma VADER.
5. Penggabungan kedua dataset, kemudian membagi data menjadi data latih dan data uji.
6. Membuat model LSTM untuk prediksi harga saham.
7. Fine-tuning hyperparameter model.
8. Melatih dan mengevaluasikan model yang telah dibuat.

1.5 Tujuan Penelitian

Tujuan dilaksanakannya penelitian ini yakni:

1. Mengimplementasikan model optimal untuk prediksi harga saham dengan bantuan analisis sentimen investor menggunakan metode VADER-LSTM.
2. Mengetahui tingkat ketepatan prediksi harga saham apabila diimplementasikan analisis sentimen dari sosial media dengan metode VADER-LSTM.

3. Mengetahui pengaruh sentimen investor melalui tweet terhadap performa model prediksi harga saham menggunakan metode VADER-LSTM.

1.6 Manfaat Penelitian

1.6.1 Bagi Penulis

1. Penulis dapat mengaplikasikan ilmu-ilmu akademis yang didapat selama perkuliahan ke dalam penelitian ini.
2. Penulis dapat memahami bagaimana membuat sebuah penelitian yang baik serta menambah ilmu pengetahuan terutama di bidang analisis sentimen.
3. Sebagai salah satu syarat dalam penyelesaian gelar Strata Satu (S1) program studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Syarif Hidayatullah Jakarta.

1.6.2 Bagi Universitas

1. Menambah kumpulan skripsi dari salah satu mahasiswa Teknik Informatika Fakultas Sains dan Teknologi mengenai *sentiment analysis* dengan metode VADER-LSTM.
2. Menjadi sebuah tolok ukur bagi universitas dalam menentukan keberhasilan dan kemampuan penulis dalam mengimplementasikan ilmu yang sudah didapatkan selama menempuh pendidikan perkuliahan di universitas.

1.6.3 Bagi Masyarakat

Menjadi salah satu pedoman atau *recommendation system* bagi penggunanya untuk pembelian saham yang tepat.

1.7 Metodologi Penelitian

Pada penelitian berjudul "Implementasi Metode VADER-LSTM dalam Pengujian Pengaruh Sentimen Investor Terhadap Prediksi Harga Saham" ini penulis melakukan pengumpulan data-data dan bahan materi yang diperlukan dengan metode sebagai berikut:

1.7.1 Metode Pengumpulan Data

Pada penelitian ini, penulis menggunakan metode pengumpulan data yang dilakukan dengan studi pustaka dan studi literatur, yakni dengan mempelajari buku-buku serta jurnal sebagai referensi terkait dengan topik bahasan penelitian.

1.7.2 Metode Implementasi

Di penelitian ini, penulis menggunakan metode penelitian yang merujuk oleh penelitian (Dewantoro, 2018), yakni terdiri atas:

1. Pengumpulan data tweet dan stock quotes
2. *Pre-processing* data
3. *Feature Selection* dan *Construction*
4. Prediksi data dengan model VADER-LSTM
5. Evaluasi hasil prediksi

1.8 Sistematika Penulisan

Sistematika penelitian yang dilakukan pada penelitian ini terdiri atas beberapa bagian, di antara lain:

BAB I PENDAHULUAN

Bab ini membahas hal umum dalam penelitian, seperti latar belakang dari dari sebuah permasalahan yang diangkat, tujuan penelitian, manfaat penelitian, rumusan masalah, batasan masalah, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini menjelaskan beberapa materi dan teori yang dibutuhkan dalam melaksanakan penelitian ini.

BAB III METODE PENELITIAN

Bab ini menjelaskan metode yang dipakai untuk mendapatkan data dan metode untuk pengembangan sistem yang telah dibuat serta kerangka berpikir pembuatan tugas akhir ini.

BAB IV IMPLEMENTASI

Bab ini menjelaskan proses implementasi dari metode yang digunakan untuk menyelesaikan permasalahan penelitian.

BAB V HASIL DAN PEMBAHASAN

Bab ini membahas hasil yang telah didapat dari proses implementasi dan eksperimen pada bab sebelumnya.

BAB VI PENUTUP

Bab ini menjelaskan kesimpulan dari hasil yang telah didapat dan menjawab semua pokok permasalahan yang dirancang serta saran-saran yang digunakan untuk penelitian lebih lanjut.

BAB II

LANDASAN TEORI

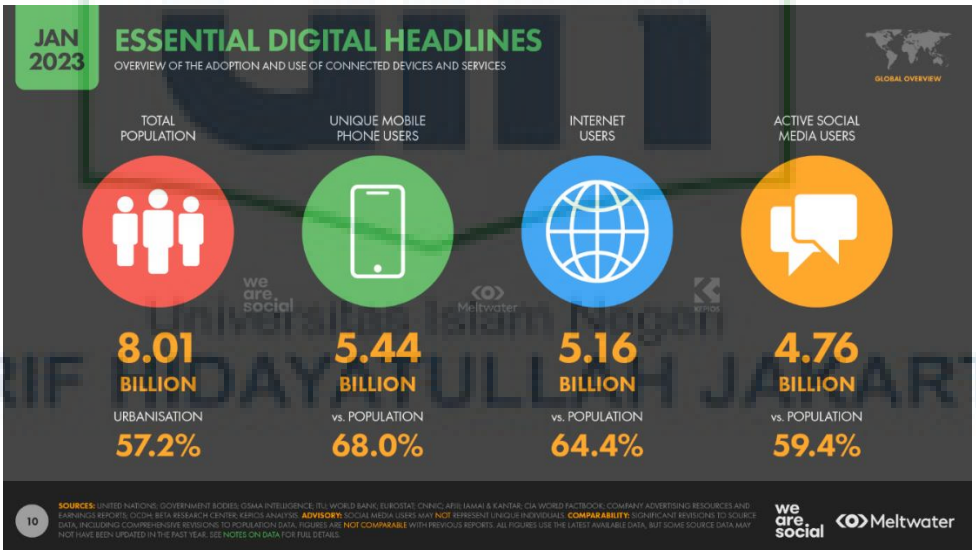
2.1 Media Sosial

Media sosial didefinisikan secara formal oleh (Carr & Hayes, 2015) sebagai saluran komunikasi massal berbasis internet, yang tidak terlatih, dan serta berjalan terus-menerus yang berfungsi memfasilitasi interaksi di antara pengguna, sehingga memperoleh nilai/value dari konten yang dibuat pengguna. Dalam kata lain, media sosial adalah kumpulan teknologi yang beragam dan berkembang pesat yang menciptakan ruang komunal online di mana sekelompok orang dapat berinteraksi, berdiskusi, berkoordinasi, atau bekerja sama. Struktur sosial dan jaringan komunitas online ini beragam seperti struktur sosial manusia dan dapat berupa apa saja mulai dari longgar, terbuka, dan oportunistik hingga tertutup, rapat, dan tertutup (Coiera, 2013).

Definisi media sosial menurut KBBI adalah laman atau aplikasi yang memungkinkan pengguna dapat membuat dan berbagi isi atau terlibat dalam jaringan sosial. Media sosial memiliki beragam bentuk, seperti aplikasi pesan online, situs *microblogging*, web berbagi konten video, hingga blog pribadi. Semua jenis sosial media memiliki peran masing-masing untuk mengekspresikan penggunanya. Oleh karena itu, media sosial berbeda dari media penyiaran tradisional karena secara langsung mendukung atau membuat jejaring sosial menggunakan teknologi informasi dan komunikasi. Beberapa situs media sosial yang cukup terkenal dan familier seperti Facebook, Twitter, YouTube, TikTok, Instagram.

Terdapat banyak sekali potensi-potensi yang dapat diambil dari media sosial, seperti menjalankan proyek kolaboratif melalui *crowdsourcing*, mengekspresikan diri sendiri menggunakan blog pribadi, hingga komunitas berbasis konten seperti video, *artwork*, musik, serta kemunculan dunia virtual yang berjalan di dunia maya seperti *video game hubs* dan dunia sosial virtual (Kaplan & Haenlein, 2010).

Pada saat ini media sosial merupakan bagian yang tak dapat dihindari dalam kehidupan sehari-hari. Penetrasi media sosial di dunia juga semakin tinggi dengan berkembangnya teknologi dan meningkatnya penggunaan barang-barang elektronik. Pada Januari 2023 sebuah artikel yang melaporkan penggunaan media sosial di dunia oleh kolaborasi *Meltwater* dengan *We Are Social*.



Gambar 2.1 Data pengguna media sosial di dunia (Kemp, 2023a)

Dari Gambar 2.1, dilaporkan bahwa dari sekitar 8 milyar penduduk di Bumi, sekitar 5.16 milyar orang atau lebih dari 64% penduduk pernah

menggunakan internet, dengan hampir 60% penduduk atau sekitar 4.76 milyar orang merupakan pengguna media sosial aktif (Kemp, 2023a).

2.2 Twitter

Twitter merupakan layanan microblogging di mana penggunanya dapat mengirim dan menerima pesan dalam bentuk *tweet*. Sebuah *tweet* dapat berupa teks, gambar, ataupun video, namun untuk *tweet* berupa teks memiliki batas maksimum sepanjang 140 karakter (Kwak et al., 2010). *Tweet* dari suatu pengguna dapat diterima oleh pengguna lain apabila mereka mengikuti pengguna yang meng-*tweet* tersebut, dan sebaliknya kita hanya dapat melihat *tweet* dari pengguna yang diikuti. Pada November 2017 Twitter mengumumkan untuk menambahkan limit teks setiap *tweet* menjadi 280 karakter (Gligorić et al., 2018).

Berbeda dengan portal media sosial kebanyakan seperti Facebook atau MySpace, hubungan mengikuti dan diikuti oleh pengguna lain tidak memerlukan persetujuan maupun timbal balik. Pengguna siapapun dapat mengikuti pengguna lainnya tanpa perlu diterima maupun harus diikuti balik (kecuali untuk pengguna yang menggunakan akun dengan mode privat). Menjadi pengikut di Twitter berarti pengguna tersebut menerima semua pesan dari yang pengguna yang diikuti. Penggunaan Twitter yang umum saat ini telah berkembang menjadi budaya berpesanan teks yang mencolok, seperti:

1. *Retweet* yang sering disingkat RT, yakni membagikan *tweet* dari pengguna lain kepada pengikut.

2. *Mention* atau menyebutkan akun pengguna lainnya dengan menggunakan karakter "@" diikuti nama penggunanya.
3. *Hashtag* atau tagar, yang berguna sebagai penanda kata kunci sebuah *tweet* dengan menggunakan karakter "#" diikuti kata kunci.
4. *Trends* yakni topik maupun *hashtag* yang sedang hangat dibicarakan oleh pengguna Twitter dalam suatu waktu.

Twitter merupakan salah satu media sosial yang sangat populer. Berdasarkan data dari artikel yang diterbitkan oleh *Kepios* melalui situs web *Data Reportal*, pengguna aktif Twitter pada bulan April 2023 mencapai 372.9 juta orang berdasarkan tools iklan Twitter (Kemp, 2023b).

2.3 Analisis Sentimen

Analisis sentimen adalah pengaplikasian dari *natural language processing* yang berfokus pada pengidentifikasian ekspresi yang mencerminkan sikap berbasis opini penulis terhadap suatu entitas atau aspek lainnya (Cambria et al., 2017). Sedangkan berdasarkan (Devika et al., 2016) Analisis sentimen adalah proses untuk mendeteksi polaritas kontekstual dari teks. Ini menentukan apakah teks yang diberikan bersifat positif, negatif atau netral. Analisis sentimen juga sering disebut sebagai opinion mining, karena analisis sentimen pada dasarnya mencari dan mengungkapkan opini atau sikap teks atau pembicara. Oleh karena itu, analisis sentimen memiliki peran penting dalam *natural language processing* yakni bidang studi ilmu komputer dan kecerdasan buatan yang berkaitan dengan interaksi bahasa manusia-komputer.

Analisis sentimen pada dasarnya adalah cabang dari ilmu komputer, *natural language processing*, dan text mining yang berfungsi untuk mengklasifikasikan sentimen atau opini dari suatu teks ke dalam suatu kelas atau label positif dan negatif.

2.4 Teknik Analisis Sentimen

Ada beberapa cara dan teknik yang dapat diimplementasikan untuk membuat model analisis sentimen. Secara umum, terdapat dua pendekatan untuk menganalisis sentimen sebuah dokumen:

2.4.1 Pendekatan Lexicon

Analisis sentimen berbasis leksikon adalah metode analisis sentimen yang dilakukan dengan menggunakan kata dan frasa opini tanpa pengetahuan sebelumnya. Kata-kata yang mengandung opini disusun dan dikumpulkan. Kata-kata positif dan negatif digabungkan dengan kata-kata opini yang secara kolektif disebut leksikon. Pendekatan berbasis leksikon menggunakan lexicon dan data yang tidak berlabel. Kata-kata dalam teks dievaluasi berdasarkan opini leksikon untuk menentukan orientasinya dan selanjutnya sentimen teks (Qi & Shabrina, 2023).

Contoh pendekatan lexicon dalam analisis sentimen yakni metode VADER, Text-Blob, dan sebagainya. Keuntungan dari metode ini yakni implementasinya tidak memerlukan data yang berlabel, serta analisis yang lebih cepat daripada pendekatan lain.

2.4.2 Pendekatan Machine Learning

Pendekatan sentimen analisis ini didasarkan dengan membangun pengklasifikasi (*classifier*) dari contoh tulisan tekstual berlabel. Rata-rata metode machine learning yang digunakan berbasis *supervised learning*, sehingga memerlukan data yang berlabel, kemudian dibangun sebuah sistem yang dapat mengklasifikasi kelas dari data lain berdasarkan kategori yang telah ditetapkan (negatif atau positif) (Ribeiro et al., 2016).

Contoh implementasi sentimen analisis dengan pendekatan machine learning yakni menggunakan model Naive Bayes, Linear classifier, Support Vector Machine, dan lain sebagainya. Kelebihan dari pendekatan machine learning yakni nilai akurasi klasifikasi lebih baik ketimbang metode yang lain.

2.4.3 Komparasi Teknik Analisis Sentimen

Berikut rangkuman perbandingan beberapa teknik sentimen analisis yang cukup terkenal (Verma & Thakur, 2018).

Tabel 2.1 Perbandingan Teknik Analisis Sentimen

Teknik	Contoh model	Kelebihan	Kekurangan
Pendekatan Leksikon	VADER, Text-Blob	• Tidak memerlukan data yang dilabeli • Cakupan *term* dan frasa beropini yang luas • Biaya komputasi yang lebih kecil	• Memerlukan kamus kosakata yang cukup besar • Berpotensi tidak mencakup frasa dan kata-kata

		daripada metode machine learning	gaul yang terbaru.
Pendekatan machine-learning	KNN, Linear Classifier, SVM	• Tingkat akurasi yang relatif lebih tinggi • Hasil training model dapat digunakan berulang kali	• Model yang didapatkan hanya cocok digunakan pada jenis dataset yang serupa dengan yang dilatih • Pengumpulan dataset berlabel dapat menghambat jalannya proses

Oleh karena alasan di atas, penulis akan menggunakan pendekatan leksikon dengan metode VADER untuk menganalisis sentimen pengguna Twitter terkait dengan saham Tesla.

2.5 Pasar Saham

KBBI mendefinisikan pasar modal sebagai: 1) seluruh kegiatan yang mempertemukan penawaran dan permintaan dana jangka panjang; 2) pusat keuangan, bank, dan firma yang meminjamkan uang secara besar-besaran; dan 3) pusat keuangan, bank, dan firma yang meminjamkan uang secara besar-besaran. Pasar saham merupakan kumpulan pasar dan pertukaran di mana aktivitas reguler pembelian, penjualan, dan penerbitan saham perusahaan publik berlangsung. Menurut (Sulia, 2017) pasar modal (*capital market*) adalah pasar yang memfasilitasi penerbitan dan perdagangan surat

berharga keuangan seperti saham dan obligasi. Pasar modal mempunyai dua fungsi utama, yakni sebagai sarana pendanaan usaha bagi perusahaan dan sebagai sarana berinvestasi bagi pemilik modal (investor). Perkembangan ekonomi suatu negara dapat diukur dengan berbagai cara, yaitu dengan mengetahui tingkat perkembangan pasar modal dan perkembangan berbagai jenis industri pada negara tersebut.

Perkembangan harga saham di pasar modal merupakan suatu indikator penting untuk mempelajari tingkah laku pasar yaitu investor. Investor akan mendasarkan keputusan investasinya pada informasi – informasi yang dimilikinya termasuk informasi keuangan perusahaan. Informasi keuangan yang digunakan untuk menganalisis harga saham antara lain price to book value, price earning ratio, dan pertumbuhan aset.

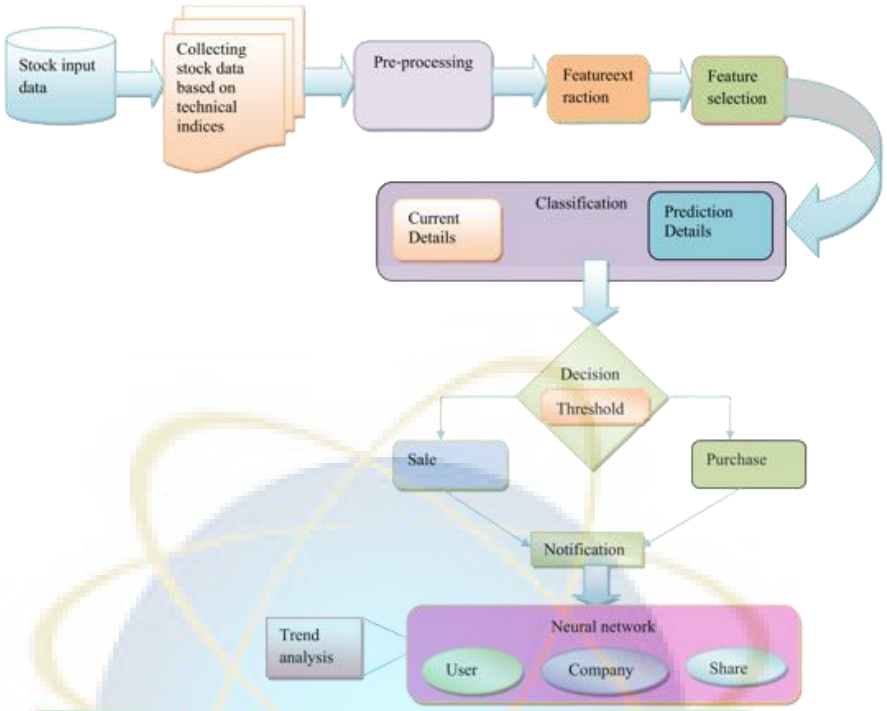
2.6 Prediksi Harga Saham

Pasar saham telah menumbuhkan daya tarik dari investor karena teknologi yang muncul, di mana peramalan dapat menghasilkan prediksi pasar yang sukses. Prediksi tren saham bergantung pada investasi dan perdagangan data saham. Alat yang digunakan untuk prediksi pasar saham dapat memantau, memprediksi, serta mengatur pasar, yang dapat digunakan untuk mengambil keputusan yang tepat. Pasar saham harus berurusan dengan beberapa data saham industri, yang mencakup seluruh pasar keuangan. Berdasarkan kondisi pasar, tindakan yang dilakukan oleh investor akan selalu beradaptasi baik untuk membeli maupun menjual saham. Beberapa faktor yang mempengaruhi kondisi pasar adalah perkiraan pendapatan dari

perusahaan, berita laba yang diterbitkan, pengumuman dividen, perubahan manajemen, dan sebagainya. Penelitian dalam isu-isu perdagangan saham ujungnya mengarah pada prediksi dari beberapa faktor yang dianggap oleh para ahli dapat mempengaruhi harga saham (Gandhmal & Kumar, 2019).

Prediksi atau prakiraan harga saham sangat penting dalam menarik investor baru serta mempertahankan investor yang sudah ada. Dengan meningkatnya akurasi alat prediksi harga saham akan membantu para investor dalam mengambil keputusan yang lebih tepat. Sudah lama upaya untuk membangun sistem prediksi saham dilakukan. Upaya awal dimulai dengan analisis fundamental dan analisis teknis. Di mana analisis fundamental menilai harga saham berdasarkan nilai intrinsiknya yaitu nilai wajar, sedangkan analisis teknikal merupakan analisis berbasis grafik dan tren. Setelah itu, model linier diperkenalkan sebagai solusi untuk prediksi pasar saham, yang meliputi *autoregressive integrated moving average* (ARIMA) dan *generalized autoregressive conditional heteroskedasticity* (GARCH). Dengan berkembangnya model *machine learning*, diterapkan juga model untuk prediksi pasar saham, misalnya *logistic regression* dan *support vector machine* (Jiang, 2021).

Berikut merupakan gambar blok diagram dari alur proses sistem prediksi harga saham modern.



Gambar 2.2 Blok diagram alur fungsi sistem prediksi saham (Gandhmal & Kumar, 2019)

Pada awalnya, data riwayat saham dikumpulkan dari berbagai dataset, yang kemudian di lakukan pre-proses untuk menghilangkan *noise*, kemudian dilakukan seleksi fitur signifikan dari dataset yang berpotensi dapat membantu prediksi, Kemudian, fitur yang dipilih dianalisis untuk mengekstraksi pengetahuan menggunakan sejumlah besar data. Selanjutnya penganalisis data menampilkan analisis yang lebih baik dalam antarmuka yang mudah digunakan (Gandhmal & Kumar, 2019).

2.7 Machine Learning

Berdasarkan (Bell, 2014) *machine learning* didefinisikan sebagai cabang dari kecerdasan buatan (*artificial intelligence*). Dengan menggunakan komputasi, dapat merancang sebuah sistem yang dapat "belajar" dari data

dengan cara dilatih. Sistem tersebut dapat mempelajari dan serta meningkatkan pengalaman serta pemahaman dari data, dan seiring dengan waktu, akan menyempurnakan model yang dapat digunakan untuk memprediksi hasil pertanyaan berdasarkan pembelajaran sebelumnya. Algoritma *machine learning* merupakan proses komputasi yang menggunakan data input untuk mencapai tugas yang diinginkan tanpa harus diprogram secara langsung (*hard-coded*) untuk menghasilkan hasil tertentu. Algoritma ini dalam artian "*soft-coded*" yang secara otomatis mengubah atau mengadaptasi arsitektur melalui iterasi sehingga mereka menjadi lebih baik dalam mencapai tugas yang diinginkan (El Naqa & Murphy, 2015). Algoritme pembelajaran mesin membangun model berdasarkan data sampel, yang disebut data pelatihan, untuk membuat prediksi atau keputusan tanpa diprogram secara eksplisit untuk melakukannya (Koza et al., 1996).

2.7.1 Tipe Algoritma Machine Learning

Menurut (Zhang, 2010), algoritma-algoritma *machine learning* dikelompokkan berdasarkan hasil yang diharapkan dari modelnya. Tipe algoritma *machine learning* yang umum adalah:

1. *Supervised learning*

   Supervised learning merupakan algoritma yang menciptakan fungsi yang memetakan *input* kepada *output* yang diharapkan. Dengan kata lain, *supervised learning* merupakan algoritma yang proses pembelajarannya mencakup klasifikasi. Pada model *supervised learning*, data input yang digunakan telah

dipetakan/dilabelkan sehingga tujuan dari model ini untuk mencari hasil klasifikasi berdasarkan dataset yang telah dilabeli.

2. *Unsupervised learning*

   Pada *unsupervised learning* dataset yang digunakan tidak diberikan label dan model yang dibuat berfungsi mengidentifikasikan pola antara data tanpa bantuan pengguna.

3. *Semi-supervised learning*

   Model *semi-supervised learning* mengkombinasikan dataset yang telah dilabeli dan tidak dilabeli untuk menghasilkan algoritma yang dapat mengklasifikasikan/melabelkan data yang sebelumnya tidak dilabeli.

4. *Reinforcement learning*

   Reinforcement learning merupakan model di mana algoritma mempelajari bagaimana cara bertindak terhadap lingkungan yang diamati. Setiap tindakan yang diambil algoritma akan berdampak pada lingkungan, baik positif maupun negatif. Sistem timbal balik ini yang akan memandu pembelajaran algoritma.

2.7.2 Komponen Machine Learning

Berdasarkan (Vasilev et al., 2019), terdapat beberapa komponen yang penting dalam membangun sebuah sistem *machine learning*:

1. *Learner* atau pembelajar

   Learner merupakan algoritma yang digunakan dengan dasar filosofi pembelajaran. Pilihan algoritma ini ditentukan oleh

masalah yang akan dipecahkan, karena masalah yang berbeda mungkin lebih tepat digunakan algoritma yang berbeda pula.

2. *Training data* atau data untuk dilatih

   Training data adalah kumpulan data mentah yang akan dipelajari. Data-data tersebut dapat berlabel maupun tak berlabel. Memiliki jumlah sampel data yang cukup sangat penting bagi pembelajar untuk memahami struktur masalah.

3. *Representation* atau representasi data

   Representasi data merupakan cara menggambarkan data berdasarkan fitur yang dipilih, yang selanjutnya akan digunakan untuk proses pembelajaran. Misalnya, untuk mengklasifikasikan gambar angka tulisan tangan, maka gambar tersebut dapat digambarkan sebagai *array* berisi nilai, di mana setiap sel berisi nilai warna satu piksel. Pilihan representasi data yang baik penting untuk mencapai hasil yang lebih baik.

4. *Goal* atau tujuan

   Tujaun merupakan alasan untuk belajar dari data untuk menyelesaikan masalah yang dihadapi. Tujuan sangat terkait dengan sasaran, dan membantu menentukan bagaimana dan apa yang harus digunakan pembelajar dan representasi data apa yang patut digunakan. Misalnya, tujuan suatu sistem yakni membersihkan kotak email dari email yang tidak diinginkan, dan

dari tujuan ini ditentukan apa target pembelajarnya. Dalam hal ini, targetnya adalah deteksi email spam.

5. *Target* atau sasaran

   Target atau sasaran mewakili apa yang sedang dipelajari serta hasil akhir yang diinginkan. Target dapat berupa klasifikasi data tidak berlabel, representasi data input berdasarkan pola atau karakteristik tersembunyi, simulator prediksi masa depan, atau respons terhadap timbal balik dari luar (dalam *reinforcement learning*).

2.8 Deep Learning

Deep Learning merupakan bagian dari keluarga besar metode *machine learning*, yang pembelajarannya menggunakan rangkaian jaringan syaraf tiruan (*artificial neural network*) dan pembelajaran representasi. Sama dengan metode *machine learning* klasik, *deep learning* dapat dilakukan secara *supervised*, *semi-supervised*, ataupun *non-supervised* (LeCun et al., 2015). *Deep learning* adalah pendekatan yang muncul dan telah diterapkan secara luas dalam domain *machine learning* tradisional. Ada tiga alasan penting utama maraknya *deep learning* saat ini: kemampuan pemrosesan chip yang meningkat secara dramatis (misalnya unit GPU), biaya perangkat keras komputasi yang secara signifikan lebih rendah, dan kemajuan yang cukup besar dalam algoritma *machine learning* umum (Guo et al., 2016).

Deep learning diimplementasikan dengan menggunakan sistem jaringan syaraf tiruan (*artificial neural network*), dengan beberapa arsitektur yang

cukup terkenal seperti *deep neural networks*, *deep belief networks*, *deep reinforcement learning*, *recurrent neural networks*, *convolutional neural networks* and *transformers*. *Deep learning* telah diterapkan pada bidang ilmu pengetahuan yang cukup luas, mulai dari visi komputer, pengenalan suara, pemrosesan bahasa alami, terjemahan mesin, bioinformatika, desain obat-obatan, analisis citra medis, ilmu sains iklim, pemeriksaan material, hingga program permainan papan, di mana implementasinya mampu menghasilkan hasil yang setara bahkan mengungguli kinerja manusia (Ciresan et al., 2012).

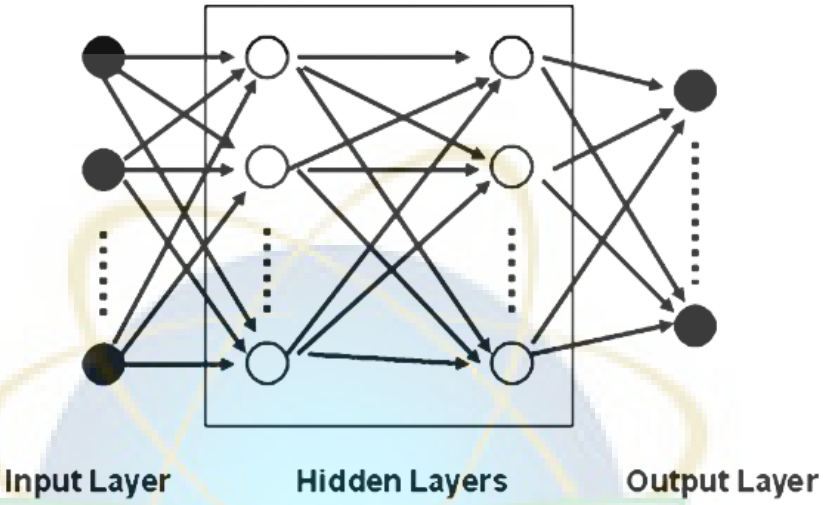
Komponen sistem yang membangun *deep learning* berupa:

a. **Neural Network**

Neural network atau jaringan saraf, juga dikenal sebagai jaringan saraf buatan (*artificial neural network*/ANN) merupakan model komputasi yang terinspirasi dari struktur dan fungsi otak manusia. Jaringan ini terdiri dari neuron buatan yang saling terhubung, atau simpul, yang bekerja bersama untuk memproses dan menganalisis informasi. Jaringan saraf dirancang untuk mengenali pola, belajar dari data, dan membuat prediksi atau keputusan berdasarkan pembelajaran tersebut. Menurut (Russell & Norvig, 2022) jaringan saraf adalah sistem komputasi yang terdiri dari beberapa elemen pemrosesan sederhana yang sangat terhubung, yang memproses informasi melalui respons dinamis mereka terhadap masukan eksternal.

Jaringan syaraf tiruan pada awalnya muncul pada tahun 1943 oleh Warren McCulloch dan Walter Pitts, yang memperkenalkan model

matematika dari neuron buatan, yang membentuk dasar pengembangan jaringan syaraf tiruan, serta menunjukkan bagaimana neuron yang saling terhubung dapat melakukan operasi logika (McCulloch & Pitts, 1943).



Gambar 2.3 Topologi Neural Network (Zou et al., 2008)

Gambar 2.3 menunjukkan arsitektur umum *neural network*. Node diatur ke dalam array linier, yang disebut *layer*. Dalam sistem jaringan syaraf tiruan umumnya terdapat *input layer*, *output layer*, dan *hidden layer*. Jaringan ini biasanya disebut jaringan syaraf *multi-layer*. Sedangkan terdapat sistem jaringan syaraf yang lebih sederhana di mana tidak terdapat *hidden* layer, yang biasanya disebut jarinagn syaraf *single-layer*. Merancang topologi jaringan syaraf melibatkan penentuan jumlah node pada setiap *layer*, jumlah *layer* dalam jaringan, dan jalur koneksi antar node (Zou et al., 2008).

Setiap node menerima banyak input dari yang lain melalui koneksi yang memiliki bobot terkait, yang berbanding lurus dengan kekuatan sinapsis (penghubung). Ketika jumlah input total melebihi nilai ambang

node, node tersebut menjadi aktif dan meneruskan sinyal melalui fungsi aktivasi dan mengirimkannya ke node tetangga. Proses ini dapat dinyatakan sebagai model matematika:

$$y = f\left(\sum_{i=0}^{n} w_i x_i - T\right) \quad (1)$$

dimana *y* adalah *output* dari node, *f* merupakan fungsi transfer, w_i adalah bobot dariinput x_i, dan *T* adalah nilai ambang batas. Fungsi aktivasi memiliki banyak bentuk. Fungsi aktivasi non linier lebih berguna daripada fungsi linier, karena hanya beberapa masalah yang dapat dipisahkan secara linier.

b. **Activation Function**

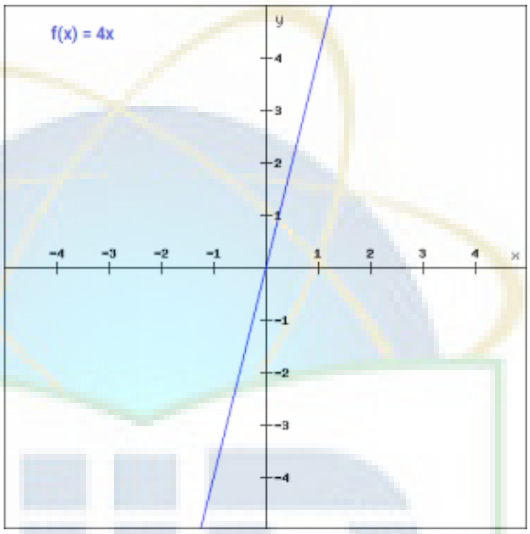
Activation functions atau fungsi aktivasi berfungsi mengatur aktivasi neuron dalam sistem jaringan syaraf tiruan. Fungsi ini diterapkan pada keluaran dari setiap neuron pada suatu *layer* jaringan saraf untuk menentukan apakah neuron tersebut akan "aktif" dan meneruskan keluarannya ke lapisan selanjutnya. Fungsi aktivasi berperan penting dalam jaringan syaraf tiruan karena menambahkan non-linearitas dalam sistem. (Sharma et al., 2017) Beberapa fungsi aktivasi yang sering digunakan yakni:

1. *Linear*

Pada fungsi aktivasi linear, keluaran fungsi berbanding lurus. Fungsi aktivasi linear dapat didefinisikan dengan:

$$f(x) = ax \quad (2)$$

di mana nilai a merupakan konstanta yang dapat dipilih pengguna. Fungsi aktivasi linear dapat digambarkan sebagai berikut.



Gambar 2.4 Fungsi Aktivasi Linear (Sharma et al., 2017)

2. *Sigmoid*

Fungsi aktivasi sigmoid merupakan fungsi aktivasi yang paling umum digunakan. Fungsi sigmoid akan selalu menghasilkan nilai di antara 0 dan 1. Fungsi sigmoid dapat didefinisikan sebagai:

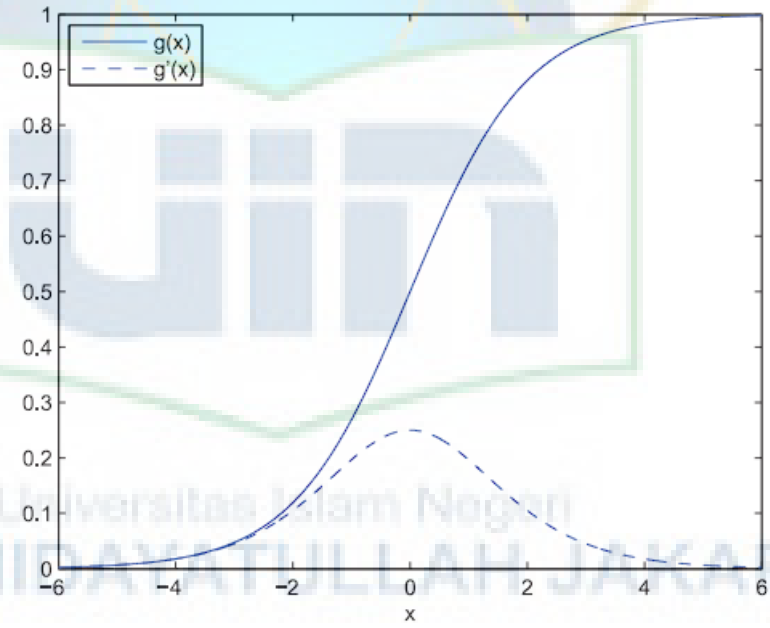
$$g(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

di mana $x \in (-\infty, \infty), g \in (0,1)$. Dalam implementasi pemodelan dan pelatihan sistem jaringan syaraf buatan *multi-layer*, terdapat forward propagation dan backward propagation. Di

backward propagation, derivatif dari fungsi aktivasi digunakan. Fungsi sigmoid adalah fungsi kontinu, yang artinya dapat dibedakan di mana-mana. Turunan fungsi sigmoid juga mudah dihitung sehingga fungsi sigmoid sering digunakan dalam sistem jaringan syaraf. (Ding et al., 2018) Turunan fungsi sigmoid didefinisikan dalam:

$$g(x) = \frac{e^{-x}}{(1+e^{-x})^2} \quad (4)$$

Fungsi sigmoid dapat digambarkan sebagai berikut.



Gambar 2.5 Fungsi Aktivasi Sigmoid (Ding et al., 2018)

3. *Softmax*

Fungsi *softmax* adalah kombinasi dari beberapa fungsi sigmoid. Seperti yang kita ketahui bahwa fungsi sigmoid

mengembalikan nilai dalam rentang 0 hingga 1, ini dapat diperlakukan sebagai probabilitas titik data kelas tertentu.

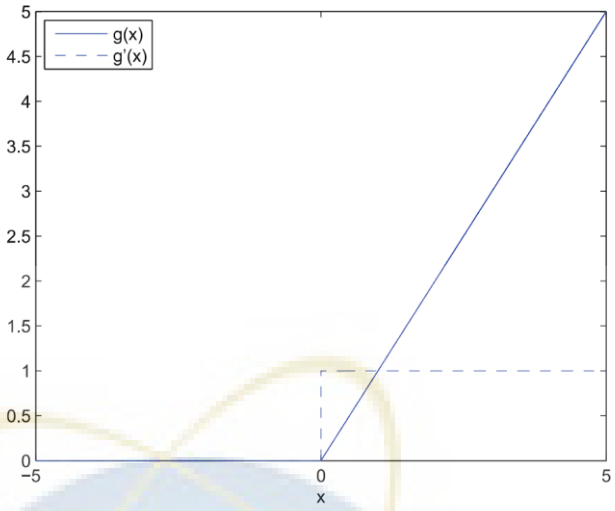
Fungsi softmax tidak seperti fungsi sigmoid yang digunakan untuk klasifikasi biner, dapat digunakan untuk masalah klasifikasi multikelas. Fungsi ini untuk setiap titik data dari semua kelas individual akan mengembalikan probabilitas. Fungsi softmax dapat dinyatakan sebagai:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \; untuk \; j = 1, \ldots \qquad (5)$$

Saat membangun jaringan atau model untuk *multi-class classification*, maka lapisan keluaran jaringan akan memiliki jumlah neuron yang sama dengan jumlah kelas di target (Sharma et al., 2017).

4. RELU

ReLU adalah singkatan dari *Rectified Linear Unit* dan merupakan fungsi aktivasi non-linier yang banyak digunakan dalam jaringan saraf. Keuntungan menggunakan fungsi ReLU adalah bahwa semua neuron tidak diaktifkan pada waktu yang bersamaan. Neuron akan dinonaktifkan hanya ketika keluaran transformasi linier adalah nol. (Sharma et al., 2017) Fungsi ReLU digambarkan dalam gambar berikut.



Gambar 2.6 Fungsi Aktivasi ReLU (Ding et al., 2018)

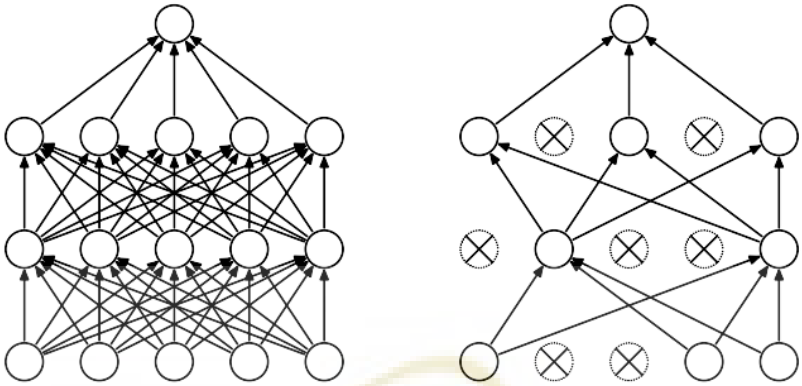
ReLU didefinisikan sebagai:

$$g(x) = \max(0, x) = \begin{cases} x \; jika \; x \geq 0 \\ 0 \; jika \; x < 0 \end{cases} \quad (6)$$

ReLU lebih efisien daripada fungsi lainnya karena semua neuron tidak diaktifkan pada waktu yang sama, melainkan sejumlah neuron diaktifkan pada satu waktu.

c. **DropOut**

DropOut merupakan salah satu teknik yang dapat digunakan untuk mencegah *overfitting* dalam jaringan syaraf tiruan. Nama DropOut mengacu pada neuron yang "di-drop" atau dihilangkan secara acak (baik yang tersembunyi maupun yang terlihat) selama proses pelatihan jaringan saraf tiruan (Srivastava et al., 2014), dengan cara menghilangkan neuron sementara, dan memutuskan semua sambungan dengan neuron lain. DropOut dapat dijelaskan dalam gambar berikut.



Gambar 2.7 DropOut (Srivastava et al., 2014)

d. **Loss Function**

Loss function adalah ukuran seberapa baik model jaringan saraf dalam melakukan tugas tertentu. Fungsi ini juga dikenal sebagai *error function* (Hastie et al., 2009). *Loss function* merupakan parameter yang ingin diminimalkan oleh jaringan saraf dengan memperbarui bobot (*weight*) di dalam jaringan menggunakan teknik yang disebut *stochastic gradient descent* (SGD). Semakin rendah nilai *loss function*, semakin baik model jaringan saraf.

e. **Backpropagation**

Backpropagation, kependekan dari "backward propagation of error," adalah algoritma untuk *supervised learning* dari jaringan syaraf tiruan yang menggunakan penurunan gradien. Dibeli suatu jaringan saraf tiruan dan *error function*, metode ini menghitung gradien *error function* sehubungan dengan bobot jaringan saraf. Berdasarkan (Cilimkovic, 2015) terdapat empat tahapan untuk melakukan *backpropagation*, yakni:

1. Perhitungan algoritma feed-forward

2. *Back-propagation* menuju lapisan *output*
3. *Back-propagation* menuju lapisan *hidden*
4. Perhitungan bobot (*weight*) ulang

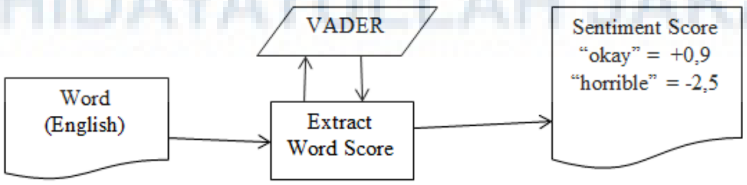
2.9 Algoritma VADER

Valence Aware Dictionary and Sentiment Reasoner (VADER) merupakan *library* kamus berbasis *lexicon* yang mampu menganalisis sentimen dari suatu teks tanpa memerlukan label di teks (Borg & Boldt, 2020). VADER diperkenalkan oleh C.J Hutto dan Eric Gilbert pada tahun 2014. Berbeda dengan pendekatan analisis sentimen *machine learning*, VADER merupakan alat analisis sentimen berbasis *lexicon* dan peraturan yang secara khusus dirancang untuk membaca sentimen di media sosial. VADER dibangun dengan dasar *human-centric*, dengan menggabungkan analisis kualitatif dengan validasi empiris serta investigasi eksperimental yang menggunakan kebijaksanaan orang banyak (Hutto & Gilbert, 2014).

Kamus *lexicon* adalah daftar leksikal yang mencakup kata-kata yang dapat digolongkan secara umum berdasarkan orientasi semantik mereka, yaitu positif, negatif, atau netral. Pendekatan berbasis leksikon adalah salah satu metode dalam analisis sentimen yang menggunakan kamus yang berisi daftar kata dengan konten opini (Nafan & Amalia, 2019). Setiap kata dalam kamus tersebut telah diberikan skor polaritas dari -1 (untuk kategori negatif) hingga +1 (untuk kategori positif). Pendekatan berbasis lexicon memungkinkan pelatihan model tanpa memerlukan data yang diberi label.

Metode VADER adalah salah satu metode analisis sentimen berbasis leksikon. Keunggulan menggunakan deteksi polaritas VADER yakni sudah tersedia kamus yang berisi nilai untuk setiap kata. Proses penentuan polaritas kalimat didapatkan dengan menggabungkan atribut "*compound*" dari setiap kata yang ada (Ghiassi & Lee, 2018). Perhitungan sentimen ini dibagi menjadi empat kelas, yaitu positif, negatif, netral, dan skor gabungan (*compound score*). Nilai positif, negatif, dan netral menggambarkan rasio proporsi valensi setiap teks, sehingga dapat menggambarkan seberapa porsi setiap sentimen yang terkandung dalam teks.

Keuntungan menggunakan VADER polarity detection adalah tersedianya kamus yang berisi nilai dari setiap kata. Hasil Preprocess text akan di nilai berdasarkan lexicon apakah itu positif, negatif atau neutral dan menambahkan skor total (*compound*). Beberapa perintah VADER yang menggunakan bahasa pemrograman phyton akan dikerjakan, dan VADER akan memanggil data lexicon dari server NLTK untuk menghitung polarity class sentimen. Berikut ini adalah flowchart proses penentuan polaritas dari suatu kalimat.



Gambar 2.8 Flowchart Penentuan Polarity Score

Penentuan polaritas kalimat didapatkan dari penyatuan attribute *compound*. Nilai *compound* adalah normalisasi jumlah nilai valensi dari

setiap kata yang muncul dalam teks, dengan rentang antara -1 (paling negatif) hingga +1 (paling positif). Jika nilai *compound* ≥ 0.5, maka sentimen tersebut dianggap positif dan diwakili oleh angka 1. Jika nilai *compound* > -0.5 dan < 0.5, maka sentimen tersebut dianggap netral. Jika nilai *compound* ≤ -0.5, maka sentimen tersebut dianggap negatif dan diwakili oleh angka -1 (Karim & Das, 2018).

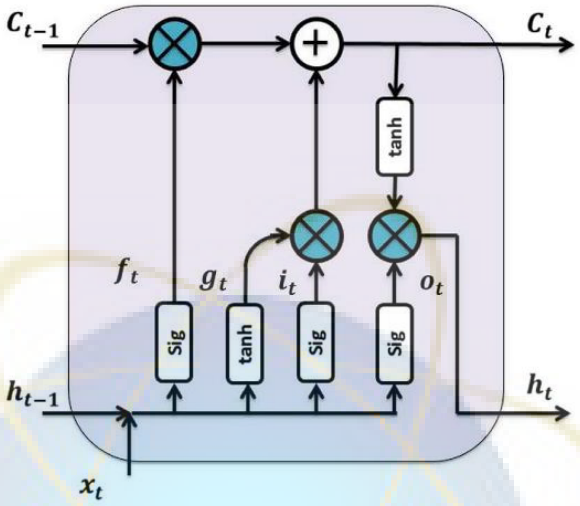
2.10 Algoritma Long-Short Term Memory

Long-Short Term Memory (LSTM) adalah sebuah sistem jaringan saraf tiruan (*artificial neural network*) yang digunakan di bidang kecerdasan buatan dan *deep learning*. Merupakan sejenis *recurrent neural network* (RNN), LSTM memiliki jaringan feedback dalam pembelajarannya, dan tak hanya dapat memproses titik data tunggal (seperti gambar), tetapi juga data berupa urutan (seperti ucapan atau video). Karakteristik ini menjadikan jaringan LSTM ideal untuk memproses dan memprediksi data berupa *time series* (Hochreiter & Schmidhuber, 1997).

Sebuah model LSTM pada umumnya terdiri dari sel (*cell*), gerbang masukan (*input gate*), gerbang keluaran (*output gate*), dan gerbang lupa (*forget gate*). Sel berfungsi untuk mengingat nilai dalam interval waktu tertentu, dan tiga gerbang berfungsi mengatur aliran informasi yang berkaitan dengan sel (Van Houdt et al., 2020).

LSTM dapat mempelajari pola panjang dari data berurut karena mencegah situasi vanishing gradient (Putra, 2020). Tetapi, LSTM tetap memiliki prinsip yang sama dengan RNN dan yang membedakan dengan

RNN yaitu isi cellnya. Recurrent Neural Network (RNN) sederhana karena dengan cell yang hanya berisi 1 layer neuron dengan fungsi aktivasi tanh.



Gambar 2.9 Arsitektur unit LSTM

Ide dasar dari LSTM adalah adanya jalur yang menghubungkan antara cell state (C_{t-1}) sebelumnya dengan cell state yang sekarang (C_t). Dengan jalur tersebut, suatu informasi pada cell state dapat dengan mudah diteruskan ke cell state berikutnya dengan beberapa modifikasi yang diperlukan. Nilai cell state merupakan vektor yang dirancang untuk menyimpan informasi tentang konteks dari suatu dari suatu sekuen data.

Langkah pertama dalam LSTM adalah menentukan informasi yang akan dibuang dari cell state (C_{t-1}) menggunakan fungsi sigmoid yang disebut sebagai forget gate (f_t). Nilai nol menandakan bahwa informasi akan dibuang sedangkan satu berarti informasi diteruskan. Formulasinya terdapat pada persamaan di bawah.

$$f_t = \sigma\left(w_f \cdot [s_{t-1}, x_t] + b_f\right) \quad (7)$$

Di mana f_t adalah forget gate, σ adalah fungsi sigmoid, w_f adalah nilai weight untuk forget gate, s_{t-1} adalah nilai output sebelum order ke t, x_t adalah nilai input pada order ke t, b_f adalah nilai bias pada forget gate.

Langkah berikutnya adalah menentukan informasi yang akan ditambahkan ke cell state (C_t). Langkah ini memproses hasil penggabungan dari S_{t-1}dan x_t menggunakan dua fungsi, yaitu fungsi sigmoid sebagai input gate dan fungsi tanh sebagai intermediate gate. Hasil dari kedua fungsi tersebut dikalikan untuk mendapatkan informasi yang akan ditambahkan pada cell state (C_t). Persamaan matematika dari kedua fungsi terdapat pada persamaan berikut.

$$i_t = \sigma(w_i \cdot [s_{t-1}, x_t] + b_i) \quad (8)$$

$$C_t = \tanh(w_c \cdot [s_{t-1}, x_t] + b_c) \quad (9)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \quad (10)$$

Dimana i_t adalah input gate, w_i adalah nilai weight untuk input gate, s_{t-1} adalah nilai output sebelum order ke t, x_t adalah nilai input pada order ke t, b_i adalah nilai bias pada input gate. C_t merupakan cell state, $tanh$ merupakan fungsi tanh, w_c merupakan nilai weight untuk cell state, b_c merupakan nilai bias untuk cell state, C_{t-1} adalah cell state sebelum order ke t.

Setelah itu, tambahkan dengan output dari forget gate pada langkah pertama. Langkah terakhir adalah menentukan output dari unit LSTM. Untuk menghasilkan output, perlu menghitung sigmoid dari gabungan s_{t-1} dan x_t

yang disebut sebagai output gate. Output gate ini menentukan seberapa besar nilai dari cell state akan dihasilkan pada s_t. kemudian hitung nilai fungsi tanh dari cell state (C_t) dan kalikan dengan nilai dari output gate. Hasil perkaliannya tersebut menjadi output dari unit LSTM yang terdapat pada persamaan di bawah.

$$o_t = \sigma(w_o \cdot [s_{t-1}, x_t] + b_o) \quad (11)$$

$$s_t = o_t \cdot \tanh(C_t) \quad (12)$$

Dimana o_t adalah output gate, σ adalah fungsi aktivasi sigmoid, w_o adalah nilai weight untuk output gate, s_{t-1} adalah nilai output sebelum order ke t. x_t adalah nilai input pada order ke t. b_o adalah nilai bias pada output gate, s_t adalah nilai output order t, o_t adalah output gate, $tanh$ adalah fungsi tanh C_t adalah cell state.

2.11 Metrik Performa

Mengevaluasikan performa dari model *machine learning* merupakan salah satu langkah penting dalam pembuatan model yang efektif. Untuk menilai kinerja dan kualitas dari model *machine learning*, dapat digunakan beberapa metode dan metrik yang berbeda. Metrik ini biasa disebut metrik performa atau *performance metrics*.

Pada sebuah dataset berupa *time series*, diperlukan metode pengujian dengan metrik yang berbeda dengan metrik konvensional, di karenakan data bersifat urutan, bukan satuan. Berikut beberapa metrik performa yang cocok digunakan untuk dataset *time series* (Adhikari & Agrawal, 2013).

2.11.1 *Mean Absolute Error*

Dalam statistika, *mean absolute error* (MAE) adalah ukuran kesalahan antara observasi berpasangan yang mengamati fenomena yang sama. Contoh pengaplikasian MAE termasuk perbandingan antara hasil diprediksi dan hasil diamati (hasil sebenarnya), perbandingan antara waktu seterusnya dengan waktu permulaan, ataupun perbandingan antara dua teknik pengukuran yang berbeda. MAE dihitung sebagai jumlah kesalahan absolut dibagi dengan ukuran sampel (Willmott & Matsuura, 2005). *Mean absolute error* dapat dihitung dengan rumus sebagai berikut:

$$MAE = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n} = \frac{\sum_{i=1}^{n}|e_i|}{n} \quad (13)$$

Dengan y_i adalah nilai sebenarnya, x_i yakni nilai yang diprediksi, dan n adalah jumlah data dalam dokumen tersebut.

2.11.2 *Mean Absolute Percentage Error*

Mean absolute percentage error atau juga dikenal sebagai *mean absolute percentage deviation* (MAPD), adalah ukuran akurasi prediksi dari metode prediksi dalam statistik. MAPE biasanya menyatakan akurasi sebagai rasio yang ditentukan oleh rumus:

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - x_i}{y_i}\right| \quad (14)$$

Dengan y_i adalah nilai sebenarnya, x_i yakni nilai yang diprediksi, dan n adalah jumlah data dalam dokumen tersebut.

2.11.3 *Mean Square Error*

Dalam statistika, *mean square error* (MSE) atau *mean square deviation* (MSD) dari suatu estimator (prosedur untuk memperkirakan kuantitas tak teramati) mengukur rata-rata kuadrat dari kesalahan—yaitu, perbedaan kuadrat rata-rata antara perkiraan nilai dan nilai sebenarnya. MSE adalah fungsi risiko, sesuai dengan nilai yang diharapkan dari *squared error loss* (Doksum & Bickel, 2015). Dalam *machine learning*, khususnya bidang *empirical risk minimization*, MSE mengacu pada risiko empiris (kerugian rata-rata pada subset yang diamati), sebagai perkiraan MSE yang sebenarnya (kerugian rata-rata pada populasi sebenarnya). *Mean squared error* dapat diukur dengan perhitungan:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - x_i)^2}{n} = \frac{\sum_{i=1}^{n} {e_i}^2}{n} \qquad (15)$$

Dengan y_i adalah nilai sebenarnya, x_i yakni nilai yang diprediksi, dan n adalah jumlah data dalam dokumen tersebut.

2.11.4 *Root Mean Square Error*

Root-mean-square error (RMSE) atau *Root mean square* deviation (RMSD) adalah ukuran yang sering digunakan untuk mengukur perbedaan antara nilai (nilai sampel atau populasi) yang diprediksi oleh model atau estimator dan nilai yang diamati (nilai sebenarnya). RMSE mewakili akar kuadrat dari kuadrat perbedaan antara nilai prediksi dan nilai yang diamati dibagi jumlah sampel.

Deviasi ini disebut *residual* apabila perhitungan dilakukan pada sampel data yang digunakan untuk estimasi, dan disebut *error* (atau kesalahan prediksi) ketika dihitung di luar sampel. RMSE berfungsi untuk mengumpulkan besaran kesalahan dalam prediksi untuk berbagai titik data menjadi satu ukuran kekuatan prediksi. RMSE adalah salah satu ukuran akurasi, untuk membandingkan kesalahan model prediksi yang berbeda untuk dataset tertentu, dan bukan antara kumpulan dataset, karena bergantung pada skala (Hyndman & Koehler, 2006). RMSE dapat dihitung dengan:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i)^2}{n}} \quad (16)$$

Dengan y_i adalah nilai sebenarnya, x_i yakni nilai yang diprediksi, dan n adalah jumlah data dalam dokumen tersebut.

2.12 Penelitian Sejenis

Penelitian sejenis merupakan penelitian terdahulu yang pernah dilakukan dengan topik dan bahasan yang relevan dengan penelitian yang akan dilakukan. Penelitian sejenis ini diambil dari berbagai media seperti jurnal, prosiding dan conference paper. Sumber-sumber ini akan digunakan sebagai landasan teori dan referensi untuk penyelesaian masalah dari penelitian yang dilakukan. Penelitian-penelitian tersebut dan metode yang akan diusulkan pada penelitian ini dapat dilihat pada tabel berikut.

Tabel 2.2 Penelitian sejenis

No	Judul	Tahun	Metode	Hasil
1	Stock closing price prediction based on sentiment analysis and LSTM (Jin et al., 2020)	2019	Long-Short Term Memory	Studi ini mengimplementasikan model LSTM novel untuk prediksi pasar saham yang ditargetkan untuk saham AAPL. Hasil penelitian menunjukkan bahwa model yang diusulkan mengungguli model yang dibandingkan secara konsisten dalam tiga aspek utama, yakni harga penutupan yang diprediksi lebih dekat, akurasi klasifikasi kenaikan dan penurunan yang lebih tinggi, dan offset waktu yang lebih rendah.
2	Stock Price Movement Prediction Using Technical Analysis and Sentiment Analysis (Sagala et al., 2020)	2020	Support Vector Machine, K-Nearest Neighbor, Naïve Bayes	Model technical analysis feature dan sentiment label yang dibuat menggunakan metode SVM dan dataset ASII memiliki akurasi tertinggi dari model lain yakni sebesar 57.5% dalam jangka waktu 5-hari kerja.

3	Analisis Sentimen Penerapan PSBB di DKI Jakarta dan Dampaknya terhadap Pergerakan IHSG (Lengkong et al., 2021)	2021	Regresi Logistik, K-Nearest Neighbor, Naïve Bayes, Random Forest	Model yang dibuat memperhitungkan sentimen dari masyarakat Indonesia dan mencari hubungannya dengan bagaimana kinerja pasar IHSG di saat diberlakukannya PSBB. Ini berpatokan bahwa disaat sentimen masyarakat buruk maka kinerja IHSG juga akan turun dan sebaliknya. Dari seluruh model yang digunakan, model klasifikasi yang memiliki akurasi tertinggi yakni regresi logistik, yang tidak jauh disusul oleh K-Nearest Neighbor dan Naive Bayes.
4	Prediksi Harga Saham Menggunakan BiLSTM dengan Faktor Sentimen Publik (Afrianto et al., 2022)	2022	Bidirectional Long-Short Term Memory	Dari model BiLSTM yang dibangun, digunakan empat skenario dataset yang akan dilatih di mana tiap dataset berbeda dalam jangka waktu data latihnya. Dari metriks Mean Square Error dan Root Mean Square Error, skenario 4 memberikan hasil prediksi paling baik dengan nilai eror masing-masing 0.094 dan 0.306. Namun berdasarkan metriks

				Direction statistics (Dstat), skenario 1 memiliki nilai terbesar yakni 68%.
5	A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction (Jing et al., 2021)	2021	Convolutional Neural Network, Long Short-Term Memory	Dari model sentiment analysis yang digunakan yakni CNN, dibandingkan dengan beberapa model sentiment analysis yakni logistic, SVM, RNN dan LSTM. Setelah melakukan validasi dengan 10-fold cross, dibuktikan bahwa metode CNN jauh lebih unggul ketimbang model lain dengan nilai rata-rata F-measure sebesar 0.8482. Dan dari model prediksi hybrid CNN dan LSTM, dihasilkan nilai MAPE sebesar 0.0449, yang lebih baik daripada nilai MAPE model LSTM saja.
6	BERT-Based Stock Market Sentiment Analysis (Lee et al., 2020)	2020	Bidirectional Encoder Representations from Transformers (BERT)	Model BERT membuktikan adanya korelasi sentimen masyarakat terutama di media sosial dengan naik turunnya harga saham. Model yang telah dibuat mencapat akurasi lebih dari 87.3% dari data training

				yang diberikan setelah enam kali iterasi.
7	Sentiment analysis and prediction of Indian stock market amid Covid-19 pandemic (Gondaliya et al., 2021)	2020	Decision Tree, K-Nearest Neighbor, Logistic Regression, Naïve Bayes, Random Forest, Support Vector Machine	Dari seluruh model klasifikasi yang dibuat, algoritma Decision Tree, Logistic Regression, Naïve Bayes, Random Forest dan Support vector machine memiliki nilai akurasi yang lebih tinggi pada metrik Bag-of-Word dengan Logistic Regression dan SVM memiliki akurasi tertinggi yakni 78%. Sedangkan untuk metric TF-IDF, algoritma KKN memiliki nilai akurasi yang lebih tinggi ketimbang metrik Bag-of-Word sebesar 68% dibandingkan 52%.
8	Stock Market Increase and Decrease using Twitter Sentiment Analysis and ARIMA Model (Kedar, 2021)	2021	Linear regression, (Autoregressive Integrated Moving Average) ARIMA	Model ARIMA merupakan algoritma yang dapat menganalisis dan meramalkan nilai sebuah set di masa depan berdasarkan nilai di masa lalu. Dari dataset yang diberikan, model ini menghasilkan nilai akurasi yang cukup baik yakni 90.137% dan nilai Mean

				Square Error sebesar 0.0065. Dan dari yang ada, sentimen yang diberikan oleh pengguna media sosial dapat meramalkan 'tren' dari harga pasar saham.
9	Sentiment Analysis Using Twitter Data Regarding BPJS Cost Increase and Its Effect on Health Sector Stock Prices (Wardhani et al., 2020)	2020	Regresi Logistik Biner	Berdasarkan model regresi logistik biner, apabila diaplikasikan untuk prediksi harga saham KAEF mendapatkan akurasi 69,6% dan untuk prediksi harga saham INAF akurasinya sebesar 73,9%. Dari permasalahan yang ada, memang benar ada korelasi yang cukup jelas di antara respon masyarakat mengenai jalannya kebijakan BPJS dengan perubahan nilai saham perusahaan-perusahaan kesehatan di Indonesia.
10	Stock movement prediction with sentiment analysis based on deep learning networks (Shi et al., 2021)	2020	Convoluted Neural Network (CNN), Recurrent Neural Network (RNN), Linear Regression	Klasifikasi sentimen menggunakan metode deep learning seperti CNN dan GRU (RNN) memiliki hasil akurasi polaritas terbaik. Namun hasil metode deep learning sebelumnya tidak jauh beda dengan metode machine learning biasa

				yakni Linear Regression. Dari metode hybrid yang menggabungkan riwayat pasar saham dengan informasi sentimen sebelumnya menghasilkan peningkatan sebesar 1.25%. Jangka waktu yang paling baik diambil yakni 5 hari kerja atau seminggu hari kerja.
11	Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine (Ren et al., 2019)	2019	Support Vector Machine	Penelitian ini mengintegrasikan analisis sentimen ke dalam metode machine learning berbasis support vector machine, dimana digunakan pertimbangan hari libur ke dalam model. Hasil penelitian menggambarkan bahwa prediksi pergerakan saham Indeks SSE50 bisa setinggi 89,93% dengan kenaikan 18,6% setelah memasukkan variabel sentimen.
12	Stock Market Prediction Using Microblogging Sentiment Analysis and	2022	TextBlob/VADER + K-Nearest Neighbors, Support Vector Machine, Logistic Regression,	Penelitian ini berhasil menggabungkan dua metode analisis sentimen leksikon dengan berbagai macam model machine learning. Hasil terbaik

	Machine Learning (Koukaras et al., 2022)		Naïve Bayes, Decision Tree, Random Forest and Multilayer Perceptron	didapatkan dengan menganalisis data menggunakan VADER dan SVM dengan nilai F-score 76.3%, dan AUC 67%.

Dari beberapa jurnal yang telah dipaparkan di tabel di atas terdapat beberapa penelitian yang telah membahas prediksi harga saham dengan analisis sentimen sebelumnya. Beberapa penelitian menggunakan metode machine learning konvensional, dan juga ada yang menggunakan metode hybrid seperti penelitian (Kedar, 2021) yang menggunakan metode hybrid *linear regression*-ARIMA, penelitian (Shi et al., 2021) mengkombinasikan model SVM-CNN serta *linear regression*-CNN. Ada juga penelitian oleh (Jing et al., 2021) yang menggunakan berbagai jenis metode hybrid seperti *genetic algorithm-support vector regression* dan hybrid CNN-LSTM.

Secara keseluruhan, metode hybrid yang digunakan pada jurnal-jurnal tersebut menghasilkan hasil prediksi yang memiliki tingkat akurasi yang lebih tinggi daripada metode *machine learning* konvensional. Namun, dikarenakan kompleksnya metode hybrid ini menyebabkan model yang diciptakan sulit diadaptasikan ke proyek yang lain. Model *machine learning* juga memerlukan data berlabel untuk mengimplementasikan model analisis sentimen. Usulan penulis dan kontribusi skripsi ini yakni mencari cara untuk memanfaatkan data tweet pengguna Twitter yang belum berlabel sentimennya untuk meningkatkan hasil prediksi saham. Caranya dengan menggabungkan model analisis sentimen berbasis leksikon dengan model machine learning.

Penelitian sebelumnya belum mengkombinasikan algoritma klasifikasi sentimen leksikon seperti *VADER*, dengan model prediksi berbasis *time series* LSTM. Oleh karena itu, penulis akan mengimplementasikan model VADER-LSTM sebagai metode kombinasi usulan untuk memprediksikan harga saham.

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Penelitian kali ini menggunakan metode penelitian kuantitatif. Sebagaimana yang telah dikemukakan (Sugiyono, 2008) mengenai metode kuantitatif merupakan "Metode penelitian yang berlandaskan pada filsafat positivisme, digunakan untuk meneliti pada populasi atau sampel tertentu, pengumpulan data menggunakan instrumen penelitian, analisis data bersifat kuantitatif/statistik, dengan tujuan untuk menguji hipotesis yang telah ditetapkan". Metode penelitian kuantitatif merupakan jenis metode penelitian yang bersifat sistematis, terencana dan terstruktur dengan dari awal hingga pembuatan desain penelitian.

3.2 Waktu dan Objek Penelitian

Objek penelitian dalam penelitian ini adalah prediksi harga saham dan analisis sentimen. Penelitian ini berfokus pada saham Tesla, Inc. (TSLA) yang merupakan perusahaan terdaftar di NASDAQ. Tesla juga terdaftar dalam NASDAQ 100 yakni indeks pasar saham yang terdiri dari 100 perusahaan non-finansial terbesar yang terdaftar di NASDAQ, sehingga peranan perusahaan ini cukup besar dalam pasar modal negara Amerika Serikat. Tesla adalah perusahaan otomotif yang saat ini berkembang pesat dengan meningkatnya popularitas mobil listrik. *Lineup* mobil listrik yang dikeluarkan Tesla juga cukup hangat dibincangkan di jaringan media sosial

seperti Twitter sehingga sentimen dari pengguna maupun investor dari peruasahaan Tesla cocok untuk dianalisis. Periode data yang dicakup dalam penelitian ini sepanjang satu tahun, yakni waktu yang cukup panjang untuk menganalisis pergerakan harga saham dan potensi pengaruh sentimen yang dikeluarkan pengguna/konsumen Tesla yang terkini/kontemporer, tanpa harus mengetahui riwayat Tesla sebelumnya.

3.3 Metode Pengumpulan Data

Pada penelitian ini penulis mengumpulkan data dan informasi mengenai objek penelitian yang dapat membantu jalannya penelitian. Adapun metode pengumpulan data yang dilakukan adalah:

3.3.1 Studi Pustaka dan Literatur

Dalam penelitian ini peneliti memanfaatkan studi pustaka dan literatur sebagai acuan dan juga referensi pada penelitian ini, teori-teori, metode, informasi dan data yang dijadikan sebagai data pendukung dan pembanding dalam penelitian yang sedang dilakukan. Selain itu penulis juga mengakses situs-situs pada internet untuk mempelajari konsep analisis sentimen, *text preprocessing*, algoritma VADER dan algoritma *deep learning* LSTM.

3.3.2 Data

Penelitian ini menggunakan dua data sekunder, yaitu data saham dari Yahoo Finance dan data sentimen dari Twitter. Kedua data tersebut diambil dari situs penyedia dataset gratis Kaggle. Dataset tersebut berisi *tweet* hasil *crawling* untuk 25 ticker saham paling banyak ditonton di

situs *Yahoo Finance* dari 30-09-2021 hingga 30-09-2022, selain itu ditambahkan harga pasar saham dan data volume untuk tanggal dan saham yang sesuai. Kedua dataset tersebut kemudian akan disaring kembali agar hanya menampilkan data yang terkait dengan saham Tesla, Inc.

3.4 Pre-processing Data

Preprocessing data adalah langkah-langkah yang digunakan untuk persiapan data dan transformasi data. Proses preprocessing berguna untuk output lebih baik dan efisien (Bhaya, 2017). Preprocessing data adalah langkah-langkah yang digunakan untuk persiapan data dan transformasi data yang berguna untuk output lebih baik dan efisien. Proses tahapan dalam preprocessing, antara lain:

3.4.1 Pre-processing Data Twitter

Pre-processing data Twitter dilakukan supaya data sesuai dengan yang dibutuhkan dalam penelitian ini. Pada tahap ini, dilakukan preprocessing data twitter meliputi:

a. Filtering data tweet

Sama halnya dengan dataset saham, dataset Twitter mengandung data tweet yang tidak berkaitan dengan perusahaan Tesla, Inc. Oleh karena itu diperlukan penyaringan kembali kepada dataset untuk menghapus data-data yang tidak diperlukan.

b. Data cleansing

Data cleansing merupakan proses untuk menghapus karakter yang tidak diperlukan, yakni karakter spesial dan simbol-simbol, URL, tag HTML, emote, dan lain-lain. Data cleansing bertujuan untuk mengurangi noise data.

c. Case Folding

Case folding adalah proses mengubah semua huruf dalam dokumen menjadi huruf kecil. Huruf ‘a’ sampai dengan ‘z’ dapat diterima, sedangkan karakter selain huruf dihilangkan dan dianggap delimiter. Fungsi lower() digunakan dalam tahap case folding.

d. Tokenizing

Tokenizing atau tokenisasi merupakan proses memecah kalimat dalam suatu dokumen menjadi kumpulan kata atau token. Dalam tahap ini setiap tweet dibagi menjadi bagian-bagian lebih kecil yakni setiap kata, dengan menggunakan spasi sebagai pemisah.

e. Normalization (Stemming dan Lemmatization)

Text normalization merupakan proses untuk menghapus identifikasi kata silang dan penulisan kata berlebihan yang kemudian diganti dengan kata baku berdasarkan kamus Bahasa Inggris NLTK. *Stemming* merupakan proses penghapusan imbuhan yang ada pada token menjadi kata dasarnya. Contohnya, untuk kata “*programmer*”, “*programming*” dan “*programs*” dapat

disederhanakan menjadi kata dasar "*program*" menggunakan *stemming*. Sedangkan lemmatization merupakan teknik lain yang digunakan untuk mengurangi kata-kata berinfleksi menjadi kata dasarnya. Proses ini mengidentifikasi "*lemma*" (bentuk kamus) kata berinfleksi berdasarkan makna yang dimaksudkan. Contohnya dari kata "*better*" dapat dilakukan lemmatization menjadi kata "*good*".

f. Filtering stopwords

Stopwords merupakan kata-kata yang tidak memiliki arti atau makna dalam kalimat, sehingga tidak penting dalam dataset. Oleh karena itu dilakukan filtering untuk menghapus stopword dalam tweet. Library NTLK menyediakan list stopword Bahasa Inggris yang selanjutnya akan digunakan dalam penelitian ini. Penghapusan stopword dalam data dapat mempermudah pengolahan data dan mengurangi kesalahpahaman dalam proses analisis.

3.4.2 Pre-processing Data Saham

Pre-processing data histori saham bertujuan agar data tersebut sesuai dengan yang dibutuhkan dalam penelitian ini. Langkah-langkah dalam pre-processing data history saham adalah:

a. Filtering data saham

Dari dataset yang telah didapatkan, terdapat sebanyak 25 jenis saham perusahaan yang paling populer di situs Yahoo Finance.

Karena fokus penelitian ini hanya menyangkut kepada saham perusahaan Tesla, Inc., maka dilakukan filtrasi atau penyaringan data saham yang tidak diperlukan.

b. Menghilangkan missing value

Missing value dapat ditangani dengan cara mengabaikan nilai missing value, mengisi nilai dengan manual, menggunakan konstanta global, dan menggunakan nilai rata-rata / median.

Nilai untuk mengisi missing value dapat ditentukan dengan menggunakan regresi, alat berbasis inferensi menggunakan bayes formalism, atau decision tree. Namun, pada penelitian ini, cara yang digunakan untuk mengatasi missing value yaitu dengan menghilangkan data yang memiliki missing value yang terdapat pada data history saham.

c. Normalisasi data

Normalisasi data dilakukan dalam rentang yang kecil seperti [0,1] atau [-1, 1] sehingga semua atribut mempunyai bobot yang sama. Min-max merupakan metode normalisasi dengan melakukan transformasi linear terhadap data asli.

Metode min-max menggunakan nilai minimum dan maksimum untuk melakukan konversi data secara linear. Misalkan, A adalah atribut bertipe numerik, maka min_a adalah nilai minimum dalam atribut A dan, max_a adalah nilai maksimum dalam atribut A. Suatu nilai x_i dapat dinormalisasi menjadi nilai

baru x_i^1 yang berada dalam rentang [minbaru A, maksbaru A] sebagaimana disajikan dalam persamaan:

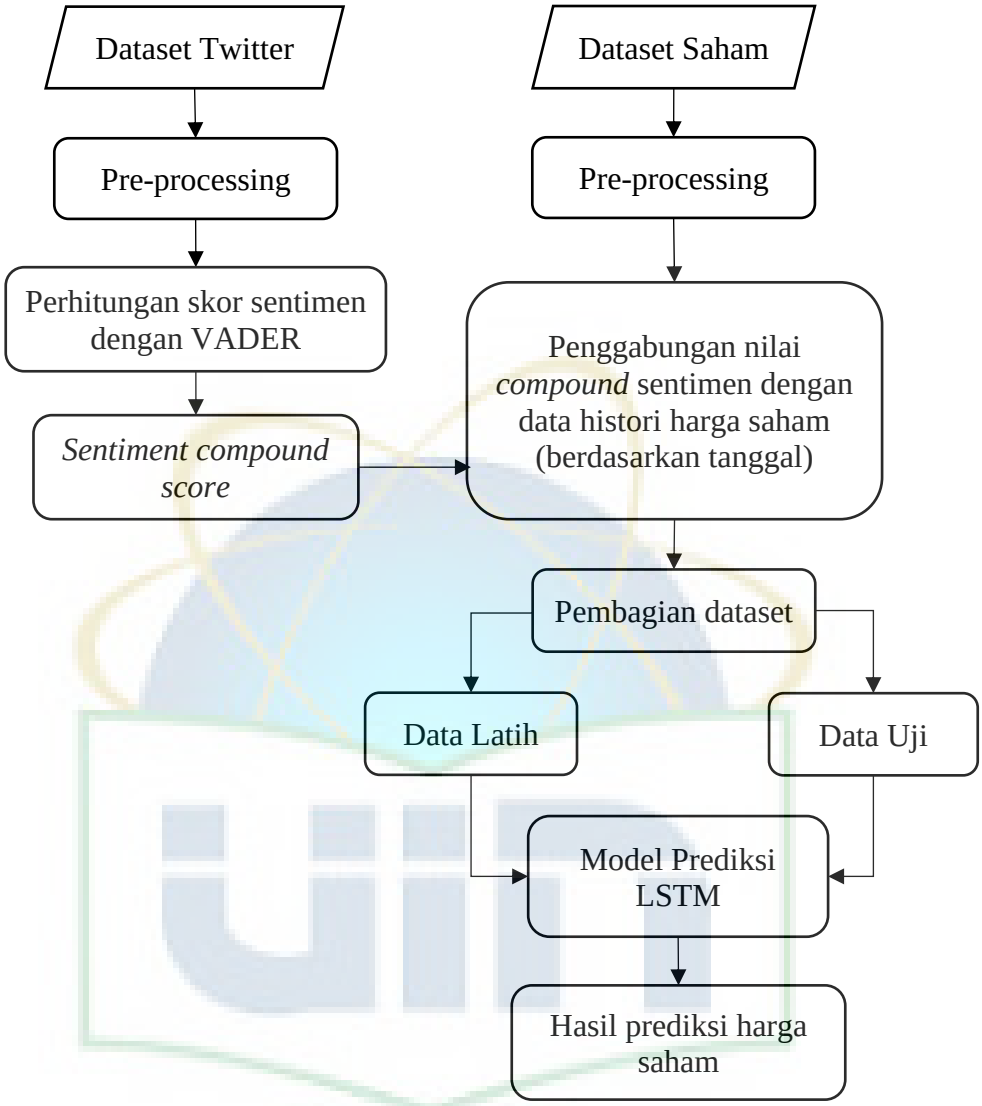
$$x_i^1 = \frac{x_i - min_a}{max_a - min_a} \quad (17)$$

Di mana x_i^1 adalah data atribut yang akan dinormalisasi, x_i adalah data yang belum dinormalisasi, min_a adalah nilai terkecil pada suatu atribut, dan max_a adalah nilai terbesar pada suatu atribut.

Metode min-max telah banyak digunakan secara praktis. Deangan metode min-max dapat dilakukan transformasi linier terhadap data asli sehingga menghasilkan keseimbangan nilai perbandingan antar data saat sebelum dan sesudah proses.

3.5 Perancangan Model Prediksi Harga Saham dengan Sentimen Analisis

Kedua dataset yang sudah dipreprocess siap digunakan sebagai dataset pelatihan dan ujicoba algoritma hybrid. Dataset tweet kemudian dilakukan analisis sentimennya menggunakan algoritma VADER yang kemudian hasil analisisnya dirangkum berdasarkan tanggal dan digabungkan dengan dataset saham untuk digunakan sebagai parameter prediksi harga saham.



Gambar 3.1 Implementasi Model

3.5.1 Pemodelan Analisis Sentimen dengan Algoritma VADER

Untuk dapat melakukan prediksi harga saham dengan parameter atau feature set sentimen pengguna twitter, diperlukan data tweet yang telah dilabeli sentimennya. Di tahapan ini, setiap tweet yang telah dilakukan pre-processing selanjutnya diberikan label bobot sentimen, baik sentimen negatif, positif, maupun netral. Terdapat beberapa cara yang dapat dilakukan untuk memberikan bobot sentimen pada

dokumen, seperti metode primitif dengan menilai teks secara manual melalui pendapat pengecek. Ada cara lain untuk membobotkan nilai sentimen yakni dengan menggunakan metode analisis sentimen dengan pendekatan lexicon dan rule-based scoring bernama VADER.

VADER adalah alat yang dapat menilai sentimen dan valensi tiap tweet dengan mencocokkan kata-kata yang muncul dengan kamus lexicon bawaan librarynya. Pada penelitian ini, penulis menggunakan metode VADER untuk melabelkan rasio sentimen positif, negatif, netral, dan skor sentimen gabungan dari tweet yang telah dipre-process sebelumnya.

3.5.2 Penggabungan Dataset Saham dan Dataset Tweet yang telah dilabeli

Setelah dataset tweet pengguna dilabelkan nilai sentimennya, langkah selanjutnya yakni menyatukan keduanya menjadi satu dataset utuh. Caranya yakni dengan merata-ratakan dataset tweet berdasarkan nilai compound dalam satu hari (per tanggal), kemudian menggabungkannya ke dalam dataset riwayat harga saham.

3.5.3 Pemodelan Prediksi Harga Saham dengan Algoritma LSTM

Kedua dataset yang telah digabungkan selanjutnya direshape dan dinormalisasikan sebelum digunakan dalam algoritma LSTM.

Algoritma deep learning yang digunakan dalam penelitian ini disediakan oleh library Keras, dengan fungsi Sequence(). Model dibangun menggunakan campuran layer LSTM, Dense, dan Dropout.

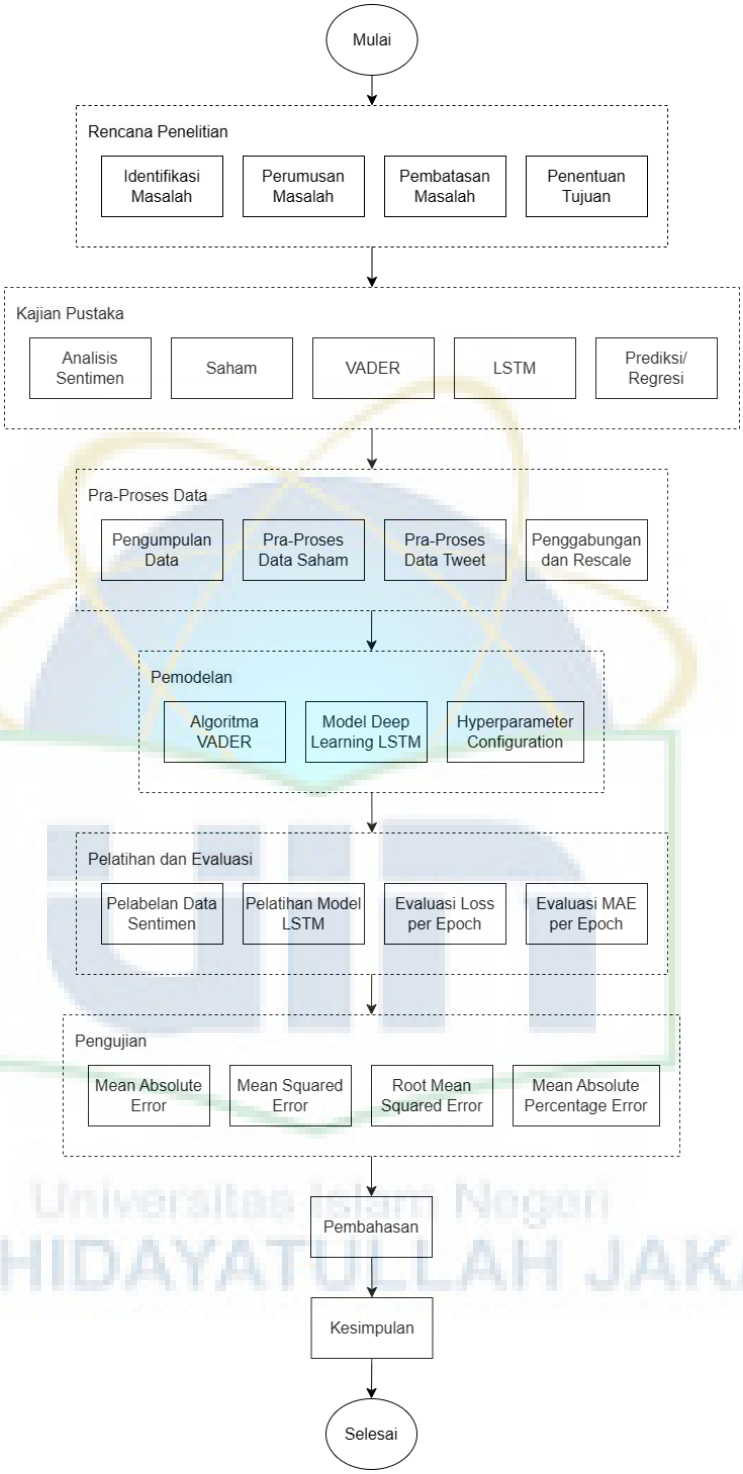
Model yang telah dibuat kemudian dilakukan fine-tuning hyperparameter untuk mencari konfigurasi parameter deep learning yang paling optimal.

3.6 Pengujian dan Evaluasi

Proses pengujian pada penelitian ini dilakukan dengan menguji hasil prediksi dari model hybrid usulan, serta hasil prediksi model LSTM tanpa fitur ekstra sentimen. Hasil prediksi tersebut kemudian akan dievaluasi, dengan membandingkan hasil kedua model menggunakan *metrics* pengukuran performa yang cocok untuk model, yakni *mean absolute error* (MAE), *mean absolute percentage error* (MAPE), *mean square error* (MSE), dan *root mean square error* (RMSE). MAE menghitung rata-rata simpangan nilai prediksi dengan nilai sebenarnya dalam dokumen. MAPE menampilkan rata-rata persentase simpangan antara nilai prediksi dengan nilai asal, MSE merupakan rata-rata kuadrat simpangan nilai prediksi dengan nilai sebenarnya, dan RMSE adalah nilai akar dari rata-rata kuadrat simpangan nilai prediksi. Ketiga metrik tersebut dijadikan sebagai pembanding antara performa model *hybrid* yang diusulkan dengan model LSTM biasa.

3.7 Kerangka Penelitian

Kerangka penelitan yang dilakukan pada penelitian ini terangkum pada Gambar 3.2 dibawah ini.



Gambar 3.2 Kerangka penelitian

BAB IV

IMPLEMENTASI SISTEM

4.1 Pengumpulan Data

Pada penelitian ini, data yang diperlukan ada dua jenis, yakni data riwayat harga saham perusahaan TSLA, serta data kumpulan tweet yang membahas perusahaan tersebut. Perlu dicatat bahwa kedua data tersebut harus memiliki atribut tanggal dan waktu sehingga dapat digabungkan menjadi satu dataset yang akan diolah selanjutnya. Setelah melakukan pencarian di internet, ditemukan beberapa dataset yang sesuai dengan kebutuhan penelitian. Dataset yang akan digunakan pada penelitian ini merupakan dataset dengan judul "*Stock Tweets for Sentiment Analysis and Prediction*" yang telah disediakan oleh akun pengguna *equinxx* di situs web dataset ternama yakni Kaggle.

```
import pandas as pd
import numpy as np

all_stocks = pd.read_csv("stock_yfinance_data.csv")
all_stocks
```

Tabel 4.1 Raw dataset saham

	Date	Open	High	Low	Close	Adj Close	Volume	Stock Name
0	2021-09-30	260.333344	263.043335	258.333344	258.493347	258.493347	53868000	TSLA
1	2021-10-01	259.466675	260.260010	254.529999	258.406677	258.406677	51094200	TSLA
...	...	...	...	...	...	...	...	...
6298	2022-09-28	13.050000	13.421000	12.690000	13.330000	13.330000	31799400	XPEV

6299	2022-09-29	12.550000	12.850000	11.850000	12.110000	12.110000	33044800	XPEV

```
all_tweets = pd.read_csv("stock_tweets.csv")
all_tweets
```

Tabel 4.2 Raw dataset tweets

	Date	Tweet	Stock Name	Company Name
1	2022-09-29 23:41:16+00:00	Mainstream media has done an amazing job at br...	TSLA	Tesla, Inc.
2	2022-09-29 23:24:43+00:00	Tesla delivery estimates are at around 364k fr...	TSLA	Tesla, Inc.
...	...	...	...	...
80791	2021-10-01 00:03:32+00:00	We delivered 10,412 Smart EVs in Sep 2021, rea...	XPEV	XPeng Inc.
80792	2021-09-30 10:22:52+00:00	Why can XPeng P5 deliver outstanding performan...	XPEV	XPeng Inc.

Dataset tersebut terdiri atas 25 riwayat harga saham paling populer di situs *stock ticker* (pemantau harga saham) *Yahoo! Finance* dalam rentang waktu antara 30-09-2021 hingga 30-09-2022. Terdapat pula sejumlah lebih dari 80 ribu tweet yang me-*mention* 25 perusahaan tersebut.

4.2 Pre-processing Data

Dalam *pre-processing*, dataset yang telah didapat diolah terlebih dahulu sebelum digunakan dalam proses labeling sentimen dan prediksi.

4.2.1 Pre-processing Data Tweet

a. Filter data tweet untuk perusahaan TSLA

Dataset tweet yang berjumlah sekitar 80.000 tweet pertama-tama disaring terlebih dahulu untuk menghapus/drop data tweet dari perusahaan yang tidak diperlukan. Lalu dilakukan juga format tanggal dataset agar dapat dataset dapat ditransformasikan menjadi *time series*.

```
tweet_df = tweet_df.drop(['Company Name', 'Stock
Name'], axis=1)
tweet_df['Date'] =

pd.to_datetime(tweet_df['Date']).dt.date
```

Tabel 4.3 Dataset tweet yang telah difilter

	Date	Tweet
0	2022-09-29	Mainstream media has done an amazing job at br...
1	2022-09-29	Tesla delivery estimates are at around 364k fr...
...	...	...
37420	2021-09-30	Get ready for a $TSLA _ _ _ _ _ _ Q3 delivery...
37421	2021-09-30	In other words, AMD has been giving Tesla pref...

b. Data cleansing

Data cleansing dilakukan untuk membersihkan dataset tweet dan bertujuan untuk mengurangi kesalahan serta mengurangi noise pada dataset. Karakter-karakter yang tidak diperlukan contohnya *mention*, *emoticon*, *hashtag*, *retweet*, tanda baca, karakter angka, URL *website*, *tag* HTML, dan lain-lain.

```
def preprocess_tweet(tweet):
```

```
    import emoji
    import re

    new_tweet = tweet

    new_tweet = re.sub(r'https?://[^ ]+', '',
new_tweet)
    new_tweet = re.sub(r'@[^ ]+', '', new_tweet)
    new_tweet = re.sub(r'#', '', new_tweet)
    new_tweet = re.sub(r'([A-Za-z])\1{2,}', r'\1',
new_tweet)
    new_tweet = emoji.demojize(new_tweet)
    new_tweet = re.sub(r' 0 ', 'zero', new_tweet)
    new_tweet = re.sub(r'[^A-Za-z ]', '',
new_tweet)
```

Tabel 4.4 Dataset tweet yang telah dibersihkan

	Date	Tweet
0	2022-09-29	Mainstream media has done an amazing job at br...
1	2022-09-29	Tesla delivery estimates are at around k from ...
...	...	...
37420	2021-09-30	Get ready for a TSLA Q delivery numberH...
37421	2021-09-30	In other words AMD has been giving Tesla prefe...

c. Case folding

Case folding merupakan proses untuk mengubah semua huruf pada dokumen menjadi huruf kecil. Case folding dalam bahasa pemrograman python dilakukan dengan fungsi lower().

```
new_tweet = new_tweet.lower()
```

Tabel 4.5 Dataset tweet yang telah di-casefold

	Date	Tweet

0	2022-09-29	mainstream media has done an amazing job at br...
1	2022-09-29	tesla delivery estimates are at around k from ...
...	...	...
37420	2021-09-30	get ready for a tsla q delivery numberh...
37421	2021-09-30	in other words amd has been giving tesla prefe...

d. Tokenization

Tokenisasi merupakan proses memecah kalimat dalam suatu dokumen menjadi kumpulan kata atau token. Dalam tahap ini setiap tweet dibagi menjadi bagian-bagian lebih kecil yakni setiap kata, dengan menggunakan spasi sebagai pemisah.

```
from nltk import word_tokenize
nltk.download('punkt')

tokens = word_tokenize(new_tweet)
```

Tabel 4.6 Dataset tweet yang telah ditokenisasi

	Date	**Tweet**
0	2022-09-29	[mainstream, media, has, done, an, amazing, jo...
1	2022-09-29	[tesla, delivery, estimates, are, at, around, ...
...	...	...
37420	2021-09-30	[get, ready, for, a, tsla, q, delivery, number...
37421	2021-09-30	[in, other, words, amd, has, been, giving, tes...

e. Stopword filtering

Stopword filtering merupakan proses penghapusan kata-kata yang tidak memiliki arti atau makna dalam kalimat, sehingga tidak penting dalam dataset.

```
from nltk.corpus import stopwords
nltk.download('stopwords')
from nltk.stem import PorterStemmer

for token in tokens:
    if token in stopwords.words('english'):
        tokens.remove(token)
    token = porter.stem(token)
```

Tabel 4.7 Dataset tweet yang telah difilter stopword

	Date	Tweet
0	2022-09-29	[mainstream, media, done, amazing, job, brainw...
1	2022-09-29	[tesla, delivery, estimates, at, around, k, th...
...	...	...
37420	2021-09-30	[get, ready, a, tsla, q, delivery, numberhave,...
37421	2021-09-30	[other, words, amd, been, giving, tesla, prefe...

f. Normalization

Normalisasi data yakni mengubah data tiap-tiap token menjadi kata baku. Normalisasi data terdapat dua jenis yakni stemming, dan lemmatization. Stemming yakni penghapusan imbuhan pada kata, sedangkan lemmatization yakni pengubahan kata menjadi kata dasarnya.

```
from nltk.corpus import stopwords
```

```
nltk.download('stopwords')
from nltk.stem import PorterStemmer

for token in tokens:
    if token in stopwords.words('english'):
        tokens.remove(token)
    token = porter.stem(token)
```

Tabel 4.8 Dataset tweet yang telah dinormalisasi

	Date	Tweet
0	2022-09-29	[mainstream, media, has, done, an, amazing, jo...
1	2022-09-29	[tesla, delivery, estimates, are, at, around, ...
...	...	...
37420	2021-09-30	[get, ready, for, a, tsla, q, delivery, number...
37421	2021-09-30	[in, other, words, amd, has, been, giving, tes...

Terakhir, data dikembalikan menjadi berupa text biasa dengan fungsi join(). Ini dikarenakan tipe data yang diperlukan untuk model pelabelan sentimen VADER merupakan string utuh, bukan token-token pecahan kalimat.

```
return ' '.join(tokens)
```

Tabel 4.9 Dataset tweet yang telah digabungkan

	Date	Tweet
0	2022-09-29	mainstream media done amazing job brainwashing...
1	2022-09-29	tesla delivery estimates at around k the analy...
...	...	...

37420	2021-09-30	get ready a tsla q delivery numberhave ur answ...
37421	2021-09-30	other words amd been giving tesla preferential...

g. Penyimpanan dataset tweet yang telah dipre-process

Dikarenakan dataset tweet yang cukup besar yakni sekitar 37 ribu data tweet yang berkaitan dengan perusahaan TSLA, metode pre-processing dataset ini memerlukan waktu yang cukup panjang (sekitar 10 menit). Oleh karena itu, penulis berinisiatif menyimpan data yang telah dipre-process menjadi file csv, dan apabila selanjutnya diperlukan untuk training/testing dapat diimport kembali.

```
processed_df.to_csv('tweet_processed.csv',
index=False)

processed_df = pd.read_csv('tweet_processed.csv')
processed_df['Date'] =
pd.to_datetime(processed_df['Date'])
processed_df['Date'] = processed_df['Date'].dt.date
processed_df
```

4.2.2 Pre-Processing Data Saham

a. Filtering data saham

Langkah pertama dalam pemrosesan data saham yakni menghapus/drop data riwayat harga saham dari perusahaan selain TSLA. Kemudian dataset diubah menjadi bentuk *timeseries* (deret waktu) dengan menetapkan tanggal pada setiap baris sebagai indeks.

```
stock_df = all_stocks[all_stocks['Stock Name'] ==
company]
stock_df = stock_df.drop('Stock Name', axis=1)
stock_df['Date'] =
pd.to_datetime(stock_df['Date']).dt.date
stock_df = stock_df.set_index("Date")
stock_df
```

Tabel 4.10 Dataset saham yang telah difilter

Date	Open	High	Low	Close	Adj Close	Volume
2021-09-30	260.333344	263.043335	258.333344	258.493347	258.493347	53868000
2021-10-01	259.466675	260.260010	254.529999	258.406677	258.406677	51094200
...	...	...	...	...	...	...
2022-09-28	283.079987	289.000000	277.570007	287.809998	287.809998	54664800
2022-09-29	282.760010	283.649994	265.779999	268.209991	268.209991	77620600

b. Drop data yang kosong atau null

Langkah selanjutnya yakni menghapus baris dengan data kosong. Apabila ada baris yang dihapus, maka nilai kosong tersebut bisa diaugmentasikan dari data yang sudah ada.

```
stock_df.dropna()
```

Tabel 4.11 Dataset yang telah didrop null

Date	Open	High	Low	Close	Adj Close	Volume
2021-09-30	260.333344	263.043335	258.333344	258.493347	258.493347	53868000
2021-10-01	259.466675	260.260010	254.529999	258.406677	258.406677	51094200

...	...	...	...	...	...	...
2022-09-28	283.079987	289.000000	277.570007	287.809998	287.809998	54664800
2022-09-29	282.760010	283.649994	265.779999	268.209991	268.209991	77620600

Dari data yang diberikan ternyata tidak ada baris kosong, sehingga tidak diperlukan augmentasi dari data yang sudah ada untuk mengisi celah data.

c. Normalisasi data

Langkah selanjutnya yakni normalisasi data dengan mengubah nilai-nilai dari dataset menjadi rentang yakni [0,1]. Proses normalisasi ini menggunakan library *MinMaxScaler* bawaan NLTK. Cara kerjanya yakni mencari patokan nilai tertinggi dan terendah dalam data, kemudian melakukan transformasi linier terhadap seluruh data lain berdasarkan nilai tertinggi dan terendah tersebut. Hasilnya seluruh nilai dataset akan berbobot antara 0 hingga 1.

Langkah normalisasi data disini dilakukan setelah kedua dataset yakni saham dan tweet digabungkan, agar keduanya dapat dilakukan normalisasi secara bersamaaan.

4.3 Implementasi Model Pelabelan Data dengan VADER

Data twitter yang sudah melalui tahap pre-processing selanjutnya diberikan label. Dalam melakukan pelabelan data, digunakan tools *SentimentIntensityAnalyzer* yakni *library vader lexicon* yang tersedia dalam

package NLTK. Vader merupakan library yang mampu melabelkan data dengan mencocokkan daftar kamus lexicon bawaan library dengan kata-kata yang muncul pada dokumen. Skor yang dihasilkan dari algoritma yakni skor positif, skor negatif dan skor negatif. Ketiga skor tersebut kemudian digabungkan menjadi *compound score* (skor gabungan).

Compound score dihitung dengan menjumlahkan skor sentimen setiap kata dalam leksikon, disesuaikan dengan aturan, lalu dinormalisasi menjadi antara -1 (negatif paling ekstrem) dan +1 (positif paling ekstrem). Ini adalah metrik yang paling berguna apabila menginginkan satu ukuran sentimen satu dimensi untuk kalimat tertentu.

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')

sentiment_analyzer = SentimentIntensityAnalyzer()
for indx, row in processed_df.T.items():
    try:
        sentence_sentiment =
sentiment_analyzer.polarity_scores(processed_df.loc[indx,
'Tweet'])
        processed_df.at[indx,'Negative'] =
sentence_sentiment['neg']
        processed_df.at[indx,'Neutral'] =
sentence_sentiment['neu']
        processed_df.at[indx,'Positive'] =
sentence_sentiment['pos']
        processed_df.at[indx,'Compound'] =
sentence_sentiment['compound']
    except TypeError:
        print (processed_df.loc[indx, 'Tweet'])
        print (indx)
        break

processed_df
```

Tabel 4.12 Dataset tweet yang telah dinilai sentimennya

	Date	Tweet	Negative	Neutral	Positive	Compound

0	2022-09-29	mainstream media done amazing job brainwashing...	0.181	0.655	0.164	0.0772
1	2022-09-29	tesla delivery estimates at around k the analy...	0.000	1.000	0.000	0.0000
...	...	...	...	...	...	...
37420	2021-09-30	get ready a tsla q delivery numberhave ur answ...	0.000	0.737	0.263	0.3612
37421	2021-09-30	other words amd been giving tesla preferential...	0.000	0.876	0.124	0.3400

Selanjutnya dapat ditentukan sentimen dari setiap tweet berdasarkan nilai compound. Aturan yang umumnya ditetapkan yakni apabila nilai compound ≥ 0.5 maka sentimen tweet ditetapkan sebagai positif, nilai compound di antara -0.5 hingga 0.5 ditetapkan sebagai netral, dan nilai compound ≤ -0.5 tweetnya sentimennya negatif.

```
labeled_tweets_df = processed_df.copy()

for indx, row in processed_df.T.items():
    if (processed_df.at[indx, 'Compound'] > 0.5):
        labeled_tweets_df.at[indx, 'Sentiment'] = 'Positive'
    elif (processed_df.at[indx, 'Compound'] < -0.5):
        labeled_tweets_df.at[indx, 'Sentiment'] = 'Negative'
    else:
        labeled_tweets_df.at[indx, 'Sentiment'] = 'Neutral'

labeled_tweets_df
```

Tabel 4.13 Dataset tweet yang telah dilabelkan

	Date	Tweet	Negative	Neutral	Positive	Compound	Sentiment
0	2022-09-29	mainstream media done amazing job brainwashing...	0.181	0.655	0.164	0.0772	Neutral
1	2022-09-29	tesla delivery estimates at around k the analy...	0.000	1.000	0.000	0.0000	Neutral
...	...	...	...	...	...	...	
37420	2021-09-30	get ready a tsla q delivery numberhave ur answ...	0.000	0.737	0.263	0.3612	Neutral
37421	2021-09-30	other words amd been giving tesla preferential...	0.000	0.876	0.124	0.3400	Neutral

4.4 Implementasi Prediksi Harga Saham dengan LSTM

Pada tahapan ini penulis mengimplementasikan model LSTM yang akan digunakan untuk sistem prediksi harga saham. Secara umum, model LSTM yang digunakan dalam penelitian ini memanfaatkan library Keras. Keras menyediakan fungsi Sequence yang dapat digunakan dalam membangun model LSTM.

4.4.1 Penggabungan, Reshaping dan Split Data

Kedua data yang telah dipre-process dan divalidasi selanjutnya digabungkan menjadi dataset yang selanjutnya akan digunakan oleh model LSTM. Data tweet sebelumnya dirata-ratakan terlebih dahulu berdasarkan tanggal. Karena perata-rataan hanya dibatasi oleh jenis data numerik, maka kolom non-numerik akan otomatis di-drop saat perata-rataan.

```
daily_sentiments_df =
labeled_tweets_df.groupby([labeled_tweets_df['Date']]).m
ean(numeric_only=True)
daily_sentiments_df
```

Tabel 4.14 Dataset tweet yang telah digabungkan per harian

Date	Negative	Neutral	Positive	Compound
2021-09-30	0.081300	0.739611	0.179067	0.241496
2021-10-01	0.051798	0.811106	0.137106	0.198523
...	...	...	...	...
2022-09-28	0.085640	0.779133	0.135227	0.110548
2022-09-29	0.104089	0.765241	0.130670	0.078079

Selanjutnya kedua dataset digabung berdasarkan tanggalnya. Perlu diperhatikan bahwa dataset harga saham hanya mencakup hari kerja (Senin – Jumat), sehingga terdapat dataset tweet harian yang tidak dimasukkan menjadi dataset akhir.

```
dataset_df = stock_df.copy()
dataset_df = dataset_df.join(daily_sentiments_df,
how="left", on="Date")
dataset_df
```

Tabel 4.15 Dataset gabungan

Date	Open	High	Low	Close	...	Positive	Compound
2021-09-30	260.333344	263.043335	258.333344	258.493347	...	0.179067	0.241496
2021-10-01	259.466675	260.260010	254.529999	258.406677	...	0.137106	0.198523
...	...	...	...	...	...	...	...
2022-09-28	283.079987	289.000000	277.570007	287.809998	...	0.135227	0.110548
2022-09-29	282.760010	283.649994	265.779999	268.209991	...	0.130670	0.078079

Data yang diperlukan oleh sebuah input layer di dalam model Keras harus mengikuti bentuk yang telah ditentukan. Pada model LSTM, data yang diperlukan berupa array 3 dimensi (x, y, z), dengan dimensi x menentukan jumlah batch yang dikirim pada setiap iterasi learning, y menggambarkan jumlah timestep atau seberapa jauh data akan diakses tiap iterasi, dan z merupakan jumlah feature yang ada dalam data.

Dikarenakan data yang ada dalam dataset berupa dataframe 2 dimensi, maka data tersebut harus ditransformasikan menjadi data yang dapat digunakan model. Proses ini juga sering disebut sebagai 'reshaping'. Data juga dinormalisasikan agar dapat meningkatkan kinerja dan hasil prediksi.

Pada penelitian ini, proses reshaping dan normalisasi dilakukan bersamaan dengan proses splitting data menjadi data pelatihan dan data pengujian. Porsi pemecahan data yang di penelitian ini yakni 80% data pelatihan dan 20% data uji. Berikut fungsi yang bekerja untuk menormalisasikan dataset, kemudian memecah data menjadi data latih dan data uji.

```
def SplitData(data, train_size, timestep):

    data_values = data.values
    training_data_len = math.ceil(len(data)* train_size)

    scaler = MinMaxScaler(feature_range=(0,1))
    scaled_data = scaler.fit_transform(data_values)

    train_data = scaled_data[0: training_data_len, :]
    test_data = scaled_data[training_data_len-timestep:
, :]

    train_data_x = train_data[0: training_data_len, :]
    train_data_y = train_data[0: training_data_len, 0:1]

    x_train = []
    y_train = []

    for i in range(timestep, len(train_data_x)):
        x_train.append(train_data_x[i-timestep:i])
        y_train.append(train_data_y[i][0])

    x_train, y_train = np.array(x_train),
np.array(y_train)
    x_train = np.reshape(x_train, (x_train.shape[0],
x_train.shape[1], x_train.shape[2]))
```

```
    test_data = scaled_data[training_data_len-timestep:
, : ]
    x_test = []
    y_test = data_values[training_data_len: , 0]

    for i in range(timestep, len(test_data)):
        x_test.append(test_data[i-timestep:i])

    x_test = np.array(x_test)
    x_test = np.reshape(x_test, (x_test.shape[0],
x_test.shape[1], x_test.shape[2]))

    return x_train, y_train, x_test, y_test, scaler
```

4.4.2 Pembangunan Model

Tahapan selanjutnya yakni pembangunan model deep learning LSTM. Berikut adalah model dasar LSTM yang dibangun untuk penelitian ini.

```
regressor = Sequential()
regressor.add(LSTM(units=neuron_units,
return_sequences=True, input_shape=(x_data.shape[1],
x_data.shape[2])))
regressor.add(Dropout(0.2))

regressor.add(LSTM(units=neuron_units,
return_sequences=False))
regressor.add(Dropout(0.2))
regressor.add(Dense(units=1, activation='linear'))
```

Berikut beberapa penjelasan pada beberapa lapisan yang muncul pada model:

1. Model yang digunakan pada penelitian ini berupa Sequential() model, yang memungkinkan lapisan library keras yang tersedia ditumpuk, sehingga data yang masuk kedalam satu lapisan dapat bergerak ke lapisan berikutnya.
2. Pada lapisan LSTM pertama, terdapat variabel *neuron_units* yang menentukan banyaknya neuron dalam lapisan tersebut.

Ditambahkan juga parameter return_sequences=True agar data yang telah diproses dalam lapisan ini dapat mengalir ke lapisan berikutnya. Parameter input_shape mengatur bentuk data yang akan digunakan dalam lapisan. x_data.shape[1] menggambarkan jumlah timestep pada data, dan x_data.shape[2] menunjukkan jumlah input atau feature set yang ada dalam data.

3. Ditambahkan lapisan Dropout(0.2) untuk secara acak mengatur input yang dialirkan dari lapisan sebelumnya menjadi nol. Lapisan dropout digunakan untuk mencegah overfitting dan mengurangi ketergantungan terhadap feature spesifik.
4. Kemudian ditambahkan lapisan LSTM lagi, namun dengan parameter return_sequences=False.
5. Lapisan Dropout(0.2) ditambahkan kembali.
6. Terakhir, lapisan Dense ditambahkan dengan unit sejumlah 1 unit. Fungsi aktivasi dari lapisan ini adalah fungsi linear, sehingga data yang dihasilkan berupa deretan nilai, sehingga cocok digunakan dalam penelitian ini.

Tabel berikut menggambarkan arsitektur model yang akan digunakan dalam penelitian ini.

Tabel 4.16 Arsitektur model LSTM

No	Layer (type)	Output Shape	Param #
1	lstm (LSTM)	(None, 60, 64)	16896
2	dropout (Dropout)	(None, 60, 64)	0

3	lstm_1 (LSTM)	(None, 64)	33024
4	dropout (Dropout)	(None, 64)	0
5	dense (Dense)	(None, 1)	65

4.4.3 Konfigurasi Model

Pada tahapan ini, penulis akan merancang beberapa skenario yang akan digunakan untuk pelatihan model. Penulis sebelumnya membuat fungsi untuk melatih model.

```
def TrainModel (x_data, y_data, epoch, neuron_units):
```

Dalam fungsi tersebut, terdapat beberapa parameter yang digunakan, di antaranya:

- x_data, yakni kumpulan data yang memetakan input yang akan digunakan oleh model. Pada model ini, data yang digunakan berupa data saham dan sentimen yang berbentuk array numpy 3 dimensi.
- y_data adalah data yang memetakan output yang dihasilkan berdasarkan data yang dimasukkan oleh x_data. Dalam penelitian ini, y_data merupakan data harga closing saham.
- epoch, yakni jumlah iterasi/perulangan data yang beredar dalam model.
- neuron_units mengatur jumlah units dari tiap layer LSTM yang ada dalam model.

Di dalam fungsi pelatihan model terdapat beberapa bagian, di antara lain:

a. Rancangan model dasar

Di bagian pertama dimasukkan rancangan model LSTM yang telah dibuat di bagian 4.4.2 sebelumnya. Di sini variabel yang didapatkan dari parameter fungsi akan digunakan dalam proses pelatihan model.

```
regressor = Sequential()
    ...
```

b. Pengaturan kompilasi model

Pada bagian ini model dikompilasi/dibangun menggunakan metode compile(). Di sini juga beberapa parameter yang dapat mempengaruhi kinerja model diatur, di antaranya:

- optimizer menentukan algoritma pengoptimalan yang digunakan dalam pelatihan. Pada penelitian ini digunakan algoritma optimizer Adam, yakni algoritma optimizer berbasis stochastic gradient descent dimodifikasi yang umum dipakai.
- loss menentukan loss function yang memengaruhi jalannya proses training. Fungsi loss harus sesuai dengan jenis data dan pada penelitian ini digunakan metrik mean squared error sebagai fungsi lossnya.
- metrics menentukan metrik performa apa saja yang akan dilacak selama dilaksanakan pelatihan.

```
regressor.compile(optimizer='adam',
                  loss='mean_squared_error',
                  metrics=['mae'])
```

c. Pelatihan model

Model yang telah dicompile selanjutnya di lakukan pelatihan menggunakan method .fit(), dengan beberapa parameter yang dibutuhkan:

- x_data dan y_data, mewakilkan dataset yang akan digunakan untuk pelatihan. x_data merupakan kumpulan variabel independen yang digunakan untuk melatih model dengan y_data adalah variabel dependen yang ingin dilatih/diprediksi.
- batch_size menunjukkan seberapa banyak potongan data yang akan dikirim kepada layer input dalam satu waktu.
- epoch mengatur seberapa banyak iterasi yang akan dilakukan model terhadap dataset yang diberikan.
- validation_split menetapkan porsi data validasi yang akan digunakan dari data training.

```
history = regressor.fit(x_data, y_data,
                        batch_size=batch_size,
                        epochs=epoch,
                        validation_split=0.2)
```

4.4.4 Fine Tuning Hyperparameter

Pada bagian pelatihan model, digunakan beberapa parameter yang dapat digunakan untuk mengoptimalkan hasil prediksi dan kinerja model.

a. Jumlah epoch

Nilai epoch memengaruhi jumlah iterasi dataset diberikan kepada dataset. Semakin banyak epoch yang dimasukkan pada model, maka akan semakin banyak pula dataset akan diiterasikan pada model, dan semakin lama pula runtime pelatihan model.

Pada pelatihan model LSTM ini, ada beberapa konfigurasi hyperparameter jumlah epoch yang digunakan: 5 epoch, 10 epoch, 20 epoch, 30 epoch, 40 epoch, 50 epoch.

```
epochs = [5, 10, 20, 30, 40, 50]
```

b. Jumlah unit LSTM

Jumlah unit pada setiap layer LSTM memengaruhi jumlah neuron yang terkandung di setiap layernya. Ini kemungkinan dapat mengubah kinerja dan kemampuan dari model untuk "mengingat" data-data yang disediakan sebelumnya.

Di pelatihan model ini, terdapat beberapa konfigurasi jumlah unit LSTM-nya: 16 unit, 32 unit, 64 unit, dan 128 unit.

```
neurons = [16, 32, 64, 128]
```

4.4.5 Pelatihan Model

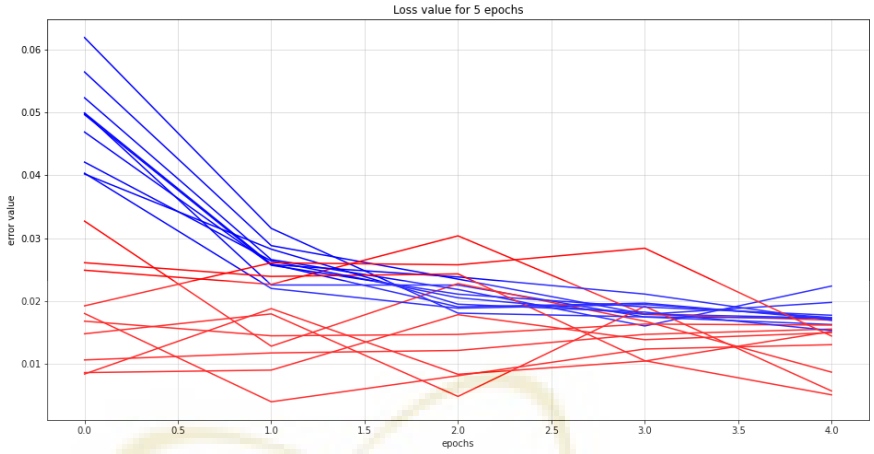
Pada tahapan ini, model dilatih untuk mencari hasil metrik pelatihan dan pengujian yang paling baik untuk setiap hyperparameter yang dikonfigurasikan. Dan untuk menghindari hasil-hasil outlier, pelatihan model untuk setiap hyperparameter dilakukan berulang kali untuk menghasilkan sebaran prediksi yang didapatkan.

Berikut contoh pelatihan untuk hyperparameter *epochs* pada konfigurasi model.

```
epoch_finetune_history = []

for i in range(len(epochs)):
    for j in range(10):

        x_train, y_train, x_test, y_test, scaler =
SplitData(combined_data, train_portion, timestep)

        start_time = time.time()
        sentimentModel, history = TrainModel(x_train,
y_train, epochs[i], 32)
        end_time = time.time()
        training_time = end_time - start_time
```

Proses pelatihan akan dimonitor untuk setiap iterasi pelatihan serta run model. Metrik yang dimonitor selama proses pelatihan berlangsung yakni metrik *training loss*, *validation loss*, *training MAE*, serta *validation MAE*. Setelah semua percobaan/run telah berhasil untuk satu konfigurasi, seluruh data metrik akan diplot untuk seluruh run. Berikut blok code penyimpanan metrik pelatihan dan hasil plotnya pada gambar.

```
epoch_finetune_history.append([epochs[i], j,
history_data['loss'], history_data['val_loss'],
history_data['mae'], history_data['val_mae'],
training_time])
```



Gambar 4.1 Contoh hasil metrik pelatihan model

4.4.6 Pengujian dan Evaluasi Model

Setiap kali model dilatih, selanjutnya dilakukan pengujian model menggunakan dengan data yang belum pernah dilihat oleh model sebelumnya.

```
predictions = model.predict(x_test)
predictions
```

```
> 2/2 [==============================] - 2s 22ms/step
> array([[0.26855218],
         [0.29853794],
         ...
         [0.4600806 ],
         [0.4459479 ]], dtype=float32)
```

Dan dikarenakan data yang dihasilkan berupa data yang telah dinormalisasikan, maka diperlukan post-process data hasil prediksi dengan scaler yang telah digunakan untuk mereshape dataset sebelumnya.

```
predictions = scaler.inverse_transform(predictions)
predictions
```

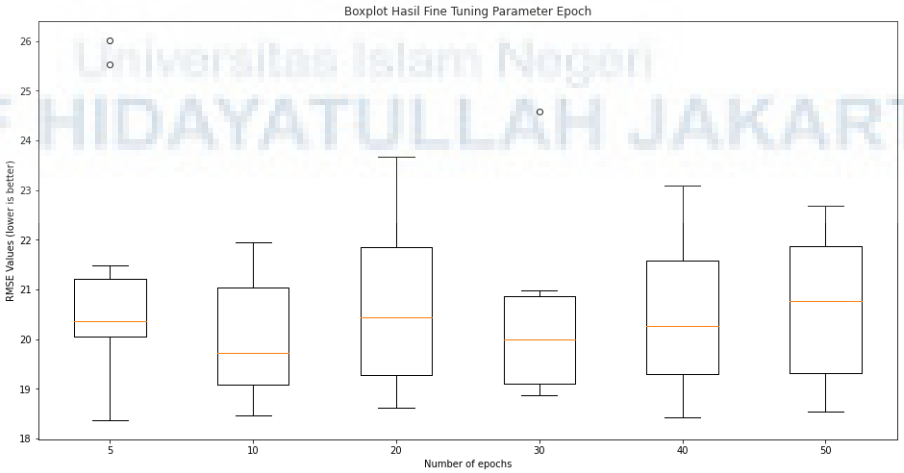
```
> array([[263.25375],
       [269.2684 ],
       ...
```

```
[301.67117],
[298.83636]], dtype=float32)
```

Setelah data direscale kembali ke skala semula, maka dapat dihitung setiap metrik yang digunakan terhadap data hasil prediksi, yang akan dibandingkan dengan nilai harga saham sebenarnya. Metrik yang digunakan di antaranya mean absolute error, mean squared error, root mean squared error, dan mean absolute percentage error.

```
mae = np.mean(np.abs(predictions - y_test))
mse = np.mean((predictions - y_test)**2)
rmse = np.sqrt(mse)
mape = np.mean(np.abs((y_test - predictions)/y_test)) *
100

epoch_finetune_test.append([epochs[i], j, mae, mse,
rmse, mape, training_time])
```

Data hasil pengujian setiap konfigurasi hyperparameter selanjutnya dapat dibuat box-plot untuk melihat sebaran data dan mendeteksi outlier. Dan apabila diperlukan, seluruh data hasil pelatihan dan pengujian dapat disimpan kedalam dokumen berbentuk csv. Berikut contoh gambar box-plot untuk salah-satu hyperparameter yang dites.



Gambar 4.2 Contoh metrik hasil pengujian model

Hasil pengujian model-model yang telah dilatih tersebut akan menjadi bagian penting dalam analisis yang dilakukan pada Bab V. Dalam bab tersebut, model-model yang telah dilatih akan dievaluasi secara mendalam dengan mempertimbangkan metrik evaluasi kinerja seperti *mean absolute error*, *mean squared error*, *root mean squared error*, dan *mean absolute percentage error*. Analisis ini bertujuan untuk mengukur sejauh mana model-model yang dikembangkan berhasil dalam melakukan prediksi harga saham menggunakan bantuan data sentimen pengguna Twitter.

BAB V

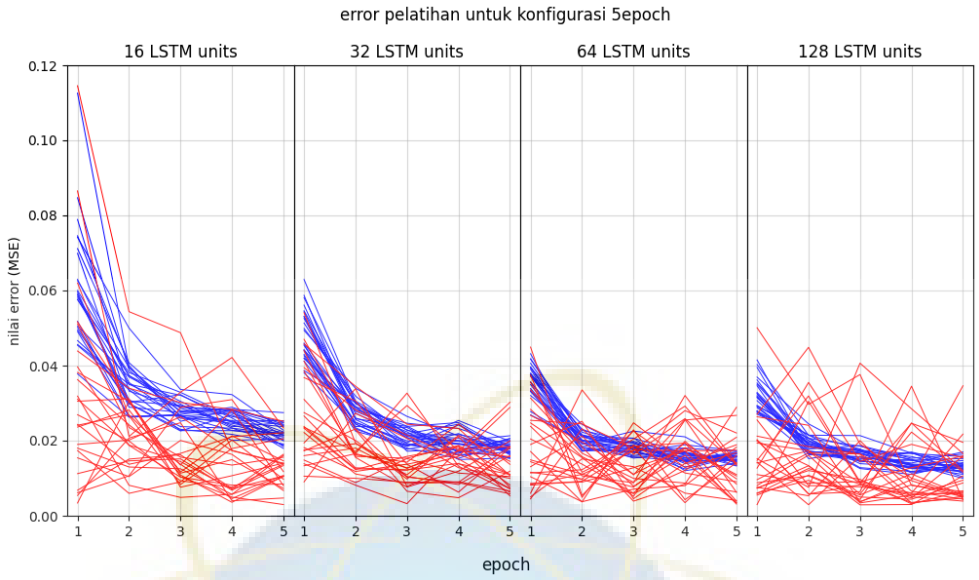
HASIL DAN PEMBAHASAN

Pada bab ini penulis akan menganalisis hasil implementasi yang telah dibangun sebelumnya. Terdapat tiga poin analisis yang dapat diambil dari penelitian ini. Analisis pertama mengacu terhadap dataset yang digunakan dan korelasi antara dua dataset. Analisis kedua dilakukan berdasarkan konfigurasi dan fine-tuning hyperparameter yang diaplikasikan pada model. Selanjutnya analisis ketiga bertujuan untuk membandingkan kinerja model saat digunakan dataset gabungan dan dengan dataset saham murni. Analisis dilakukan berdasarkan data grafik hasil pelatihan dan evaluasi. Kinerja setiap model dan konfigurasinya juga dijelaskan dalam metrik performa seperti MAE, MSE, RMSE, dan MAPE.

5.1 Analisis Hasil Fine Tuning Hyperparameter Model LSTM

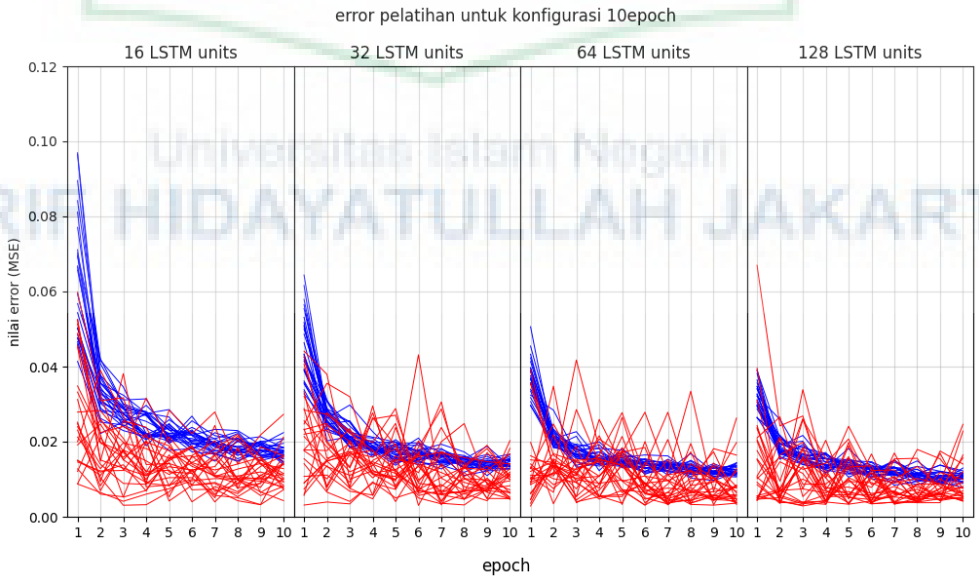
Berikut adalah plot hasil fine-tuning parameter epoch dari konfigurasi yang telah disebutkan. Setiap parameter yang dites dijalankan sebanyak 10 kali untuk mendapatkan sebaran nilai loss dan validasi setiap parameter. Ini berguna untuk mencari outlier dan menentukan seberapa besar sebaran nilai yang mungkin dapat terjadi saat melatih model. Nilai loss untuk pelatihan digambarkan dalam garis biru, dan nilai loss untuk validasi digambarkan dengan garis merah.

Dari keenam jenis parameter epoch yang di-finetune, dengan masing-masing parameter dilakukan pengetesan ulang 25 kali, dihasilkan plot nilai loss setiap parameter.



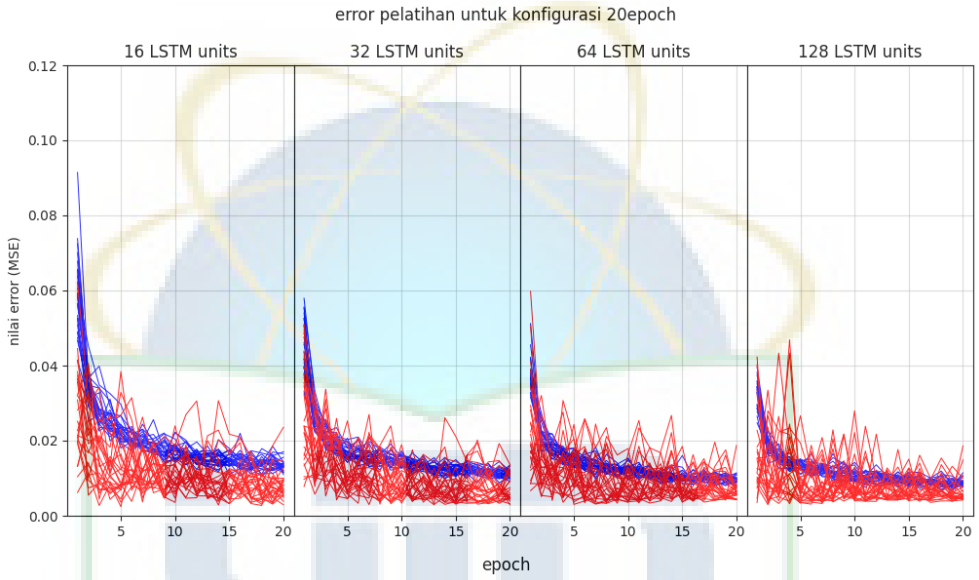
Gambar 5.1 Error pelatihan konfigurasi 5 epoch

Dari graf eror pelatihan untuk konfigurasi 5 epoch di atas, terlihat bahwa semakin banyak jumlah unit LSTM maka nilai erornya juga semakin berkurang, yang berarti jumlah unit LSTM berbanding terbalik dengan nilai error dalam pelatihan. Namun terlihat juga bahwa nilai error validasi juga semakin *noisy* seiring meningkatnya jumlah LSTM.



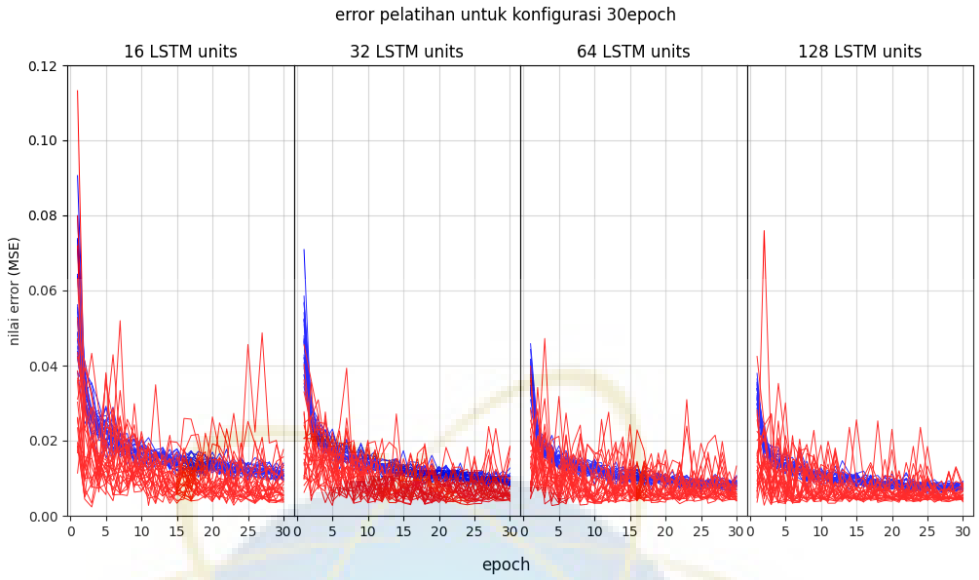
Gambar 5.2 Error pelatihan konfigurasi 10 epoch

Pada konfigurasi 10 epoch terlihat penurunan yang signifikan terhadap nilai error pelatihan pada hasil akhir model dibandingkan dengan konfigurasi 5 epoch, namun terlihat juga nilai error validasi semakin *noisy*, di mana faktornya kemungkinan besar data awal yang kurang bersih, dan juga mengindikasikan overfitting mulai muncul.



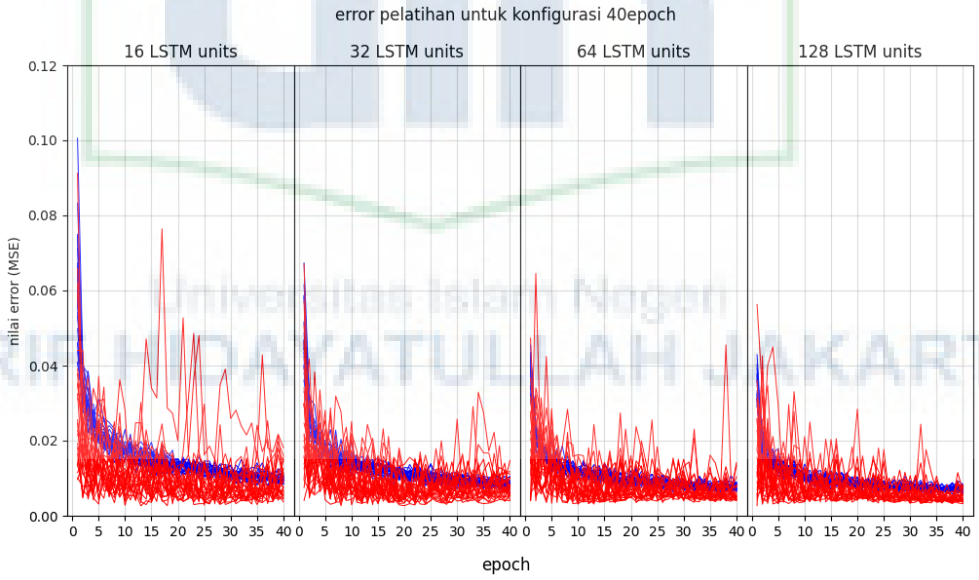
Gambar 5.3 Error pelatihan konfigurasi 20 epoch

Di konfigurasi 20 epoch dan 128 unit LSTM terlihat bahwa noise dari nilai error validasinya lebih tinggi daripada konfigurasi-konfigurasi lainnya. Mengikuti konfigurasi lainnya, tren error pelatihan semakin menurun seiring bertambahnya jumlah unit LSTM dan perulangan epoch.



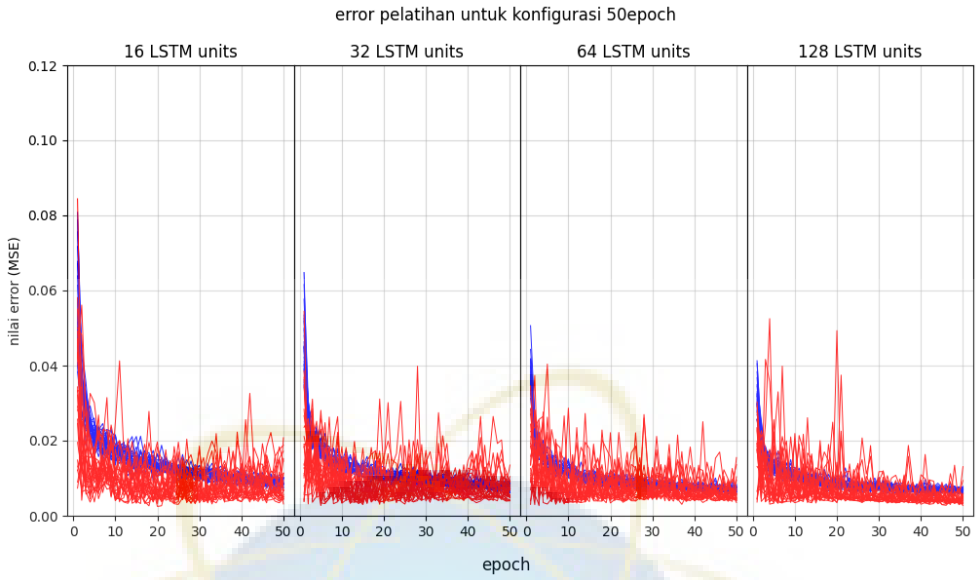
Gambar 5.4 Error pelatihan konfigurasi 30 epoch

Di konfigurasi 30 epoch juga menunjukkan nilai error validasi yang semakin acak dan random, meskipun nilai eror pelatihannya semakin menurun.



Gambar 5.5 Error pelatihan konfigurasi 40 epoch

Pada konfigurasi 40 epoch, terdapat *spike* nilai error validasi pada model yang mengalami overfitting parah di model dengan 16 unit LSTM.



Gambar 5.6 Error pelatihan konfigurasi 50 epoch

Pada konfigurasi terakhir yakni dengan epoch sebanyak 50 kali menunjukkan bahwa terdapat gejala beberapa model yang mengalami overfitting, sehingga terjadi spike nilai validasi pada grafik. Nilai error validasi juga relatif sama di antara seluruh konfigurasi, meskipun nilai error pelatihan semakin menurun dengan bertambahnya jumlah epoch dan unit LSTM.

Tabel berikut menggambarkan rata-rata nilai metrik performa dari 25 pengujian setiap konfigurasi grid search. Metrik yang digunakan yakni *mean absolute error* (MAE), *mean squared error* (MSE), *root mean squared error* (RMSE), dan *mean absolute percentage error* (RMSE). Apabila model semakin bagus kinerjanya, maka nilai metrik akan semakin kecil.

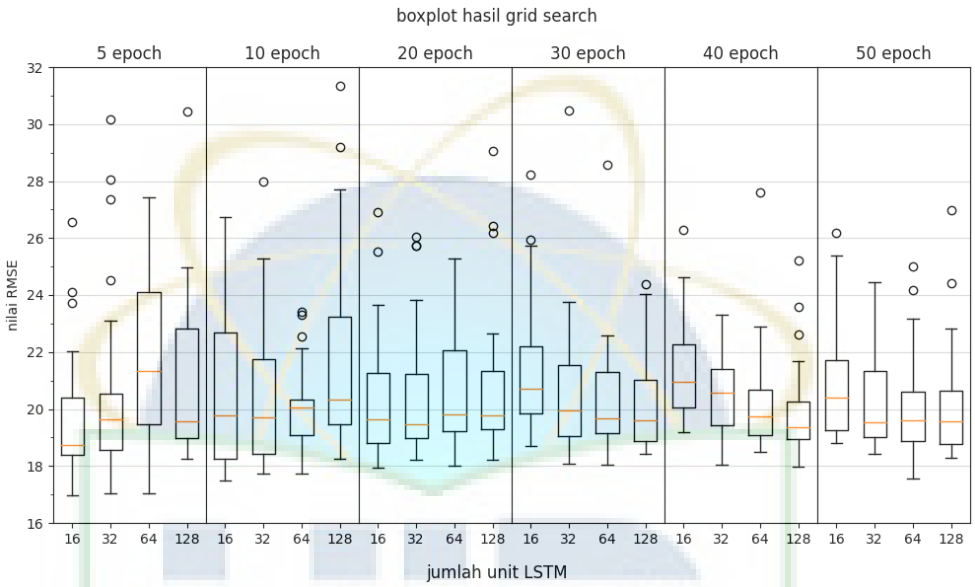
Tabel 5.1 Hasil Finetuning Parameter Epoch

No	Epoch	LSTM Units	MAE	MSE	RMSE	MAPE (%)	Train Time (s)
1	5	16	15.85	393.68	19.71	5.54	17.25

2	5	32	16.73	434.71	20.58	5.85	18.98
3	5	64	17.62	477.37	21.63	6.16	20.84
4	5	128	16.82	439.26	20.77	5.88	21.7
5	10	16	16.53	424.47	20.44	5.79	32.86
6	10	32	16.57	427.32	20.5	5.8	34.66
7	10	64	16.09	405.29	20.07	5.62	35.21
8	10	128	17.87	491.95	21.91	6.27	39.84
9	20	16	16.33	416.17	20.28	5.71	65.34
10	20	32	16.61	430.39	20.61	5.81	61.25
11	20	64	16.59	429.37	20.61	5.79	69.17
12	20	128	16.82	441.18	20.84	5.87	75.84
13	30	16	17.25	459.52	21.31	6.04	83.23
14	30	32	16.69	435.92	20.73	5.83	93.52
15	30	64	16.45	421.3	20.42	5.75	108.84
16	30	128	16.34	413.06	20.24	5.72	112.03
17	40	16	17.3	459.21	21.36	6.06	118.83
18	40	32	16.47	420.38	20.46	5.75	118.99
19	40	64	16.2	410.28	20.17	5.64	126.34
20	40	128	16.08	401.07	19.95	5.61	141.53
21	50	16	17.05	448.72	21.08	5.97	138.54
22	50	32	16.27	410.49	20.2	5.68	141.62
23	50	64	16.11	403.57	20.01	5.62	164
24	50	128	16.27	412.24	20.2	5.66	178.64

Terlihat dari tabel bahwa performa model dengan konfigurasi finetuning epoch sebanyak 5 kali dan jumlah LSTM sebanyak 16 unit mendapatkan rata-rata nilai keempat metrik yang paling bagus, yakni nilai *mean absolute error* sebanyak 15.85 poin, *mean squared error* sebanyak 393.68 poin, *root mean squared error* sebanyak 19.71 poin, dan nilai *mean absolute percentage error* sebanyak 5.54%. Sedangkan konfigurasi yang memiliki kinerja paling buruk

yakni konfigurasi 10 epoch dan 128 unit LSTM, dengan nilai *mean absolute error* sebanyak 17.87 poin, *mean squared error* sebanyak 491.95 poin, *root mean squared error* sebanyak 21.91 poin, dan nilai *mean absolute percentage error* sebanyak 6.27%.



Gambar 5.7 Box plot sebaran hasil grid search

Diagram di atas menunjukkan boxplot yang menunjukkan sebaran hasil RMSE dari 24 konfigurasi yang dilakukan sebanyak masing-masing 25 kali. Dari diagram di atas terlihat bahwa konfigurasi 10 epoch dan 16 unit LSTM memiliki nilai median (garis merah setiap box) paling rendah, sebesar 18.74 dan dengan sebaran hasil RMSE yang cukup baik dibanding konfigurasi-konfigurasi yang lain. Meskipun demikian, banyak konfigurasi yang berkecenderungan memiliki outlier.

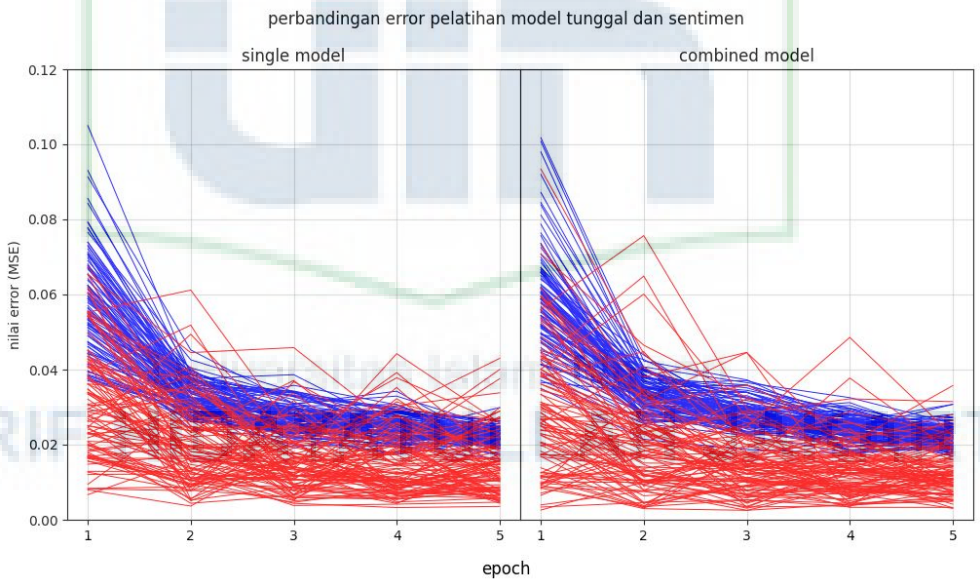
Terlihat juga dari boxplot di atas bahwa model-model dengan konfigurasi epoch kecil seperti 5 dan 10 epoch memiliki sebaran hasil pengujian yang lebih besar, sehingga berpotensi menghasilkan model yang bervariasi

akurasinya. Namun, model-model dengan konfigurasi epoch lebih banyak cenderung memiliki rataan dan median yang lebih tinggi.

Oleh karena itu dari seluruh konfigurasi yang telah dibuat, maka model dengan hasil terbaik dari finetuning hyperparameter ini yakni model dengan jumlah epoch 5, dan jumlah neuron di LSTM sebanyak 16 unit.

5.2 Analisis Pengujian Pengaruh Sentimen terhadap Model Prediksi

Berikut hasil pengujian model yang telah dioptimalisasi menggunakan *hyperparameter* epoch=5 dan neuron=16 terhadap dua skenario perbandingan, yakni digunakan kedua dataset saham dan sentimen sebagai feature set model, dan digunakan hanya dataset saham saja. Pelatihan dilakukan sebanyak 100 kali untuk setiap skenario.



Gambar 5.8 Error pelatihan dua dataset yang dibandingkan

Gambar di atas menyajikan perbandingan terhadap parameter error (MSE) saat pelatihan model. Dari gambar tersebut terlihat bahwa kedua nilai error pelatihan dan validasi cenderung mirip, namun nilai error validasi pada

model dengan dataset kombinasi memiliki lebih banyak *noise* dan *error spike*, yang menandakan bahwa kemungkinan besar dataset *Twitter* yang digunakan belum bersih.

Selanjutnya yakni data mengenai hasil uji dari seluruh pelatihan. Berikut adalah tabel hasil rata-rata metrik pengujian setiap skenario.

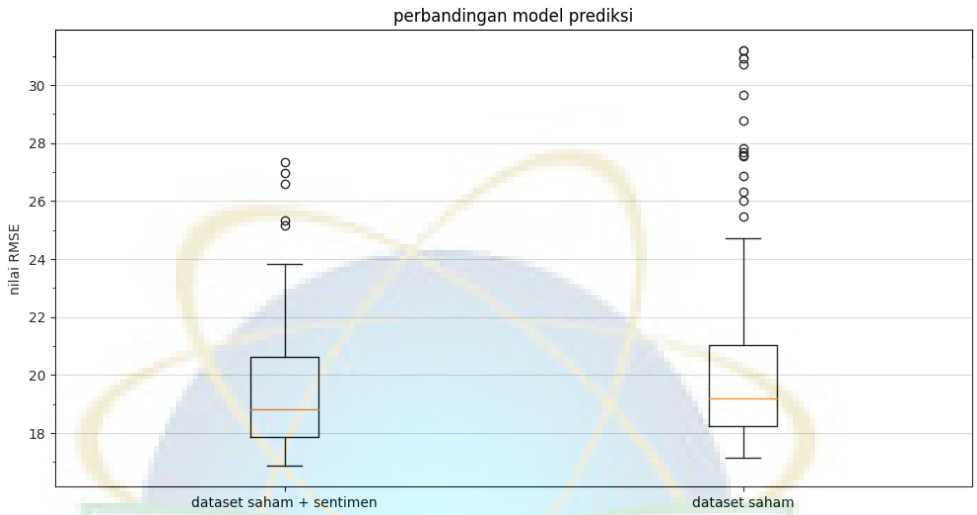
Tabel 5.2 Perbandingan hasil uji model terhadap dua skenario dataset

Skenario	Epoch	LSTM Units	Dataset	MAE	MSE	RMSE	MAPE
1	5	16	Saham	16.71	435.23	20.56	5.83%
2	5	16	Gabungan	15.75	387.15	19.54	5.48%

Dari tabel di atas, didapatkan hasil bahwa dataset gabungan (saham dan sentimen) memiliki keunggulan kecil dalam semua metrik dibandingkan dataset murni saham. Rincian peningkatannya yakni seperti berikut:

- Pada metrik *mean absolute error*, nilainya turun dari **16.71** pada dataset tunggal menjadi **15.75** pada dataset gabungan, dengan peningkatan metrik sebanyak **5.75%** dari nilai MAPE tunggal.
- Di metrik *mean squared error*, nilainya turun dari **435.23** menjadi **387.15**, dengan penurunan sebanyak **11.05%**.
- Metrik *root mean squared error* mengalami penurunan dari **20.56** menjadi **19.54**, yang berarti nilainya turun sebanyak **4.96%**.
- Metrik *mean absolute percentage error* menghitung persentasi deviasi nilai prediksi dari nilai sebenarnya, mengalami penurunan error rate dari **5.83%** menjadi **5.48%**.

Selanjutnya sebaran hasil pengujian data dapat ditampilkan, berikut boxplot dari 100 run pelatihan dan uji model terhadap kedua skenario dataset yang diberikan.



Gambar 5.9 Boxplot perbandingan skenario dataset

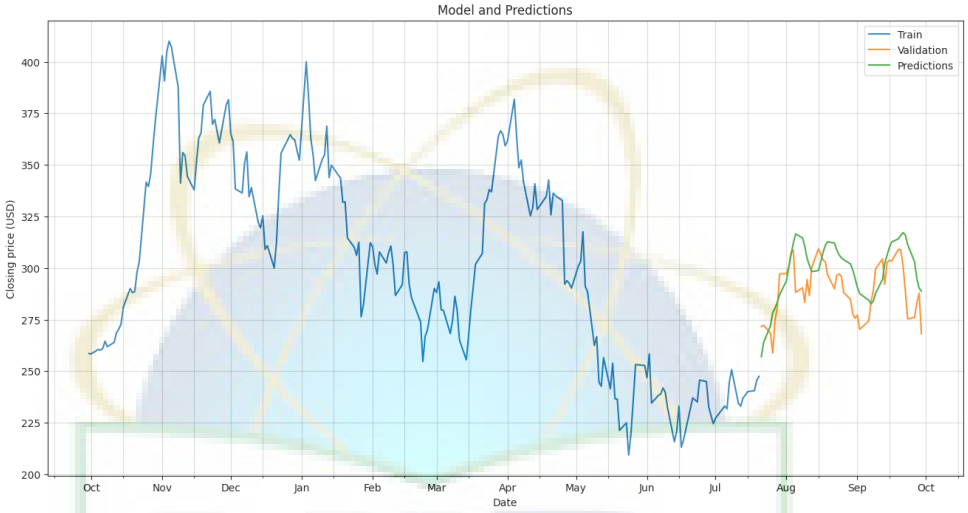
Boxplot yang didapatkan dari 100 kali sesi pelatihan menghasilkan wawasan bahwa tidak hanya nilai metrik rata-ratanya mengalami peningkatan, namun juga hasil model yang dilatih dengan dataset gabungan memiliki sebaran yang lebih kecil, yang dibuktikan dengan nilai median serta kuartil atas yang lebih rendah ketimbang model dengan dataset tunggal. Model dengan dataset gabungan juga memiliki lebih sedikit outlier ketimbang dataset tunggal.

Berikut salah satu contoh data dan plot graph yang dapat membandingkan kedua hasil model.

Tabel 5.3 Data harga closing dan prediksi pada dataset tunggal

Date	Close	Predicted
2022-07-21	271.706665	257.101074
2022-07-22	272.243347	263.904877

2022-07-25	268.433319	271.992340
...	...	...
2022-09-28	287.809998	290.218323
2022-09-29	268.209991	288.827271



Gambar 5.10 Plot grafik prediksi dataset tunggal

Pada gambar di atas, garis biru merupakan data training, garis oranye merupakan data testing (data harga saham sebenarnya), sedangkan garis hijau merupakan prediksi dari model yang diberikan dataset saham saja. Dapat dilihat dari gambar di atas, bentuk graph prediksi model cukup mengikuti data training, namun memiliki bias lebih tinggi dari nilai sebenarnya.

Berikut adalah rekapitulasi data prediksi dibandingkan data sebenarnya. Data prediksi yang digunakan diambil dari model yang diberikan data saham dan sentimen.

Tabel 5.4 Analisis data prediksi dengan dataset gabungan

Tanggal	**Close**	**Predict**	**Diff**	**% Diff**	**Daily Returns**	**Predict Daily Returns**	**% Diff Daily Returns**

7/21/2022	271.71	247.29	24.42	8.99%			
7/22/2022	272.24	253.76	18.48	6.79%	0.00198	0.02619	1226.01%
7/25/2022	268.43	262.67	5.76	2.15%	-0.01399	0.03510	350.82%
7/26/2022	258.86	270.37	11.51	4.45%	-0.03566	0.02932	182.22%
7/27/2022	274.82	271.85	2.97	1.08%	0.06166	0.00546	91.15%
7/28/2022	280.90	274.58	6.32	2.25%	0.02212	0.01004	54.63%
7/29/2022	297.15	280.13	17.02	5.73%	0.05785	0.02023	65.04%
8/1/2022	297.28	287.28	10.00	3.36%	0.00043	0.02551	5883.97%
8/2/2022	300.59	293.25	7.34	2.44%	0.01113	0.02077	86.57%
8/3/2022	307.40	299.81	7.59	2.47%	0.02266	0.02239	1.17%
8/4/2022	308.63	305.50	3.13	1.02%	0.00402	0.01897	371.55%
8/5/2022	288.17	308.80	20.63	7.16%	-0.06630	0.01079	116.27%
8/8/2022	290.42	307.28	16.85	5.80%	0.00782	-0.00492	162.95%
8/9/2022	283.33	303.79	20.46	7.22%	-0.02441	-0.01135	53.52%
8/10/2022	294.36	297.33	2.97	1.01%	0.03891	-0.02126	154.65%
8/11/2022	286.63	294.47	7.84	2.74%	-0.02625	-0.00960	63.42%
8/12/2022	300.03	291.60	8.43	2.81%	0.04675	-0.00977	120.90%
8/15/2022	309.32	291.66	17.66	5.71%	0.03096	0.00021	99.33%
8/16/2022	306.56	294.18	12.38	4.04%	-0.00891	0.00865	197.06%
8/17/2022	304.00	297.42	6.58	2.16%	-0.00837	0.01101	231.53%
8/18/2022	302.87	298.05	4.82	1.59%	-0.00371	0.00211	156.87%
8/19/2022	296.67	298.87	2.20	0.74%	-0.02048	0.00276	113.50%
8/22/2022	289.91	300.48	10.57	3.65%	-0.02276	0.00540	123.72%
8/23/2022	296.45	298.24	1.79	0.60%	0.02256	-0.00746	133.05%
8/24/2022	297.10	295.04	2.05	0.69%	0.00217	-0.01073	594.38%
8/25/2022	296.07	293.10	2.97	1.00%	-0.00346	-0.00659	90.61%
8/26/2022	288.09	292.15	4.06	1.41%	-0.02695	-0.00326	87.92%
8/29/2022	284.82	289.47	4.65	1.63%	-0.01135	-0.00916	19.27%
8/30/2022	277.70	286.72	9.02	3.25%	-0.02500	-0.00950	61.98%
8/31/2022	275.61	283.26	7.65	2.77%	-0.00753	-0.01207	60.37%
9/1/2022	277.16	279.71	2.55	0.92%	0.00562	-0.01253	322.72%
9/2/2022	270.21	276.54	6.33	2.34%	-0.02508	-0.01132	54.86%
9/6/2022	274.42	274.31	0.11	0.04%	0.01558	-0.00808	151.84%
9/7/2022	283.70	271.98	11.72	4.13%	0.03382	-0.00850	125.14%
9/8/2022	289.26	272.26	17.00	5.88%	0.01960	0.00105	94.64%
9/9/2022	299.68	274.65	25.03	8.35%	0.03602	0.00877	75.65%

9/12/2022	304.42	280.70	23.72	7.79%	0.01582	0.02201	39.17%
9/13/2022	292.13	286.70	5.43	1.86%	-0.04037	0.02139	152.99%
9/14/2022	302.61	290.84	11.77	3.89%	0.03587	0.01443	59.77%
9/15/2022	303.75	295.38	8.37	2.75%	0.00377	0.01563	314.76%
9/16/2022	303.35	299.14	4.21	1.39%	-0.00132	0.01273	1066.70%
9/19/2022	309.07	302.40	6.67	2.16%	0.01886	0.01089	42.26%
9/20/2022	308.73	303.99	4.74	1.54%	-0.00110	0.00526	577.76%
9/21/2022	300.80	304.10	3.30	1.10%	-0.02569	0.00035	101.37%
9/22/2022	288.59	302.31	13.72	4.75%	-0.04059	-0.00589	85.49%
9/23/2022	275.33	297.71	22.38	8.13%	-0.04595	-0.01521	66.90%
9/26/2022	276.01	290.62	14.61	5.29%	0.00247	-0.02381	1064.01%
9/27/2022	282.94	283.78	0.84	0.30%	0.02511	-0.02354	193.75%
9/28/2022	287.81	280.05	7.76	2.70%	0.01721	-0.01315	176.39%
9/29/2022	268.21	277.98	9.77	3.64%	-0.06810	-0.00738	89.16%

Tabel diatas memuat data hasil prediksi 50 hari terakhir dataset saham, di mana model menggunakan dataset gabungan. Terlihat bahwa nilai *Closing* tertinggi dari saham Tesla, Inc yakni sebesar 309.32 USD pada tanggal 15 Agustus 2022. Pada hari yang sama, model memprediksikan bahwa harga penutupan saham sebesar 291.66 USD, dengan perbedaan sebesar 5.71% dibanding harga sebenarnya.

Nilai rata-rata persentase error absolut prediksi dari harga penutupan sebenarnya sebesar 0.55%. Dengan kata lain, dalam 50 hari yang diprediksikan oleh model, agregat kesalahan absolute prediksi kurang lebih 3.31%. Range kesalahannya juga cukup baik, dengan kesalahan terburuk sebesar 8.99% pada tanggal 21 Juli 2022, dan kesalahan terbaik sebesar 0.04% pada 6 September 2022.

Daily returns menghitung perubahan harga penutupan (closing price) terhadap harga sebelumnya. Model juga mampu menghasilkan nilai daily returns yang cukup baik, dengan rata-rata agregat kesalahan daily returnsnya sebesar 322.65% dari nilai semula. Kesalahan prediksi daily returns berada pada tanggal 1 Agustus 2022 dengan kesalahan sebesar 5883.97%, sedangkan kesalahan terbaik dengan kisaran 1.17% pada tanggal 3 Agustus 2022.

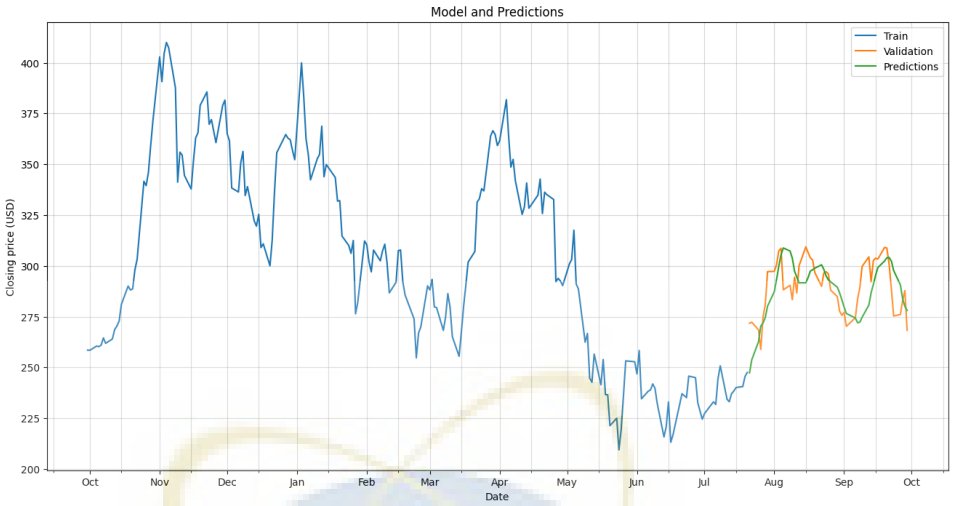
Berikut barplot harga penutupan saham yang diprediksi oleh model, beserta data perubahan tiap harinya.

Tabel 5.5 Data prediksi dan pergerakan harian

Tanggal	Prediksi	Perubahan
7/21/2022	247.29	-0.21▼
7/22/2022	253.76	+6.47▲
7/25/2022	262.67	+8.91▲
7/26/2022	270.37	+7.70▲
7/27/2022	271.85	+1.48▲
7/28/2022	274.58	+2.73▲
7/29/2022	280.13	+5.55▲
8/1/2022	287.28	+7.15▲
8/2/2022	293.25	+5.97▲
8/3/2022	299.81	+6.56▲
8/4/2022	305.50	+5.69▲
8/5/2022	308.80	+3.30▲
8/8/2022	307.28	-1.52▼
8/9/2022	303.79	-3.49▼
8/10/2022	297.33	-6.46▼
8/11/2022	294.47	-2.86▼
8/12/2022	291.60	-2.87▼
8/15/2022	291.66	+0.06▲
8/16/2022	294.18	+2.52▲
8/17/2022	297.42	+3.24▲

Gambar 5.11 Candleplot prediksi harga saham dengan dataset gabungan

8/18/2022	298.05	+0.63▲
8/19/2022	298.87	+0.82▲
8/22/2022	300.48	+1.61▲
8/23/2022	298.24	-2.24▼
8/24/2022	295.04	-3.20▼
8/25/2022	293.10	-1.94▼
8/26/2022	292.15	-0.95▼
8/29/2022	289.47	-2.68▼
8/30/2022	286.72	-2.75▼
8/31/2022	283.26	-3.46▼
9/1/2022	279.71	-3.55▼
9/2/2022	276.54	-3.17▼
9/6/2022	274.31	-2.23▼
9/7/2022	271.98	-2.33▼
9/8/2022	272.26	+0.28▲
9/9/2022	274.65	+2.39▲
9/12/2022	280.70	+6.05▲
9/13/2022	286.70	+6.00▲
9/14/2022	290.84	+4.14▲
9/15/2022	295.38	+4.54▲
9/16/2022	299.14	+3.76▲
9/19/2022	302.40	+3.26▲
9/20/2022	303.99	+1.59▲
9/21/2022	304.10	+0.11▲
9/22/2022	302.31	-1.79▼
9/23/2022	297.71	-4.60▼
9/26/2022	290.62	-7.09▼
9/27/2022	283.78	-6.84▼
9/28/2022	280.05	-3.73▼
9/29/2022	277.98	-2.07▼



Gambar 5.12 Plot grafik prediksi dengan dataset gabungan

Berdasarkan gambar di atas, graph hasil prediksi model gabungan memiliki bentuk yang mirip dengan data training, dan juga memiliki bias yang lebih kecil ketimbang hasil prediksi menggunakan dataset saham tunggal. Oleh karena itu, hasil prediksi model gabungan yakni menggunakan dataset saham dan sentimen lebih akurat daripada dataset saham saja.

Namun, perlu dicatat bahwa hasil prediksi dari model deep learning LSTM ini berfluktuatif dan bergantung dengan nilai random, sehingga pada umumnya tidak dapat mengambil simpulan apakah model yang menggunakan dataset gabungan memiliki keunggulan ketimbang model yang diberi dataset tunggal. Keunggulan kecil baru muncul apabila model dilatih berkali-kali dan dianalisis hasilnya keseluruhan.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan yang telah dilakukan di bab sebelumnya, ada beberapa kesimpulan yang dapat ditarik dari penelitian ini:

1. Penelitian ini menghasilkan sebuah sistem prediksi harga saham dengan model LSTM, yang menggunakan dataset saham dan sentimen dari pengguna Twitter yang dianalisis menggunakan algoritma VADER. Konfigurasi model optimal yang dihasilkan menggunakan *hyperparameter epoch* sebanyak 5, dan jumlah unit LSTM sebanyak 16.
2. Dari pelatihan dan pengujian yang dilakukan, performa model optimal yang dihasilkan memperoleh nilai metrik *mean absolute error* (MAE) sebesar 15.85, nilai *mean squared error* (MSE) sebesar 393.68, nilai metrik *root mean squared error* (RMSE) sebesar 19.71, dan *mean absolute percentage error* (MAPE) sebesar 5.54%.
3. Pengujian model dengan menggabungkan data sentimen menunjukkan pola hasil yang mengungguli model dengan data saham saja, dengan model tersebut mendapatkan peningkatan nilai metrik MAE sebesar 5.75%, MSE sebesar 11.05%, RMSE sebesar 4.96%, dan nilai MAPE yang menurun dari 5.83% menjadi 5.48% berdasarkan rata-rata dari 100 kali pengujian.

6.2 Saran

Dari hasil penelitian yang dilakukan, penulis dapat memberikan saran untuk penelitian selanjutnya, yakni meningkatkan kualitas dataset tweet dengan mengacukan *benchmark* performa pada dataset menggunakan metode analisis sentimen alternatif. Penulis kedepannya juga dapat menggunakan lebih dari satu feature yang diambil dari dataset saham dan tweet untuk digunakan sebagai dataset pelatihan model.

DAFTAR PUSTAKA

Adhikari, R., & Agrawal, R. K. (2013). An Introductory Study on Time Series Modeling and Forecasting. *ArXiv Preprint ArXiv:1302.6613*. https://doi.org/https://doi.org/10.48550/arXiv.1302.6613

Afrianto, N., Fudholi, D. H., & Rani, S. (2022). Prediksi Harga Saham Menggunakan BiLSTM dengan Faktor Sentimen Publik. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *6*(1), 41–46. https://doi.org/10.29207/resti.v6i1.3676

Bell, J. (2014). *Machine Learning: Hands-On for Developers and Technical Professionals* (1st ed., Vol. 1). Wiley.

Bhaya, W. (2017). Review of Data Preprocessing Techniques in Data Mining. *Journal of Engineering and Applied Sciences*, *12*, 4102–4107. https://doi.org/10.3923/jeasci.2017.4102.4107

Borg, A., & Boldt, M. (2020). Using VADER sentiment and SVM for predicting customer response sentiment. *Expert Systems with Applications*, *162*, 113746. https://doi.org/10.1016/j.eswa.2020.113746

Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2017). *A Practical Guide to Sentiment Analysis* (E. Cambria, D. Das, S. Bandyopadhyay, & A. Feraco, Eds.; Vol. 5). Springer International Publishing. https://doi.org/10.1007/978-3-319-55394-8

Carr, C. T., & Hayes, R. A. (2015). Social Media: Defining, Developing, and Divining. *Atlantic Journal of Communication*, *23*(1), 46–65. https://doi.org/10.1080/15456870.2015.972282

Cilimkovic, M. (2015). Neural networks and back propagation algorithm. *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin*, *15*(1).

Ciresan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3642–3649. https://doi.org/10.1109/CVPR.2012.6248110

Cohen-Charash, Y., Scherbaum, C. A., Kammeyer-Mueller, J. D., & Staw, B. M. (2013). Mood and the Market: Can Press Reports of Investors' Mood Predict Stock Prices? *PLoS ONE*, *8*(8), e72031. https://doi.org/10.1371/journal.pone.0072031

Coiera, E. (2013). Social networks, social media, and social diseases. *BMJ*, *346*(may22 16), f3007–f3007. https://doi.org/10.1136/bmj.f3007

Devika, M. D., Sunitha, C., & Ganesh, A. (2016). Sentiment Analysis: A Comparative Study on Different Approaches. *Procedia Computer Science*, *87*, 44–49. https://doi.org/10.1016/j.procs.2016.05.124

Dewantoro, R. (2018). *Prediksi Arah Harga Saham Menggunakan Sentimen Analisis Pada Social Media*. https://doi.org/10.13140/RG.2.2.15084.41607

Ding, B., Qian, H., & Zhou, J. (2018). Activation functions and their characteristics in deep neural networks. *2018 Chinese Control And Decision Conference (CCDC)*, 1836–1841. https://doi.org/10.1109/CCDC.2018.8407425

Doksum, K. A., & Bickel, P. J. (2015). *Mathematical Statistics: Basic Ideas and Selected Topics* (2nd ed., Vol. 1).

El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In *Machine Learning in Radiation Oncology* (pp. 3–11). Springer International Publishing. https://doi.org/10.1007/978-3-319-18305-3_1

Gandhmal, D. P., & Kumar, K. (2019). Systematic analysis and review of stock market prediction techniques. *Computer Science Review*, *34*, 100190. https://doi.org/10.1016/j.cosrev.2019.08.001

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/10.1016/j.eswa.2018.04.006

Gligorić, K., Anderson, A., & West, R. (2018). How Constraints Affect Content: The Case of Twitter's Switch from 140 to 280 Characters. *Proceedings of the International AAAI Conference on Web and Social Media*, *12*(1). https://doi.org/10.1609/icwsm.v12i1.15079

Gondaliya, C., Patel, A., & Shah, T. (2021). Sentiment analysis and prediction of Indian stock market amid Covid-19 pandemic. *IOP Conference Series: Materials Science and Engineering*, *1020*(1), 012023. https://doi.org/10.1088/1757-899X/1020/1/012023

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. https://doi.org/10.1016/j.neucom.2015.09.116

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hutto, C., & Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *Proceedings of the International AAAI Conference on Web and Social Media*, *8*(1), 216–225. https://doi.org/10.1609/icwsm.v8i1.14550

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications*, *184*, 115537. https://doi.org/10.1016/j.eswa.2021.115537

Jin, Z., Yang, Y., & Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and LSTM. *Neural Computing and Applications*, *32*(13), 9713–9729. https://doi.org/10.1007/s00521-019-04504-2

Jing, N., Wu, Z., & Wang, H. (2021). A hybrid model integrating deep learning with investor sentiment analysis for stock price prediction. *Expert Systems with Applications*, *178*, 115019. https://doi.org/10.1016/j.eswa.2021.115019

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons*, *53*(1), 59–68. https://doi.org/10.1016/j.bushor.2009.09.003

Karim, M., & Das, S. (2018). Sentiment Analysis on Textual Reviews. *IOP Conference Series: Materials Science and Engineering*, *396*, 012020. https://doi.org/10.1088/1757-899X/396/1/012020

Kedar, S. V. (2021). Stock Market Increase and Decrease using Twitter Sentiment Analysis and ARIMA Model. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(1S), 146–161. https://doi.org/10.17762/turcomat.v12i1S.1596

Kemp, S. (2023a, January). *Digital 2023: Global Overview Report*. Data Reportal. https://datareportal.com/reports/digital-2023-global-overview-report

Kemp, S. (2023b, April). *Twitter Users, Stats, Data & Trends*. Data Reportal. https://datareportal.com/essential-twitter-stats

Koukaras, P., Nousi, C., & Tjortjis, C. (2022). Stock Market Prediction Using Microblogging Sentiment Analysis and Machine Learning. *Telecom*, *3*(2), 358–378. https://doi.org/10.3390/telecom3020019

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In *Artificial Intelligence in Design '96* (pp. 151–170). Springer Netherlands. https://doi.org/10.1007/978-94-009-0279-4_9

Kwak, H., Lee, C., Park, H., & Moon, S. (2010). What is Twitter, a social network or a news media? *Proceedings of the 19th International Conference on World Wide Web*, 591–600. https://doi.org/10.1145/1772690.1772751

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, C.-C., Gao, Z., & Tsai, C.-L. (2020). BERT-Based Stock Market Sentiment Analysis. *2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, 1–2. https://doi.org/10.1109/ICCE-Taiwan49838.2020.9258102

Lengkong, N. C., Safitri, O., Machsus, S., Putra, Y. R., Syahadati, A., & Nooraeni, R. (2021). ANALISIS SENTIMEN PENERAPAN PSBB DI DKI JAKARTA DAN DAMPAKNYA TERHADAP PERGERAKAN IHSG. *Jurnal Teknoinfo*, *15*(1), 20. https://doi.org/10.33365/jti.v15i1.866

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

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/10.30865/mib.v3i4.1283

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), 101–119. https://doi.org/10.1016/j.jfds.2017.11.002

Putra, J. W. G. (2020). *Pengenalan Pembelajaran Mesin dan Deep Learning*. https://www.researchgate.net/publication/323700644_Pengenalan_Pembelajaran_Mesin_dan_Deep_Learning

Pyeong Kang Kim, D., Lee, J., Lee, J., & Suh, J. (2021). Elon Musk's Twitter and Its Correlation with Tesla's Stock Market. *International Journal of Data Science and Analysis*, *7*(1), 13. https://doi.org/10.11648/j.ijdsa.20210701.14

Qi, Y., & Shabrina, Z. (2023). Sentiment analysis using Twitter data: a comparative application of lexicon- and machine-learning-based approach. *Social Network Analysis and Mining*, *13*(1), 31. https://doi.org/10.1007/s13278-023-01030-x

Ren, R., Wu, D. D., & Liu, T. (2019). Forecasting Stock Market Movement Direction Using Sentiment Analysis and Support Vector Machine. *IEEE Systems Journal*, *13*(1), 760–770. https://doi.org/10.1109/JSYST.2018.2794462

Ribeiro, F. N., Araújo, M., Gonçalves, P., André Gonçalves, M., & Benevenuto, F. (2016). SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, *5*(1), 23. https://doi.org/10.1140/epjds/s13688-016-0085-1

Russell, S., & Norvig, P. (2022). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Sagala, T. W., Saputri, M. S., Mahendra, R., & Budi, I. (2020). Stock Price Movement Prediction Using Technical Analysis and Sentiment Analysis. *Proceedings of the 2020 2nd Asia Pacific Information Technology Conference*, 123–127. https://doi.org/10.1145/3379310.3381045

Seedhouse, E. (2013). Elon Musk: The space industry's Tony Stark. In *SpaceX* (pp. 1–15). Springer New York. https://doi.org/10.1007/978-1-4614-5514-1_1

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, *6*(12), 310–316.

Shi, Y., Zheng, Y., Guo, K., & Ren, X. (2021). Stock movement prediction with sentiment analysis based on deep learning networks. *Concurrency and Computation: Practice and Experience*, *33*(6). https://doi.org/10.1002/cpe.6076

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sugiyono. (2008). *Metode penelitian pendidikan : (pendekatan kuantitatif, kualitatif dan R & D)* (6th ed.). Alfabeta.

Sulia. (2017). Analisis Faktor-Faktor yang Mempengaruhi Harga Saham pada Perusahaan yang Terdaftar di Bursa Efek Indonesia. *Jurnal Wira Ekonomi Mikroskil*, *7*(2), 129–140.

Tarczyński, W., Mentel, U., Mentel, G., & Shahzad, U. (2021). The Influence of Investors' Mood on the Stock Prices: Evidence from Energy Firms in Warsaw Stock Exchange, Poland. *Energies*, *14*(21), 7396. https://doi.org/10.3390/en14217396

Teti, E., Dallocchio, M., & Aniasi, A. (2019). The relationship between twitter and stock prices. Evidence from the US technology industry. *Technological Forecasting and Social Change*, *149*, 119747. https://doi.org/10.1016/j.techfore.2019.119747

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Vasilev, I., Slater, D., Spacagna, G., Roelants, P., & Zocca, V. (2019). *Python deep learning: Exploring deep learning techniques and neural network architectures with pytorch, Keras, and tensorflow*. Packt Publishing.

Verma, B., & Thakur, R. S. (2018). *Sentiment Analysis Using Lexicon and Machine Learning-Based Approaches: A Survey* (pp. 441–447). https://doi.org/10.1007/978-981-10-8198-9_46

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. https://doi.org/10.24014/ijaidm.v3i1.8245

Willmott, C., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, *30*, 79–82. https://doi.org/10.3354/cr030079

Zhang, Y. (2010). *New Advances in Machine Learning* (Y. Zhang, Ed.). InTech.

Zou, J., Han, Y., & So, S.-S. (2008). *Overview of Artificial Neural Networks* (pp. 14–22). https://doi.org/10.1007/978-1-60327-101-1_2