

SISTEM ANALISA HARGA SAHAM MENGGUNAKAN ALGORITMA *LONG SHORT TERM MEMORY (LSTM)*

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan
Program Pendidikan Sarjana

Oleh:

Carel Anthony Filemon Patra
2019130033



JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER – LIKMI
BANDUNG
2022

SISTEM ANALISA HARGA SAHAM MENGGUNAKAN ALGORITMA *LONG SHORT TERM MEMORY (LSTM)*

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Kelulusan
Program Pendidikan Sarjana

Oleh:

Carel Anthony Filemon Patra
2019130033

Bandung, 31 Agustus 2022
Menyetujui,

Dhanny Setiawan, S.T., M.T.
Pembimbing

Dhanny Setiawan, S.T., M.T.
Ketua Jurusan

JURUSAN INFORMATIKA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA & KOMPUTER – LIKMI
BANDUNG
2022

ABSTRAK

Saham adalah modal dan bukti kepemilikan yang ditanamkan seseorang pada suatu badan usaha atau perseroan terbatas. Saham sendiri memiliki tujuan untuk menambah modal suatu perusahaan dari publik. Jenis saham terbagi menjadi 2 yakni saham biasa dan saham preferen. Berinvestasi saham kini semakin populer dan berkembang pesat di Indonesia, dengan berinvestasi saham, investor akan mendapatkan keuntungan yang cukup cepat dan besar dalam waktu yang cukup singkat, dalam beberapa hari, minggu, bulan atau tahun dari menginvestasikan dananya. Investasi juga memiliki resiko yang cukup besar, untuk mengantisipasinya, investor wajib memiliki bekal ilmu analisa yang baik, terdapat 2 analisa yang banyak dipakai oleh para investor mulai yakni analisa fundamental dan teknikal.

Forecasting adalah proses memprediksi peristiwa masa depan dengan memperoleh data historis dan merencanakan masa depan. *Forecasting* adalah alat yang akan membantu pengambil keputusan dalam penyusunan rencana nyata untuk masa depan. *Forecasting* banyak di antaranya dapat digunakan, terutama di bidang ekonomi, bisnis, dan pemasaran, mulai dari memprediksi produk masa depan, membantu perusahaan memprediksi kinerja perusahaan, dan membantu mereka membuat keputusan bisnis.

Long Short Term Memory (LSTM) berasal dari metode *RNN (Recurrent Neural Network)*, yang dirancang khusus untuk memproses Data Sequence. Namun, RNN memiliki masalah yaitu jika terjadi perubahan range nilai dari satu level ke level berikutnya dalam arsitektur. LSTM dirancang dan dirancang untuk mengatasi masalah ini.

Hasil analisis menunjukkan bahwa sistem dapat memprediksi harga saham dengan angka keakuratan prediksi tertinggi yaitu 99%.

Kata kunci : Saham, *Forecasting*, *Long Short Term Memory*

KATA PENGANTAR

Pertama-tama panjatkan Puji dan Syukur kepada Tuhan Yesus karena anugerah dan kasih-Nya yang telah membimbing dan menuntun penulis dari awal masa perkuliahan hingga penyusunan tugas akhir ini. Begitu banyak ilmu, pengalaman, dan pembelajaran berharga yang penulis dapatkan selama masa perkuliahan di STMIK LIKMI. Penghargaan dan terima kasih diberikan terutama pada keluarga, terutama kedua orang tua terkasih yang selalu memberikan dukungan, semangat, doa, dan saran sehingga penulisan tugas akhir ini dapat berjalan dengan baik.

Penyusunan tugas akhir ini juga tidak lepas dari bantuan berbagai pihak, karena itu penulis mengucapkan terima kasih kepada :

1. Pak Dhanny Setiawan, S.T.,M.T. selaku pembimbing yang telah bersedia meluangkan waktu, pikiran, dan tenaga untuk membimbing dan memberikan masukan dan saran serta dorongan demi kelancaran penulisan tugas akhir ini.
2. Kepada teman-teman seperjuangan di masa perkuliahan angkatan 2019, khususnya teman dekat yang dibanggakan yaitu Nathanael Mario, Kevin Gideon, Charles Sebastian, Serious Harefa, Delfiero Matius yang telah mendukung dan memberikan semangat selama pengerjaan tugas akhir ini dan pada saat masa perkuliahan. Kepada teman-teman masa kecil yang saya kasihi yaitu Venanceus Fortunatus dan Erik Mario. Kepada kekasih yaitu Debby Margarettha yang selalu memberikan semangat, saran, dukungan moral, dan doa selama masa perkuliahan dan saat menulis tugas akhir ini.
3. Kepada seluruh dosen dan staff STMIK LIKMI, serta semua pihak yang tidak dapat disebutkan satu persatu, atas ilmu dan bantuannya selama masa perkuliahan.

Bandung, 31 Agustus 2022

Penulis

DAFTAR ISI

ABSTRAK	iii
KATA PENGANTAR	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
DAFTAR SIMBOL	x
DAFTAR LAMPIRAN	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Maksud dan Tujuan Penelitian	3
1.5 Kegunaan dan Hasil	4
1.6 Sistematika Penulisan	4
BAB II LANDASAN TEORI	5
2.1 Rekayasa Perangkat Lunak	5
2.2 Metodologi Penelitian	6
2.3 Model Pengembangan	7
2.4 Teori Khusus	8
2.5 Perangkat Lunak yang Menjadi Alat Bantu Perancangan Perangkat Lunak ...	13
BAB III ANALISIS DAN PERANCANGAN PERANGKAT LUNAK	22
3.1 Gambaran Umum	23
3.2 Spesifikasi Kebutuhan Perangkat Lunak	23
3.2.1 Kebutuhan Fungsional	24
3.2.2 Kebutuhan Non Fungsional	24
3.3 Diagram-Diagram Perancangan Perangkat Lunak	24
3.3.1 Use Case Diagram	24
3.3.2 Skenario Use Case	25
3.3.3 Activity Diagram	26
3.3.4 Class Diagram	27
3.3.5 Sequence Diagram	28
BAB IV HASIL DAN PEMBAHASAN	31
4.1 Spesifikasi Perangkat Lunak	30

4.2	Menganalisis Pengujian Analisa Harga Saham.....	30
BAB V KESIMPULAN DAN SARAN		41
5.1	Kesimpulan	41
5.2	Saran.....	41
DAFTAR PUSTAKA.....		42
LAMPIRAN.....		45

DAFTAR GAMBAR

Gambar 3. 1 <i>Use Case Diagram</i> sistem analisa harga saham menggunakan algoritma <i>Long Short Term Memory (LSTM)</i>	24
Gambar 3. 2 <i>Activity Diagram</i> sistem analisa harga saham menggunakan algoritma <i>Long Short Term Memory (LSTM)</i>	26
Gambar 3. 3 <i>Class Diagram</i> sistem analisa harga saham menggunakan algoritma <i>Long Short Term Memory (LSTM)</i>	28
Gambar 3. 4 <i>Sequence Diagram</i> sistem analisa harga saham menggunakan algoritma <i>Long Short Term Memory (LSTM)</i>	29
Gambar 4. 1 Grafik Data <i>Real</i> Harga <i>Close</i> Saham Tesla, Inc	31
Gambar 4. 2 Grafik Prediksi dengan 60% <i>Data Training</i> dan 40% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 10.....	32
Gambar 4. 3 Grafik Prediksi dengan 60% <i>Data Training</i> dan 40% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 20.....	33
Gambar 4. 4 Grafik Prediksi dengan 60% <i>Data Training</i> dan 40% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 30.....	33
Gambar 4. 5 Grafik Prediksi dengan 60% <i>Data Training</i> dan 40% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 40.....	34
Gambar 4. 6 Grafik Prediksi dengan 60% <i>Data Training</i> dan 40% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 50.....	34
Gambar 4. 7 Grafik Prediksi dengan 70% <i>Data Training</i> dan 30% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 10.....	35
Gambar 4. 8 Grafik Prediksi dengan 70% <i>Data Training</i> dan 30% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 20.....	35
Gambar 4. 9 Grafik Prediksi dengan 70% <i>Data Training</i> dan 30% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 30.....	36
Gambar 4.10 Grafik Prediksi dengan 70% <i>Data Training</i> dan 30% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 40.....	36
Gambar 4.11 Grafik Prediksi dengan 80% <i>Data Training</i> dan 20% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 10.....	37
Gambar 4.12 Grafik Prediksi dengan 80% <i>Data Training</i> dan 20% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 10.....	37
Gambar 4.13 Grafik Prediksi dengan 80% <i>Data Training</i> dan 20% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 20.....	38
Gambar 4.14 Grafik Prediksi dengan 80% <i>Data Training</i> dan 20% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 30.....	38
Gambar 4.15 Grafik Prediksi dengan 80% <i>Data Training</i> dan 20% <i>Data Testing</i> dan Jumlah <i>Epoch</i> 40.....	39

Gambar 4.16 Grafik Prediksi dengan 80% Data Training dan 20% Data Testing dan
Jumlah Epoch 50 39

DAFTAR TABEL

Tabel 3.1 Skenario normal <i>use case</i> melihat data saham.....	25
Tabel 3.2 Skenario alternatif <i>use case</i> melihat data saham.....	25
Tabel 3.3 Skenario <i>use case</i> melihat grafik prediksi harga saham	25
Tabel 4.1 Analisis pengujian analisa harga saham	40

DAFTAR SIMBOL

Use Case Diagram

Nama Simbol	Simbol	Keterangan
<i>Use Case</i>		Sebuah <i>behavior</i> yang dilakukan sistem perangkat lunak
Aktor		Entitas atau <i>roll</i> diluar sistem yang berhubungan dengan sistem
Asosiasi		Hubungan antara aktor dan <i>use case</i>

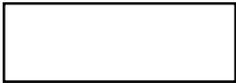
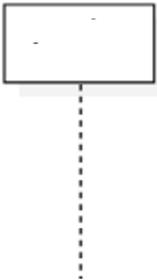
Activity Diagram

Nama Simbol	Simbol	Keterangan
Kondisi awal		Menunjukkan kondisi awal aktivitas sebuah sistem
Kondisi akhir		Menunjukkan kondisi akhir aktivitas berakhirnya sistem
<i>Activity</i>		Kegiatan sistem lakukan
<i>Decision</i>		Percabangan untuk memilih sebuah keputusan
garis		Menunjukkan arah <i>action state</i> selanjutnya

Class Diagram

Nama Simbol	Simbol	Keterangan
<i>Class</i>		Kelas (Class) memiliki tiga bagian yang di atas nama kelas, kedua atribut kelas, dan proses di dalam kelas
<i>Asosiasi</i>		Garis penghubung antar kelas

Sequence Diagram

Nama Simbol	Simbol	Keterangan
<i>Activation box</i>		Waktu yang dibutuhkan suatu objek untuk menyelesaikan pekerjaannya
<i>Object</i>		Garis penghubung antar kelas
<i>Synchronous message</i>		Simbol yang memberi isyarat alur selanjutnya yang akan dikerjakan
<i>Lifelines</i>		Menunjukkan kejadian berurutan yang terjadi pada sebuah objek selama proses pembuatan

DAFTAR LAMPIRAN

LISTING PROGRAM..... 45

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saham adalah modal dan bukti kepemilikan yang ditanamkan seseorang pada suatu badan usaha atau perseroan terbatas. Saham sendiri memiliki tujuan untuk menambah modal suatu perusahaan dari publik. Jenis saham terbagi menjadi 2 yakni saham biasa dan saham preferen. Berinvestasi saham kini semakin populer dan berkembang pesat di Indonesia, dengan berinvestasi saham, investor akan mendapatkan keuntungan yang cukup cepat dan besar dalam waktu yang cukup singkat, dalam beberapa hari, minggu, bulan atau tahun dari menginvestasikan dananya (Izzah et al., 2021). Investasi juga memiliki resiko yang cukup besar, untuk mengantisipasinya, investor wajib memiliki bekal ilmu analisa yang baik, terdapat 2 analisa yang banyak dipakai oleh para investor mulai yakni analisa fundamental dan teknikal.

Pasar saham merupakan salah satu sumber pembiayaan terpenting bagi perusahaan. Perusahaan yang terdaftar dapat meningkatkan pendanaan jangka panjang agar perusahaan dapat mengoptimalkan sumber dayanya, langkah-langkah ini untuk meningkatkan efisiensi kinerja (Saraswati, 2020). Dana yang diperoleh dari proses penjualan saham merupakan modal jangka panjang sehingga perusahaan wajib untuk mengatur keuangannya sehingga memiliki kinerja yang baik. Perkembangan pasar saham sangat berpengaruh dalam proses bertumbuhnya ekonomi di suatu kawasan, dengan semakin banyaknya masyarakat yang dapat membeli saham itu menandakan terjadi kenaikan kemampuan masyarakat untuk melakukan sebuah transaksi, dan ini juga menjadi instrumen yang penting dimana semakin banyak masyarakat yang melakukan investasi saham, berarti semakin banyak pula masyarakat yang memiliki kesadaran tentang pentingnya investasi.

Saham adalah merek dagang yang dimiliki oleh orang atau badan yang dimiliki korporasi atau perseroan terbatas berwujud sebuah kertas yang menjelaskan pemilik kertas tersebut merupakan salah satu pemilik perusahaan tersebut.

Pengertian kepemilikan saham dan tujuan saham menurut Darmaji adalah sebagai berikut: *“saham boleh dikatakan sebagai surat berharga karena mendefinisikan sebagai tanda kepemilikan seseorang atas suatu badan perusahaan. Pemilik surat berharga tersebut memiliki hak untuk menerima dividen dari suatu badan perusahaan yang telah dibeli sahamnya, dan untuk pembagian dividen”*.(Hendarsih, 2016) Pembagian dividen biasanya dibagi dalam beberapa termin waktu. Saham merupakan sektor investasi yang menjanjikan tetapi harus diperlukan komitmen kepercayaan dari investor untuk membeli saham dari suatu perusahaan. Pergerakan harga saham sendiri sangat fleksibel, itulah diperlukan kemampuan analisa yang baik agar para investor meminimalisir kerugian dan memaksimalkan keuntungan.

Investor membuat prediksi tentang perusahaan untuk berinvestasi dengan melihat laporan keuangan perusahaan, terutama perusahaan yang sudah go public. Laporan keuangan dilihat investor dan nilainya sebagai acuan bagi investor untuk menanamkan modalnya pada perusahaan, aspek yang dilihat investor dalam laporan keuangan seperti kinerja usaha, jalannya kegiatan usaha, kapasitas usaha, melaksanakan kegiatannya secara efisien dan efektif (Prakoso, 2016). Nilai *return* dari saham merupakan ketika harga saat jual saham lebih tinggi saat investor membelinya. Tentu saja jika investor menginginkan return yang besar diiringi dengan resiko yang besar pula, sebagai contoh investor yang ingin mendapat *return* yang besar harus memborong saham dengan banyak, ini diartikan investor berani *all-in*. Analisa yang tepat dapat memberikan penilaian dalam investor mengambil keputusan, itulah mengapa investor harus melihat kinerja perusahaannya dulu sebelum membeli sahamnya.

Tujuan utama investor adalah untuk mendapatkan keuntungan investasi (*return*) yang tinggi. Keuntungan adalah satu dari beberapa banyak parameter evaluasi manfaat dari suatu tindakan investor yang pertama kali akan berinvestasi. Pertama lihat saham perusahaan mana yang paling menguntungkan dengan cara mengevaluasi kinerja perusahaan (Nabila, 2017). Perkembangan teknologi dan informasi juga menjadi poin penting dalam perkembangan pengetahuan tentang pasar saham. Telah terdapat banyak sekali aplikasi-aplikasi broker yang menawarkan kemudahan dalam membeli saham suatu

perusahaan. Tentu saja *influencer social media* turut andil dalam menaikkan pamor tentang investasi saham.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dibahas, adapun rumusan masalah dari tugas akhir ini adalah:

1. Berapa komposisi data pada proses prediksi harga saham menggunakan algoritma *Long Short Term Memory (LSTM)* agar mendapatkan hasil yang terbaik?
2. Berapa tingkat akurasi terbaik dari prediksi menggunakan algoritma *Long Short Term Memory (LSTM)*?

1.3 Batasan Masalah

Terdapat banyak aspek yang dapat menjadi bahan analisa dalam melihat pergerakan pasar saham, ada analisa fundamental, analisa teknikal, dan sentimen pasar. Agar penelitian lebih terfokus dan tidak meluas dari pembahasan, maka penelitian ini akan berfokus pada pembuatan sistem analisa harga saham dengan kemampuan untuk melakukan analisa menggunakan algoritma LSTM (*Long Short Term Memory*), serta mengambil dataset harga saham dari Yahoo Finance dengan rentang waktu pengambilan data dari 1 Januari 2015 – 1 Januari 2022.

1.4 Maksud dan Tujuan Penelitian

Beberapa maksud dan tujuan dari penelitian yang telah dibuat:

1. Melakukan sebuah analisa untuk mencari komposisi data terbaik yang memiliki tingkat akurasi tertinggi.
2. Menguji bahwa dengan algoritma tertentu, bisa mendapatkan data yang keakuratannya mendekati data asli.

1.5 Kegunaan dan Hasil

Pembuat dapat mempelajari bagaimana cara kerja dari algoritma *Long Short Term Memory (LSTM)* dalam penyelesaian prediksi harga saham dan mengetahui perbandingan harga *close* saham dengan hasil prediksi.

1.6 Sistematika Penulisan

Untuk menyusun tugas akhir ini dibutuhkan sistematika penyajian sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, maksud dan tujuan, batasan masalah, kegunaan hasil, dan sistematika penulisan tugas akhir.

BAB II LANDASAN TEORI

Bab ini akan membahas rangkuman atas teori-teori yang mendukung berhubungan dengan prediksi harga saham. Teori-teori tersebut diantaranya *forecasting, data time series, algoritma Long Short Term Memory (LSTM), Root Mean Square Error*, bahasa pemrograman yang digunakan, serta beberapa *library* pendukung.

BAB III ANALISIS DAN RANCANGAN SISTEM

Bab ini menjelaskan tentang gambaran umum rancangan sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)* dan diagram-diagram untuk perancangan sistem.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini, akan menjelaskan hasil dari pengujian analisa harga saham menggunakan algoritma *Long Short Term Memory(LSTM)*.

BAB V KESIMPULAN

Bab ini berisi kesimpulan dan saran untuk penelitian ini yang dapat digunakan untuk perkembangan pada penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Rekayasa Perangkat Lunak

Perangkat lunak adalah program komputer yang terkait dengan dokumentasi perangkat lunak, seperti dokumentasi analisis persyaratan, model desain, dan panduan pengguna. Rekayasa perangkat lunak adalah ilmu yang berhubungan dengan produksi perangkat lunak berdasarkan analisis kebutuhan pengguna, definisi definisi kebutuhan pengguna, desain, pembuatan kode, pengujian dan pemeliharaan (Hasanah & Untari, 2018). Perangkat lunak adalah seperangkat program, proses, dan catatan data yang terintegrasi ke dalam aplikasi yang ditujukan untuk merespons semua kendala kinerja di dunia nyata. Rekayasa perangkat lunak boleh diartikan sebagai disiplin ilmu yang membahas segala aspek dari perangkat lunak. Aspek yang diperhatikan dalam ilmu ini mulai dari spesifikasi sistem hingga pemeliharaan sistem.

Rekayasa perangkat lunak adalah pendekatan matematis untuk membuat dan memelihara program berkualitas tinggi dan berbiaya rendah (Irmayani, 2019). Rekayasa perangkat lunak merupakan disiplin rekayasa yang berkonsentrasi pada banyak faktor. Rekayasa perangkat lunak sendiri tidak hanya berkonsentrasi pada *source code*, dokumentasi aspek produk juga sesuatu yang perlu diperhatikan selama pengembangan, instalasi, penggunaan, dan pemeliharaan sistem. Rekayasa atau *Engineering* merupakan pendekatan sistematis berdasar ilmu pengetahuan dan matematis.

Rekayasa perangkat lunak adalah proses pengembangan perangkat lunak dengan mengeksekusi perintah dalam program menggunakan bahasa pemrograman komputer sehingga komputer dapat memahami dan bekerja secara efisien dan efektif. Rekayasa perangkat lunak merupakan proses pengembangan perangkat lunak dengan menjalankan perintah dari program dalam sebuah komputer, sehingga dapat bekerja dengan efisien dan efektif (Pitrawati & Ningsih, 2017). Jelas jika rekayasa perangkat lunak ini ada bertujuan

untuk membantu kinerja dari *user*. Oleh karena itu rekayasa perangkat lunak jangan sampai sulit untuk digunakan oleh *user* karena pada dasarnya untuk membantu kinerja.

Rekayasa perangkat lunak adalah proses menggunakan mesin kinerja manusia untuk membuat perangkat lunak yang dirancang untuk dijalankan lebih efisien oleh orang-orang. Perangkat lunak adalah sebuah program komputer yang terintegrasi dengan dokumentasi perangkat lunak yang terdiri dari dokumentasi kebutuhan, model sistem, dan cara penggunaannya (Hendrajati & Widyatmoko, 2013). Tujuan utama dari dibuatnya perangkat lunak yakni untuk membantu kinerja *user* sehingga lebih efektif dan efisien, tetapi kita tidak bisa mengesampingkan unsur bisnisnya, sebuah perangkat lunak juga harus mengandung unsur ekonomis. Unsur ekonomis dimaksudkan, biaya perawatannya tidak mahal, efisien dengan sumber daya, serta memiliki harga yang bersaing.

Dengan demikian, rekayasa perangkat lunak dapat diartikan sebagai proses pengembangan sistem atau program yang dapat meningkatkan produktivitas pengguna, karena melibatkan operasi pemrosesan data dan informasi. Untuk rekayasa perangkat lunak, pengguna harus terlebih dahulu memesan penggunaannya.

2.2 Metodologi Penelitian

Penjelasan mengenai *Programming Oriented (OOP)* adalah sebagai berikut: "*Programming Oriented (OOP)* adalah metode pemrograman yang menggunakan objek dan kategori." *OOP* adalah teori yang menganalisis sistem dan masalah aplikasi. Dalam *OOP* setiap bagian program adalah *object*. Sebuah *object* mewakili bagian program yang akan diselesaikan. *Object Oriented Programming* adalah suatu cara penyelesaian yang berlandaskan pada *object*.

Keuntungan dari *Object Oriented Programming* sendiri yakni; alamiah, *reliable*, *reusable*, *maintainable*, *extendable* (Retnoningsih et al., 2017). *Object* merupakan unit terkecil dalam pemrograman *OOP*, *object* merupakan entitas dari keadaan, perilaku, dan identitas yang dirumuskan dalam suatu lingkup. *Object Oriented Programming (OOP)* adalah sistem perangkat lunak yang menggunakan objek dan kelas. *OOP* memfasilitasi pembuatan aplikasi, dan manfaatnya meliputi:

1. *Reuseability*, kode yang dibuat dapat dipakai kembali.
2. *Extensibility*, pemrogram dapat membuat metode baru dan mengubah yang sudah ada tanpa perlu membuat kode dari awal.
3. *Maintainability*, kode yang sudah dibuat lebih mudah untuk mengelola menangani adanya *error* dalam pengembangannya

2.3 Model Pengembangan

Model pengembangan yang dipakai untuk membangun sistem analisa harga saham menggunakan model spiral. Model proses yang menggabungkan wujud pengulangan dari model prototyping dengan pengendalian dan sistematika dari *linear sequential model* dengan penambahan elemen baru. Model ini memiliki cara kerja dimana jika terjadi ketidakpastian terhadap kebutuhan, maka simulasi dan permodelan lain dapat digunakan untuk menjelaskan masalah dan melengkapi maupun membenahi kebutuhan. *User* dapat melakukan pengecekan, evaluasi, dan saran untuk perbaikan.

1. Spiral Model

Model spiral adalah model proses yang menggabungkan bentuk iteratif dari model prototipe dengan aspek kontrol dan sistematik dari model sekuensial linier. Model ini memiliki empat kegiatan penting, yaitu (Bolung & Tampangela, 2017):

a. Perencanaan

Penentuan tujuan, alternatif, dan batasan.

b. Analisis Resiko

Analisis alternatif dan identifikasi atau pemecahan resiko.

c. Rekayasa

Pengembangan level berikutnya dari produk.

d. Evaluasi

Penilaian terhadap hasil rekayasa bentuk spiral menggambarkan semakin besar iterasinya, maka semakin lengkap versi dari perangkat lunak yang dibuat.

2.4 Teori Khusus

Teori khusus yang terdapat dalam pembuatan sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)* terdapat *forecasting*, *data time series*, algoritma *Long Short Term Memory (LSTM)*, dan *Root Mean Square Error (RMSE)*.

2.4.1 Forecasting

Forecasting adalah proses memprediksi peristiwa masa depan dengan memperoleh data historis dan merencanakan masa depan (Wiranda & Sadikin, 2019). *Forecasting* adalah alat yang akan membantu pengambil keputusan dalam penyusunan rencana nyata untuk masa depan. *Forecasting* banyak di antaranya dapat digunakan, terutama di bidang ekonomi, bisnis, dan pemasaran, mulai dari memprediksi produk masa depan, membantu perusahaan memprediksi kinerja perusahaan, dan membantu mereka membuat keputusan bisnis.

Forecasting atau peramalan adalah prediksi komponen pengelolaan pengambilan keputusan. *Forecasting* mengurangi ketergantungan perkiraan pada hal-hal yang tidak atau belum pasti (Rizkiyani, 2014). Peramalan merupakan kegiatan untuk aspek dalam pengambil keputusan. Peramalan memiliki fungsi untuk mengurangi segala bentuk ketergantungan pada hal-hal yang belum pasti. Metode *time series* merupakan yang paling populer untuk diterapkan pada proses *forecasting*.

Prediksi merupakan salah satu elemen penting saat mengambil keputusan. Prediksi dibuat berdasarkan masa lalu yang kemudian menganalisisnya menggunakan metode tertentu (Mulyani et al., 2021). Data dari masa lalu dikumpulkan, dipelajari, dianalisis, dan berhubungan dengan perjalanan karena faktor waktu kemudian hasil analisis data itu dapat memprediksi apa yang akan terjadi di masa depan (Fauzi, 2019). Pengaruh yang diperoleh saat melakukan prediksi sangatlah besar, jika berkecimpung di dunia bisnis pasti akan mempengaruhi penyusunan anggaran pada penjualan, memprediksi menentukan

anggaran untuk bulan atau tahun yang mendatang, bahkan dapat memprediksi banyaknya atensi masyarakat dengan produk yang dijual. Prediksi sendiri dapat dicari menggunakan tren, pada sektor saham diperlukan data saham dari masa lalu hingga masa sekarang, sehingga dapat diprediksi lalu disimpulkan, bagaimana tren kinerja dari perusahaan.

Forecasting adalah proses memperkirakan menggunakan nilai silang untuk memprediksi peristiwa masa depan dan untuk mengambil data historis dan mewakili mereka menggunakan perhitungan matematis (Nasution, 2019). Data historis yang dimiliki akan diproyeksikan ke masa depan menggunakan perhitungan matematis. Data yang dipakai berbentuk *time series*, yang mengandalkan data-data terdahulu dan bobot yang berbeda-beda untuk memperoleh prediksi di masa mendatang.

Dengan demikian, *forecasting* merupakan proses prediksi yang memerlukan data historis, dengan tujuan untuk memproyeksikan masa depan dan diolah menggunakan perhitungan matematis, sehingga dapat membantu pengambil keputusan.

2.4.2 Data Time Series

Ini adalah serangkaian pengamatan berdasarkan urutan blok peristiwa yang ditutup dalam periode waktu tertentu. Pada data time series, model dibagi menjadi tiga yaitu siklus, trend, dan model musiman (Al'afi et al., 2020). Berdasarkan *zoning* (domain), informasi deret waktu dibagi menjadi dua domain, domain waktu dan domain frekuensi. *Data time series* adalah urutan kronologis yang panjang. Jenis informasi ini umum dalam kehidupan sehari-hari karena informasi yang dikumpulkan dari waktu ke waktu, mulai dari setiap hari, setiap minggu, setiap bulan, maupun setiap tahun. Data yang dikumpulkan menggambarkan hal ini.

Data time series adalah suatu model penelitian aktif yang bertujuan mengumpulkan dan menganalisis pengamatan masa lalu. Tujuannya adalah untuk mengembangkan modelnya serta mendefinisikan struktur seri (Audina et al., 2021). Rangkaian pengamatan data dalam kurun waktu tertentu, pendekatan *time series* sangat cocok untuk menyelesaikan masalah data linier. Tujuan dari pengembangan model ini adalah untuk

menggambarkan struktur dari deret waktu. Deret waktu dapat dibedakan dengan cara pengolahannya, terdapat analisis deret waktu dan ada peramalan deret waktu.

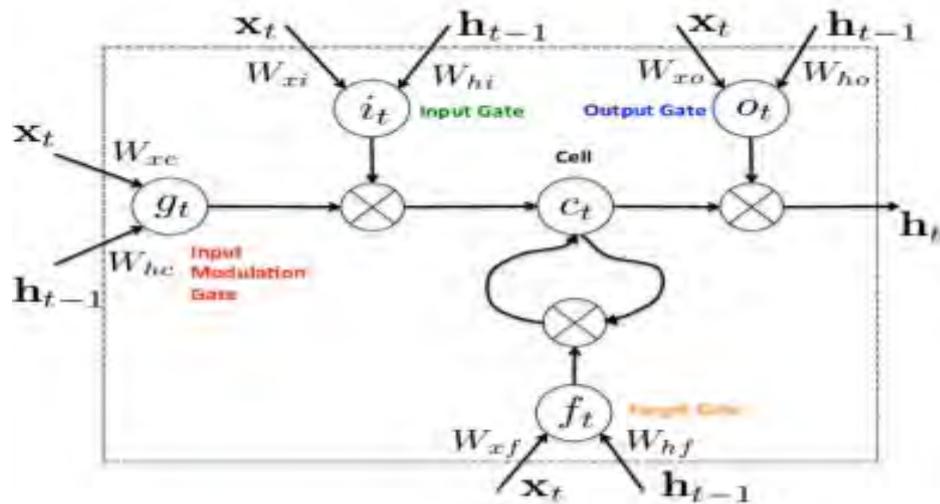
Data time series dianalisis untuk menemukan pola dan seri di hitung sehingga menjadi data untuk masa depan berdasarkan pola. Ekstrapolasi pola peristiwa masa depan berdasarkan pola adalah subjek utama analisis rangkaian waktu (Harlina & Usman, 2020). Rangkaian waktu sendiri merupakan sebuah urutan data numerik yang terekam dalam beberapa rentang waktu, yang dapat dihitung dalam rentang harian, mingguan, bulanan, triwulan, maupun tahunan.

Data time series adalah analisis rangkaian atau rangkaian peristiwa yang terjadi selama periode waktu tertentu. Karakteristik *data time series* memiliki tujuan untuk menemukan suatu keteraturan pola yang digunakan untuk memprediksi masa mendatang (Hansun, 2013). Banyak sekali metode atau proses untuk memproses *data time series*, mulai dari sistem *fuzzy*, *neural network*, *genethic algorithm*, dan *hybrid*.

Dapat disimpulkan bahwa *data time series* merupakan kumpulan kejadian yang terjadi dalam periode tertentu, dan waktunya dapat meliputi hari, bulan, hingga tahunan, yang digunakan untuk penelitian.

2.4.3 Algoritma Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) berasal dari metode *RNN (Recurrent Neural Network)*, yang dirancang khusus untuk memproses *Data Sequence*. Namun, *RNN* memiliki masalah yaitu jika terjadi perubahan *range* nilai dari satu level ke level berikutnya dalam arsitektur (Wahyu, 2021). *LSTM* dirancang dan dirancang untuk mengatasi masalah ini. Arsitektur *LSTM* terdiri dari lapisan-lapisan yang terdiri dari:



Gambar 2. 1 Arsitektur LSTM

Sumber: (Wahyu, 2021)

1. Input Gate (i_t)

Input gate berperan mengambil *output* sebelumnya dan *input* baru serta melewati mereka melalui lapisan *sigmoid*. Gate ini mengembalikan nilai 0 atau 1. Rumus dari (i_t) adalah:

$$i_t = \sigma(W_i s_{t-1} + W_i x_t)$$

Penjelasan,

W_i = Bobot dari *Input Gate*.

s_{t-1} = *State* sebelumnya atau *state* pada waktu $t-1$.

x_t = *Input* pada waktu t .

σ = Fungsi aktivasi *sigmoid*.

Nilai gerbang *input* dikalikan dengan *output* dari lapisan kandidat (\tilde{C}). Rumus dari (\tilde{C}) adalah:

$$\tilde{C} = \tanh(W_c s_{t-1} + W_c x_t)$$

$$c_t = (i_t * \tilde{C}_t + f_t * c_{t-1})$$

Penjelasan,

\tilde{c} = *Intermediate cell state*.

W_c = Bobot dari *cell state*.

$St-1$ = *State* sebelumnya atau *state* pada waktu $t - 1$.

X_t = *Input* pada waktu t .

2. *Forget Gate* (f_t)

Forget Gate adalah lapisan *sigmoid* yang mengambil *output* pada waktu $t-1$ dan *input* pada waktu t dan menggabungkannya serta menerapkan fungsi aktivasi *sigmoid*. Karena *sigmoid*, *output gate* ini adalah 0 atau 1. Jika $f_t = 0$, maka keadaan (*state*) sebelumnya akan dilupakan, sementara jika $f_t = 1$ *state* sebelumnya tidak berubah. Rumus dari f_t adalah:

$$f_t = \sigma(W_f S_{t-1} + W_f X_t)$$

Penjelasan,

W_f = Bobot dari *forget gate*.

$St-1$ = *State* sebelumnya atau *state* pada waktu $t - 1$.

X_t = *Input* pada waktu t .

σ = Fungsi aktivasi *sigmoid*.

3. *Output Gate* (o_t)

Output Gate mengontrol seberapa banyak *state* yang lewat *output* dan bekerja dengan cara yang sama dengan *gate* lainnya. Dan terakhir menghasilkan *cell state* baru (h_t). Rumus dari o_t dan h_t adalah:

$$o_t = \sigma(W_o S_{t-1} + W_o X_t)$$

$$h_t = o_t * \tanh(ct)$$

Penjelasan,

W_o = Bobot dari *output gate*.

$St-1$ = State sebelumnya atau *state* pada waktu $t - 1$.

Xt = *Input* pada waktu t .

σ = Fungsi aktivasi *sigmoid*.

2.4.4 Root Mean Square Error (RMSE)

Root mean square error (RMSE) adalah metode atau metode umum yang biasa digunakan untuk menguji perbedaan nilai prediksi data, penjumlahan kuadrat selisih nilai aktual, dan nilai. dapat menentukan jumlah *error* pada nilai *forecast*, besar kecilnya nilai *RMSE*, besarnya *error* pada nilai *forecast*. skala *RMSE* (Wahyu, 2021).

$$\sqrt{\sum(P_i - O_i)^2 / n}$$

Penjelasan,

Σ = simbol untuk mengartikan total

P_i = permintaan aktual

O_i = peramalan permintaan

n = jumlah periode peramalan yang terlibat

2.5 Perangkat Lunak yang Menjadi Alat Bantu Perancangan Perangkat Lunak

Perangkat lunak yang digunakan untuk membuat sistem analisa harga saham menggunakan *algoritma Long Short Term Memory (LSTM)* adalah menggunakan bahasa pemrograman *Python, Anaconda, Jupyter Notebook, Pandas, Numeric Python (Numpy), Keras, Matplotlib, Yahoo Finance, dan Star UML*.

2.5.1 Python

Python adalah pemrograman berorientasi objek dan deskriptif yang termasuk dalam bahasa perantara untuk memfasilitasi pembelajaran. *Python* menjadi populer ketika digunakan secara luas dalam penelitian termasuk ilmu data, kecerdasan buatan, penelitian, permainan, pengembangan perangkat lunak, dan pembelajaran mesin (Pajankar, 2021). *Python* merupakan bahasa yang paling populer saat ini karena ada banyak industri yang perlu mengembangkan sistem yang hanya dapat digunakan dengan

Python atau dapat berhasil dikembangkan, baik dalam sains, kedokteran, atau institusi. Menurut Countench dan Helian, "*Python terkenal karena dapat digunakan di mesin pencari apa pun, dari analitik otomatis* (Rolon-Mérette et al., 2020)." Inilah mengapa *Python* bagus untuk penelitian, pemrosesan data, pembelajaran mesin, dan kecerdasan buatan.

Python merupakan bahasa pemrograman yang tergolong *high-level and interpreted programming language* (Sharma & Rengarajan, 2020), yang pertama kali diperkenalkan oleh Guido van Rossum pada tahun 1991. Bahasa pemrograman ini mendukung berbagai paradigma pemrograman, termasuk terstruktur, berorientasi objek, dan pemrograman fungsional.

Kode program yang ditulis dengan *Python* membutuhkan penerjemah, biasanya disebut penerjemah. Translator sendiri adalah program yang diinstal di komputer mana pun untuk merancang dan membuat kode, untuk membuatnya berfungsi, kode *Python* diubah menjadi kode dan diubah menjadi *Python Virtual Machine (PVM)* (Nelli, 2018).

2.5.2 Anaconda

Anaconda adalah *software* gratis yang memberikan *toolkit* yang dapat digunakan untuk kebutuhan penelitian dan *sains* (Rolon-Mérette et al., 2020). *Anaconda* bisa di *download* untuk sistem operasi *Windows, Mac, Linux* secara gratis, dengan meng-*install anaconda* dapat memberikan akses untuk melakukan *coding* dalam bahasa *python* atau *R*. *Anaconda* memudahkan proses dalam menjaga semua *libraries* untuk terus *up to date*, itu semua dapat *Anaconda* lakukan dalam sekali instalasi.

Anaconda adalah *software* gratis yang digunakan pengembangannya dalam bahasa pemrograman *python* dan *R*. *Anaconda* sendiri digunakan untuk berbagai tujuan ilmiah, *Machine Learning, Deep Learning*, dan banyak lagi, dengan akses lebih dari 300 *library* (Sharma & Rengarajan, 2020). Sangat memungkinkan untuk melakukan instalasi *python* beserta *library*-nya. Untuk *library*-nya sendiri terdapat *open cv, numpy*, dan lain-lain.

Anaconda digunakan untuk mempercepat dalam perhitungan aljabar linier yang akan membantu kerja dalam mengerjakan analisis data (Sheppard, 2013). Dapat

menyediakan modul dengan kinerja yang tinggi untuk membaca data ataupun *file* dalam ukuran yang besar. Untuk penginstallan sendiri *Anaconda* sangat mudah, dan jangan dilupakan untuk *update*-nya pun sangat mudah. *Library* yang terdapat pada *anaconda* ada banyak terdapat *numpy*, *pandas*, *Jupyter*, *Scikit learn*, *Matplotlib*, *Keras*, dan lain-lain.

Anaconda merupakan *software open source* dan gratis, untuk mengolah *data science* maupun *machine learning* (Bhevsar & Manglani, 2019) yang didistribusikan dalam bahasa pemrograman *python* dan *R*. *Konda* merupakan sistem manajemen paket *open source* dan lintas platform.

2.5.3 Jupyter Notebook

Jupyter notebook adalah *Integrated Development Environment* (IDE) yang berbasis *web*, setiap blok kode dapat dijalankan secara terpisah sehingga sangat fleksibel dan memudahkan untuk bereksperimen, sehingga *output* kode, visualisasi, persamaan, dan teks normal dapat digunakan semua di satu tempat (Pérez & Granger, 2015). *Jupyter notebook* membuat efektif dalam hal membuat dan berbagi dokumen, dan untuk menghasilkan data yang terorganisir dan estetis saat menyajikan kode program, dan dikarenakan berbasis *web* tentu saja *Jupyter notebook* sangat memudahkan untuk melakukan kolaborasi dengan lebih optimal.

Jupyter Notebook bersifat interaktif di saat yang bersamaan dokumen pemrograman akan dipelajari dan akan menjalankan dokumen tersebut. *Jupyter Notebook* terdiri dari *cells*, dan untuk menjalankan *cells*-nya menggunakan *kernel* (Pimentel et al., 2019), kemudian dalam eksekusi, kernel memberikan nomor ke penghitung yang menunjukkan pelaksanaan tugas. Pengguna dapat menjalankannya di dalam *cells* perintah apapun, dan sel tertentu dapat dieksekusi beberapa kali.

Jupyter Notebook adalah *web-based environment* yang dapat digunakan untuk mengetik kode secara interaktif. *Jupyter Notebook* sendiri berjalan di dalam *lpython environment*, yang dirilis pada 2001 (Suárez-García et al., 2021), dan *Jupyter Notebook*

memungkinkan pengguna menuliskan kode dengan bahasa pemrograman *python*, *R*, dan *Julia*, dan *Matlab*.

Jupyter Notebook adalah aplikasi *web open source* yang memungkinkan pengguna menuliskan teks deskriptif, perbandingan, dan ilustrasi (Wang et al., 2020). *Jupyter Notebook* juga menjadi pilihan utama para *data scientist* dikarenakan membuat analisis data lebih mudah untuk direkam dan dipahami. *Jupyter Notebook* memiliki popularitas yang meningkat akhir-akhir ini. Data mencatat sebanyak 2.5 juta proyek *jupyter notebook* yang terdapat di *github* pada tahun 2018, ini lebih banyak 10 kali lipat dari tahun 2015.

Jupyter notebook memungkinkan untuk menulis program dalam bentuk program matematika, *jupyter notebook* juga memungkinkan pembuatan, penghapusan, pengeditan kode menjadi lebih interaktif. *Jupyter Notebook* berjalan di server (Sheppard, 2013), sehingga memerlukan koneksi internet untuk mengaksesnya. *Browser* untuk mengaksesnya beragam, ada *chrome*, *firefox*, *safari*, *edge*, dan lain-lain.

2.5.4 Pandas

Pandas merupakan *library python* yang digunakan untuk melakukan analisis data dan merupakan bagian dari *SciPy* (Pajankar, 2021). *Pandas* berisi struktur data yang biasa digunakan untuk membuat tabel, mengubah dimensi data, mengecek data, dan lain sebagainya.

Pandas Datareader adalah *sub package* dari *Pandas* yang bekerja mengambil *access key* dari sebuah sumber data dari internet. Berikut beberapa contoh sumber data yang didukung oleh *Pandas Datareader*, terdiri dari; *AlphaVantage*, *Federal Reserve Economic Data (FRED)*, *Fama-French Data*, *Bank of Canada*, *Econdb*, *Enigma*, *Eurostat*, *The Investors Exchange (IEX)*, *Moscow Exchange (MOEX)*, *NASDAQ*, *Naver Finance*, *Organisation for Economics Co-operation and Development (OECD)*, *Quandl*, *Stooq.com*, *Tiingo*, *Thrift Saving Plan (TSP)*, *World Bank*, *Yahoo Finance*, *Google Finance*, dan lainnya (Gananjay, 2021).

Pandas Dataframe merupakan representasi tabular data pada *python*. *Pandas Dataframe* memiliki baris dan kolom. *Pandas Dataframe* juga memiliki label yang biasa disebut dengan indeks (Pajankar, 2021), sehingga memudahkan untuk melakukan manipulasi data ataupun seleksi data. *Pandas Dataframe* juga memungkinkan pengguna untuk membuat *dataframe* dari *array*, seri, daftar, dan kamus.

2.5.5 Numerical Python (Numpy)

Numpy digunakan untuk membuat fungsi matematika khusus. *Python (Numpy)* adalah *open source* dan bagian dari paket *Python* yang menggunakan angka dan matematika *Numpy* yang dihasilkan dengan cepat. Tujuan dari implementasi *Numpy* adalah untuk mengontrol *array* dan *matriks*. Modul *numpy* berisi berbagai pustaka standar untuk melakukan perhitungan matematis menggunakan *array* dan *matriks* (Sheela et al., 2019). *Matriks numpy* menyediakan alat dimensi yang kuat untuk objek. Objek *matriks* digunakan untuk perhitungan *matriks*. *Numpy* mendukung fitur-fitur canggih untuk melakukan tugas-tugas yang dapat digunakan kembali. *Numpy* menyajikan paket dasar untuk *aljabar linier*, yang sangat berguna untuk perhitungan matematis.

Numpy sebuah *library python* yang digunakan untuk keperluan analisis numerik dalam *python* dan menyimpan datanya dalam bentuk *array* dimensi (Nongthombam & Sharma, 2021). *Numpy* menyediakan satu set tipe data *array* dan *matriks* yang dimanfaatkan untuk pengolahan statistik, *ekonometrik*, dan analisis data.

Sebagai bagian dari *library Python* dalam analisis data, banyak bahasa *Python* dan pemrograman sampel digunakan, sehingga memungkinkan untuk membantu dalam pemrosesan data (Nelli, 2018). *Numpy* digunakan untuk melakukan perhitungan *array multidimensi*, sehingga sangat memungkinkan untuk melakukan perhitungan matematika tingkat tinggi dengan sangat efisien.

2.5.6 Keras

Keras adalah *high-level neural network API*, yang dasarnya ditulis dalam bahasa pemrograman *python*. *Keras* dikembangkan dengan fokus untuk melakukan eksperimen

yang cepat, pengguna dimungkinkan untuk mengakses seluruh *API Keras* (B Arnold, 2017). Fungsi yang dimiliki menunjukkan cara kerja lapisan atau objek. Kumpulan data kecil diperhatikan untuk memastikan apakah model sudah benar dan cocok untuk suatu tugas.

Keras dapat membantu untuk mengembangkan model *deep learning* yang berfungsi penuh dengan kurang dari 15 baris dari kode. *Keras* memiliki kesederhanaan dan berusaha membantu pengguna untuk mengembangkan dan mengirimkan model *Deep Learning* dengan cepat fleksibilitas tinggi meskipun *API* tingkat tinggi (Moolayil, 2019).

Keras adalah *library python* untuk mengembangkan *deep learning* yang dapat berjalan diatas *TensorFlow*, *Theano*, maupun *CNTK*. Pengembang dapat fokus dalam konsep *deep learning* (Manaswi, 2018), seperti membuat *neural network*, sembari menjaga detail dari *tensor*, bentuknya, dan detail matematisnya. *TensorFlow*, *Theano*, dan *CNTK* berperan menjadi *backend* dari *Keras*.

Keras memiliki beberapa modul, terdiri dari modul *function modules*, *layer modules*, *preprocessing modules*, *objective function modules*, *optimization method selection modules*, dan lain-lain (Jiang & Shen, 2019). Banyaknya modul yang dimiliki oleh *Keras* memungkinkan membangun mode jaringan dengan lebih mudah dan parameter kunci dari *neural network* dapat lebih ditingkatkan.

2.5.7 Matplotlib

Matplotlib adalah sebuah *library python* Bertujuan untuk menggambar grafik untuk merepresentasikan sesuatu dan sangat berguna untuk membangun sebuah aplikasi yang interaktif dan *user-friendly* (Sheela et al., 2019). *Matplotlib* termasuk sebuah *library* yang fleksibel dimana biasanya digunakan untuk menghasilkan *bar*, *line*, *column*, *pie charts*, dan *scatterplots*, sehingga sangat baik untuk menyajikan data *histogram*.

Matplotlib merupakan bagian dari *library python* yang bertujuan untuk melakukan visualisasi data dan menyajikannya dalam grafik. Dalam *matplotlib* tersedia plot untuk 2

dimensi dan 3 dimensi (Nongthombam & Sharma, 2021), tetapi memiliki keterbatasan dalam segi tampilan diagram yang tidak terlalu menarik karena hasil diagramnya standar.

Matplotlib adalah *library python* yang dirancang untuk mengoptimalkan *Matlab* sesuai dengan presentasi grafis dan sintaksis, untuk membantu pengguna melihat data untuk pertama kalinya (Nelli, 2018), terutama bagi mereka yang tidak terbiasa dengannya. *Matplotlib* mampu mewariskan segi interaktif dari *matlab*, dalam artian pengguna dapat memberikan perintah untuk mengontrol perkembangan bertahap dari representasi grafis data.

2.5.8 Yahoo Finance

Yahoo finance banyak berita keuangan, informasi bisnis dan akses informasi dengan perusahaan. Pada tahun 2008, *Yahoo Finance* terdaftar untuk membantu bekerja pada penelitian berita dan dari Amerika Serikat, itu bertahan tiga hari sehari. *Yahoo finance* yang berpotensi memberikan informasi perusahaan di bursa, terutama bagi pengguna yang sangat mengandalkan *Yahoo Finance*, yang dapat diakses oleh pengguna saat ini mencapai 30 juta orang per hari (Lawrence et al., 2016), dibandingkan dengan *EDGAR* dan 60.000 pengguna per hari. . Tidak seperti penelitian *Google*, di mana informasi keuangan tidak tersedia, *Yahoo* bekerja secara ekonomis dengan cara perusahaan mengakses informasi dan informasi keuangan. *Yahoo Search* memiliki informasi paling mutakhir di pencarian *Google* atau *EDGAR*, terhitung 35% dari pencarian *Yahoo* sementara pencarian *Google* atau *EDGAR* hanya berarti 6 persen dari perbedaan dalam pencarian *Yahoo*.

Yahoo Finance sangat terkenal sebagai media informasi *financial* di Amerika Serikat. Alasan mengapa *Yahoo Finance* sangat terkenal dan sering kali dijadikan sebagai media informasi terutama di bidang ekonomi karena, para investor sangat memperhatikan sekali literatur keuangan secara umum, dan mengandalkan studi *observasional* untuk menyelidiki sentimen antara apa saja yang terjadi di dunia dan aktivitas pasar (Lawrence et al., 2018). Pada dasarnya ekonomi sangat bergantung dengan keadaan dunia. *Yahoo Finance* selalu memberikan berita tentang pasar saham, dan suasana dunia yang mempengaruhi

pergerakan saham. Tak hanya itu *Yahoo Finance* menyajikan data-data pergerakan saham yang *real* dan hampir seluruh perusahaan yang melantai di bursa saham, dapat dilihat kinerjanya melalui *Yahoo Finance*.

2.5.9 *Star UML*

Star UML adalah bahasa yang digunakan untuk membuat rencana, menggambarkan proses dan rencana.

UML menyediakan diagram yang dapat membantu menjelaskan penggunaan. Ada beberapa jenis diagram di *Star UML*, yaitu (Yusman, 2017):

1. *Use Case*

Use case diagram digunakan untuk menentukan interaksi apa yang dimiliki pemain pengguna dengan suatu sistem. Gunakan diagram cerita yang digunakan untuk menggambarkan fitur-fitur yang tersedia dalam sistem

2. *Activity Diagram*

Diagram digunakan untuk memahami alur kerja suatu proyek, aktivitas sistem, aktor, atau proses bisnis, dan dapat dikatakan bahwa diagram aktivitas merupakan perpanjangan dari diagram *use case* (Wardani, 2012) dimana diagram aktivitas menjelaskan setiap *use case* secara detail. . *Activity diagram* digunakan untuk memvisualisasikan interaksi antar *use case*.

3. *Sequence Diagram*

Sequence diagram menggambarkan perilaku suatu objek yang digunakan, menggambarkan masa pakai objek dan pesan yang dikirim dan diterima oleh objek, menunjukkan urutan pertukaran pesan antar objek, tugas yang dilakukan, dan urutan kejadian dalam sistem (Wardani, 2012). *Sequence diagram* dapat menggambarkan interaksi antar objek dalam suatu sistem yang terdiri dari dimensi waktu dan objek terkait.

4. *Class Diagram*

Class diagram adalah kumpulan objek yang memiliki atribut dan metode yang menggambarkan struktur dan definisi serta mendefinisikan hubungan antar kelas, termasuk penyimpanan, pewarisan, dan integrasi (Wardani, 2012).

BAB III

ANALISIS DAN PERANCANGAN PERANGKAT LUNAK

3.1 Gambaran Umum

Sistem analisa harga saham menggunakan *algoritma Long Short Term Memory (LSTM)* merupakan sistem analisa untuk harga saham memakai analisa teknikal, sehingga kita perlu mengambil data perusahaan yang melantai di bursa pasar saham dan mengolah data perusahaannya sehingga menyajikan sebuah grafik analisa.

Pengguna dapat menggunakan sistem ini dengan cara memilih saham yang telah di sediakan di dalam sistemnya, disaat pengguna memilih saham langsung tersedia grafik dari saham perusahaan yang dipilih, sehingga pengguna dapat menggunakan grafik tersebut untuk menganalisa kapan saatnya untuk membeli maupun menjual sahamnya.

Untuk proses pengolahan data menggunakan teknik *forecasting* dengan mengambil *data time series* dari *Yahoo Finance*, dimana saat pengambilan *data time series* terdapat *time domain* dan *frequency domain* untuk mengetahui kondisi data saham suatu perusahaan.

Pada akhir akan tersaji sebuah evaluasi hasil prediksi dengan cara membandingkan data real dengan data prediksi, menghitung jumlah *error*, sehingga dapat disimpulkan, apakah algoritma yang dipakai pada pembuatan sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)* memiliki rasio ketepatan prediksi yang baik atau tidak.

3.2 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak terdiri dari kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional merupakan penjelasan mengenai fungsi utama dan proses apa yang terjadi pada sebuah program. Kebutuhan non fungsional berisi tentang spesifikasi perangkat lunak yang diperlukan dalam membangun sebuah program.

3.2.1 Kebutuhan Fungsional

Sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)* memiliki fungsi untuk mencari tahu metode-metode apa saja yang diperlukan untuk melakukan prediksi harga saham, mulai dari algoritma yang dipakai, indikator yang digunakan, serta yang paling penting adalah bagaimana apa yang kita analisa memiliki prediksi yang tidak berbeda jauh dengan data sesungguhnya.

Belajar banyak tentang cara pengambilan data, mengolah data, dan menyajikan data, sehingga tentu sangat berguna bagi masyarakat yang ingin belajar tentang penerapan *data science*, juga menjadi suatu sarana bagi masyarakat untuk memulai belajar memprediksi harga saham dengan lebih baik.

3.2.2 Kebutuhan Non Fungsional

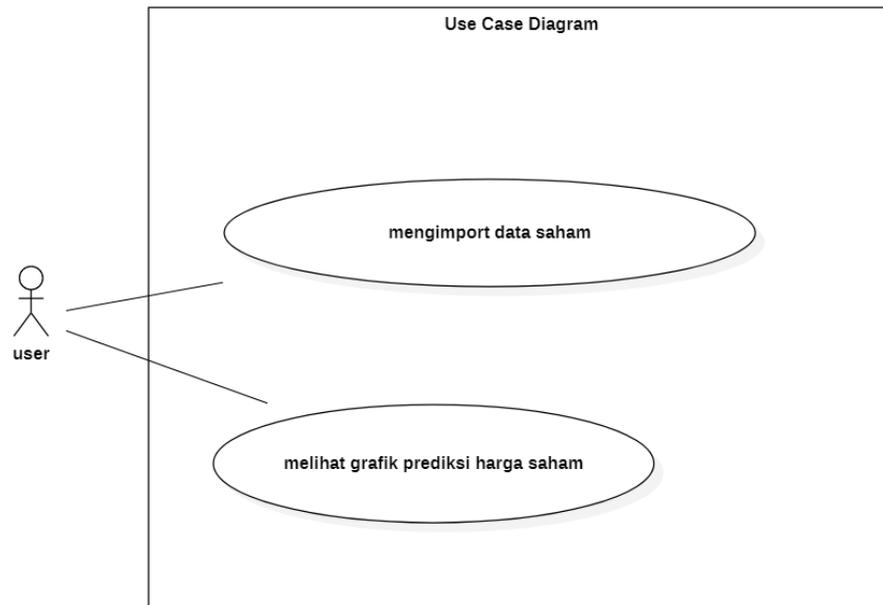
Untuk melakukan analisa harga saham tentu saja dibutuhkan spesifikasi sistem yang khusus dalam menunjang pengerjaan analisa ini. Berikut spesifikasi yang diperlukan.

1. *Operating System* : *Windows 8, macOS 10.13+, Linux.*
2. *System Architecture* : *Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86; Linux- 64-bit x86, 64-bit aarch64 (AWS Graviton2 / arm64), 64-bit Power8/Power9, s390x (Linux on IBM Z & LinuxONE).*
3. Minimal memiliki ruang penyimpanan 5 GB.

3.3 Diagram – diagram Perancangan Perangkat Lunak

Diagram-diagram yang dipakai pada pengerjaan Tugas Akhir kali ini terdiri dari *Use Case, Skenario Use Case, Activity Diagram, Class Diagram, dan Sequence Diagram.*

3.3.1 Use Case Diagram



Gambar 3. 1 *Use Case Diagram* sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)*

Penjelasan mengenai Gambar 3.1 *Use Case Diagram* sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)*.

1. User

User berperan sebagai aktor yang bertugas melakukan *input code* sehingga komputer dapat melakukan *import* data hingga menjadi data dalam bentuk grafik.

2. Mengimpor data saham

Mengambil data saham dari *Yahoo Finance*

3. Melihat grafik prediksi harga saham

Penyajian *data time series* yang telah diambil dan disajikan menjadi sebuah grafik prediksi.

3.3.2 Skenario Use Case

Skenario Use Case merupakan bagian *use case* yang berfungsi menunjukkan segala proses yang ada di dalam sebuah *use case*. Dimulai dari pengguna memberikan perintah pada setiap bagian dan mencatat apa saja respon dari sistem terhadap sebuah perintah.

Tabel 3. 1 Skenario normal *use case* melihat data saham

<i>User</i>	Sistem
1. Menulis kode untuk mengambil data saham dari <i>Yahoo Finance</i>	
	2. Mengimpor <i>dataset</i> saham
3. melihat <i>dataset</i> saham	

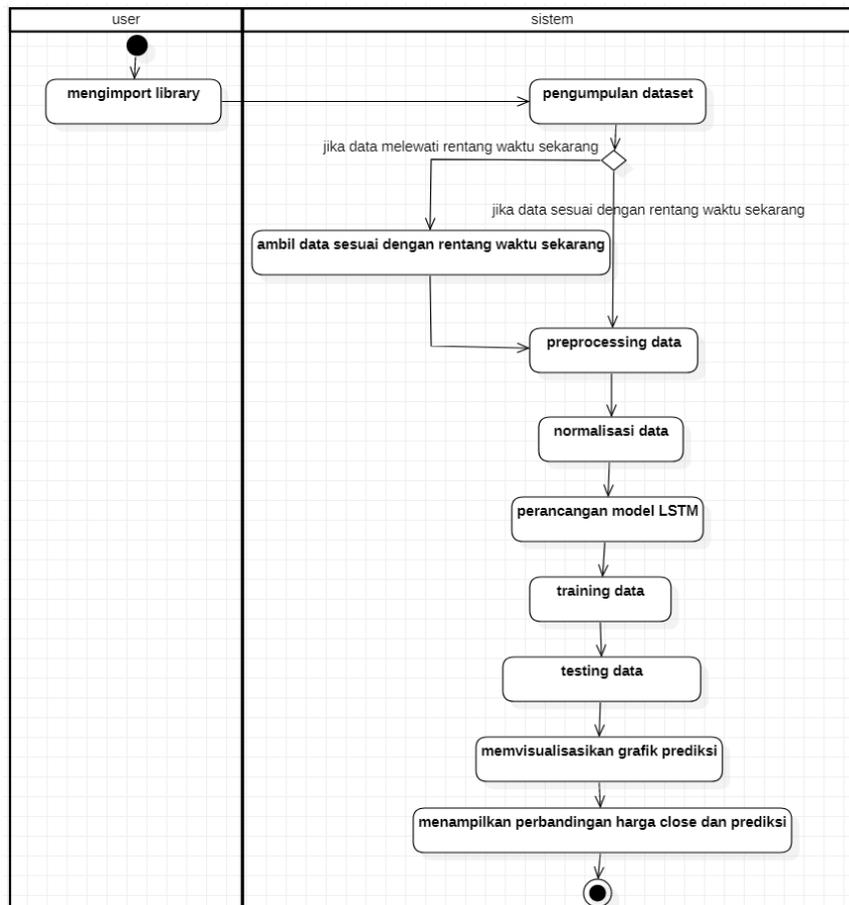
Tabel 3. 2 Skenario alternatif *use case* melihat data saham

<i>User</i>	Sistem
1. Menulis kode untuk mengambil data saham dari <i>Yahoo Finance</i>	
	2. Mengimpor <i>dataset</i> saham
	3. Sistem tidak dapat mengimpor data saham jika rentang waktu harga saham melebihi tanggal sekarang
	4. Mengimpor data saham hingga rentang waktu sekarang
	5. Menampilkan <i>dataset</i> saham

Tabel 3. 3 Skenario *use case* melihat grafik prediksi harga saham

<i>User</i>	Sistem
	1. Mengolah <i>dataset</i> yang telah diimpor
	2. Melakukan <i>preprocessing data</i>
	3. Melakukan normalisasi data
	4. Perancangan model <i>LSTM</i>
	5. Melakukan <i>Training data</i>
	6. Melakukan <i>Testing data</i>
	7. Visualisasi hasil prediksi

3.3.3 Activity Diagram



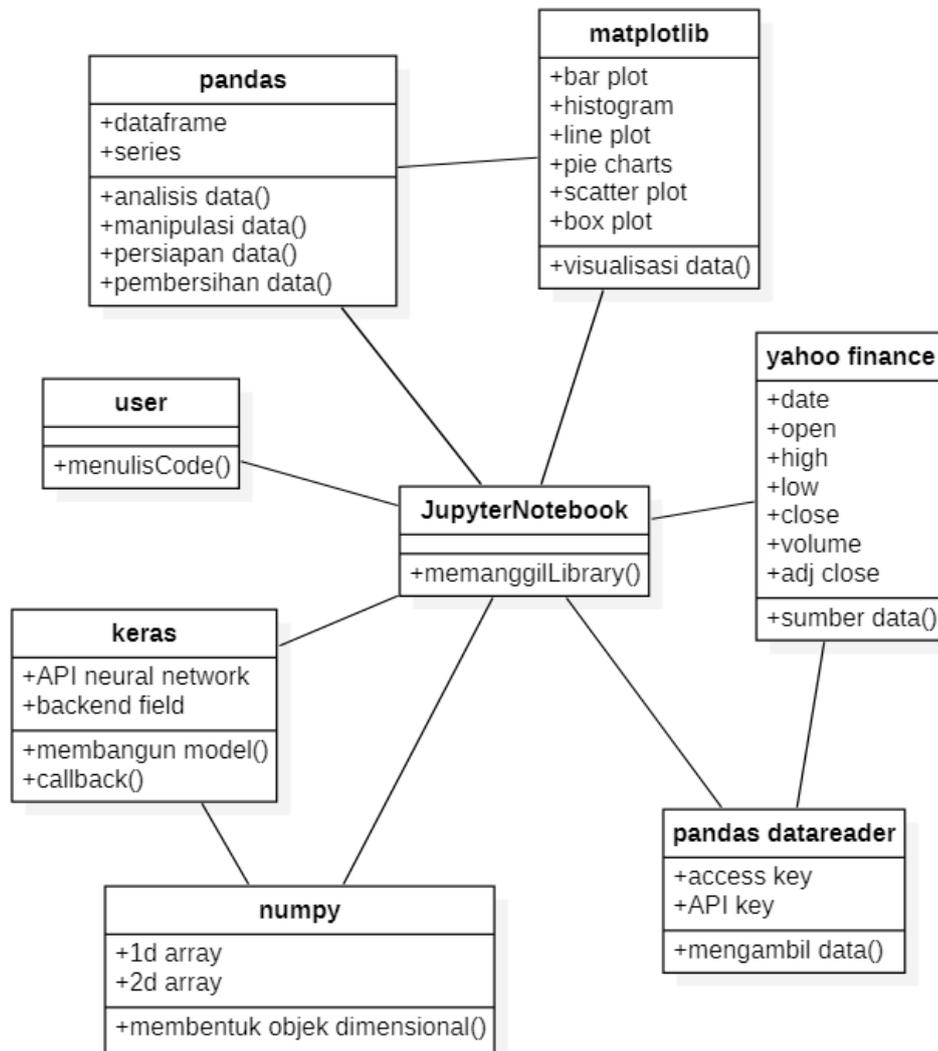
Gambar 3. 2 Activity Diagram sistem analisa harga saham menggunakan algoritma Long Short Term Memory (LSTM)

Penjelasan Gambar 3.2 Activity Diagram sistem analisa harga saham menggunakan algoritma Long Short Term Memory (LSTM).

User ditugaskan untuk meng-*import library python* untuk membuat prediksi harga saham ini, mulai dari *pandas, numpy, keras, matplotlib, dan pandas datareader*. Untuk meng-*import dataset* sistem akan membaca code yang telah dituliskan dan dengan otomatis akan membaca *dataset*, untuk mengambil data saham menggunakan *pandas datareader*. Menampilkan *dataset* berupa tabel yang berisi *date, high, low open, close, volume, dan adj close*. Dalam memprediksi harga saham kali ini, yang digunakan hanya data *close* saja. Menentukan *data training* dimaksudkan untuk memisahkan porsi data untuk di *training* dan data untuk di *testing*, bagian ini dinamakan sebagai *preprocessing data*. Normalisasi data berperan agar model dapat belajar lebih cepat dan meningkatkan

akurasi model, dan meminimalisir *error*, untuk melakukan normalisasi data menggunakan *MinMaxScaler*. Perancangan model *LSTM* dimulai dengan mengawali proses oleh *LSTM*, yang digunakan *hidden layer*, *neuron* (memori sel), *epoch* (iterasi), dan *batch size* (jumlah sampel data). Fase *Training data LSTM* dimulai dengan pemrosesan bobot awal sel *LSTM*, diikuti oleh menentukan nilai yang tidak digunakan, menentukan nilai masukan, memperbarui sel, menentukan nilai keluaran, menghitung nilai *loss*, dan proses ini dilakukan sebanyak yang telah ditentukan. *Testing data* lanjutan proses *training data*, dimana proses ini mengolah parameter hasil *testing* dengan menggunakan data uji. Visualisasi data ini dimaksudkan untuk menampilkan grafik data *real* dengan data dari hasil prediksi, jadi memudahkan untuk dipahami.

3.3.4 Class Diagram



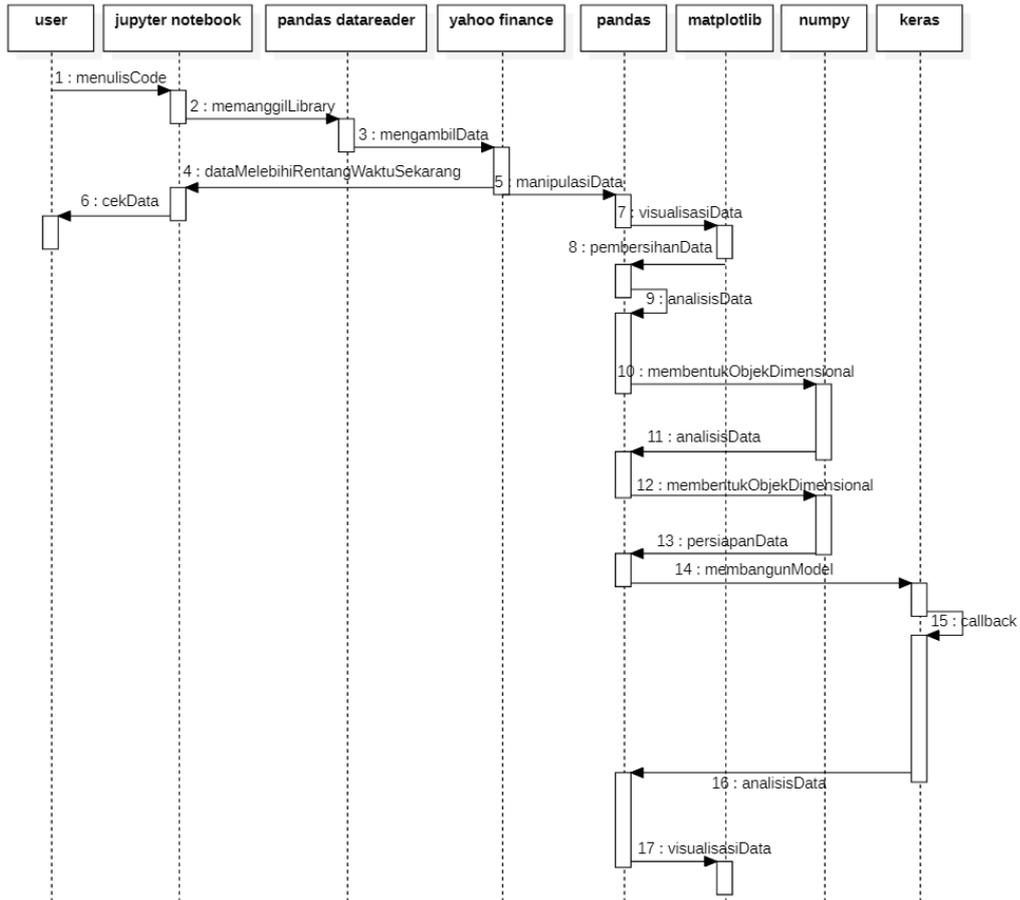
Gambar 3. 3 *Class Diagram* sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)*

Penjelasan Gambar 3.3 *Class Diagram* sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)*.

Pandas datareader berfungsi untuk mengambil dataset yang bersumber dari *Yahoo Finance*, dimana *Pandas datareader* ini bekerja dengan cara mengambil *access key* dari *Yahoo Finance* sehingga data saham dapat diambil dan ditampilkan. *Pandas* merupakan *library python* yang memiliki fungsi untuk menganalisis data, mempersiapkan data, manipulasi data, dan membersihkan data yang tidak terpakai. *Matplotlib* digunakan untuk memvisualisasikan data sehingga data dapat disajikan dalam bentuk grafik sehingga lebih

mudah untuk dilihat. *Numpy* sendiri merupakan *library* yang fungsinya membuat objek dimensional menjadi 1 dimensi *array* atau 2 dimensi *array*. *Keras* bertujuan untuk membangun model *LSTM*.

3.3.5 Sequence Diagram



Gambar 3. 4 Sequence Diagram sistem analisa harga saham menggunakan algoritma Long Short Term Memory (LSTM)

Penjelasan Gambar 3.4 Sequence Diagram sistem analisa harga saham menggunakan algoritma Long Short Term Memory (LSTM).

Pada proses analisa harga saham menggunakan algoritma Long Short Term Memory (LSTM) user bertugas menuliskan code di Jupyter Notebook sehingga Jupyter Notebook dapat meng-import library yang diperlukan untuk proses analisa. Terutama meng-import Pandas Datareader untuk mengambil data saham dari Yahoo Finance, jika data yang terambil tidak lengkap, maka user harus mengecek datanya dan mengecek rentang waktu data saham yang diambil, karena data yang terambil tergantung dari

kekuatan *memory device user*. Setelah data terambil maka *Pandas* akan melakukan manipulasi data dalam bentuk sebuah tabel untuk mempresentasikan data saham yang telah terambil. *Matplotlib* melakukan tugas untuk menampilkan grafik dari tabel yang telah disajikan oleh *Pandas*. Proses selanjutnya *Pandas* akan melakukan pembersihan data dimana, *Pandas* hanya menyeleksi data yang dipakai saja dan membuang data yang tidak diperlukan, pada analisa ini, menggunakan harga *close* pada saham. *Pandas* lanjut memproses data untuk menentukan jumlah *data training* yang diperlukan. Selanjutnya *Numpy* membuat array dari *data training* yang telah di skala kan. Setelah dari *Numpy*, *data training* akan dibagi menjadi dua data yaitu data x dan data y. *Keras* akan membangun model *Long Short Term Memory (LSTM)* dan memproses jumlah *epoch* untuk mendapatkan kesalahan data dalam proses pengujian data, dan hasilnya disebut *callback*. *Pandas* akan menganalisis data untuk menentukan *data testing* lalu diolah untuk menampilkan hasil prediksi. Hasil prediksi di visualisasikan oleh *Matplotlib* sehingga lebih mudah dilihat.

BAB IV

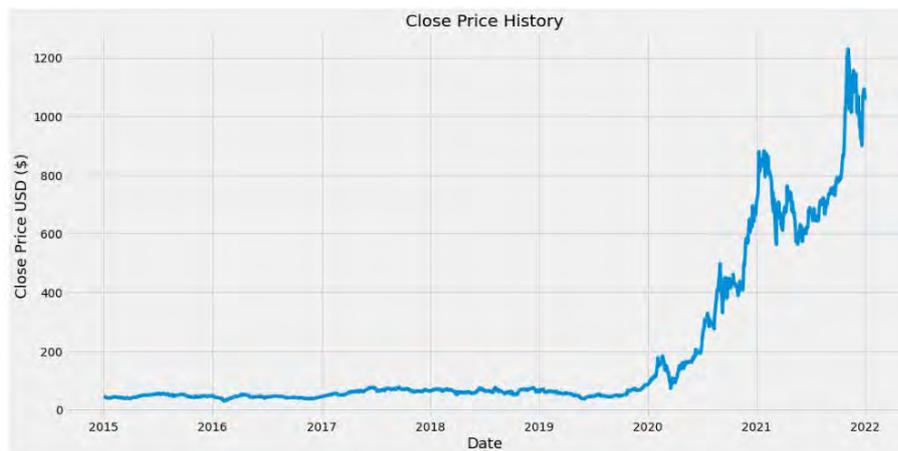
HASIL DAN PEMBAHASAN

4.1 Spesifikasi Perangkat Lunak

Pada pembuatan dan pengujian sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)* diperlukan spesifikasi perangkat keras sebagai berikut.

1. RAM 8 GB
2. Ruang Penyimpanan SSD 256 GB
3. Processor Intel Core i3 Gen 11 (4 CPUs) 3.00 GHz
4. VGA Intel UHD Graphics

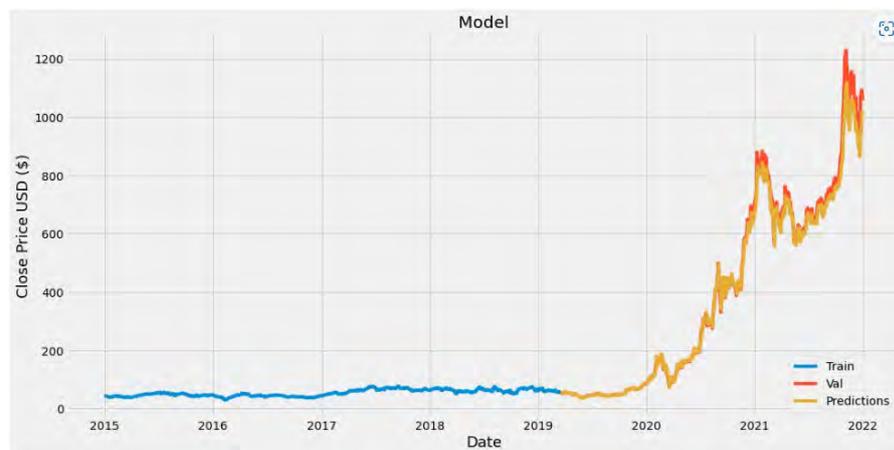
4.2 Menganalisis Pengujian Analisa Harga Saham



Gambar 4. 1 Grafik Data *Real* Harga *Close* Saham Tesla, Inc

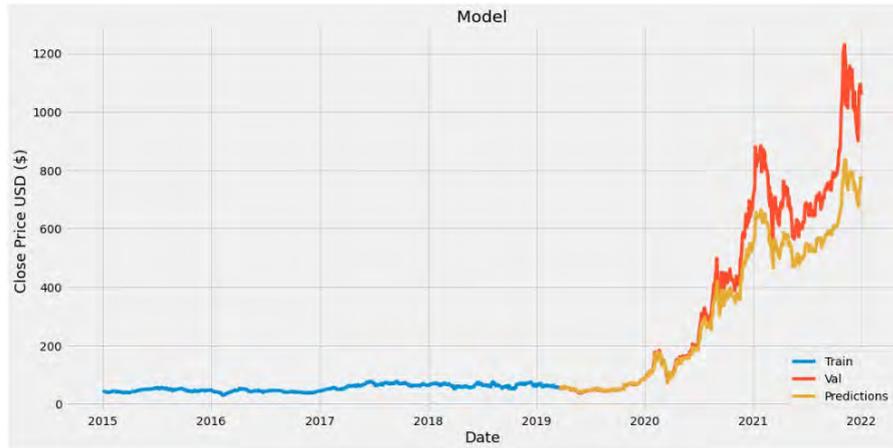
Pada pembuatan dan pengujian data yang digunakan berupa data harga *close* saham *Tesla, Inc. (TSLA)* terhadap *US Dollar*. Data ini dapat disebut juga data *time series*, karena pada data ini memuat nilai pengamatan dari suatu periode waktu yang datanya diambil dari *Yahoo Finance* (dapat diakses melalui <https://finance.yahoo.com/>), data yang digunakan dari rentang waktu 1 Januari 2015 – 1 Januari 2022, dengan jumlah data yang terambil ada 1764 data. Setelah data terambil maka dibagi menjadi *data training* dan *data testing*. Setelah memiliki data harga *close* saham proses selanjutnya adalah melakukan *forecasting* atau melakukan sebuah prediksi dimana saat melakukan prediksi diharapkan grafik prediksi memiliki tingkat kesamaan yang mendekati data aslinya. Algoritma *Long*

Short Term Memory (LSTM) bekerja dengan cara mengingat kumpulan informasi dengan rentang waktu yang panjang dan yang telah ditentukan (*data time series*). Algoritma bekerja menghapus informasi yang tidak relevan dan tidak diperlukan. Algoritma *Long Short Term Memory* efisien dalam memproses, memprediksi, serta mengklasifikasikan *data time series*. Bobot komposisi pembagian *dataset* terbagi menjadi tiga komposisi. Pertama 60% *data training* dan 40% *data testing*. Kedua 70% *data training* dan 30% *data testing*. Ketiga 80% *data training* dan 20% *data testing*. Masing-masing *dataset* diuji dengan komposisi *epoch* yang dimulai dari besar *epoch* 10 hingga 50.



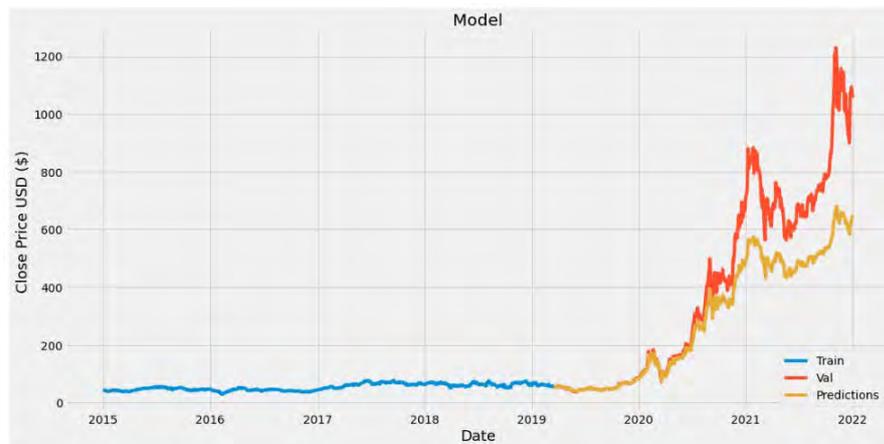
Gambar 4. 2 Grafik Prediksi dengan 60% *Data Training* dan 40% *Data Testing* dan Jumlah *Epoch* 10

Pada pengujian dengan komposisi data 60% *data training* dan 40% *data testing* serta jumlah *epoch* 10 didapat hasil grafik (Gambar 4.2). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 11,167. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 11,167 USD.



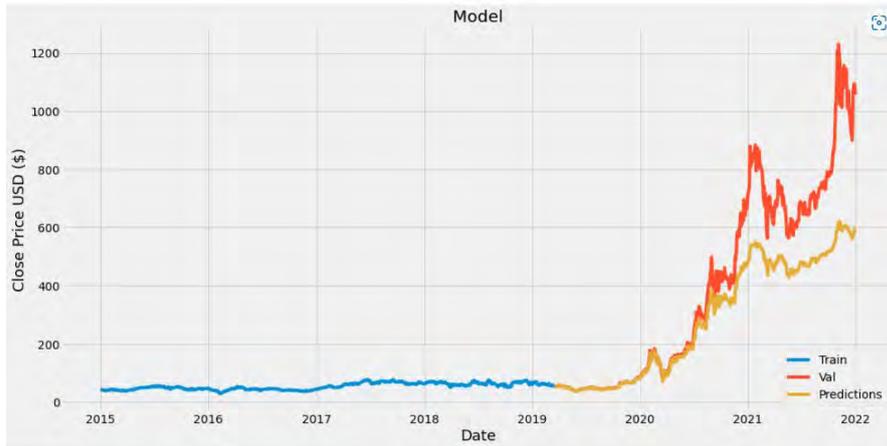
Gambar 4. 3 Grafik Prediksi dengan 60% *Data Training* dan 40% *Data Testing* dan Jumlah *Epoch* 20

Pada pengujian dengan komposisi data 60% *data training* dan 40% *data testing* serta jumlah *epoch* 20 didapat hasil grafik (Gambar 4.3). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 76,446. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 76,446 USD.



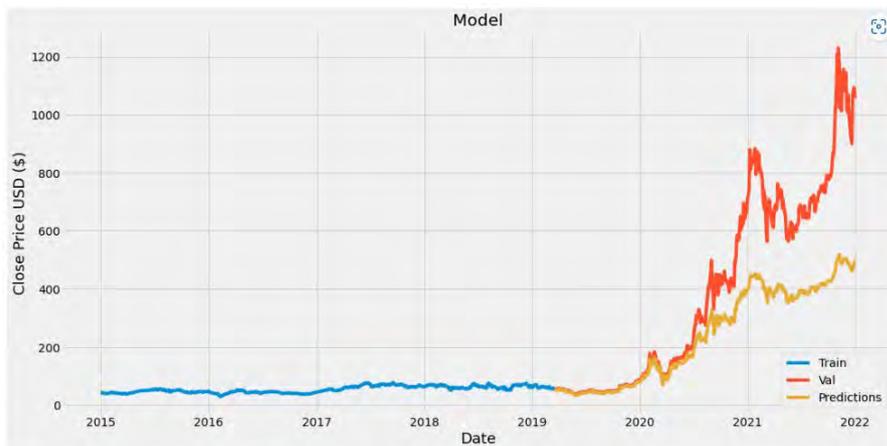
Gambar 4. 4 Grafik Prediksi Dengan 60% *Data Training* dan 40% *Data Testing* dan Jumlah *Epoch* 30

Pada pengujian dengan komposisi data 60% *data training* dan 40% *data testing* serta jumlah *epoch* 30 didapat hasil grafik (Gambar 4.4). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 106,880. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 106,88 USD.



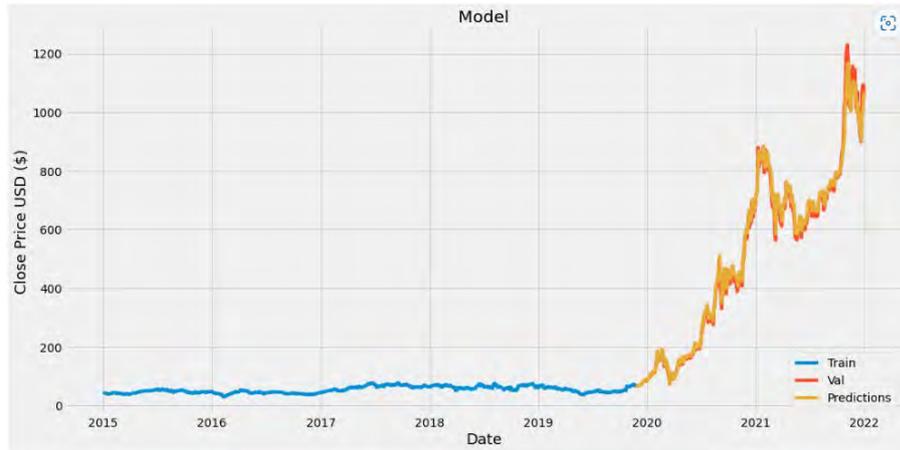
Gambar 4. 5 Grafik Prediksi dengan 60% *Data Training* dan 40% *Data Testing* dan Jumlah *Epoch* 40

Pada pengujian dengan komposisi data 60% *data training* dan 40% *data testing* serta jumlah *epoch* 40 didapat hasil grafik (Gambar 4.5). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 112,995. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 112,996 USD.



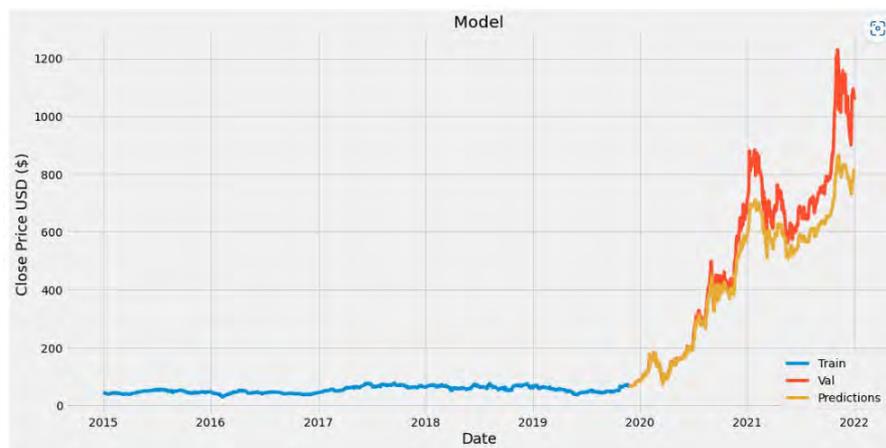
Gambar 4. 6 Grafik Prediksi dengan 60% *Data Training* dan 40% *Data Testing* dan Jumlah *Epoch* 50

Pada pengujian dengan komposisi data 60% *data training* dan 40% *data testing* serta jumlah *epoch* 50 didapat hasil grafik (Gambar 4.6). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 157,867. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 157,867 USD.



Gambar 4. 7 Grafik Prediksi dengan 70% *Data Training* dan 30% *Data Testing* dan Jumlah *Epoch* 10

Pada pengujian dengan komposisi data 70% *data training* dan 30% *data testing* serta jumlah *epoch* 10 didapat hasil grafik (Gambar 4.7). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 5,587. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 5,587 USD.



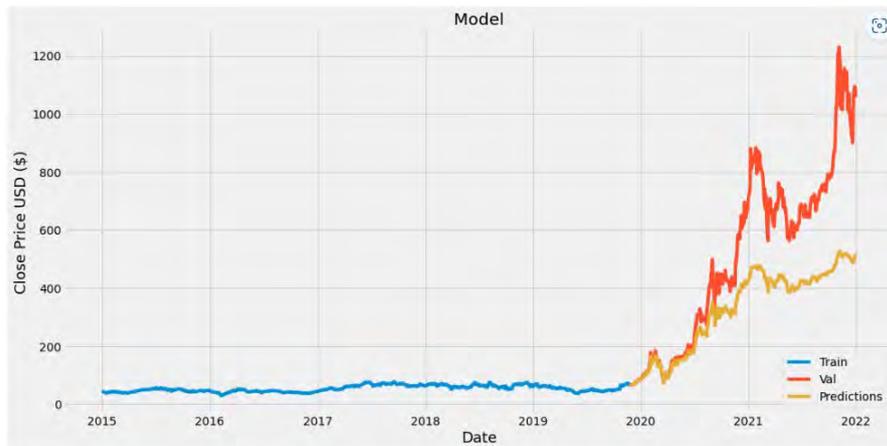
Gambar 4. 8 Grafik Prediksi dengan 70% *Data Training* dan 30% *Data Testing* dan Jumlah *Epoch* 20

Pada pengujian dengan komposisi data 70% *data training* dan 30% *data testing* serta jumlah *epoch* 20 didapat hasil grafik (Gambar 4.8). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 74,165. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 74,165 USD.



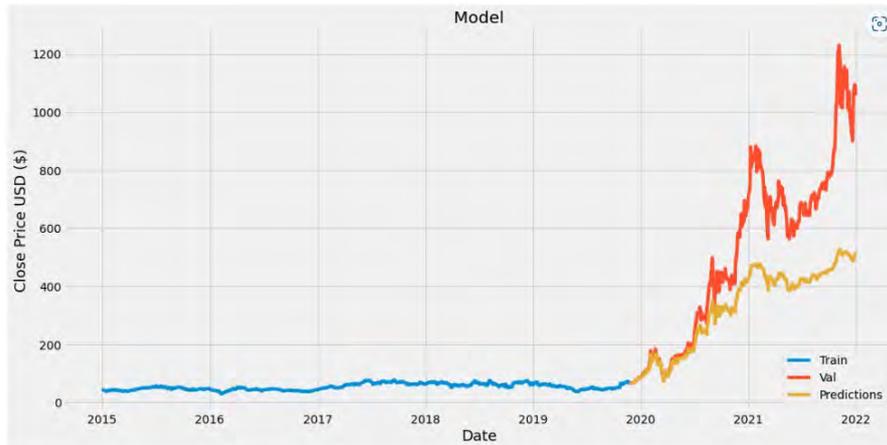
Gambar 4. 9 Grafik Prediksi dengan 70% *Data Training* dan 30% *Data Testing* dan Jumlah *Epoch* 30

Pada pengujian dengan komposisi data 70% *data training* dan 30% *data testing* serta jumlah *epoch* 30 didapat hasil grafik (Gambar 4.9). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 172,720. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 172,720 USD.



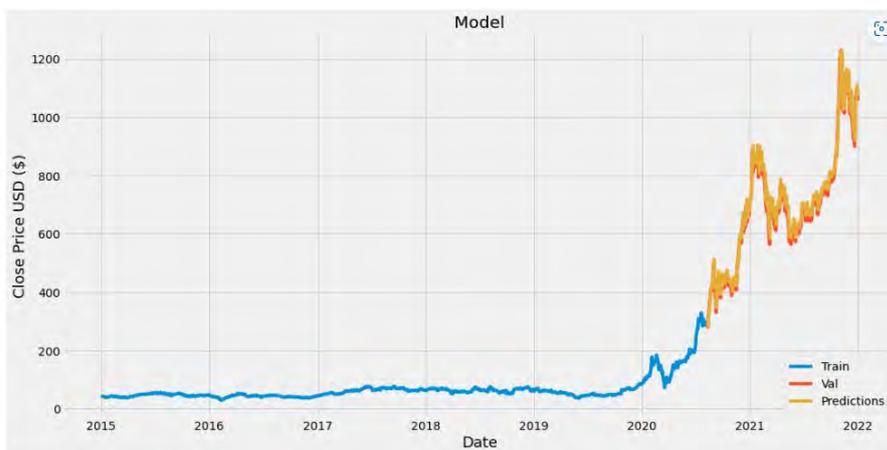
Gambar 4.10 Grafik Prediksi dengan 70% *Data Training* dan 30% *Data Testing* dan Jumlah *Epoch* 40

Pada pengujian dengan komposisi data 70% *data training* dan 30% *data testing* serta jumlah *epoch* 30 didapat hasil grafik (Gambar 4.9). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 172,720. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 172,720 USD.



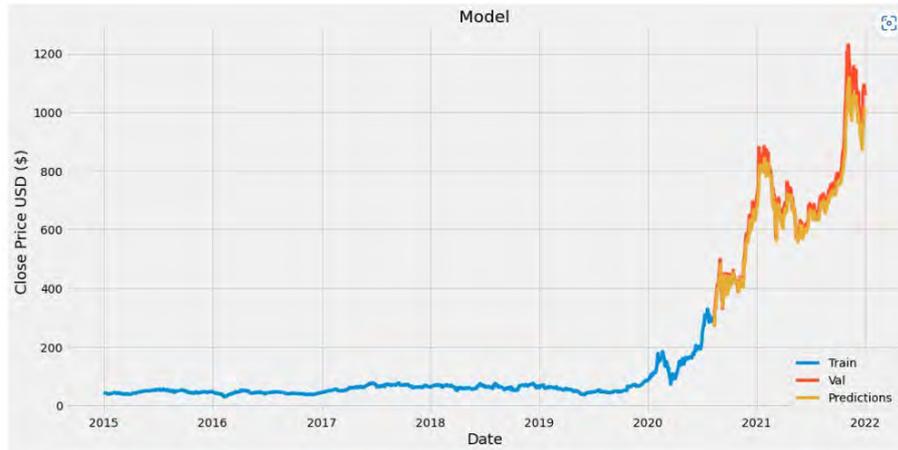
Gambar 4.11 Grafik Prediksi dengan 80% *Data Training* dan 20% *Data Testing* dan Jumlah *Epoch* 10

Pada pengujian dengan komposisi data 70% *data training* dan 30% *data testing* serta jumlah *epoch* 50 didapat hasil grafik (Gambar 4.11). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 186,850. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 186,85 USD.



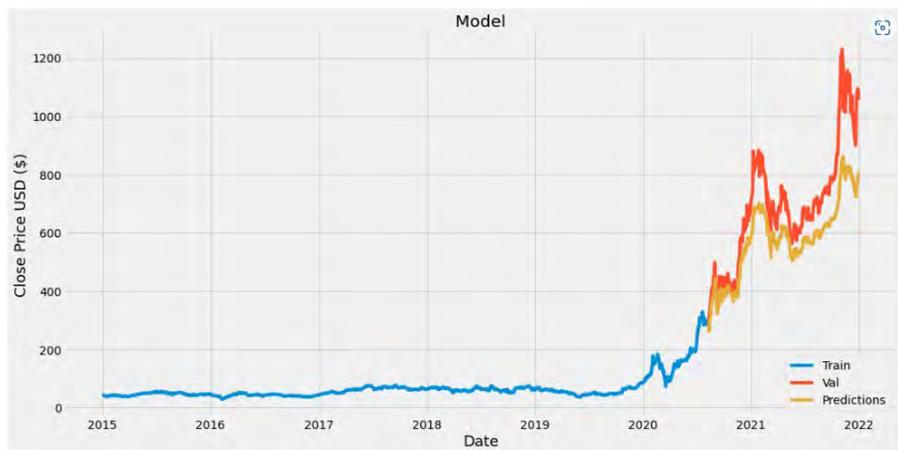
Gambar 4.12 Grafik Prediksi dengan 80% *Data Training* dan 20% *Data Testing* dan Jumlah *Epoch* 10

Pada pengujian dengan komposisi data 80% *data training* dan 20% *data testing* serta jumlah *epoch* 10 didapat hasil grafik (Gambar 4.12). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 15,792. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 15,793 USD.



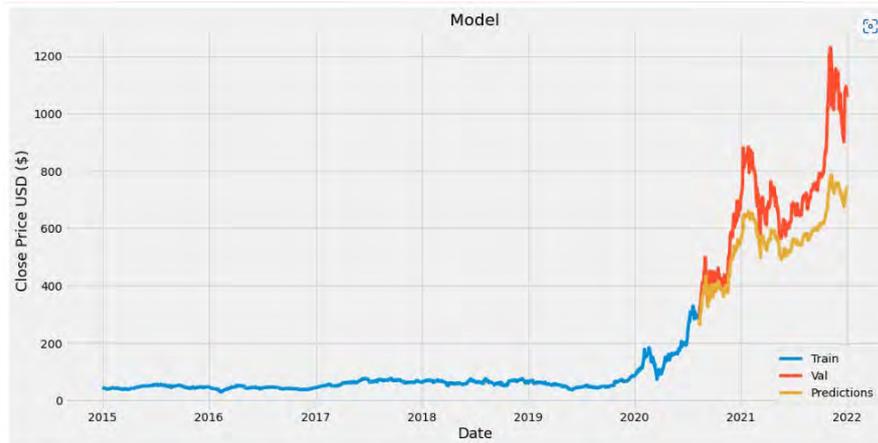
Gambar 4.13 Grafik Prediksi dengan 80% *Data Training* dan 20% *Data Testing* dan Jumlah *Epoch* 20

Pada pengujian dengan komposisi data 80% *data training* dan 20% *data testing* serta jumlah *epoch* 20 didapat hasil grafik (Gambar 4.13). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 25,409. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 25,409 USD.



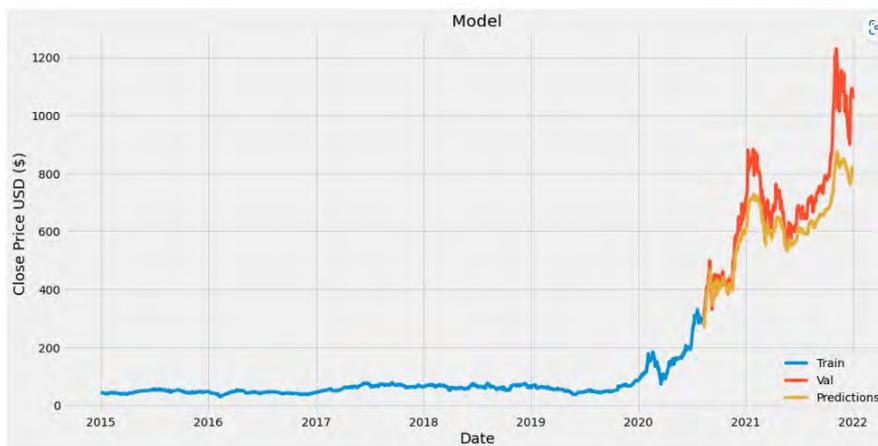
Gambar 4.14 Grafik Prediksi dengan 80% *Data Training* dan 20% *Data Testing* dan Jumlah *Epoch* 30

Pada pengujian dengan komposisi data 80% *data training* dan 20% *data testing* serta jumlah *epoch* 30 didapat hasil grafik (Gambar 4.14). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 144,418. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 114,419 USD.



Gambar 4.15 Grafik Prediksi dengan 80% *Data Training* dan 20% *Data Testing* dan Jumlah *Epoch* 40

Pada pengujian dengan komposisi data 80% *data training* dan 20% *data testing* serta jumlah *epoch* 40 didapat hasil grafik (Gambar 4.15). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 140,989. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 140,989 USD.



Gambar 4.16 Grafik Prediksi dengan 80% *Data Training* dan 20% *Data Testing* dan Jumlah *Epoch* 50

Pada pengujian dengan komposisi data 80% *data training* dan 20% *data testing* serta jumlah *epoch* 50 didapat hasil grafik (Gambar 4.16). Besaran nilai *RMSE* dari hasil pengujian tersebut sebesar 89,416. Selisih harga prediksi dengan harga *real* dari hasil pengujian sekitar 89,416 USD.

Tabel 4. 1 Analisis pengujian analisa harga saham

No	Komposisi Data	Epoch	Rata-rata harga close (\$)	Rata-rata harga prediction (\$)	keakuratan (%)
1	60% data training	10	397,9126091	386,7450048	97%
	40% data testing	20	397,9126091	321,4667755	81%
		30	397,9126091	291,0324576	73%
		40	397,9126091	284,9168395	72%
		50	397,9126091	240,0454064	60%
2	70% data training	10	513,6336775	519,2209223	99%
	30% data testing	20	513,6336775	439,4682901	85%
		30	513,6336775	340,9127369	66%
		40	513,6336775	325,145793	63%
		50	513,6336775	326,7830925	64%
3	80% data training	10	692,6265839	708,4192761	98%
	20% data testing	20	692,6265839	667,217393	96%
		30	692,6265839	578,2077391	83%
		40	692,6265839	551,6372954	80%
		50	692,6265839	603,2103895	87%

Berdasarkan data dari tabel yang tersaji komposisi data dengan 70% *data training* dan 30% *data testing*, data dengan pengujian menggunakan 10 *epoch* memiliki keakuratan yang lebih tinggi daripada data yang lain, dengan tingkat keakuratan mencapai 99% dengan selisih rata-rata harga prediksi dengan rata-rata harga *real close* sebesar 5,587 USD.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan pengujian dan menganalisa hasil uji pada pengerjaan sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)*.

1. Berdasarkan hasil analisa data dengan komposisi 70% *data training* dan 30% *data testing* dengan jumlah *epoch* 10 memiliki keakuratan yang lebih tinggi dengan angka keakuratan sekitar 99%.
2. Berdasarkan hasil analisa tingkat keakuratan tertinggi yaitu 99% dengan selisih 5,587 USD

5.2 Saran

Adapun dalam pengerjaan tugas akhir yang telah penulis buat. Maka penulis berharap kedepannya dapat mengembangkan sistem analisa harga saham menggunakan algoritma *Long Short Term Memory (LSTM)*.

1. Melakukan prediksi dengan pengambilan data yang lebih banyak dari yang sekarang.
2. Membandingkan kinerja algoritma *Long Short Term Memory (LSTM)* dengan algoritma lain.

DAFTAR PUSTAKA

- Izzah, N. A., Martia, D. Y., Imaculata, M., Hidayatullah, M. I., Pradana, A. B., Setiyani, D. A., & Sapuri, E. (2021). Analisis Teknikal Pergerakan Harga Saham Dengan Menggunakan Indikator Stochastic Oscillator Dan Weighted Moving Average. *Keunis*, 9(1), 36. <https://doi.org/10.32497/keunis.v9i1.2307>
- Al'afi, A. M., Widiarti, Kurniasari, D., & Usman, M. (2020). Peramalan Data Time Series Seasonal Menggunakan Metode Analisis Spektral. *Jurnal Siger Matematika*, 1(1), 10–15. <https://doi.org/10.23960/jsm.v1i1.2484>
- Audina, B., Fatekurohman, M., & Riski, A. (2021). Peramalan Arus Kas dengan Pendekatan Time Series Menggunakan Support Vector Machine. *Indonesian Journal of Applied Statistics*, 4(1), 34. <https://doi.org/10.13057/ijas.v4i1.47953>
- B Arnold, T. (2017). kerasR: R Interface to the Keras Deep Learning Library. *The Journal of Open Source Software*, 2(14), 296. <https://doi.org/10.21105/joss.00296>
- Bhevsar, H., & Manglani, R. (2019). IRJET- Sentiment Analysis of Twitter Data using Python. *International Research Journal of Engineering and Technology (IRJET)*, 06(03).
- Bolung, M., & Tampangela, H. R. K. (2017). Analisa Penggunaan Metodologi Pengembangan Perangkat Lunak. *Jurnal ELTIKOM*, 1(1), 1–10. <https://doi.org/10.31961/eltikom.v1i1.1>
- Fauzi, A. (2019). Forecasting Saham Syariah Dengan Menggunakan Lstm. *Al-Masraf: Jurnal Lembaga Keuangan Dan Perbankan*, 4(1), 65. <https://doi.org/10.15548/al-masraf.v4i1.235>
- Gananjay, T. (2021). Analysis and Evaluation of Technical Indicators for Prediction of Stock Market. *International Journal of Engineering Research & Technology (IJERT)*, 10(May), 341–344. <https://doi.org/10.13140/RG.2.2.25727.53924>
- Hansun, S. (2013). Peramalan Data IHSG Menggunakan Fuzzy Time Series. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 7(1), 79–88. <https://doi.org/10.22146/ijccs.2155>
- Harlina, S., & Usman. (2020). Analisa Prediktif Curah Hujan Data Time Series Berbasis Metode Neural Network. *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, 10(2), 163. <https://doi.org/10.35585/inspir.v10i2.2586>
- Hasanah, F., & Untari, R. (2018). Buku Ajar Rekayasa Perangkat Lunak. In *Buku Ajar Rekayasa Perangkat Lunak*. <https://doi.org/10.21070/2018/978-602-5914-09-6>
- Hendarsih, I. (2016). Analisis Perubahan Harga Saham Dengan Menggunakan Grafik Candlestick. *Jurnal Akuntansi Dan Keuangan*, 3(2), 186–197.
- Hendrajati, A., & Widyatmoko, K. (2013). Jurnal Rekayasa Perangkat Lunak Sentra Pelayanan Kepolisian Terpadu (SPKT) Pada POLRESTABES Semarang. *Polrestabes Semarang*, 11.
- Irmayani, D. (2019). Rekayasa Perangkat Lunak. *Jurnal Informatika*, 2(3), 1–9. <https://doi.org/10.36987/informatika.v2i3.201>
- Jiang, Z., & Shen, G. (2019). Prediction of house price based on the back propagation neural network in the keras deep learning framework. *2019 6th International Conference on Systems and Informatics, ICSAI 2019, Icsai*, 1408–1412. <https://doi.org/10.1109/ICSAI48974.2019.9010071>
- Lawrence, A., Ryans, J. P., Sun, E., & Laptev, N. (2016). Yahoo Finance Search and Earnings Announcements. *SSRN Electronic Journal*.

<https://doi.org/10.2139/ssrn.2804353>

- Lawrence, A., Ryans, J., Sun, E., & Laptev, N. (2018). Earnings announcement promotions: A Yahoo Finance field experiment. *Journal of Accounting and Economics*, 66(2–3), 399–414. <https://doi.org/10.1016/j.jacceco.2018.08.004>
- Manaswi, N. K. (2018). Deep Learning with Applications Using Python. *Deep Learning with Applications Using Python*, 31–43. <https://doi.org/10.1007/978-1-4842-3516-4>
- Moolayil, J. (2019). Learn Keras for Deep Neural Networks. *Learn Keras for Deep Neural Networks, DI*, 1–16. <https://doi.org/10.1007/978-1-4842-4240-7>
- Mulyani, S., Hayati, D., & Sari, A. N. (2021). Analisis Metode Peramalan (Forecasting) Penjualan Sepeda Motor Honda Dalam Menyusun Anggaran Penjualan Pada Pt Trio Motor Martadinata Banjarmasin. *Jurnal Ekonomi Dan Bisnis*, 14(1), 178–188.
- Nabila, F. (2017). *Jurnal faktor-faktor yang mempengaruhi*. 01(03).
- Nasution, A. (2019). METODE WEIGHTED MOVING AVERAGE DALAM M-FORECASTING PENDAHULUAN Teknologi mobile sekarang ini berkembang pesat , seperti terlihat dari lembaga riset digital marketing , emarketer yang memperkirakan pada 2018 jumlah pengguna aktif smartphone di Indonesia le. V(2), 119–124.
- Nelli, F. (2018). Python Data Analytics. In *Python Data Analytics*. <https://doi.org/10.1007/978-1-4842-3913-1>
- Nongthombam, K., & Sharma, D. (2021). *Data Analysis using Python - Sales Analysis*. 10(07), 463–468. <https://www.storiesondata.com/post/data-analysis-using-python-sales-analysis>
- Pajankar, A. (2021). Practical Python Data Visualization. In *Practical Python Data Visualization*. <https://doi.org/10.1007/978-1-4842-6455-3>
- Pérez, F., & Granger, B. E. (2015). Project Jupyter : Computational Narratives as the Engine of Collaborative Data Science. *UC Berkeley*, April, 1–24. <http://archive.ipython.org/JupyterGrantNarrative-2015.pdf>
- Pimentel, J. F., Murta, L., Braganholo, V., & Freire, J. (2019). A large-scale study about quality and reproducibility of jupyter notebooks. *IEEE International Working Conference on Mining Software Repositories, 2019-May*, 507–517. <https://doi.org/10.1109/MSR.2019.00077>
- Pitrawati, P., & Ningsih, M. W. A. (2017). Rekayasa Perangkat Lunak Pada Informasi Pemetaan Lokasi Wisata Di Lampung. *Jurnal Cendikia*, 14(2), 27–32. <https://jurnal.dcc.ac.id/index.php/JC/article/view/8>
- Prakoso, R. (2016). Analisis Faktor-Faktor yang Mempengaruhi Return Saham (Studi Empiris pada Perusahaan Manufaktur yang Go Public di Bursa Efek Indonesia Tahun 2011-2014). *Jurnal Bisnis Indonesia*, 05(April 2016), 5–20.
- Retnoningsih, E., Shadiq, J., & Oscar, D. (2017). Pembelajaran Pemrograman Berorientasi Objek (Object Oriented Programming) Berbasis Project Based Learning. *Informatics for Educators and Professionals*, 2(1), 95 – 104. <http://ejournal-binainsani.ac.id/index.php/ITBI/article/view/668>
- Rizkiyani, M. (2014). PENERAPAN FORECASTING METHODS UNTUK MENINGKATKAN STRATEGI Sistem Informasi Fakultas Ilmu Komputer. *Sistem Informasi Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semarang*, 12. http://eprints.dinus.ac.id/13341/1/jurnal_13927.pdf
- Rolon-Mérette, D., Ross, M., Rolon-Mérette, T., & Church, K. (2020). Introduction to Anaconda and Python: Installation and setup. *The Quantitative Methods for Psychology*, 16(5), S3–S11. <https://doi.org/10.20982/tqmp.16.5.s003>

- Saraswati, H. (2020). Dampak Pandemi Covid-19 Terhadap Pasar Saham Di Indonesia. *JAD: Jurnal Riset Akuntansi Dan Keuangan Dewantara*, 3(2), 153–163.
- Sharma, S., & Rengarajan, A. (2020). Hand Gesture Recognition Using OpenCv and Python. *New Trends in Computational Vision and Bio-Inspired Computing*, 5(2), 1711–1719. https://doi.org/10.1007/978-3-030-41862-5_174
- Sheela, A., Meena, K., & Ranjani, J. (2019). Combination of NumPy , SciPy and Matplotlib / Pylab - a good alternative methodology to MATLAB - A Comparative analysis. *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 1–5.
- Sheppard, K. (2013). Introduction to Python for Econometrics, Statistics and Data Analysis. *Report*, 363.
- Suárez-García, A., Arce-Fariña, E., Álvarez Hernández, M., & Fernández-Gavilanes, M. (2021). Teaching structural analysis theory with Jupyter Notebooks. *Computer Applications in Engineering Education*, 29(5), 1257–1266. <https://doi.org/10.1002/cae.22383>
- Wahyu, W. (2021). *RECURRENT NEURAL NETWORK-LONG SHORT TERM MEMORY*.
- Wang, J., Li, L., & Zeller, A. (2020). Better Code, Better Sharing: On the Need of Analyzing Jupyter Notebooks. *Proceedings - 2020 ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results, ICSE-NIER 2020*, 1, 53–56. <https://doi.org/10.1145/3377816.3381724>
- Wardani, R. (2012). Program Pendidikan dan Pelatihan Profesi Guru Teknologi Informasi dan Komunikasi Rekayasa Perangkat Lunak. *Program Pendidikan Dan Pelatihan Profesi Guru Teknologi Informasi Dan Komunikasi*, 1–43.
- Wiranda, L., & Sadikin, M. (2019). Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk Pt. Metiska Farma. *Jurnal Nasional Pendidikan Teknik Informatika*, 8(3), 184–196.
- Yusman, N. (2017). Perancangan Sistem Informasi Berbasis Orientasi Objek Menggunakan Star Uml. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699.

LAMPIRAN

LISTING PROGRAM

```
#library
import pandas as pd
import math
import numpy as np
import pandas_datareader as pdd
from sklearn.preprocessing import MinMaxScaler
from keras.layers import Dropout
from keras.models import Sequential
from keras.layers import Dense, LSTM
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')

#dataset
df = pdd.DataReader('TSLA', data_source='yahoo',start='2015-01-01',end='2022-01-01')
df

#ukuran dataset
df.shape

#visualisasi close price
plt.figure(figsize=(16,8))
plt.title('Close Price History')
plt.plot(df['Close'])
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price USD ($)',fontsize=18)
plt.show()

#dataframe close price
data = df.filter(['Close'])
#mengubah dataframe menjadi array(numpy)
dataset = data.values
#menentukan data training *math.ceil untuk mengelompokkan angka ke atas
training_data_len = math.ceil(len(dataset)*.8)
training_data_len

#menskalakan data dengan rentang antara 0 dan 1
scaler = MinMaxScaler(feature_range=(0,1))
#menampung objek dataset yang telah diskalakan
scaled_data = scaler.fit_transform(dataset)
scaled_data

#membuat penskalaan training dataset
train_data = scaled_data[0:training_data_len,:]
#membagi data training (x dan y)
x_train = []
y_train = []
for i in range(60,len(train_data)):
    x_train.append(train_data[i-60:i,0])
    y_train.append(train_data[i,0])
    if i<= 61:
        print(x_train)
        print(y_train)

#konversi x_train y_train menjadi array
```

```

x_train, y_train = np.array(x_train), np.array(y_train)

#reshape data [0]=1175, [1]=60
x_train = np.reshape(x_train,(x_train.shape[0],x_train.shape[1],1))
x_train.shape

#membuat model LSTM
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(x_train.shape[1],1)))
model.add(LSTM(50,return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))

#compile model
model.compile(optimizer='adam',loss='mean_squared_error')

#train data
model.fit(x_train,y_train,batch_size=1,epochs=10)

#create testing dataset
test_data = scaled_data[training_data_len - 60:, :]
x_test = []
y_test = dataset[training_data_len:, :]
for i in range(60, len(test_data)):
    x_test.append(test_data[i-60:i,0])

x_test = np.array(x_test)
x_test = np.reshape(x_test,(x_test.shape[0],x_test.shape[1],1))
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
rmse = np.sqrt(np.mean(predictions - y_test)**2)
rmse

train = data[:training_data_len]
valid = data[training_data_len:]
valid['Predictions'] = predictions
plt.figure(figsize=(16,8))
plt.title('Model')
plt.xlabel('Date', fontsize = 18)
plt.ylabel('Close Price USD ($)', fontsize = 18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Val', 'Predictions'], loc='lower right')
plt.show()

valid

```