

**PREDIKSI HARGA BITCOIN MENGGUNAKAN METODE  
*LONG SHORT TERM MEMORY***

S K R I P S I



**MUHAMMAD FAJRI RAMADHAN**

**F1E119141**

**PROGRAM STUDI SISTEM INFORMASI  
JURUSAN TEKNIK ELEKTRO DAN INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS JAMBI**

**2024**

**SURAT PERNYATAAN**

Dengan ini saya menyatakan bahwa skripsi ini benar-benar karya sendiri. Sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain kecuali sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim.

Tanda tangan yang tertera dalam halaman pengesahan adalah asli. Jika tidak asli, saya siap menerima sanksi sesuai dengan peraturan yang berlaku.

Jambi

Yang menyatakan

MUHAMMAD FAJRI RAMADHAN

F1E119141

## ABSTRAK

Investasi menjadi salah satu kegiatan yang digemari masyarakat karena dapat meningkatkan keuntungan dari kegiatan tersebut, Pada saat ini, sudah banyak ragam instrument-instrumen investasi yang tersedia seperti saham, obligasi, emas, properti dan yang terbaru adalah *cryptocurrency* (mata uang kripto). Sejak didirikan pada 2008, bitcoin telah muncul sebagai mata uang digital utama dalam hal kapitalisasi pasar dan terus menarik perhatian investor, Transaksi bitcoin pertama kali terjadi pada Januari 2009. Lebih dari dua tahun kemudian, berbagai laporan memperkirakan sirkulasi bitcoin menjadi lebih dari 6,5 juta dengan sekitar 10.000 pengguna, Hal ini menyebabkan semakin banyak masyarakat yang tertarik untuk melakukan investasi terhadap bitcoin. Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, Berdasarkan fakta tersebut, penting dilakukan penelitian mengenai prediksi harga bitcoin untuk mengetahui bagaimana perkembangan harga bitcoin pada masa yang akan datang, dengan memprediksi harga bitcoin pada masa yang akan datang, investor dapat memperkirakan harga bitcoin dari prediksi harga yang telah dilakukan agar tidak terjadi kesalahan dalam melakukan investasi sehingga dapat mencegah besarnya potensi kerugian berdasarkan hasil penelitian didapatkan tingkat akurasi RMSE sebesar 478.237 serta nilai MAE yang dihasilkan sebesar 330.322 dan juga didapatkan hasil akurasi MAPE sebesar 99.01%. Sehingga dapat disimpulkan bahwa model dan website yang telah mampu melakukan prediksi harga bitcoin

**Kata Kunci:** LSTM, Bitcoin, Cryptocurrency, Akurasi, Deep Learning

**PREDIKSI HARGA BITCOIN MENGGUNAKAN METODE  
*LONG SHORT TERM MEMORY***

**S K R I P S I**

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana pada  
program studi sistem informasi



**MUHAMMAD FAJRI RAMADHAN**

**F1E119141**

**PROGRAM STUDI SISTEM INFORMASI  
JURUSAN TEKNIK ELEKTRO DAN INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS JAMBI**

**2024**

## PENGESAHAN

Skripsi dengan judul **PREDIKSI HARGA BITCOIN MENGGUNAKAN METODE LONG SHORT TERM MEMORY** yang disusun oleh **MUHAMMAD FAJRI RAMADHAN, NIM: F1E119141** telah dipertahankan di depan tim penguji pada tanggal 03 januari 2024 dan dinyatakan lulus.

### Susunan Tim Penguji:

Ketua : Edi Saputra, S.T., M.Sc.  
Anggota :1. Zainil Abidin, S.T., M.Eng.  
2. Yolla Noverina, S.Kom., M.Kom.

### Disetujui:

Pembimbing Utama

Pembimbing Pendamping

Dewi Lestari, S.Kom., M.S.I  
NIP. 198410092019032014

Ulfa Khaira, S.Komp., M.Kom.  
NIP. 198912292019032018

### Diketahui:

Dekan Fakultas Sains dan Teknologi Ketua Jurusan Teknik Elektro dan Informatika

Drs. Jefri Marzal, M.Sc., D.I.T  
NIP. 196806021993031004

Nehru, S.Si., M.T  
NIP. 197602082001121002

## RIWAYAT HIDUP



Muhammad Fajri Ramadhan, lahir di Kuala Tungkal, pada tanggal 11 Desember 2001. Penulis merupakan anak Kedua dari 5 bersaudara dari pasangan Harnuni dan Nasiah. Jalur pendidikan formal yang ditempuh penulis adalah sebagai berikut:

1. Sekolah Dasar di SD Negeri 187 Tebing Tinggi Tahun 2006-2012,
2. Madrasah Tsanawiyah di Pondok Pesantren Darunnajah Jakarta 2012-2015,
3. SMAN 11 Kota Jambi tahun 2015-2018.

Saat ini Penulis melanjutkan pendidikannya di salah satu Perguruan Tinggi Negeri, yaitu Universitas Jambi. Penulis sedang menjalani semester 9 di program studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi. Selama menempuh pendidikan jenjang S1, penulis cukup aktif dalam bidang akademik dan non akademik.

## PRAKATA

Puji dan syukur penulis panjatkan atas berkat dan rahmat Allah SWT yang telah memberikan rahmat dan karunia-Nya, sehingga terselesaikannya skripsi yang berjudul "**PREDIKSI HARGA BITCOIN MENGGUNAKAN METODE *LONG SHORT TERM MEMORY***".

Selama pelaksanaan penelitian dan penyusunan tugas akhir ini penulis banyak dibantu dan didukung oleh berbagai pihak. Untuk itu penulis ingin mengucapkan terima kasih kepada:

1. Drs. Jefri Marzal, M.Sc., D.I.T. selaku Dekan Fakultas Sains dan Teknologi Universitas Jambi.
2. Ibu Dewi Lestari S.Kom., M.S.I. selaku pembimbing utama dan Ibu Ulfa Khaira, S.Komp., M.Kom. selaku pembimbing pendamping yang memberikan banyak dukungan dan ilmu dalam proses penyelesaian skripsi ini.
3. Bapak Edi Saputra, S.T., M.Sc. , Zainil Abidin, S.T., M.Eng. , Yolla Noverina, S.Kom., M.Kom. Sebagai Tim Penguji, yang telah banyak memberikan banyak saran dan masukan untuk penyelesaian skripsi ini.
4. Bapak Edi Saputra, S.T., M.Sc, selaku Ketua Prodi Sistem Informasi.
5. Kedua orang tua saya yang tercinta ayah(Harnuni) dan ibu (Nasiah), serta yang saya sayangi kakak dan adik-adik saya yang telah mendukung saya dan memberikan doa yang terbaik bagi saya selama mengerjakan skripsi ini.
6. Seluruh dosen Fakultas Sains dan Teknologi Universitas Jambi, terkhusus untuk dosen Prodi Sistem Informasi yang telah memberikan banyak ilmu kepada penulis selama masa perkuliahan
7. Kepada rekan-rekan seperjuangan yaitu mahasiswa program studi Sistem Informasi angkatan 2019 Fakultas Sains dan Teknologi Universitas Jambi yang saling mendukung dan berjuang bersama hingga saat ini.
8. Semua pihak yang turut membantu penulis selama menyelesaikan skripsi ini baik secara langsung ataupun tidak langsung, yang tidak dapat disebutkan satu persatu.

Penulis menyadari masih terdapat banyak kekurangan dalam pembuatan tugas akhir ini. Oleh sebab itu, penulis berharap mendapat masukan dan saran yang bersifat membangun guna memperbaiki dan melengkapi tugas akhir ini. Penulis berharap laporan ini dapat memberikan manfaat kepada pembaca karena sebaik-baik makhluk adalah yang bermanfaat bagi makhluk lainnya.

## DAFTAR ISI

SURAT PERNYATAAN .....	ii
ABSTRAK.....	iii
PENGESAHAN .....	v
RIWAYAT HIDUP .....	vi
PRAKATA.....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN .....	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan Penelitian .....	4
1.4 Manfaat Penelitian .....	4
1.5 Batasan Penelitian.....	4
BAB II LANDASAN TEORI.....	5
2.1 Prediksi .....	5
2.2 <i>Deep Learning</i> .....	5
2.3 <i>Neural Network</i> .....	6
2.4 Time Series.....	6
2.5 <i>Artificial Neural Network</i> .....	9
2.5 <i>Perbandingan Metode</i> .....	11
2.6 <i>Long Short Term Memory</i> .....	12
2.7 <i>Adaptive Moment Estimation (Adam)</i> .....	17
2.8 Evaluasi .....	17
2.9 <i>Bitcoin</i> .....	18
2.10 Penelitian Terdahulu.....	19
BAB III METODE PENELITIAN .....	22

3.1 Tempat dan Waktu Penelitian.....	22
3.2 Perangkat Penelitian.....	22
3.3 Tahapan Penelitian.....	22
3.4 Jenis dan Sumber Data.....	26
3.5 Teknik Pengumpulan Data.....	27
3.6 Metode Analisis Data.....	27
BAB IV HASIL DAN PEMBAHASAN .....	29
4.1 Penyiapan Data.....	29
4.2 Praproses Data .....	30
4.3 Pengolahan Model LSTM.....	32
4.4 Prediksi Data .....	35
4.5 Implementasi GUI( <i>Graphical User Interface</i> ).....	37
4.6 Evaluasi Model.....	40
BAB V KESIMPULAN DAN SARAN .....	44
5.1 Kesimpulan.....	44
5.2 Saran.....	44
DAFTAR PUSTAKA.....	45
LAMPIRAN .....	48

## DAFTAR GAMBAR

<b>Gambar</b> 1. Pola Data Trend(Andini & Auristandi, 2016).....	7
<b>Gambar</b> 2. Pola Data Siklis(Andini & Auristandi, 2016) .....	8
<b>Gambar</b> 3. Pola Data Musiman(Andini & Auristandi, 2016).....	8
<b>Gambar</b> 4. Pola Data Fluktuasi Tak Beraturan(Darmawan & dkk, 2016) .....	9
<b>Gambar</b> 5. Blockchain Sederhana(Pate1, 2019).....	10
<b>Gambar</b> 6. Struktur LSTM(Qaiu, Wang, & Zhou, 2020).....	13
<b>Gambar</b> 7. Struktur LSTM Forget Gate((Olah, 2015) .....	14
<b>Gambar</b> 8. Struktur LSTM Input Gate(Olah, 2015).....	15
<b>Gambar</b> 9. Struktur LSTM Cell State(Olah, 2015) .....	15
<b>Gambar</b> 10. Struktur LSTM Output Gate(Olah, 2015).....	16
<b>Gambar</b> 11. Alur Penelitian.....	23
<b>Gambar</b> 12. Grafik harga bitcoin dari tahun 2021-2023.....	30
<b>Gambar</b> 13. Hasil Bobot Dari Setiap Neuron.....	34
<b>Gambar</b> 14. Bias Untuk Setiap Neuron.....	35
<b>Gambar</b> 15. Perbandingan Data Prediksi dan Data Aktual .....	36
<b>Gambar</b> 16. Nilai MAPE Dari Model Terbaik .....	37
<b>Gambar</b> 17. Halaman Utama GUI Prediksi Harga Bitcoin .....	38
<b>Gambar</b> 18. GUI Chart Harga Bitcoin Yang di Tambahkan Teknikal Analisis .....	39
<b>Gambar</b> 19. <i>Chart</i> Hasil Prediksi Harga Bitcoin dan Harga Asli Bitcoin .....	39
<b>Gambar</b> 20. Hasil Prediksi Harga Bitcoin.....	40

**DAFTAR TABEL**

<b>Tabel 1.</b> Perbandingan Metode.....	11
<b>Tabel 2.</b> Variabel yang terdapat pada bitcoin.....	18
<b>Tabel 3.</b> Datar Penelitian Terdahulu .....	19
<b>Tabel 4.</b> Sampel Dataset Bitcoin.....	24
<b>Tabel 5.</b> Definisi Operasional Variabel Penelitian .....	27
<b>Tabel 6.</b> Atribut Dataset.....	29
<b>Tabel 7.</b> Sample Dataset Bitcoin.....	29
<b>Tabel 8.</b> Hasil Normalisasi Data Harga Bitcoin .....	31
<b>Tabel 9.</b> Hasil Pembagian Data Latih dan Data Uji.....	32
<b>Tabel 10.</b> Hasil Pengujian Data Harga Bitcoin .....	33
<b>Tabel 11.</b> Hasil Pengujian .....	36
<b>Tabel 12.</b> Perhitungan RMSE Oprimasi Adam .....	41
<b>Tabel 13.</b> Perhitungan MAE Oprimasi Adam .....	42

**DAFTAR LAMPIRAN**

<b>Lampiran 1</b> .....	48
<b>Lampiran 2</b> .....	52

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Investasi menjadi salah satu kegiatan yang digemari masyarakat karena dapat meningkatkan keuntungan dari kegiatan tersebut. investasi memiliki tujuan, yaitu menciptakan keuntungan finansial, mendapatkan profit maksimum, dan menciptakan kemakmuran bagi investor. Kegiatan investasi ini telah banyak dijalankan oleh badan usaha maupun perseorangan yang berupaya mendapatkan keuntungan dari nominal dana yang dimiliki. Pada saat ini, sudah banyak ragam instrument-instrumen investasi yang tersedia seperti saham, obligasi, emas, properti dan yang terbaru adalah *cryptocurrency* (mata uang kripto). Kripto sebagai salah satu mata uang yang memiliki wujud kode kriptografi yang digunakan untuk menjaga keamanan transaksi dan dapat disimpan dalam sebuah perangkat komputer dapat ditransfer atau dipindah tangankan semacam surat elektronik yang memungkinkan kripto dapat dimanfaatkan sebagai alat pembayaran. Keunggulan dari mata uang kripto karena dilihat dari sisi kerugian atau kerusakannya tidak mudah dihancurkan hingga disembunyikan karena kode didalamnya mengandung algoritma kriptografi yang sekaligus mampu menghindari pemalsuan (Dharma and Gusniati, 2023)

*Cryptocurrency* merupakan salah satu bentuk mata uang elektronik yang paling cepat digemari dan diminati di kehidupan masyarakat saat ini. Bentuk uang elektronik tersebut sudah diketahui sebagian besar masyarakat serta merupakan fenomena baru di seluruh dunia. Hingga saat ini diketahui bahwa tingkat perkembangan mata uang digital ini meningkat secara signifikan setiap tahun, dan bitcoin (BTC) dianggap sebagai salah satu hal yang paling berharga akhir-akhir ini. Sondakh Alfered berpendapat 700 (tujuh ratus) lebih variasi *cryptocurrency* telah muncul di pasar dalam periode lima tahun terakhir (Arwono, Iskandar and Wardana, 2023). Beberapa *cryptocurrency* yang banyak dikenal selain bitcoin seperti *ethereum, binance, tether* dan *ripple*.

Sejak didirikan pada 2008, bitcoin telah muncul sebagai mata uang digital utama dalam hal kapitalisasi pasar dan terus menarik perhatian investor. Meningkatnya volatilitas *cryptocurrency* dapat dilihat pada pergeseran ekonomi global seperti market keuangan dunia yang menurun, perusahaan mengalami krisis keuangan, tingkat pengangguran meningkat, serta perubahan perilaku konsumen. Perubahan tersebut secara signifikan berpengaruh pada *cryptocurrency*, hal ini menunjukkan fluktuasi *cryptocurrency* mengalami ketidakstabilan dan tidak teratur dibandingkan ekuitas, sehingga pasar

*cryptocurrency* lebih beresiko dan sulit untuk diprediksi (Kuncara and Anugrah, 2023). Transaksi bitcoin pertama kali terjadi pada Januari 2009. Lebih dari dua tahun kemudian, berbagai laporan memperkirakan sirkulasi bitcoin menjadi lebih dari 6,5 juta dengan sekitar 10.000 pengguna. Bitcoin mulai mendapatkan perhatian ketika mulai digunakan sebagai alat pembayaran di *silkroad*, yaitu sebuah pasar perdagangan obat-obatan ilegal selain itu bitcoin juga mendapatkan dukungan dari beberapa orang penting dalam dunia teknologi informasi dan juga baru-baru ini negara El Salvador menjadikan bitcoin sebagai mata uang negara dan menjadi alat pembayaran yang sah (Hasani, 2022). Selanjutnya, Bohme et al. (2015) menemukan bahwa transaksi dapat ditunda hingga satu jam yang sangat mengurangi kemungkinan likuiditas. Namun, protokol bitcoin tidak membatasi transfer karena daftar pantauan atau negara yang diembargo karena identitas penggunanya tidak diketahui. Ini memberikan fleksibilitas dan kecepatan transfer internasional bitcoin yang sangat besar dibandingkan dengan mata uang lain yang dikelola oleh bank (Dyhrberg, 2016). Hal ini menyebabkan semakin banyak masyarakat yang tertarik untuk melakukan investasi terhadap bitcoin.

Prediksi adalah suatu proses memperkirakan secara sistematis tentang sesuatu yang paling mungkin terjadi di masa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara sesuatu yang terjadi dengan hasil perkiraan) dapat diperkecil, Prediksi tidak harus memberikan jawaban secara pasti kejadian yang akan terjadi, melainkan berusaha untuk mencari jawaban sedekat mungkin yang akan terjadi (Kafil, 2019). Pengertian prediksi sama dengan ramalan atau perkiraan. Menurut kamus besar bahasa Indonesia, prediksi adalah hasil dari kegiatan memprediksi atau meramal atau memperkirakan nilai pada masa yang akan datang dengan menggunakan data masa lalu. Prediksi menunjukkan apa yang akan terjadi pada suatu keadaan tertentu dan merupakan input bagi proses perencanaan dan pengambilan keputusan.

Berdasarkan fakta tersebut, penting dilakukan penelitian mengenai prediksi harga bitcoin untuk mengetahui bagaimana perkembangan harga bitcoin pada masa yang akan datang. Oleh sebab itu, peneliti menentukan topik tentang prediksi harga bitcoin. Dengan memprediksi harga bitcoin pada masa yang akan datang, investor dapat memperkirakan harga bitcoin dari prediksi harga yang telah dilakukan agar tidak terjadi kesalahan dalam melakukan investasi sehingga dapat mencegah besarnya potensi kerugian. Prediksi harga bitcoin juga berguna untuk melihat bagaimana perkembangan *cryptocurrency* dari tahun-tahun sebelumnya apakah ada perkembangan signifikan yang dapat mempengaruhi

kenaikan harga bitcoin pada masa yang akan datang. Dengan informasi data yang dapat berguna dalam memprediksi perkembangan harga bitcoin, informasi ini akan menjadi salah satu dasar dalam pengambilan keputusan bagi investor. Salah satu cara memprediksi harga bitcoin ialah dengan menggunakan metode *Long Short Term Memory* (LSTM). LSTM merupakan sebuah arsitektur *Reccurent Neural Network* (RNN) yang memiliki *memory cell*. Dengan *memory cell*, arsitektur LSTM dapat bekerja dengan lebih baik dibandingkan jaringan syaraf rekuren biasa karena memiliki kemampuan dalam mengingat informasi untuk periode waktu yang lebih lama, sehingga menjadikan algoritma yang lebih baik untuk prediksi data berjenis *time series* (Prasetyanwar and Jondri, 2018). Analisis runtun waktu atau *time series* adalah salah satu metode dalam analisis data yang berbentuk kuantitatif dengan menentukan pola data masa lampau yang telah dikumpulkan dengan mempertimbangkan waktu (Wijayanti dan Pulungan, 2012). Untuk data *time series*, dapat menggunakan beberapa model prediksi dengan teknik statistik dan teknik *deep learning*. Teknik statistik termasuk LSTM (Long Short Term Memory), AR, MA, ARIMA, SARIMA, dan lain-lain. Pada penelitian ini bertujuan untuk menguji efektivitas penggunaan LSTM dalam memprediksi harga bitcoin dan mengevaluasi kinerja model LSTM dalam memprediksi harga Bitcoin dengan menggunakan kombinasi parameter terbaik pada penelitian yang berjudul *Analisis dan Implementasi Long Short Term Memory Neural Network* untuk Prediksi Harga Bitcoin (Aldi, Jondri and Aditsania, 2018) penelitian tersebut membahas tentang metode *Long Short Term Memory* dalam memprediksi data *time series* pada kasus harga bitcoin. Dalam penelitian tersebut, didapatkan dengan menggunakan data historis harga bitcoin dari tanggal 22 Mei 2016 sampai 19 Mei 2018 didapatkan hasil terbaik yaitu dengan komposisi data latih 70% dan data uji 30%, parameter 1 pola *time series*, jumlah 25 neuron hidden, dan max epoch adalah 100 dengan akurasi rata-rata pada data latih 95.36% dan data testing 93.5%. Diharapkan pada penelitian ini akan mendapatkan hasil yang lebih baik dari penelitian sebelumnya dengan kombinasi parameter dan dataset yang berbeda. Dari latar belakang dan tinjauan serta melihat permasalahan diatas dan juga dari beberapa metode peramalan yang ada maka penelitian ini berjudul “Prediksi Harga Bitcoin Menggunakan Metode Long Short Term Memory”.

## **1.2 Rumusan Masalah**

Adapun rumusan masalah dalam melakukan penelitian ini adalah:

1. Bagaimana penggunaan metode *Deep Learning* menggunakan *Long-Short Term Memory* untuk memprediksi harga uang digital bitcoin?

2. Bagaimana tingkat akurasi yang didapatkan dari hasil prediksi harga bitcoin menggunakan algoritma *Long-Short Term Memory*?

### **1.3 Tujuan Penelitian**

Adapun Tujuan penelitian ini adalah :

1. Untuk mengetahui hasil prediksi harga bitcoin menggunakan LSTM berdasarkan informasi harga bitcoin?
2. Untuk mengetahui tingkat akurasi model data yang di latih dan di uji dengan metode *Long-Short Term Memory*?

### **1.4 Manfaat Penelitian**

Dari penelitian yang dilakukan terdapat beberapa manfaat yang bisa diambil, yaitu:

1. Penelitian ini dapat membantu dalam pengembangan strategi perdagangan yang lebih efektif. Dengan memanfaatkan informasi prediksi harga, para pegiat pasar dapat membuat keputusan perdagangan yang lebih informasional dan memberikan analisis yang lebih akurat.
2. Penelitian ini dapat memberikan wawasan baru tentang bagaimana teknologi blockchain dan data histori harga dapat dimanfaatkan secara efektif dalam prediksi harga aset digital.
3. Dapat menjadi referensi dan landasan untuk penelitian selanjutnya dalam bidang prediksi harga pasar keuangan lainnya.

### **1.5 Batasan Penelitian**

Adapun batasan masalah dalam penelitian ini adalah sebagai berikut :

1. Untuk data yang digunakan adalah *price* bitcoin karena memiliki kapitalisasi pasar yang paling besar, *date*, *open price*, *high price*, *low price*, *close price*, *Adj close*, *volume*.
2. Untuk data yang diambil adalah data harian dari periode Januari 2021 s.d Mei 2023.
3. Software yang digunakan adalah *jupyter notebook*

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Prediksi**

Menurut kamus besar Bahasa Indonesia, prediksi ialah hasil kegiatan dari meramal (KBBI, 2017). Prediksi juga dapat dikatakan sebagai suatu proses memperkirakan penjualan serta penggunaan produk sebagai akibatnya produk-produk itu dapat dirancang dalam kuantitas yang tepat. Hal ini bisa dilakukan dengan melibatkan pengambilan data masa kemudian dan menempatkannya ke masa yang akan tiba dengan suatu bentuk contoh matematis. Prediksi dapat bersifat kualitatif dalam artian tidak berbentuk angka maupun bersifat kuantitatif yang artinya berbentuk angka. Prediksi kualitatif sangat sulit dilakukan untuk mencapai hasil yang baik karena variabel pada prediksi kualitatif bersifat relatif. Prediksi kuantitatif yaitu prediksi yang menggunakan model matematis dengan dimasukan data masa lalu dan variabel sebab akibat untuk meramalkan data masa depan. Terdapat 7 langkah prediksi yaitu pertama menentukan penggunaan dari prediksi, kedua memilih produk yang akan diprediksi, ketiga menentukan horizon waktu dari prediksi (jangka pendek, menengah, dan panjang), keempat memilih model prediksi, kelima memperoleh data yang dibutuhkan untuk melakukan prediksi, keenam membuat prediksi, dan yang terakhir memvalidasi dan mengimplementasikan hasilnya. Adapun manfaat dari prediksi ialah sebagai alat bantu untuk merencanakan sesuatu sehingga lebih efektif dan efisien, selain itu untuk menetapkan kebutuhan sumber daya pada masa yang akan datang serta dapat membuat keputusan yang tepat (Ngantung *et al.*, 2019).

#### **2.2 Deep Learning**

Deep Learning merupakan salah satu bidang dari Machine Learning yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik Deep Learning memberikan arsitektur yang sangat kuat untuk Supervised Learning. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada Machine Learning terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi (Deng and Yu, 2013).

Algoritma yang digunakan pada Feature Engineering dapat menemukan pola umum yang penting untuk membedakan antara kelas Dalam Deep Learning,

Salah satu studi yang telah digunakan secara luas oleh para ahli untuk membangun model prediksi adalah data mining. Deep learning memiliki beragam algoritma yang berfokus pada pembelajaran representasi data (nonlinear) bertingkat. Salah satu algoritma deep learning yang terbukti berhasil digunakan untuk memprediksi data time series adalah algoritma Long Short Term Memory (LSTM) yang merupakan turunan dari Recurrent Neural Network (RNN) (Sianturi, Cholissodin and Yudistira, 2023).

### **2.3 Neural Network**

Neural network adalah model yang terinspirasi oleh bagaimana neuron dalam otak manusia bekerja. Neural Network terdiri dari sejumlah besar elemen pemrosesan informasi (neuron) yang saling terhubung dan bekerja bersama-sama untuk menyelesaikan sebuah masalah tertentu, yang pada umumnya adalah masalah klasifikasi ataupun prediksi (Ridho *et al.*, 2022). Otak manusia terdiri dari ratusan juta sel syaraf yang disebut neuron. Otak digambarkan sebagai sebuah mesin yang menghubungkan neuron satu dengan lainnya dalam bentuk impuls saraf sehingga dapat mengkoordinasikan berbagai fungsi tubuh.

Cara kerja neuron tersebut ditiru oleh Neural Network sebagai langkah membuat machine yang pintar. Dalam hal ini, Neural Network tidak diprogram untuk menghasilkan keluaran tertentu. Semua keluaran yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran. Pada proses pembelajaran, pola-pola input dimasukkan ke Neural Network untuk dan kemudian jaringan diajari untuk menentukan jawaban yang bisa diterima. Pembelajaran dilakukan dengan menentukan bobot masing-masing node dalam jaringan (Prasetyo *et al.*, 2017).

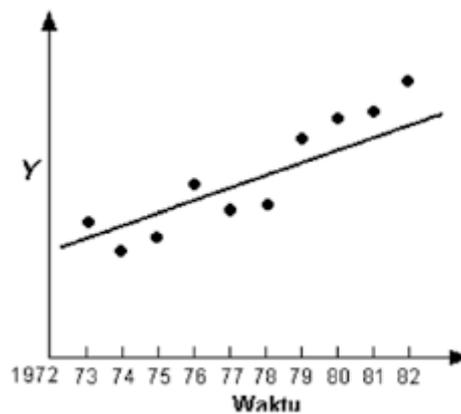
### **2.4 Time Series**

Menurut Lincoln Arsyad (2009) dalam jurnal yang ditulis oleh Nawangwulan and Angesti, (2016), setiap variabel yang terdiri dari data yang dikumpulkan, dicatat atau diobservasi sepanjang waktu yang berurutan disebut data runtut waktu (time series). Analisis runtut waktu dilakukan untuk menemukan pola pertumbuhan atau perubahan masa lalu, yang dapat digunakan untuk memperkirakan pola pada masa yang akan datang. Analisis ini cukup penting dalam proses peramalan dan membantu mengurangi kesalahan dalam peramalan tersebut. Dalam analisis runtut waktu terdapat 4 komponen yaitu (Nawangwulan and Angesti, 2016):

#### **2.3.1 Trend**

*Trend* ialah perkembangan jangka panjang dalam suatu runtut waktu yang dapat digambarkan dengan sebuah garis lurus atau sebuah kurva yang

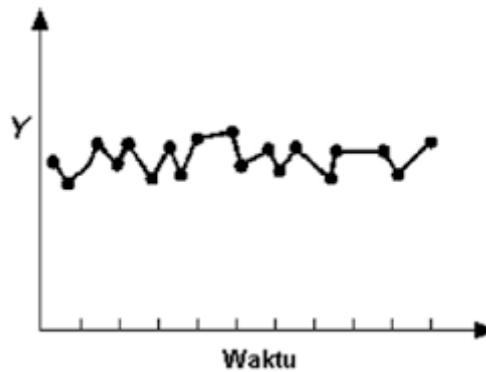
menghasilkan atau mempengaruhi trend dari suatu seri dapat berupa perubahan populasi, perubahan harga, perubahan teknologi, dan peningkatan produktivitas. Dalam analisis trend variabel bebasnya adalah waktu. Jika grafik berbentuk garis lurus dalam skala aritmatika, maka peneliti akan menggunakan persamaan linier dalam analisis datanya. Jika data dinyatakan dalam bentuk semilogaritma dan terbentuk grafik dengan garis lurus, maka peneliti akan memilih model eksponensial dalam datanya. Metode untuk menjelaskan trend linier ialah metode kuadrat terkecil.



**Gambar 1.** Pola Data Trend (Andini & Auristandi, 2016)

### 2.3.2 Variasi Siklis

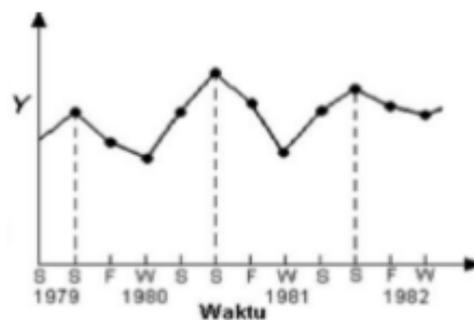
Komponen siklis ialah suatu seri fluktuasi seperti gelombang atau siklus yang mempengaruhi keadaan ekonomi selama lebih dari satu tahun. Hal tersebut dapat dilihat dari perbedaan antara nilai yang diharapkan (trend) dengan nilai yang sebenarnya yaitu variasi residual yang berfluktuasi sekitar trend. Komponen siklis dan tak beraturan dari data runtut waktu dapat diidentifikasi dengan cara menghilangkan pengaruh trend, metode ini disebut metode residual (residual method). Tahap metode residual tergantung pada dimulainya menganalisis dengan data tahunan, bulanan, atau kuartalan. Jika data yang digunakan ialah data bulanan atau kuartalan, maka pengaruh trend dan komponen-komponen musiman harus dihilangkan. Jika datanya ialah data tahunan, maka pengaruh trend yang dihilangkan.



**Gambar 2.** Pola Data Siklis(Andini & Auristandi, 2016)

### 2.3.3 Musiman

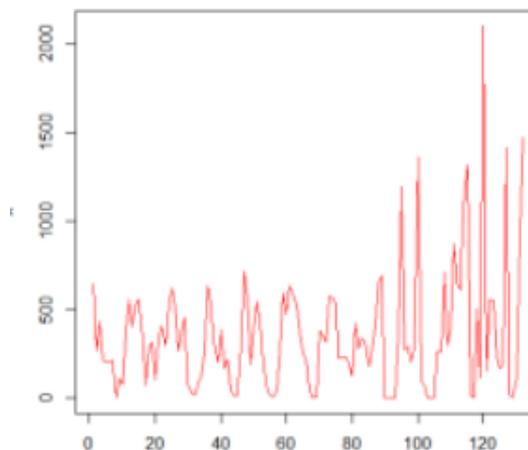
Fluktuasi musiman biasanya dijumpai pada data yang dikelompokkan secara kuartalan, bulanan, atau mingguan. Variasi musiman ini menggambarkan pola perubahan yang berulang secara teratur dari waktu ke waktu. Komponen musiman runtut waktu diukur dalam bentuk angka indeks. Interpretasi angka indeks ini, yang mencerminkan besarnya pengaruh musiman untuk suatu segmen tahun tertentu, berkaitan dengan perbandingan nilai terhitung atau nilai yang diharapkan dari segmen tersebut (bulan, kuartal, dan sebagainya).



**Gambar 3.** Pola Data Musiman(Andini & Auristandi, 2016)

### 2.3.4 Fluktuasi Tak Beraturan

Komponen tidak beraturan terbentuk dari fluktuasi-fluktuasi yang disebabkan oleh peristiwa-peristiwa yang tidak terduga seperti perubahan cuaca, pemogokan, perang, pemilihan umum, rumor perang, dan lain-lain.



**Gambar 4.** Pola Data Fluktuasi Tak Beraturan(Darmawan & dkk, 2016)

## 2.5 Artificial Neural Network

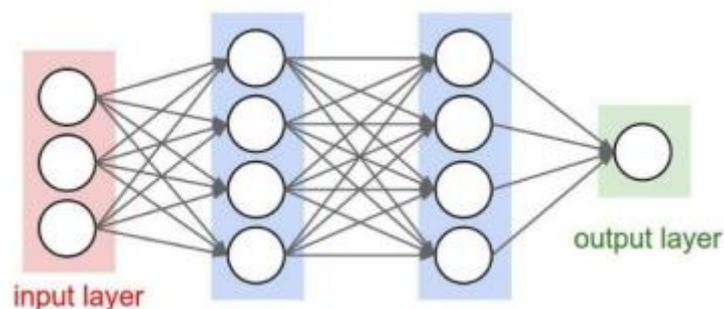
*Artificial Neural Network*(ANN) adalah teknik pembelajaran mesin yang terinspirasi oleh jaringan saraf biologis di otak manusia. Penentuan jumlah lapisan dan jumlah neuron di lapisan tersembunyi dan koneksi di antara neuron tersebut adalah hal yang penting (Paquin *et al.*, 2015). Neuron mempunyai karakteristik yang sama dalam ANN, terdiri dalam kelompok-kelompok yang disebut layer. Neuron-neuron dalam satu layer terhubung dalam layer-layerlainnya yang berdekatan. Kekuatan hubungan antar neuron yang berdekatan direpresentasikan dalam kekuatan hubungan atau bobot(Meliana, 2021).

Langkah penting dalam pengembangan sebuah model ANN adalah penentuan bobot matrik melalui pelatihan (training). Ada dua tipe mekanisme 32 training yaitu supervised training dan unsupervised training. Supervised training memerlukan supervisi dari luar untuk memandu proses training. Algoritma ini menggunakan sejumlah pasangan data input-output yang dipergunakan sebagai contoh, dimana data yang dipergunakan sebagai contoh sebaiknya menggunakan data yang sudah diketahui kebenarannya. Output dari jaringan lalu dibandingkan dengan data output yang diharapkan untuk mendapatkan selisih antara output perkiraan dengan output sebenarnya. Selisih inilah yang dipergunakan untuk mengubah bobot jaringan sehingga diperoleh output yang sama atau mendekati target. Mekanisme sebuah supervised training yang disebut algoritma backpropagation training umumnya banyak digunakan dalam aplikasi-aplikasi engineering. Karena ANN tidak mempertimbangkan fisik dari permasalahan, ANN merupakan sebuah model blackbox namun dapat mendeteksi proses fisik dalam model ANN yang sudah ditraining (Meliana, 2021).

Jaringan saraf terdiri dari lapisan input dan output, serta (dalam

kebanyakan kasus) hidden layer yang terdiri dari unit yang mengubah input menjadi sesuatu yang dapat digunakan oleh lapisan output. Input, output, dan hidden layer adalah alat yang sangat baik untuk menemukan pola yang terlalu rumit atau banyak bagi programmer untuk mengekstraksi dan mengajar mesin untuk mengenali. Sementara jaringan saraf (juga disebut "perceptrons") telah ada sejak 1940-an, hanya dalam beberapa dekade terakhir di mana mereka telah menjadi bagian utama dari kecerdasan buatan. Hal ini disebabkan oleh temuan teknik yang disebut "backpropagation" yang memungkinkan jaringan untuk menyesuaikan lapisan neuron tersembunyi mereka dalam situasi di mana hasilnya tidak sesuai dengan harapan peneliti misalnya seperti jaringan yang dirancang untuk mengenali kucing, tapi mengidentifikasi kelinci. Kemajuan penting lainnya adalah munculnya deep learning neural networks, dimana lapisan yang berbeda dari jaringan multilayer mengekstraksi fitur yang berbeda hingga dapat mengenali apa yang dicari. Untuk ide dasar tentang bagaimana deep learning neural networks, bayangkan garis pabrik. Setelah bahan baku (kumpulan data) dimasukkan, mereka kemudian diturunkan melalui ban berjalan, dengan setiap pemberhentian berikutnya atau lapisan mengekstraksi serangkaian fitur tingkat tinggi yang berbeda. Jika jaringan dimaksudkan untuk mengenali objek, lapisan pertama mungkin menganalisis kecerahan pikselnya.

Kemajuan penting lainnya adalah munculnya deep learning neural networks, dimana lapisan yang berbeda dari jaringan multilayer mengekstraksi fitur yang berbeda hingga dapat mengenali apa yang dicari. Untuk ide dasar tentang bagaimana deep learning neural networks, bayangkan garis pabrik. Setelah bahan baku (kumpulan data) dimasukkan, mereka kemudian diturunkan melalui ban berjalan, dengan setiap pemberhentian berikutnya atau lapisan mengekstraksi serangkaian fitur tingkat tinggi yang berbeda. Jika jaringan dimaksudkan untuk mengenali objek, lapisan pertama mungkin menganalisis kecerahan pikselnya.



**Gambar 5.** Blockchain Sederhana(Patel, 2019)

Lapisan selanjutnya dapat mengidentifikasi setiap tepi dalam gambar,

berdasarkan garis piksel yang sama. Setelah ini, layer lain mungkin mengenali tekstur dan bentuk, dan seterusnya. Pada saat lapisan keempat atau kelima tercapai, jaring pembelajaran yang dalam akan menciptakan detektor fitur yang kompleks. Dapat diketahui bahwa elemen gambar tertentu (seperti sepasang mata, hidung, dan mulut) umumnya ditemukan bersama. Setelah ini dilakukan, para peneliti yang telah melatih jaringan dapat memberikan label ke output, dan kemudian menggunakan backpropagation untuk memperbaiki kesalahan yang telah dibuat. Setelah beberapa saat, jaringan dapat melakukan tugas klasifikasi sendiri tanpa memerlukan manusia untuk membantu setiap saat. Namun pada Artificial Neural Network tidak dapat mengingat informasi yang diberikan sebelumnya. Hal ini diatasi oleh Recurrent Neural Networks (RNN). RNN adalah jaringan dengan loop di dalamnya, memungkinkan informasi untuk bertahan.

## 2.5 Perbandingan Metode

**Tabel 1.** Perbandingan Metode

<b>1</b>	<b>Judul Penelitian</b>	<b>Penulis dan tahun</b>	<b>Metode</b>	<b>Hasil Perbandingan</b>
	Pemodelan Time Series untuk Peramalan Web Traffic Menggunakan Algoritma Arima, LSTM, dan GRU	(Oxaichiko Arissinta, Dwi Sulistiyawati and Kurnianto Iqbal Kharisudin, 2022)	ARIMA,LSTM dan GRU	Berdasarkan hasil analisis peramalan yang dilakukan dengan menggunakan ketiga metode yaitu ARIMA, LSTM, dan GRU diperoleh hasil peramalan yang paling baik akurasiya menggunakan LSTM dan GRU
<b>2</b>	PERBANDINGAN METODE DAN GRU UNTUK KLASIFIKASI BERITA PALSU BERBAHASA INDONESIA	(Aldi, Jondri and Aditsania, 2018)	LSTM dan GRU	Nilai akurasi yang dihasilkan oleh model cukup baik yaitu 73% menggunakan metode LSTM dan 64% menggunakan metode GRU

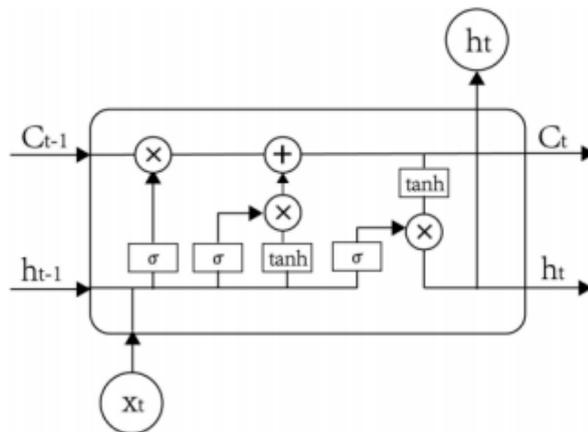
3	PERBANDINGAN PREDIKSI PENGGUNAAN LISTRIK DENGAN MENGGUNAKAN METODE LONG SHORT TERM MEMORY (LSTM) DAN RECURRENT NEURAL NETWORK (RNN)	(Selle, Yudistira and Dewi, 2022)	LSTM dan RNN	Hasil perbandingan yang didapatkan antara RNN dan LSTM, menyimpulkan bahwa LSTM memiliki kinerja yang lebih baik dan mampu menurunkan nilai eror pada penggunaan deret waktu yang lebih panjang dalam melakukan prediksi.
4	Perbandingan Algoritma Long Short-Term Memory dengan SVR pada Prediksi Harga Saham di Indonesia	(Arfan and ETP, 2020)	LSTM dan SVR	Secara umum semakin besar rentang data yang digunakan oleh SVR, maka semakin besar nilai MSE yang dihasilkan. Sedangkan nilai MSE yang dihasilkan LSTM memiliki nilai yang sama walaupun menggunakan rentang data yang berbeda. Berdasarkan hasil pengujian, LTSM mampu memprediksi harga saham pada tahun 2017-2019 dengan performa yang baik,

**Sumber: Penelitian terdahulu**

### **2.6 Long Short Term Memory**

Long Short Term Memory (LSTM) menggunakan salah satu bentuk RNN yang paling umum yang dimaksudkan untuk menghindari masalah ketergantungan jangka panjang dan sesuai untuk memproses serta memprediksi deret waktu.

Diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997, model LSTM terdiri dari serangkaian sel memori unik yang menggantikan neuron lapisan tersembunyi dari RNN, dan kuncinya adalah keadaan sel-sel memori. Model LSTM menyaring informasi melalui struktur gerbang untuk mempertahankan dan memperbarui keadaan sel memori. Struktur pintunya mencakup gerbang input, forget gate, dan gerbang output. Setiap sel memori memiliki tiga lapisan sigmoid dan satu lapisan tanh. Lapisan sigmoid menampilkan angka antara nol dan satu, menggambarkan berapa banyak masing-masing komponen yang harus dilewati. Nilai nol berarti "jangan biarkan apa pun lewat" sementara nilai satu berarti "biarkan semuanya lewat". Sigmoid akan bernilai nol jika terletak pada rentang  $0 \leq \sigma \leq 0,5$  dan bernilai satu jika terletak pada rentang  $0,5 < \sigma \leq 1$  (R, 2017).



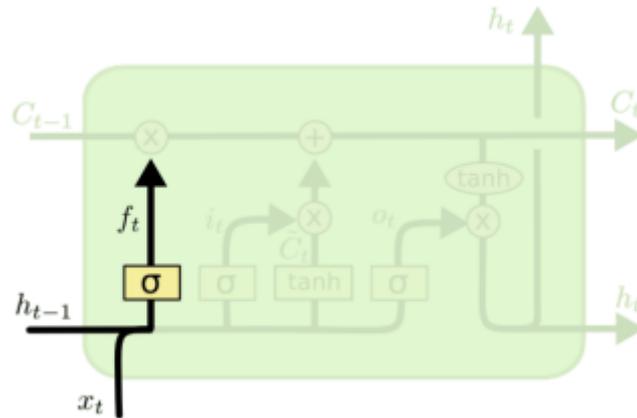
**Gambar 6.** Struktur LSTM (Qiu, Wang, & Zhou, 2020)

Fungsi sigmoiddan tanh ditunjukkan pada persamaan (1) dan (2) (Dharma, Gusniati and ..., 2023):

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (1)$$

$$\text{Tanh}(x) = 2\sigma(2x) - 1 \quad (2)$$

Dimana  $x$  adalah data input  $e$  adalah konstan matematika.



**Gambar 7.** Struktur LSTM Forget Gate(Olah, 2015)

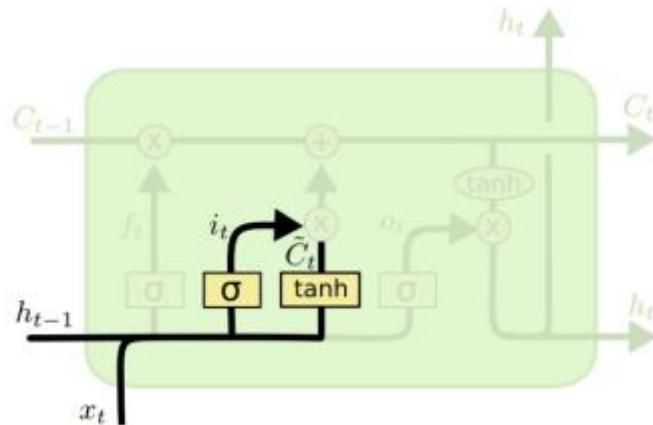
*Forget gate* dalam unit LSTM menentukan informasi status sel mana yang dibuang dari model. Seperti yang ditunjukkan pada **Gambar 7**, sel memori menerima output  $h_{t-1}$  dari momen sebelumnya dan informasi eksternal  $x_t$  dari momen saat ini sebagai input dan menggabungkannya dalam vektor panjang  $[h_{t-1}, x_t]$ . Notasi  $[h_{t-1}, x_t]$  merupakan operasi konkatenasi, artinya menambahkan baris dari  $x_t$  dengan baris dari  $h_{t-1}$  melalui transformasi  $\sigma$  menjadi:

$$f_t = \sigma (W_t \cdot [h_{t-1}, x_t] + b_f) \quad (3)$$

Keterangan:

- $f_t$  : Forget gate
- $\sigma$  : Fungsi sigmoid
- $W_t$  : Nilai weight untuk forget gate
- $h_{t-1}$  : Nilai output sebelum orde ke-t
- $x_t$  : Nilai input pada orde ke-t
- $b_f$  : Nilai bias pada forget gate

Fungsi utama dari *forget gate* adalah untuk merekam seberapa banyak status sel  $C_{t-1}$  dari waktu sebelumnya dicadangkan ke status sel  $C_t$  dari waktu saat ini. Gerbang akan menampilkan nilai antara 0 dan 1 berdasarkan pada  $h_{t-1}$  dan  $x_t$ , dimana 1 menunjukkan reservasi lengkap dan 0 menunjukkan pembuangan lengkap.



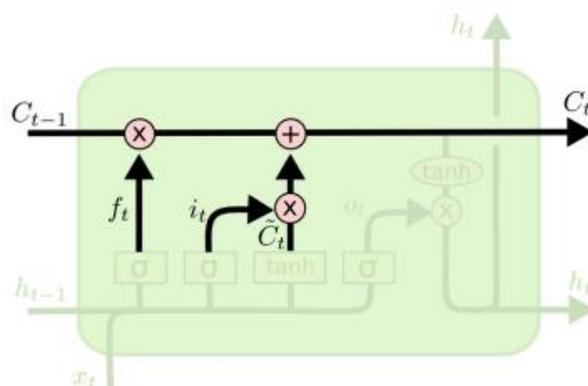
**Gambar 8.** Struktur LSTM Input Gate(Olah, 2015)

Gerbang input menentukan berapa banyak input jaringan waktu saat ini  $x_t$  dicadangkan ke dalam status sel  $C_t$ , yang mencegah konten tidak signifikan dari memasuki sel memori. Gerbang input memiliki dua fungsi, pertama adalah menemukan keadaan sel yang harus diperbarui; nilai yang akan diperbarui dipilih oleh lapisan sigmoid, seperti dalam Persamaan (4):

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i) \quad (4)$$

Keterangan:

- $i_t$  : Input gate
- $\sigma$  : Fungsi sigmoid
- $W_i$  : Nilai weight untuk input gate
- $h_{t-1}$  : Nilai output sebelum orde ke-t
- $x_t$  : Nilai input pada orde ke-t
- $b_i$  : Nilai bias pada input gate



**Gambar 9.** Struktur LSTM Cell State(Olah, 2015)

Sedangkan fungsi lainnya adalah untuk memperbarui informasi ke b diperbarui ke keadaan sel. Vektor kandidat baru  $\tilde{C}_t$  dibuat melalui lapisan tanh untuk mengontrol berapa banyak informasi baru ditambahkan, seperti dalam

Persamaan (5). Dan Persamaan (6) digunakan untuk memperbarui keadaan sel dari sel memori

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (5)$$

Keterangan :

$C_t$  : Nilai baru yang ditambahkan ke cell state

$\tanh$  : Fungsi tanh

$W_c$  : Nilai weight untuk cell state

$h_{t-1}$  : Nilai output sebelum orde ke-t

$x_t$  : Nilai input pada orde ke-t

$b_c$  : Nilai bias pada cell state  $C_t = f_t * C_{t-1} + i_t * C_t$

Keterangan:

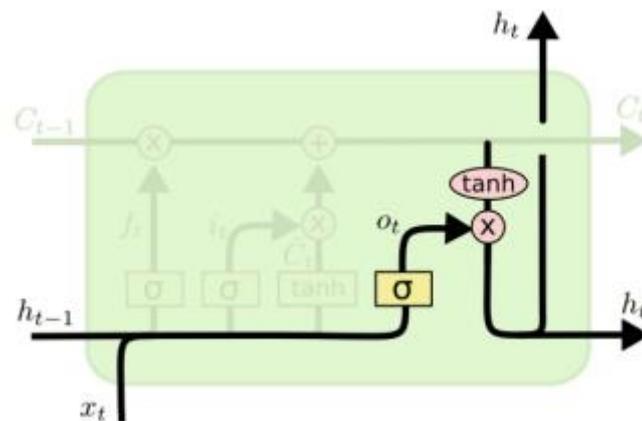
$$C_t = f_t * C_{t-1} + i_t * C_t \quad (6)$$

$C_t$  : Nilai baru yang ditambahkan ke cell state

$f_t$  : Forget gate  $C_{t-1}$  : Cell state sebelum orde ke-t

$i_t$  : Input gate

$C_t$  : Nilai baru yang ditambahkan ke cell state



**Gambar 10.** Struktur LSTM Output Gate (Olah, 2015)

Gerbang output mengontrol seberapa banyak keadaan sel saat ini dibuang. Informasi keluaran pertama-tama ditentukan oleh lapisan sigmoid, dan kemudian keadaan sel diproses oleh tanh dan dikalikan dengan keluaran lapisan sigmoid untuk mendapatkan bagian keluaran akhir:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (7)$$

Keterangan:

$O_t$  : Output gate

$\sigma$  : Fungsi sigmoid

$W_o$  : Nilai weight untuk output gate

$h_{t-1}$  : Nilai output sebelum orde ke-t

$x_t$  : Nilai input pada orde ke-t

$bo$  : Nilai bias pada output gate

Nilai output akhir sel didefinisikan sebagai:

$$h_t = O_t * \tanh(C_t) \quad (8)$$

Keterangan:

$O_t$  : Output gate

$\sigma$  : Fungsi sigmoid

$W_o$  : Nilai weight untuk output gate

$h_{t-1}$  : Nilai output sebelum orde ke-t

$x_t$  : Nilai input pada orde ke-t

$bo$  : Nilai bias pada output gate

## 2.7 Adaptive Moment Estimation (Adam)

Optimasi berbasis gradien stokastik adalah kepentingan praktis inti dalam banyak bidang sains dan teknik. Banyak masalah dalam bidang ini dapat dilemparkan sebagai optimalisasi beberapa fungsi tujuan berstandarisasi skalar yang membutuhkan maksimalisasi atau minimalisasi berkenaan dengan parameternya. Adam merupakan algoritma optimasi yang dikembangkan dengan memanfaatkan kelebihan dari algoritma Adaptive Gradient (AdaGrad) yang bekerja dengan baik dengan gradien jarang dan Root Mean Square Propagation (RMSProp) yang bekerja dengan baik secara on-line dan non Pengaturan stasioner.

Adam tidak hanya mengadaptasi tingkat pembelajaran parameter berdasarkan rata-rata pertama (mean) seperti dalam RMSProp, namun Adam juga menggunakan rata-rata kedua dari gradien (varians uncentered). Algoritma menghitung rata-rata pergerakan eksponensial dari gradien dan gradien kuadratnya, dan parameter  $\beta_1$  dan  $\beta_2$  mengontrol tingkat peluruhan rata-rata pergerakan (Aldi, Jondri, & Aditsania, 2018). Beberapa keuntungan Adam adalah bahwa besarnya pembaruan parameter tidak sama dengan penskalaan gradien, ukurannya kira-kira dibatasi oleh hyperparameter stepsize, tidak memerlukan alat tulis stasioner. Adaptive Moment Estimation (Adam) adalah metode yang menghitung adaptive learning rate untuk setiap parameter. Nilai parameter yang direkomendasikan adalah  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , dan  $\epsilon = 10^{-8}$  dengan  $\beta_1 = \beta_2 =$  tingkat penurunan eksponensial dan  $\epsilon =$  nilai epsilon untuk update parameter (Aldi, Jondri and Aditsania, 2018).

## 2.8 Evaluasi

Pengertian evaluasi secara umum dapat diartikan sebagai proses sistematis untuk menentukan nilai sesuatu (ketentuan, kegiatan, keputusan, unjuk-kerja, proses, orang, objek dan yang lainnya) berdasarkan kriteria tertentu melalui penilaian (Magdalena, Fauzi and Putri, 2020). Evaluasi dalam time series adalah

proses untuk mengevaluasi seberapa baik model yang digunakan dalam memprediksi data. Evaluasi dilakukan dengan membandingkan hasil prediksi dengan data aktual. Beberapa metrik evaluasi performa model yang sering digunakan dalam evaluasi time series adalah Mean Absolute Error (MAE), Mean Squared Error (MSE), dan Root Mean Squared Error (RMSE). Selain itu, pengujian asumsi model juga dilakukan untuk memastikan bahwa model yang digunakan sesuai dengan karakteristik data. Beberapa asumsi yang diuji antara lain normalitas, homoskedastisitas, dan independensi residual. Evaluasi time series dapat dilakukan pada berbagai bidang seperti keuangan, ekonomi, sains, teknik, statistika, dan kebijakan publik. Pemilihan teknik evaluasi yang tepat harus disesuaikan dengan jenis data yang akan diproses dan tujuan dari analisis time series tersebut.

## 2.9 Bitcoin

Bitcoin merupakan salah satu cryptocurrency yang menggunakan peer-to-peer untuk proses transaksi. Bitcoin adalah mata uang digital yang berada di dalam sistem jaringan pembayaran open source P2P (peer-to-peer). (Nakamoto, 2009) P2P adalah salah satu model jaringan komputer yang terdiri dari dua atau beberapa komputer, dimana setiap komputer yang terdapat di dalam lingkungan jaringan tersebut bisa saling berbagi. Jaringan ini memudahkan pengguna dalam bertransaksi secara langsung tanpa memerlukan jasa dari pihak ketiga seperti misalnya Bank. (Pramono *et al.*, 2022) Unsur-unsur bitcoin adalah adanya jaringan peer-to-peer, blok, blockchain, dan miners. Jaringan peer-to-peer dalam bitcoin memperbolehkan pengguna untuk mentransfer sejumlah nilai bitcoin, transaksi ini disimpan dalam file yang disebut dengan blok, blok-blok ini akan terjalin satu sama lain sehingga membentuk rantai blok yang disebut dengan blockchain, dan miners memecahkan formula matematika kompleks untuk membuktikan kepemilikan Bitcoin. bitcoin merupakan suatu uang digital yang dapat digunakan untuk bertransaksi baik sebagai penyimpanan maupun alat pembayaran yang penggunaannya menggunakan jaringan online. Pada bitcoin juga terdapat beberapa variabel penting yaitu adj price, open price, close price, high price, low price, volume, date, yang dapat di lihat seperti tabel di bawah ini.

**Tabel 2.** Variabel yang terdapat pada bitcoin

No	Variabel	Kategori Data	Definisi Operasional
1.	<i>Adj Price</i>	Harga Bitcoin	Harga Penutupan Bitcoin yang disesuaikan dikonversi kedalam mata uang <i>United</i>

			<i>State Dollar(USD)</i>
2.	<i>Open Price</i>	Informasi Pasar	Harga pertama yang tercatat pada saat pembukaan perdagangan pada periode waktu tertentu.
3.	<i>Close Price</i>	Informasi Pasar	Harga Penutupan Terakhir pada rentang waktu tertentu dari asset Bitcoin
4.	<i>High Price</i>	Informasi Pasar	Harga maksimum yang dicapai oleh bitcoin pada rentang waktu tertentu yang sudah di tentukan
5.	<i>Low Price</i>	Informasi Pasar	Harga minimum/terendah yang dicapai oleh bitcoin pada rentang waktu yang telah di tetapkan
6.	<i>Volume</i>	Informasi Pasar	Jumlah total bitcoin yang diperdagangkan dalam suatu periode waktu
7.	<i>Date</i>	Informasi Pasar	Tanggal dan waktu tertentu yang terkait dengan aktivitas pergerakan harga bitcoin

## 2.10 Penelitian Terdahulu

Terkait dengan penelitian yang dilakukan penulis, maka penelitian terdahulu menjadi sangat penting agar dapat diketahui hubungan antara penelitian yang dilakukan sebelumnya dengan penelitian yang dilakukan saat ini sehingga dapat diketahui kontribusi penelitian ini terhadap perkembangan ilmu pengetahuan. Berikut merupakan beberapa sajian tentang penelitian serupa yang terkait dengan penelitian ini.

**Tabel 3.** Datar Penelitian Terdahulu

No	Judul Penelitian	Penulis dan tahun	Metode	Hasil Penelitian
1	Analisa Prediksi Harga Saham Sektor Perbankan Menggunakan	(Riyantoko <i>et al.</i> , 2020)	Long Short Term Memory	Pada penelitian ini didapatkan beberapa percobaan diantaranya melakukan analisis pada

	Algoritma Long-Short-Term-Memory			model optimasi, variasi epoch, waktu komputasi, nilai loss dan akurasi, serta nilai RMSE. Variasi nilai epoch mempengaruhi waktu komputasi, semakin besar nilai epoch, maka semakin tinggi juga waktu komputasi yang dibutuhkan untuk menyelesaikan algoritma LSTM.
<b>2</b>	Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin	(Aldi, Jondri and Aditsania, 2018)	Long Short Term Memory	Pada Tugas Akhir ini, dibangun model LSTM untuk memprediksi harga Bitcoin dengan pengujian parameter komposisi data, jumlah pola time series, jumlah hidden neuron dan max epoch. Pada pengujian tersebut didapatkan hasil yang terbaik yaitu dengan komposisi data latih 70% dan data uji 30%, parameter 1 pola time series, jumlah 25 neuron hidden, dan max epoch adalah 100 dengan akurasi rata-rata pada data latih 95.36% dan data testing 93.5%.

3	Penerapan Algoritma Long Short Term Memory (LSTM) Berbasis Multifungsi Aktivitas Terbobot Dalam Prediksi harga Ethereum	(Sianturi, Cholissodin and Yudistira, 2023)	Long Short Term Memory	Penerapan algoritma Long-Short Term Memory dalam memprediksi harga ethereum menghasilkan nilai paling optimal dengan proporsi data training dan data testing sebesar 70:30, sequence data sebesar 14 yang menggambarkan banyak data dalam 2 minggu, nilai hidden unit sebesar 64, jumlah epoch sebesar 150, dan multi fungsi aktivasi yang digunakan adalah sigmoid.
4	PERBANDINGAN PREDIKSI PENGGUNAAN LISTRIK DENGAN MENGGUNAKAN METODE LONG SHORT TERM MEMORY (LSTM) DAN RECURRENT NEURAL NETWORK (RNN)	(Selle, Yudistira and Dewi, 2022)	Recurrent Neural Network	Berdasarkan pengujian dan analisis yang dilakukan dalam penerapan metode LSTM untuk melakukan prediksi penggunaan listrik, sehingga didapatkan kesimpulan: 1. Panjang sequence time series prediksi pengujian dengan rata-rata RMSE terbaik pada penggunaan panjang sequence 20 -30 sequence.

---

**Sumber: Penelitian terdahulu**

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Tempat dan Waktu Penelitian**

Penelitian ini akan dilakukan di Kota Jambi. Penelitian ini akan dilakukan dalam jangka waktu ± 3 (tiga) bulan.

#### **3.2 Perangkat Penelitian**

Perangkat keras yang digunakan dalam penelitian ini adalah komputer personal dengan spesifikasi sebagai berikut:

##### 3.1 Perangkat keras (Hardware)

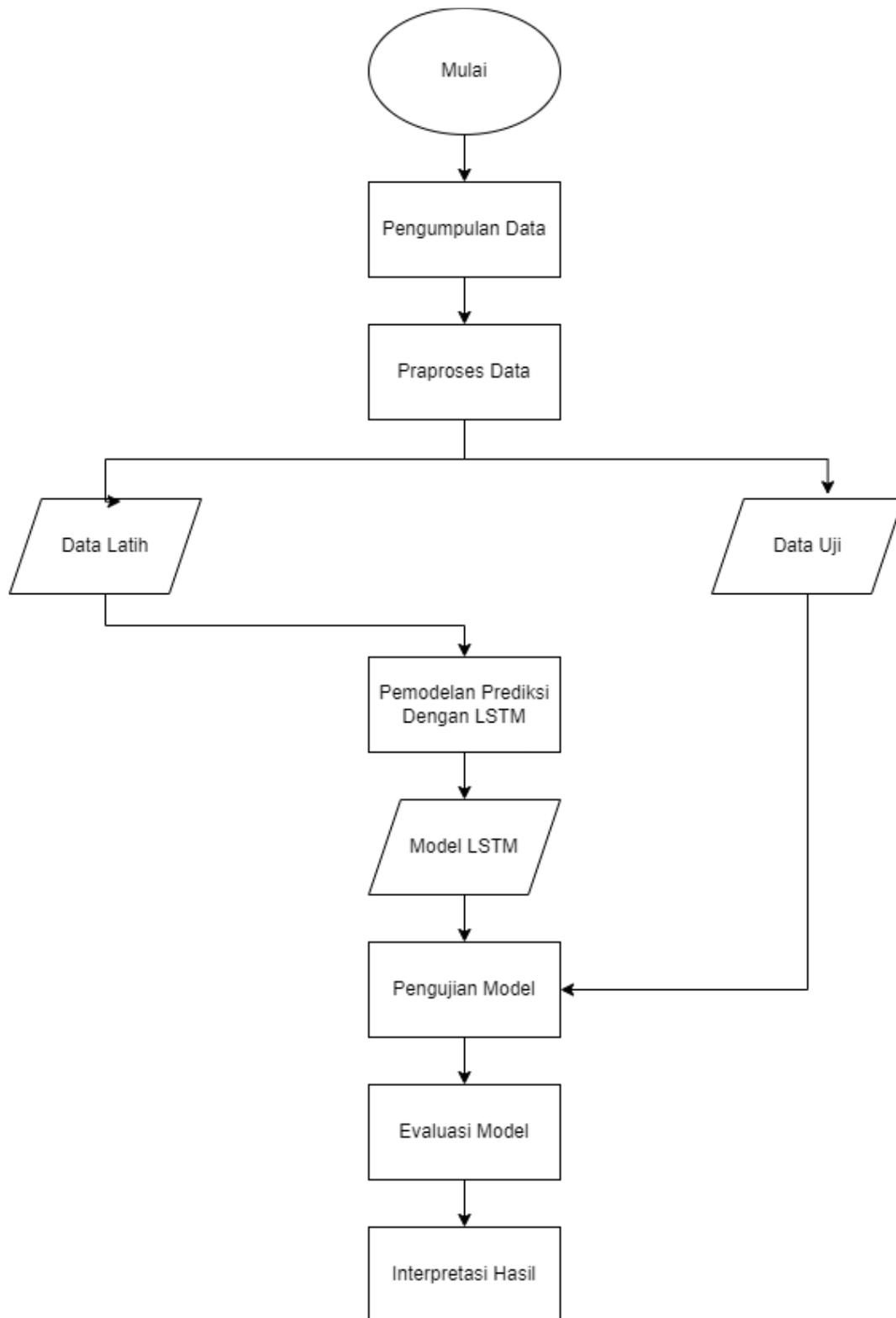
- a. Laptop Msi dengan spesifikasi processor Intel® Core™ i5-11200H CPU @ 2.4GHz, RAM 16 GB, dan system type 64-bit operating system

##### 3.2 Perangkat lunak (Software)

- a. Sistem Operasi Windows
- b. Jupyter notebook, Bahasa pemrograman Python 3.11
- c. Library numpy, pandas, streamlib, matplotlib, pylab, sklearn, keras dan tensorflow

#### **3.3 Tahapan Penelitian**

Dalam melakukan penelitian ini tentunya memerlukan adanya kerangka penelitian yang jelas supaya mempermudah dalam melakukan penelitian. maksud dari kerangka penelitian tersebut yaitu menjelaskan langkah-langkah yang akan dilakukan dalam menyelesaikan masalah dalam penelitian ini. kerangka penelitian tersebut dapat dilihat pada gambar dibawah ini.



**Gambar 11.** Alur Penelitian

### 3.1 Pengumpulan data

Dalam melakukan penelitian ini, diperlukan pengumpulan dataset yang berupa kumpulan data time series histori harga bitcoin yang dapat dilihat pada **tabel 4**.

**Tabel 4.** Sampel Dataset Bitcoin

No.	Tanggal	Harga rata-rata bitcoin(USD)
1	2021-01-01	29374.152344
2	2021-01-02	32127.267578
3	2021-01-03	32782.023438
4	2021-01-04	31971.914063
5	2021-01-05	33992.429688
6	2023-10-03	27429.978516
7	2023-10-04	27799.394531
8	2023-10-05	27415.912109
9	2023-10-06	27946.597656
10	2023-10-07	27986.648438

Sumber: [finance.yahoo.com](https://finance.yahoo.com)

### 3.2 *Preprocessing* data

Pada tahapan *preprocessing* yaitu mempersiapkan data sebelum memasuki tahap perancangan model prediksi. Pada tahap ini dilakukan normalisasi data untuk menghilangkan nilai *null* menggunakan teknik *min-max scaling* pada dataset yang disiapkan agar meminimalkan *error* pada saat melakukan uji model prediksi .

### 3.3 Data Latih dan Data Uji

Data pada penelitian ini dialokasikan menjadi dua jenis, yaitu data *training* atau data pembelajaran dan data *testing* atau data uji. Pembagian data untuk model prediksi yang dibangun dibagi dengan perbandingan 90:10, dimana sebanyak 90% dari total data akan dijadikan data *training* dan 10% lainnya akan dijadikan data *testing*. Kombinasi ini dipilih berdasarkan hasil uji coba serta pertimbangan setelah dilakukan percobaan terhadap beberapa perbandingan jumlah data *testing* dan *training*.

### 3.4 Model LSTM(*Long Short Term Memory*)

Dalam konstruksi LSTM, tiga jenis bobot digunakan. Bobot pada LSTM adalah bobot dari *input layer* ke *hidden layer*, bobot dari *hidden layer* ke *output layer* dan bobot dari *context layer* ke *hidden layer*. Selain bobot, proses LSTM juga termasuk inisialisasi parameter pembelajaran. Parameter pembelajaran pada

LSTM dalam penelitian ini adalah nilai *learning rate*, *epoch* dan *activation function*. Pembelajaran berhenti ketika nilai kesalahan mencapai target atau mencapai maksimum itersi yang ditentukan.

### 3.5 Training Data

Berikut adalah penjelasan dari proses training pada jaringan LSTM yang diusulkan:

- a. Hitung semua *fungsi gate unit* pada setiap *neuron*. Dengan berurut fungsi *gate* yang akan dihitung adalah *forget gate*, fungsi *input gate*, dan yang terakhir fungsi *output gate* dengan persamaan yang telah diketahui sebelumnya.
- b. Menghitung fungsi aktivitas linear pada *output layer*
- c. Jika telah melakukan perulangan sebanyak *epoch* yang telah ditentukan, maka berhenti. jika belum, akan dilakukan optimalisasi dengan optimasi Adam dan memperbarui bobot dan bias pada system, kemudian kembali ke langkah dua.

### 3.6 Evaluasi

Lalu akan dievaluasi kinerja dari model tersebut melalui Root Mean Square (RMSE), MAE (*Mean Absolute Error*) dan Mean Absolute Percentage Error (MAPE). RMSE adalah metode untuk mengevaluasi teknik peramalan dengan mengukur tingkat akurasi. Rumus dari RMSE yaitu (Hudzaifah, Rismayadi dan Topiq, 2019) :

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (A_t - F_t)^2}{n}} \quad (9)$$

Keterangan :

- At : nilai data aktual  
 Ft : nilai hasil peramalan  
 n : banyak data

MAE (*Mean Absolute Error*) untuk mengetahui akurasi prediksi dengan melihat nilai error. Mean Absolute Error (MAE) adalah salah satu metode yang digunakan untuk mengukur tingkat keakuratan model peramalan. Adapun untuk rumus persamaan MAE sendiri ialah sebagai berikut:

$$MAE = \sqrt{\frac{\sum_{t=1}^n (x_t - f_t)}{n}}$$

Dimana:

- $x_t$  = nilai aktual pada waktu t  
 $f_t$  = nilai prediksi pada waktu t

$t$  = waktu  
 $n$  = jumlah data

MAPE adalah rata-rata dari diferensiasi antara hasil sebenarnya dan nilai peramalan yang dinyatakan melalui persentase nilai aktual. Rumus dari MAPE yaitu (Wiranda and Sadikin, 2019):

$$PE_t = \left(\frac{e_t}{z_t}\right) \times 100 \quad (10)$$

Lalu,

$$MAPE = \sqrt{\frac{\sum_{t=1}^n (At - Ft)}{At}} \times 100\% \quad (11)$$

Keterangan :

PE<sub>t</sub> : persentase kesalahan pada waktu ke -t

E<sub>t</sub> : nilai kesalahan waktu ke -t

Z<sub>t</sub> : data aktual waktu ke -t

n : banyak data

### 3.7 Implementasi GUI Melalui *Streamlit*

Graphical User Interface (GUI) adalah antarmuka program yang bertindak sebagai media komunikasi antara pengguna dan perangkat lunak (Muhtadi, Priyadi and Rahmani, 2019). Setelah melakukan preprocessing data hingga evaluasi data secara manual untuk mengukur dataset prediksi harga bitcoin, maka peneliti membangun GUI yang dapat digunakan oleh pengguna secara mudah tanpa memiliki dasar coding. Mockup dari sistem peramalan yang akan dibangun. Aplikasi ini disusun dengan dasar dari pengujian yang telah penulis lakukan pada tahap sebelumnya. Setelah membangun GUI, maka output dari proses tersebut adalah sebuah sistem berbasis web. Melalui sistem inilah pengguna dapat melakukan proses peramalan data dengan memberikan masukan berupa data time series, kemudian akan diolah melalui sistem ini.

### 3.4 Jenis dan Sumber Data

Dalam melakukan penelitian ini, langkah pertama yaitu pengumpulan dataset histori harga bitcoin yang memiliki tujuh jenis variabel selanjutnya dataset dibagi menjadi hanya dua jenis variabel yaitu *close price* dan *date*. Dataset diambil dari sumber [finance.yahoo.com](https://finance.yahoo.com) sesuai variabel yang dibutuhkan. Dataset diambil dari tahun 2020 sampai dengan 2023.

**Tabel 5.**Definisi Operasional Variabel Penelitian

No	Variabel	Kategori Data	Definisi Operasional
1.	<i>Close Price</i>	Informasi Pasar	Harga Penutupan Terakhir pada rentang waktu tertentu dari asset Bitcoin
2.	<i>Date</i>	Informasi Pasar	Tanggal dan waktu tertentu yang terkait dengan aktivitas pergerakan harga bitcoin

Pada jenis dan sumber data terdapat beberapa variable yang mempunyai penjelasan sebagai berikut :

1. *Close Price* adalah harga terakhir bitcoin pada akhir periode perdagangan, *close price* sering menjadi variable yang paling diperhatikan karena mencerminkan nilai bitcoin pada akhir periode tertentu.
2. *Date* adalah tanggal dimana data tersebut berlaku

### 3.5 Teknik Pengumpulan Data

Dalam proses pengumpulan data dalam penelitian ini, teknik pengumpulan data sebagai berikut:

1. Teknik *time series*  
Teknik ini disusun dan dikumpulkan berdasarkan urutan waktu
2. Studi Pustaka  
Penelitian ini juga melakukan proses studi pustaka dengan membaca dan mempelajari literatur yang berkaitan dengan topik penelitian ini.

### 3.6 Metode Analisis Data

Pada penelitian ini digunakan model peramalan dari teknik statistik dan teknik Deep Learning yakni model LSTM (*Long Short Term Memory*). Tahapan analisis yang akan dilakukan peneliti untuk analisis menggunakan LSTM.

Tahapan untuk analisis melalui model peramalan LSTM yaitu:

1. Pengumpulan data historis harga bitcoin.
2. Pra-pemrosesan data untuk membersihkan dan mempersiapkan data, hal ini termasuk penghapusan data yang hilang atau tidak lengkap, normalisasi data dan pembagian data menjadi set pelatihan dan set pengujian.
3. Pembuatan model LSTM menggunakan kerangka kerja Tensorflow atau Keras.

4. pelatihan model LSTM dari set pelatihan yang telah dipersiapkan sebelumnya,model dilatih dengan urutan data waktu ke dalamnya dan menyesuaikan bobotnya melalui serangkaian iterasi.
5. selanjutnya melakukan validasi model,evaluasi model menggunakan set pengujian yang telah dipisahkan sebelumnya.memeriksa sejauh mana model dapat memprediksi harga bitcoin dengan akurat dengan menggunakan MSE dan RMSE untuk mengukur performa model.
6. Kemudian akan dilakukan prediksi harga setelah model dianggap memadai.
7. Lalu dilakukan evaluasi dan perbandingan hasil prediksi dengan metrik evaluasi menggunakan RMSE dan MAPE untuk mengevaluasi performa model secara kuantatif.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Penyiapan Data

Tahap ini diawali dengan melakukan pengumpulan data kemudian menjadikannya dalam bentuk dataset. Untuk data dalam penelitian ini sendiri diperoleh dari website [finance.yahoo.com](http://finance.yahoo.com). Bentuk dataset berupa data harga rata-rata bulanan, tanggal, harga tertinggi, harga terendah, harga penutupan dan volume. data yang di pakai adalah data historis harga bitcoin dalam kurun waktu  $\pm 2$  tahun 2021-2023 . Data yang di gunakan dikonversikan dalam bentuk file csv (comma separated values). Dataset yang terkumpul sebanyak 1010 data ditiap masing-masing kolom. Selanjutnya dataset akan diolah untuk mengambil data yang dibutuhkan untuk prediksi disini data yang digunakan mengandung dua atribut yaitu tanggal dan harga bitcoin. Penjelasan atribut dan beberapa contoh instant data disajikan pada **Tabel 6** dan **Tabel 7**.

**Tabel 6.**Atribut Dataset

No	Atribut	Nilai Atribut
1	Tanggal	2021-2022-2023
2	<i>Close Price</i> (USD)	28.994,32.810,31.977, dll

Sumber:Olah Data

**Tabel 7.**Sample Dataset Bitcoin

No	Tanggal	<i>Close Price</i>
1	2021-01-01	29374.152344
2	2021-01-02	32127.267578
3	2021-01-03	32782.023438
4	2021-01-04	31971.914063
5	2021-01-05	33992.429688
6	2023-10-03	27429.978516
7	2023-10-04	27799.394531
8	2023-10-05	27415.912109
9	2023-10-06	27946.597656
10	2023-10-07	27986.648438

Sumber: [finance.yahoo.com](http://finance.yahoo.com)

Perubahan harga bitcoin sangat dipengaruhi oleh berbagai faktor, termasuk kebijakan pemerintah dan regulasi global, seperti kebijakan Komisi Sekuritas dan Bursa (SEC) Amerika Serikat. Dari plot data, terlihat bahwa kecenderungan peningkatan harga Bitcoin dari waktu ke waktu, khususnya pada awal tahun dan akhir tahun, sedangkan harga Bitcoin cenderung menurun pada pertengahan tahun. Fluktuasi harga bitcoin dari tahun 2021 sampai 2023 mencapai 64.74%. Peningkatan harga bitcoin tertinggi terjadi pada periode November 2021 dan selanjutnya Bitcoin mengalami penurunan sampai saat ini. Beberapa komponen besar yang menyebabkan kenaikan harga bitcoin adalah ancaman inflasi yang terjadi pada tahun 2021 dan naiknya nama bitcoin yang disebabkan oleh beberapa faktor seperti banyaknya minat investor besar yang mencoba berinvestasi pada bitcoin, Berikut perjalanan data harga bitcoin selama 2 tahun terakhir ditampilkan dalam bentuk grafik dibawah ini.



**Gambar 12.** Grafik harga bitcoin dari tahun 2021-2023

## 4.2 Praproses Data

Setelah menyiapkan dataset, langkah selanjutnya ialah Praproses data. Praproses data ini dilakukan juga dengan melihat ada atau tidaknya missing value. Di dalam data yang dipakai dalam penelitian ini tidak ada *missing value* sehingga peneliti dapat melanjutkan normalisasi data. Dalam meminimalkan error, dilakukan normalisasi pada dataset dengan mengubah data aktual menjadi nilai dengan range interval  $[0,1]$ . Teknik yang digunakan pada proses normalisasi yaitu min-max scaler. Adapun untuk rumus normalisasi ialah sebagai berikut:

$$norm = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Keterangan :

$x$  = nilai yang akan dinormalisasi

$\min(x)$  = nilai minimum data

$\max(x)$  = nilai maksimum data

berikut contoh perhitungan *min-max scaler* menggunakan data harga penutupan Bitcoin:

1. Normalisasi x Variabel pertama diketahui:

$$\begin{aligned} x &= 19416 \\ \min(x) &= 15787 \\ \max(x) &= 67566 \end{aligned}$$

jawaban:

$$norm = \frac{x - \min(x)}{\max(x) - \min(x)} = \frac{19416 - 15787}{67566 - 15787} = \frac{3659}{51779} = 0,23133014$$

Berikut hasil normalisasi data dalam bentuk table:

**Tabel 8.** Hasil Normalisasi Data Harga Bitcoin

Harga Aktual	Denormalisasi
19416	0.23133014
19446	0.23323325
19051	0.21379632
19157	0.2080555
19382	0.21481367
19185	0.2291844
19067	0.21661183
19268	0.20908916
19550	0.22186639
19334	0.23988336

Sumber : Hasil Olah Data

Setelah melakukan normalisasi data (teknik normalisasi min max scaler), selanjutnya membagi data menjadi data latih dan data uji. Skenario pembagian dataset untuk eksperimen disiapkan berdasarkan komposisi data latih dan data uji dalam presentase. Skenario uji coba disajikan pada **Tabel 9**. Pada penelitian ini peneliti menggunakan eksperimen parameter epoch 200 neuron 10 dan menggunakan optimasi Adam dengan perbandingan 70% untuk data latih 30% untuk data uji, perbandingan 80 % untuk data latih 20 % untuk data uji, serta perbandingan 90 % untuk data latih 10 % untuk data uji. **Tabel 9** menunjukkan hasil perbandingan tiap pembagian data latih dan data uji yang mana lebih akurat dengan melihat nilai error terkecil menggunakan perhitungan RMSE.

**Tabel 9.** Hasil Pembagian Data Latih dan Data Uji

<b>Eksperimen</b>	<b>Data Latih</b>	<b>Data Uji</b>	<b>RMSE</b>
1	70%	30%	1249
2	80%	20%	1244
3	90%	10%	1041

Pembuatan data latih dimaksud agar meningkatkan kinerja LSTM terhadap data uji. Data latih lebih besar dibandingkan data uji dengan tujuan mesin pembelajaran atau algoritma pembelajaran lebih terlatih dengan pola data dari data latih. Hal ini berguna ketika algoritma atau mesin menghasilkan suatu model dan model tersebut diberikan kepada data uji akan memberikan prediksi data uji yang lebih akurat. Data latih yang sudah didapatkan, akan digunakan untuk proses pelatihan. Proses pelatihan dilakukan dengan menggunakan metode LSTM sehingga terbentuk suatu model yang nanti akan diuji performasinya terhadap data uji. Proses tersebut terus diulang hingga mendapatkan model dengan akurasi yang paling bagus. Kemudian setelah didapatkan model yang terbaik, model tersebut akan digunakan untuk proses prediksi. Dalam penelitian ini menggunakan data latih 90% dan data uji 10% karena mempunyai nilai error terkecil menggunakan persamaan RMSE yaitu sebesar 1041.

### **4.3 Pengelolaan Model LSTM**

Setelah Membagi data menjadi data latih dan data uji, langkah selanjutnya ialah menentukan inisialisasi parameter LSTM yang akan dibutuhkan. Langkah ini terdiri dari mendefinisikan neuron (*hidden layer*) dan mendefinisikan jumlah *epoch* untuk iterasi dalam suatu jaringan LSTM. Setelah itu proses pembelajaran jaringan LSTM dapat dilakukan. Perbandingan beberapa parameter sangat penting dilakukan agar menghasilkan model terbaik dalam mencapai prediksi yang akurat. Dalam menentukan jumlah *neuron*, tidak ada aturan yang mengatur banyaknya *neuron* yang digunakan, sehingga nilai *neuron* dapat dipakai peneliti dengan terlebih dahulu melakukan percobaan sampai mendapatkan hasil yang optimal. Begitu juga dengan *epoch*, *epoch* adalah langkah yang dilakukan pada proses pembelajaran *neural network*. Besarnya *epoch* yang telah ditetapkan akan mempengaruhi besaran proses pembelajaran dan berhenti tepat pada nilai *epoch* yang telah ditentukan tersebut. Jaringan dengan beberapa *neuron* pada *hidden layer* tidak mampu menangkap hubungan kompleks antara variabel target dan input. Namun, jika terlalu banyak *neuron* ditugaskan di bagian tersembunyi dari jaringan maka akan membuat kemampuan prediksi buruk dari data yang tidak

terlihat yang menyebabkan *overparameterization*. Oleh sebab itu, sejumlah *neuron* dalam lapisan tersembunyi harus dicoba serta diuji dan kinerja arsitektur harus dinilai setelah masing- masing dijalankan dengan sejumlah *neuron* dalam lapisan tersembunyi. Berdasarkan penelitian-penelitian sebelumnya, jumlah *neuron* serta time yang optimal dipengaruhi oleh berbagai faktor salah satunya yaitu algoritma optimasi. Pada penelitian ini data yang dipakai ialah data harga Bitcoin serta jumlah neuron pada *hidden layer* yang akan digunakan untuk percobaan yaitu 10, 20, 30, 40, dan 50 sedangkan untuk *epoch* yang akan digunakan ialah 200, 300 dan 400 dengan menggunakan optimasi Adam. Untuk mengetahui jumlah *neuron*, *epoch* dan optimasi yang tepat dapat dilihat nilai *error* yang terkecil, dimana nilai *error* yang dipakai peneliti menggunakan persamaan RMSE( *Root Mean Square Error*) dan MAE( *Mean Absolute Error*).

#### 4.3.1 Penentuan Jumlah Neuron dan Epoch

Penentuan jumlah optimal *neuron* pada *hidden layer* dan *epoch* adalah tugas penting dalam metode LSTM. Jaringan dengan beberapa *neuron* pada *hidden layer* mungkin tidak mampu menangkap hubungan kompleks antara variabel target dan input. Namun, jika terlalu banyak *neuron* ditugaskan di bagian tersembunyi dari jaringan maka akan membuat kemampuan prediksi buruk dari data yang tidak terlihat karena *overparameterization*. Oleh karena itu, sejumlah neuron dalam lapisan tersembunyi harus dicoba, dan kinerja arsitektur harus dinilai setelah masing-masing dijalankan dengan sejumlah *neuron* dalam lapisan tersembunyi.

Pada penelitian ini jaringan yang dibentuk dengan jumlah neuron pada *hidden layer* yang akan digunakan untuk percobaan yaitu 10, 20, 30, 40, dan 50. Sedangkan untuk *epoch* akan digunakan 200, 300, dan 400. Untuk mengetahui jumlah *neuron* dan *epoch* yang tepat dapat dilihat nilai loss yang terkecil, dimana nilai loss merupakan nilai MSE. Serta menggunakan optimasi Adam. Optimasi adam digunakan karena menggunakan teknik bias *correction*. Tidak ada aturan dalam menentukan jumlah neuron dan epoch sehingga jumlah neuron dan epoch didapatkan melalui percobaan sampai mendapatkan yang paling optimal dalam memprediksi time series. Berikut merupakan tabel hasil pengujian dengan beberapa jumlah *neuron* dan *epoch* tersebut.

**Tabel 10.** Hasil Pengujian Data Harga Bitcoin

Jumlah Neuron	Epoch	RMSE	MAE
10	200	1041	617.64
	300	1037	606.56

	400	1077	675.88
20	200	1106	726.57
	300	1043	631.86
	400	1053	654.21
30	200	1041	615.11
	300	1042	612.63
	400	1100	731.52
40	200	1043	633.79
	300	1182	839.93
	400	1219	903.91
50	200	1241	927.52
	300	1040	634.86
	400	1027	597.96

Berdasarkan **Tabel 10** dapat diketahui bahwa jumlah neuron 50 dan jumlah epoch (iterasi) 400 menghasilkan nilai RMSE dan MSE yang terkecil jika dibandingkan dengan jumlah neuron dan epoch lainnya. Jumlah epoch yang terlalu sedikit mengakibatkan jaringan yang terbentuk bersifat terlalu general (umum), artinya kemampuan jaringan dalam mengenali pola terlalu sedikit atau tidak ada sama sekali. Sedangkan jumlah epoch yang terlalu banyak dapat mengakibatkan jaringan mengalami kondisi overfit atau jaringan terlalu bersifat terlalu spesifik terhadap data pelatihan.

Nilai bobot dan bias yang didapatkan dari 50 neuron dan 400 epochs dapat dilihat pada **Gambar 13** dan **Gambar 14**.

```

Input Gate Weights:
W_i: [[ 0.16494784 -0.05623257 -0.00733709 -0.1062664 ]]
W_f: [[ 6.42804727e-02 -1.16170496e-01 -9.31008607e-02 7.88744539e-02]
 [ 3.25018615e-02 -2.21851602e-01 1.88105792e-01 6.73529729e-02]
 [ 9.65996608e-02 -6.21261708e-02 5.43904714e-02 -1.73027486e-01]
 [-1.28891945e-01 -4.61526625e-02 -3.70108858e-02 -3.86237465e-02]
 [ 9.03991535e-02 3.06459647e-02 -3.47728133e-02 -2.83107888e-02]
 [-2.12060098e-04 4.65988576e-01 1.02564901e-01 -4.55324426e-02]
 [-6.80684671e-02 2.24245880e-02 7.92758423e-04 6.71912730e-03]
 [-9.25345495e-02 -1.12534054e-01 1.22993588e-01 -6.61549419e-02]
 [-6.73076436e-02 -2.21242875e-01 1.93689894e-02 4.09072861e-02]
 [-2.75917593e-02 1.12838097e-01 -1.48952454e-01 -5.67767285e-02]
 [ 3.73743512e-02 -7.60522410e-02 1.50848995e-03 2.68282983e-02]
 [-8.40054676e-02 -1.96220458e-01 -7.31817260e-02 1.84159353e-01]
 [-1.16764180e-01 -8.42392147e-02 -5.53829521e-02 9.57078412e-02]
 [-6.99611008e-02 -1.05000839e-01 8.34298283e-02 -4.04170491e-02]
 [ 4.91896234e-02 5.75907826e-02 2.63826828e-02 -6.05394617e-02]
 [-7.24681318e-02 -7.97444060e-02 -9.95278582e-02 -1.12923093e-01]
 [-3.59751694e-02 9.35318619e-02 -1.09633267e-01 -9.72223654e-03]

```

**Gambar 13.** Hasil Bobot Dari Setiap Neuron

```

Input Gate Bias:
b_i: [[ 0.16494784 -0.05623257 -0.00733709 -0.1062664 ]]
b_f: [[ 6.42804727e-02 -1.16170496e-01 -9.31008607e-02 7.88744539e-02]
[ 3.25018615e-02 -2.21851602e-01 1.88105792e-01 6.73529729e-02]
[ 9.65996608e-02 -6.21261708e-02 5.43904714e-02 -1.73027486e-01]
[-1.28891945e-01 -4.61526625e-02 -3.70108858e-02 -3.86237465e-02]
[ 9.03991535e-02 3.06459647e-02 -3.47728133e-02 -2.83107888e-02]
[-2.12060098e-04 4.65988576e-01 1.02564901e-01 -4.55324426e-02]
[-6.80684671e-02 2.24245880e-02 7.92758423e-04 6.71912730e-03]
[-9.25345495e-02 -1.12534054e-01 1.22993588e-01 -6.61549419e-02]
[-6.73076436e-02 -2.21242875e-01 1.93689894e-02 4.09072861e-02]
[-2.75917593e-02 1.12838097e-01 -1.48952454e-01 -5.67767285e-02]
[ 3.73743512e-02 -7.60522410e-02 1.50848995e-03 2.68282983e-02]
[-8.40054676e-02 -1.96220458e-01 -7.31817260e-02 1.84159353e-01]
[-1.16764180e-01 -8.42392147e-02 -5.53829521e-02 9.57078412e-02]
[-6.99611008e-02 -1.05000839e-01 8.34298283e-02 -4.04170491e-02]
[ 4.91896234e-02 5.75907826e-02 2.63826828e-02 -6.05394617e-02]
[-7.24681318e-02 -7.97444060e-02 -9.95278582e-02 -1.12923093e-01]
[-3.59751694e-02 9.35318619e-02 -1.09633267e-01 -9.72223654e-03]

```

**Gambar 14.** Bias Untuk Setiap Neuron

Berdasarkan **Gambar 13**  $W_i[0]$ ,  $W_f[0]$ ,  $W_c[0]$ , dan  $W_o[0]$  berturut-turut merupakan bobot pada variabel pertama (*Close*) untuk input gate, forget gate, cell state, dan output gate. Input gate merupakan gerbang yang menentukan berapa banyak input jaringan waktu saat ini yang dicadangkan ke dalam status sel, forget gate merupakan gerbang yang menentukan informasi status sel mana yang dibuang dari model, sedangkan output gate merupakan gerbang yang mengontrol seberapa banyak keadaan sel saat ini dibuang banyaknya bobot sesuai dengan banyaknya neuron yang digunakan yaitu 30 dan masing-masing variabel memiliki bobotnya masing-masing. Sehingga ketika dijumlahkan terdapat 600 bobot untuk masing-masing input gate, forget gate, cell state, dan output gate. Sedangkan untuk **Gambar 14**  $b_i$ ,  $b_f$ ,  $b_c$ , dan  $b_o$  berturut-turut merupakan bias untuk input gate, forget gate, cell state, dan output gate. Banyaknya bias sesuai dengan banyaknya neuron yang digunakan yaitu 30 sehingga ketika dijumlahkan terdapat 120 bias secara keseluruhan

#### 4.4 Prediksi Data

Setelah mendapatkan model pada proses training, maka model tersebut akan diuji menggunakan data testing yang telah didapat pada pembentukan data training dan data testing. Untuk melihat baik atau tidaknya hasil dari model yang dibentuk dapat dilihat dari plot data aktual dan data prediksi. Plot tersebut dapat dilihat pada **Gambar 15**:



**Gambar 15.** Perbandingan Data Prediksi dan Data Aktual

Pada **Gambar 15** dapat dilihat bahwa model yang dibentuk dapat menghasilkan output yang sesuai. Hal tersebut dapat dilihat dari pola data prediksi yang mengikuti pola yang dibentuk oleh data aktual. Data aktual diwakili oleh warna biru dan data prediksi diwakili oleh warna hijau. Pada **Tabel 11** akan ditampilkan prediksi harga. Hasil prediksi sebelumnya ditampilkan dalam nilai dengan range interval  $[0;1]$ , sehingga nilai tersebut harus dikembalikan ke dalam nilai sebenarnya dengan melakukan denormalisasi. Di bawah ini merupakan hasil perbandingan harga aktual dan harga prediksi.

**Tabel 11.** Hasil Pengujian

No.	Tanggal	Harga	
		Harga Aktual	Prediksi
1.	2021-01-17	36178,14	37224,69
2.	2021-01-18	35791,28	36411,66
3.	2021-01-19	36630,07	35898,15
4.	2021-01-20	36069,8	36589,23
5.	2021-01-21	35547,75	36168,8
6.	2021-01-22	30825,7	35656,97
7.	2021-01-23	33005,76	31275,86
8.	2021-01-24	32067,64	32718,23
9.	2021-01-25	32289,38	32020,35
10.	2021-01-26	32366,39	32215,19

Hasil dari setiap model yang dilatih diukur oleh Mean Absolute Percentage Error (MAPE). Nilai MAPE yang dihasilkan dari prediksi dapat dilihat pada **Gambar 16**.

```
In [22]: # Hitung metrik untuk data latih
train_rmse = math.sqrt(mean_squared_error(original_ytrain, train_predict))
train_mse = mean_squared_error(original_ytrain, train_predict)
train_mae = mean_absolute_error(original_ytrain, train_predict)

# Hitung MAPE untuk data latih
train_mape = np.mean(np.abs((original_ytrain - train_predict) / original_ytrain)) * 100

# Tampilkan metrik untuk data latih
print("Train data RMSE:", train_rmse)
print("Train data MSE:", train_mse)
print("Train data MAE:", train_mae)
print("Train data MAPE:", train_mape, "%")

# Menghitung akurasi MAPE untuk data latih
accuracy_mape_train = 100 - train_mape
print("Accuracy MAPE (Training):", accuracy_mape_train, "%")

# Hitung metrik untuk data uji
test_rmse = math.sqrt(mean_squared_error(original_ytest, test_predict))
test_mse = mean_squared_error(original_ytest, test_predict)
test_mae = mean_absolute_error(original_ytest, test_predict)

# Hitung MAPE untuk data uji
test_mape = np.mean(np.abs((original_ytest - test_predict) / original_ytest)) * 100

# Tampilkan metrik untuk data uji
print("Test data RMSE:", test_rmse)
print("Test data MSE:", test_mse)
print("Test data MAE:", test_mae)
print("Test data MAPE:", test_mape, "%")

# Menghitung akurasi MAPE untuk data uji
accuracy_mape_test = 100 - test_mape
print("Accuracy MAPE (Testing):", accuracy_mape_test, "%")

# Hitung MAPE keseluruhan
overall_mape = (train_mape + test_mape) / 2

# Tampilkan metrik keseluruhan
print("Nilai Keseluruhan RMSE:", math.sqrt((train_rmse**2 + test_rmse**2) / 2))
print("Nilai Keseluruhan MSE:", (train_mse + test_mse) / 2)
print("Nilai Keseluruhan MAE:", (train_mae + test_mae) / 2)
print("Nilai Keseluruhan MAPE:", overall_mape, "%")

# Menghitung akurasi MAPE keseluruhan
accuracy_mape_overall = 100 - overall_mape
print("Accuracy MAPE (Overall):", accuracy_mape_overall, "%")

Train data RMSE: 1376.5338260973613
Train data MSE: 1894845.3743902403
Train data MAE: 909.6923631266101
Train data MAPE: 2.4793838760615916 %
Accuracy MAPE (Training): 97.520612359385 %
Test data RMSE: 466.4709235180734
Test data MSE: 217595.1224889095
Test data MAE: 286.2279181691176
Test data MAPE: 1.0415454039520171 %
Accuracy MAPE (Testing): 98.9584549904799 %
Nilai Keseluruhan RMSE: 1027.725765190897
Nilai Keseluruhan MSE: 1056220.2484355704
Nilai Keseluruhan MAE: 597.9681486478638
Nilai Keseluruhan MAPE: 1.760692084067595 %
Accuracy MAPE (Overall): 98.23930791599324 %
```

**Gambar 16.** Nilai MAPE Dari Model Terbaik

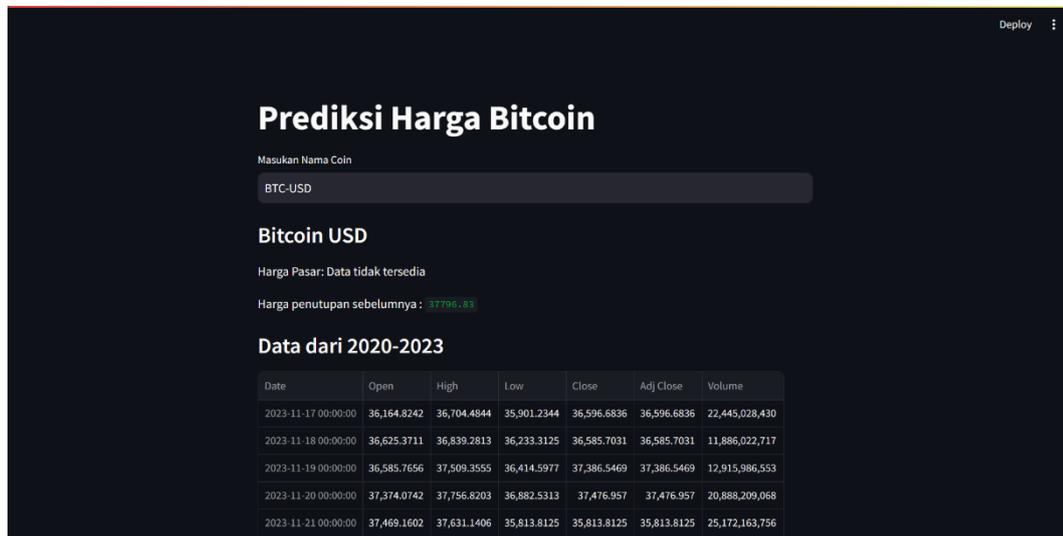
Dari nilai MAPE tersebut dapat dilihat bahwa hasil prediksi dengan menggunakan 50 neuron dan 400 epoch memiliki nilai akurasi sebesar 98,2 %.

#### 4.5 Implementasi GUI (*Graphical User Interface*)

Dalam membuat GUI (Graphical User Interface) prediksi harga bitcoin, peneliti menggunakan framework streamlit. Streamlit merupakan sebuah framework berbasis python dan bersifat open-source yang dibuat untuk memudahkan dalam membangun aplikasi web di bidang sains data dan machine learning yang interaktif. Streamlit mudah digunakan karena tidak memerlukan pengetahuan tentang pengembangan web misalnya HTML (Hypertext Markup Language) ataupun CSS (Cascading Style Sheet) sehingga tidak perlu untuk mengatur tata letak yang sedemikian rupa dan metodenya yang mudah menjadikan streamlit sebagai tools yang bagus untuk menjelajahi kumpulan data, mendemonstrasikan model pembelajaran mesin, visi komputer, pemrosesan bahasa alami, visualisasi data, dan banyak proyek berpusat pada data lainnya. Hal yang dilakukan setelah meng-install streamlit ialah meng-import nya. Setelah meng-import streamlit, selanjutnya menjalankan aplikasi streamlit dengan

mengetik streamlit run namafile.py pada terminal. Pada penelitian ini, peneliti menggunakan fungsi streamlit yaitu `st.file_uploader("pilih file")` dimana fungsi ini dapat mengunggah atau menyimpan file.

Berikut GUI yang dibuat oleh peneliti untuk prediksi data harga bitcoin dimana data yang diinput ialah data harga bitcoin yang diambil dari yahoo finance menggunakan library `yfinance` dan output yang dihasilkan merupakan nilai hasil prediksi dan grafik hasil prediksi.



**Gambar 17.** Halaman Utama GUI Prediksi Harga Bitcoin

Gambar diatas merupakan halaman utama implementasi GUI yang telah dibuat oleh peneliti. Untuk mengetahui hasil prediksi, terlebih dahulu menginputkan *cryptocurrency* apa yang ingin di prediksi karena pada penelitian ini berfokus terhadap bitcoin jadi data yang akan ditampilkan adalah data bitcoin. Selanjutnya, harga akan diproses untuk menampilkan hasil prediksi hingga 1 tahun kedepan.



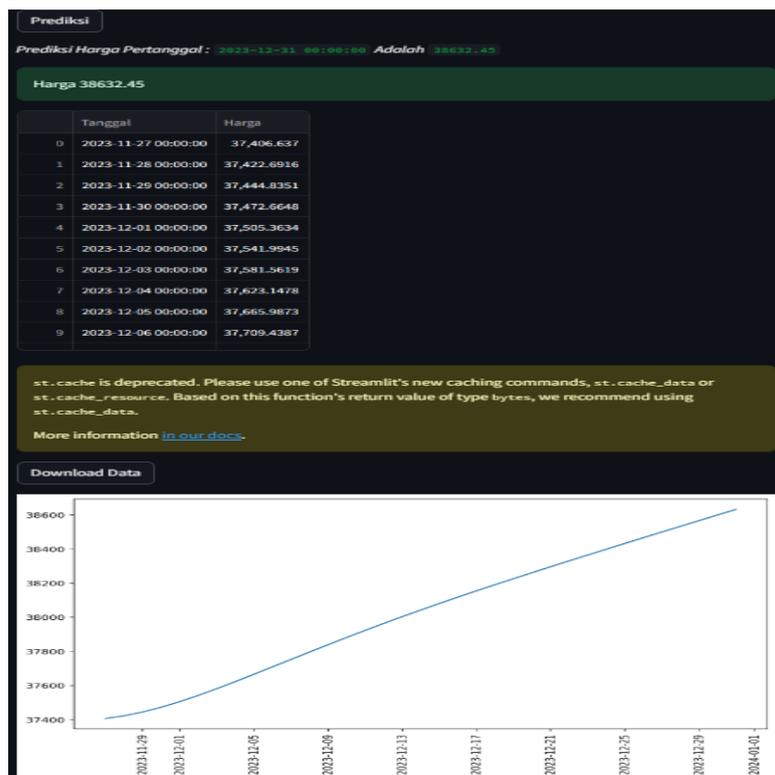
**Gambar 18.** GUI Chart Harga Bitcoin Yang di Tambahkan Teknikal Analisis

Selain menampilkan hasil prediksi selama 1 tahun kedepan, dalam implementas GUI menggunakan streamlit ini juga dapat menampilkan hasil grafik prediksi juga ditambahkan fitur teknikal analisis untuk membantu pengguna memprediksi harga dengan lebih baik lagi. Berikut hasil grafik yang ditampilkan.



**Gambar 19.** Chart Hasil Prediksi Harga Bitcoin dan Harga Asli Bitcoin

Fitur selanjutnya adalah pengguna bebas memilih sampai mana prediksi yang di inginkan dengan maksimal prediksi yaitu 1 tahun, harga prediksi akan ditampilkan dalam bentuk tabel yang dapat di unduh menjadi file excel. Yang bisa dilihat pada **Gambar 20**.



**Gambar 20.** Hasil Prediksi Harga Bitcoin

#### 4.6 Evaluasi Model

Pada penelitian ini menggunakan persamaan RMSE (*Root Mean Square Error*) untuk menghitung akurasi prediksi dengan melihat nilai error yang didapatkan. *Root Mean Square Error* (RMSE) adalah metrik yang digunakan untuk mengukur seberapa akurat suatu model dalam memprediksi nilai. RMSE mengukur seberapa besar deviasi antara nilai prediksi dengan nilai sebenarnya, dan memberikan gambaran tentang seberapa baik model dalam melakukan prediksi. RMSE dihitung dengan cara mengambil akar kuadrat dari rata-rata dari kuadrat selisih antara nilai prediksi dan nilai sebenarnya.

Kegunaan RMSE adalah untuk mengevaluasi kinerja suatu model prediksi. Semakin kecil nilai RMSE, semakin baik pula kinerja model dalam melakukan prediksi. Adapun untuk rumus persamaan RMSE ialah sebagai berikut:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (x_t - f_t)^2}{n}}$$

Dimana:

- $x_t$  = nilai aktual pada waktu  $t$
- $f_t$  = nilai prediksi pada waktu  $t$
- $t$  = waktu
- $n$  = jumlah data

Berikut perhitungan persamaan RMSE menggunakan data harga bitcoin dengan data latih 90 %, data uji 10 %, optimasi Adam, epoch 50, dan neuron 400.

Diketahui :

$$n = 10$$

$$xt1 = 26.664$$

$$xt10 = 27.415$$

$$ft1 = 26.683$$

$$ft10 = 27.754$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (xt-ft)^2}{n}} = \sqrt{\frac{\sum_{t=1}^n (26.664-26.683)^2 + (27.415-27.754)^2}{10}} = 1041$$

Berikut tabel yang menyajikan hasil perhitungan RMSE menggunakan optimasi Adam:

**Tabel 12.** Perhitungan RMSE Oprimasi Adam

<b>Jumlah Neuron</b>	<b>Epoch</b>	<b>RMSE</b>
10	200	1041
	300	1037
	400	1077
20	200	1106
	300	1043
	400	1053
30	200	1041
	300	1042
	400	1100
40	200	1043
	300	1182
	400	1219
50	200	1241
	300	1040
	400	1027

Selain menggunakan perhitungan RMSE, penulis juga menggunakan perhitungan MAE (*Mean Absolute Error*) untuk mengetahui akurasi prediksi dengan melihat nilai error. Mean Absolute Error (MAE) adalah salah satu metode yang digunakan untuk mengukur tingkat keakuratan model peramalan. MAE mengukur rata-rata dari nilai absolut selisih antara prediksi model dengan nilai yang diamati. Kegunaan MAE adalah untuk mengevaluasi seberapa akurat model

peramalan dalam memprediksi nilai. Semakin kecil nilai MAE, semakin baik pula kinerja model dalam melakukan prediksi. Adapun untuk rumus persamaan MAE sendiri ialah sebagai berikut:

$$MAE = \sqrt{\frac{\sum_{t=1}^n (xt - ft)}{n}}$$

Dimana:

- $xt$  = nilai aktual pada waktu  $t$
- $ft$  = nilai prediksi pada waktu  $t$
- $t$  = waktu
- $n$  = jumlah data

Berikut perhitungan persamaan MAE menggunakan data harga bitcoin dengan data latih 90 %, data uji 10 %, optimasi Adam, epoch 50, dan neuron 400.

Diketahui:

- $n$  = 10
- $xt1$  = 26.664
- $xt10$  = 27.415
- $ft1$  = 26.683
- $ft10$  = 27.754

MAE?

$$MAE = \sqrt{\frac{\sum_{t=1}^n (xt-ft)^2}{n}} = \sqrt{\frac{\sum_{t=1}^n (26.664-26.683)^2 + (27.415-27.754)^2}{10}} = 617.64$$

Berikut tabel yang menyajikan hasil perhitungan MAE menggunakan optimasi Adam:

**Tabel 13.** Perhitungan MAE Optrimasi Adam

Jumlah Neuron	Epoch	MAE
10	200	617.64
	300	606.56
	400	675.88
20	200	726.57
	300	631.86
	400	654.21
30	200	615.11
	300	612.63
	400	731.52
40	200	633.79
	300	839.93

	400	903.91
50	200	927.52
	300	634.86
	400	597.96

---

Selain RMSE dan MAE pada penelitian ini digunakan juga perhitungan MAPE (*Mean Absolute Percentage Error*) yang digunakan untuk mengukur tingkat keakuratan model peramalan. MAPE mengukur rata-rata dari nilai persentase selisih antara prediksi model dengan nilai yang diamati. Kegunaan MAPE adalah untuk mengevaluasi seberapa akurat model peramalan dalam memprediksi nilai dalam bentuk persentase.

$$MAPE = \sqrt{\frac{\sum_{t=1}^n (At - Ft)}{At}} \times 100\%$$

Keterangan:

MAPE = mean absolute percentage error

n = jumlah data

at = nilai hasil aktual

ft = nilai hasil prediksi

Berikut perhitungan persamaan MAE menggunakan data harga bitcoin data latih 90 %, data uji 10 %, optimasi Adam, epoch 50, dan neuron 400.

Diketahui:

$$n = 10$$

$$At = [26.664, 26.049, 26.096, 26.189, 26.124, 27.983, 27.530, 27.429, 27.799, 27.415]$$

$$Ft = [26.683, 26.612, 26.779, 27.209, 27.222, 27.073, 27.940, 27.720, 27.493, 27.754]$$

$$MAPE = \frac{1}{n} \sqrt{\frac{\sum_{t=1}^n (y_i - \hat{y}_i)}{\hat{y}_i}} \times 100\% = \frac{1}{10} \left( \frac{26.664 - 26.683}{26.664} + \frac{26.049 - 26.612}{26.049} + \frac{26.189 - 26.779}{26.189} \right) \times 100$$

$$MAPE = \frac{1}{n} \sqrt{\frac{\sum_{t=1}^n (y_i - \hat{y}_i)}{\hat{y}_i}} \times 100\% = \frac{1}{10} (0.000711 + 0.0216 + 0.0225) \times 100 = 0.982$$

Dapat diketahui dari hasil prediksi yang telah dilakukan, bahwa dengan menggunakan metode LSTM dapat mencapai hasil mendekati akurat dengan nilai error terkecil menggunakan persamaan RMSE sebesar 1027 dan persamaan MAE sebesar 597.96 untuk data harga bitcoin.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan pada penelitian yang telah dilakukan, maka dapat diperoleh kesimpulan dari penelitian mengenai prediksi harga bitcoin menggunakan metode deep learning dengan metode Long Short Term Memory (LSTM) yaitu sebagai berikut:

1. Berdasarkan hasil penelitian yang sudah dilakukan dengan menggunakan model lstm dengan komposisi jumlah neuron 50 dan epoch 400 menggunakan optimasi adam mendapatkan nilai root mean square error (RMSE) dan mean absolute error(MAE) terbaik. untuk data harga bitcoin mendapatkan nilai RMSE sebesar 1027 serta nilai MAE yang dihasilkan sebesar 597.96 dan juga dihasilkan akurasi MAPE yang baik yaitu sebesar 98,2%.
2. Berdasarkan prediksi harga bitcoin menggunakan *deep learning* dengan metode *long short term memory*(LSTM) yang telah dilakukan oleh penulis didapatkan bahwasanya percobaan prediksi yang dilakukan untuk 1 bulan kedepan harga bitcoin cenderung akan mengalami kenaikan dari prediksi yang penulis lakukan yaitu pada 28 november 2023

#### **5.2 Saran**

Dalam memprediksi harga bitcoin menggunakan metode deep learning dengan metode Long Short Term Memory (LSTM) ini masih terdapat berbagai keterbatasan dan kekurangan sehingga memerlukan pengembangan lebih lanjut agar prediksi yang dibuat lebih akurat. Berdasarkan keterbatasan waktu maka peneliti menyarankan untuk dilakukan pengembangan penelitian sebagai berikut:

1. Dapat menambahkan faktor-faktor yang dapat berpengaruh terhadap pergerakan harga bitcoin sebagai variasi dataset untuk memadukan antara dataset harga bitcoin dan juga atribut-atribut lainnya seperti menggunakan data sentimen dan lain sebagainya.
2. Penelitian selanjutnya dapat mencoba menggunakan opsi optimasi lainnya seperti AdaMax, RMSpro, dll. Sehingga dapat menghasilkan prediksi yang lebih akurat.
3. Dapat menambahkan variabel-variabel lainya untuk membantu supaya mendapatkan hasil prediksi yang lebih maksimal seperti menambahkan variabel makro ekonomi ataupun variabel volatilitas pasar bitcoin.

## DAFTAR PUSTAKA

Aldi, M.W.P., Jondri and Aditsania, A. (2018) 'Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin', *e-Proceeding of Engineering*, 5(2), pp. 3548–3555.

Arfan, A. and ETP, L. (2020) 'Perbandingan Algoritma Long Short-Term Memory dengan SVR Pada Prediksi Harga Saham di Indonesia', *Petir*, 13(1), pp. 33–43. Available at: <https://doi.org/10.33322/petir.v13i1.858>.

Arwono, D.G., Iskandar, H. and Wardana, D.J. (2023) 'Tinjauan Yuridis Regulasi Cryptocurrency Terhadap Tindak Pidana Kejahatan di Indonesia', *Amnesti Jurnal Hukum*, 5(1), pp. 110–125. Available at: <https://jurnal.umpwr.ac.id/index.php/amnesti/article/view/2759>.

Deng, L. and Yu, D. (2013) 'Deep learning: Methods and applications', *Foundations and Trends in Signal Processing*, 7(3–4), pp. 197–387. Available at: <https://doi.org/10.1561/20000000039>.

Dharma, B., Gusniati, P. and ... (2023) 'Analisis Pemanfaatan Cryptocurrency Bitcoin Sebagai Alat Alternatif Investasi', ... *Sistem Informasi dan ...*, 2(1), pp. 175–182. Available at: <https://ejournal.stie-trianandra.ac.id/index.php/jupsim/article/view/858>.

Dyhrberg, A.H. (2016) 'Bitcoin, gold and the dollar - A GARCH volatility analysis', *Finance Research Letters*, 16, pp. 85–92. Available at: <https://doi.org/10.1016/j.frl.2015.10.008>.

Hasani, M.N. (2022) 'Analisis Cryptocurrency Sebagai Alat Alternatif Dalam Berinvestasi Di Indonesia', *Jurnal Ilmiah Ekonomi Bisnis*, 8(2), pp. 329–344. Available at: <http://ejournal.stiepancasetia.ac.id/index.php/jiebJilid>.

Kafil, M. (2019) 'Penerapan Metode K-Nearest Neighbors', *Jurnal Mahasiswa Teknik Informatika (JATI)*, 3(2), pp. 59–66.

Kuncara, T. and Anugrah, K.P. (2023) 'Analisis Volatilitas Cryptocurrency Pada Sebelum Pandemi Dan Pada Saat Pandemi Covid 19 Dengan Metode Return Pada Bitcoin Dan Ethereum', *Keunis*, 11(1), p. 86. Available at: <https://doi.org/10.32497/keunis.v11i1.3981>.

Magdalena, I., Fauzi, H.N. and Putri, R. (2020) 'Pentingnya Evaluasi Dalam Pembelajaran Dan Akibat Memanipulasinya', *Jurnal Pendidikan dan Sains*, 2(2), pp. 244–257. Available at: <https://ejournal.stitpn.ac.id/index.php/bintang>.

Meliana, C. (2021) 'Perbandingan Metode Long Short Term Memory (LSTM) DAN Genetic Algorithm-Long Short Term Memory (GA-LSTM) Pada Peramalan Polutan Udara'. Available at: <http://repository.unimus.ac.id/4751/7/BABII.pdf>.

Muhtadi, M.M., Friyadi, M.D. and Rahmani, A. (2019) 'Analisis GUI Testing pada Aplikasi E-Commerce menggunakan Katalon', *Prosiding Industrial Research*

*Workshop and National Seminar*, 10(1), pp. 1387–1393.

Nawangwulan, S. and Angesti, D. (2016) 'Analisis Time Series Metode Winter Jumlah Penderita Gastroenteritis Rawat Inap Berdasarkan Data Rekam Medis Di Rsud Dr. Soetomo Surabaya', *Jurnal Manajemen Kesehatan Yayasan RS.Dr. Soetomo*, 2(1), p. 17. Available at: <https://doi.org/10.29241/jmk.v2i1.48>.

Ngantung, M. *et al.* (2019) 'Analisis Peramalan Permintaan Obat Antibiotik Pada Apotik Edelweis Tatelu', *Jurnal EMBA: Jurnal Riset Ekonomi, Manajemen, Bisnis dan Akuntansi*, 7(4), pp. 4859–4867. Available at: <https://doi.org/10.35794/emba.v7i4.25439>.

Oxaichiko Arissinta, I., Dwi Sulistiyawati, I. and Kurnianto Iqbal Kharisudin, D. (2022) 'Pemodelan Time Series untuk Peramalan Web Traffic Menggunakan Algoritma Arima', *Prosiding Seminar Nasional Matematika*, 5, pp. 693–700. Available at: <https://journal.unnes.ac.id/sju/index.php/prisma/>.

Paquin, F. *et al.* (2015) 'Multi-phase semicrystalline microstructures drive exciton dissociation in neat plastic semiconductors', *J. Mater. Chem. C*, 3, pp. 10715–10722. Available at: <https://doi.org/10.1039/b000000x>.

Pramono, A.A.D.I. *et al.* (2022) 'Memprediksi Harga Cryptocurrency Dengan Menggunakan Metode Long Short Term Memory (Lstm) Halaman Judul Skripsi Oleh Arik Adi Pramono Nim. 15610076 Program Studi Matematika Fakultas Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang'.

Prasetyanwar, H. and Jondri (2018) 'Peramalan nilai tukar IDR-USD menggunakan long short term memory', *e-Proceeding of Engineering*, 5(2), pp. 3820–3826.

Prasetyo, A.S. *et al.* (2017) 'Analisis Potensi dan Risiko Investasi Cryptocurrency di Indonesia', *Jurnal Ilmiah Information Technology d'Computare*, 7(1), pp. 17–22. Available at: <https://doi.org/10.35957/jatisi.v7i2.195>.

Ridho, I.I. *et al.* (2022) 'Metode Neural Network Untuk Penentuan Akurasi Prediksi Harga Rumah', *Technologia: Jurnal Ilmiah*, 13(1), p. 56. Available at: <https://doi.org/10.31602/tji.v13i1.6252>.

Riyantoko, P.A. *et al.* (2020) 'Analisis Prediksi Harga Saham Sektor Perbankan Menggunakan Algoritma Long-Short Terms Memory (Lstm)', *In Seminar Nasional Informatika (SEMNASIF)*, 1(1), pp. 427–435. Available at: <http://www.jurnal.upnyk.ac.id/index.php/semnasif/article/view/4135>.

Selle, N., Yudistira, N. and Dewi, C. (2022) 'Perbandingan Prediksi Penggunaan Listrik dengan Menggunakan Metode Long Short Term Memory (LSTM) dan Recurrent Neural Network (RNN)', *Jurnal Teknologi Informasi dan Ilmu Komputer*, 9(1), p. 155. Available at: <https://doi.org/10.25126/jtiik.2022915585>.

Sianturi, T.B., Cholissodin, I. and Yudistira, N. (2023) 'Penerapan Algoritma Long Short-Term Memory ( LSTM ) berbasis Multi Fungsi Aktivasi Terbobot dalam Prediksi Harga Ethereum', 7(3), pp. 1101–1107.

Wiranda, L. and Sadikin, M. (2019) 'Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk Pt. Metiska Farma', *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 8(3), pp. 184–196.

## LAMPIRAN

### Lampiran 1

```
#library yang dibutuhkan
import tensorflow as tf
import math
import numpy as np
import pandas as pd
#library untuk evaluasi hasil
from sklearn.metrics import mean_squared_error, mean_absolute_error,
explained_variance_score, r2_score
from sklearn.metrics import mean_poisson_deviance, mean_gamma_deviance,
accuracy_score
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import LSTM
from tensorflow.keras.models import load_model
#library untuk visualisasi hasil
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
from itertools import cycle
import plotly.graph_objects as go
import plotly.express as px
from plotly.subplots import make_subplots
import matplotlib.backends.backend_pdf
df=pd.read_csv('bitcoin.csv')
# melihat data
print(df)
# Periksa nama kolom di DataFrame Anda
print(df.columns)
# melihat berapa banyak data
df.shape
# cek angka null
print('Null Values:', df.isnull().values.sum())
print('NA values:', df.isnull().values.any())
# Visualisasi angka null
plt.figure(figsize=(10, 6))
plt.bar(df.columns, df.isnull().sum(), color='red')
plt.title('Null Values in DataFrame')
plt.xlabel('Columns')
plt.ylabel('Count of Null Values')
plt.show()
# visualisasi histori harga penutupan bitcoin
plt.figure(figsize=(16, 8))
plt.title('Histori Harga Penutupan')
plt.plot(df['Close'])
plt.xlabel('Date', fontsize=18)
plt.ylabel('Harga Penutupan USD ($)', fontsize=18)
plt.show()
# Ambil Semua Harga Close Price
closedf = df[['Date', 'Close']]
print("Bentuk kerangka data :", closedf.shape)
fig = px.line(closedf, x=closedf.Date,
y=closedf.Close, labels={'date': 'Date', 'close': 'Close Stock'})
```

```

fig.update_traces(marker_line_width=2, opacity=0.8, marker_line_color='orange')
fig.update_layout(title_text='Periode yang dipertimbangkan untuk memprediksi
Close Price Bitcoin',
                  plot_bgcolor='white', font_size=15, font_color='black')
fig.update_xaxes(showgrid=False)
fig.update_yaxes(showgrid=False)
fig.show()
# deleting date column and normalizing using MinMax Scaler
del closedf['Date']
scaler=MinMaxScaler(feature_range=(0,1))
closedf=scaler.fit_transform(np.array(closedf).reshape(-1,1))
print(closedf.shape)
print("Data yang sudah di-denormalisasi:")
print(closedf)
# percobaan pembagian data latih dan uji
training_size=int(len(closedf)*0.90)
test_size=len(closedf)-training_size
train_data,test_data=closedf[0:training_size:],closedf[training_size:len(closedf),1]
print("train_data: ", train_data.shape)
print("test_data: ", test_data.shape)
# mengonversi array nilai menjadi matriks kumpulan data
def create_dataset(dataset, time_step=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0]
        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])
    return np.array(dataX), np.array(dataY)
time_step = 15
X_train, y_train = create_dataset(train_data, time_step)
X_test, y_test = create_dataset(test_data, time_step)
print("X_train: ", X_train.shape)
print("y_train: ", y_train.shape)
print("X_test: ", X_test.shape)
print("y_test: ", y_test.shape)
# membentuk kembali input menjadi [samples, time steps, features] yang
diperlukan untuk LSTM
X_train =X_train.reshape(X_train.shape[0],X_train.shape[1] , 1)
X_test = X_test.reshape(X_test.shape[0],X_test.shape[1] , 1)
print("X_train: ", X_train.shape)
print("X_test: ", X_test.shape)
model=Sequential()
model.add(LSTM(50,input_shape=(None,1),activation="relu"))
model.add(Dense(1))
model.compile(loss="mean_squared_error",optimizer="adam")
history =
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=400,batch_size=
32,verbose=1)
model.save('Model Baru.keras')
import matplotlib.pyplot as plt
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(len(loss))
plt.plot(epochs, loss, 'r', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title("Training and validation loss")
plt.legend(loc=0)

```

```

plt.figure()
### prediksi dan periksa kinerja metrik
train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
train_predict.shape, test_predict.shape
# Ubah kembali ke bentuk aslinya
train_predict = scaler.inverse_transform(train_predict)
test_predict = scaler.inverse_transform(test_predict)
original_ytrain = scaler.inverse_transform(y_train.reshape(-1,1))
original_ytest = scaler.inverse_transform(y_test.reshape(-1,1))
# Hitung metrik untuk data latih
train_rmse = math.sqrt(mean_squared_error(original_ytrain, train_predict))
train_mse = mean_squared_error(original_ytrain, train_predict)
train_mae = mean_absolute_error(original_ytrain, train_predict)
# Hitung MAPE untuk data latih
train_mape = np.mean(np.abs((original_ytrain - train_predict) / original_ytrain)) *
100
# Tampilkan metrik untuk data latih
print("Train data RMSE:", train_rmse)
print("Train data MSE:", train_mse)
print("Train data MAE:", train_mae)
print("Train data MAPE:", train_mape, "%")
# Menghitung akurasi MAPE untuk data latih
accuracy_mape_train = 100 - train_mape
print("Accuracy MAPE (Training):", accuracy_mape_train, "%")
# Hitung metrik untuk data uji
test_rmse = math.sqrt(mean_squared_error(original_ytest, test_predict))
test_mse = mean_squared_error(original_ytest, test_predict)
test_mae = mean_absolute_error(original_ytest, test_predict)
# Hitung MAPE untuk data uji
test_mape = np.mean(np.abs((original_ytest - test_predict) / original_ytest)) * 100
# Tampilkan metrik untuk data uji
print("Test data RMSE:", test_rmse)
print("Test data MSE:", test_mse)
print("Test data MAE:", test_mae)
print("Test data MAPE:", test_mape, "%")
# Menghitung akurasi MAPE untuk data uji
accuracy_mape_test = 100 - test_mape
print("Accuracy MAPE (Testing):", accuracy_mape_test, "%")
# Hitung MAPE keseluruhan
overall_mape = (train_mape + test_mape) / 2
# Tampilkan metrik keseluruhan
print("Nilai Keseluruhan RMSE:", math.sqrt((train_rmse**2 + test_rmse**2) / 2))
print("Nilai Keseluruhan MSE:", (train_mse + test_mse) / 2)
print("Nilai Keseluruhan MAE:", (train_mae + test_mae) / 2)
print("Nilai Keseluruhan MAPE:", overall_mape, "%")
# Menghitung akurasi MAPE keseluruhan
accuracy_mape_overall = 100 - overall_mape
print("Accuracy MAPE (Overall):", accuracy_mape_overall, "%")
weights_lstm = model.layers[0].get_weights()
# Bobot LSTM
W_i, W_f, W_c, W_o = weights_lstm[0][:, :4], weights_lstm[0][:, 4:8],
weights_lstm[0][:, 8:12], weights_lstm[0][:, 12:]
U_i, U_f, U_c, U_o = weights_lstm[1][:, :4], weights_lstm[1][:, 4:8],
weights_lstm[1][:, 8:12], weights_lstm[1][:, 12:]
b_i, b_f, b_c, b_o = weights_lstm[2][:4], weights_lstm[2][4:8], weights_lstm[2][8:12],
weights_lstm[2][12:]

```

```

# Menampilkan bobot LSTM
gate_labels = ['Input', 'Forget', 'Cell', 'Output']
weight_labels = ['W_i', 'W_f', 'W_c', 'W_o']
bias_labels = ['b_i', 'b_f', 'b_c', 'b_o']
for gate, W, U, b in zip(gate_labels, [W_i, W_f, W_c, W_o], [U_i, U_f, U_c, U_o], [b_i,
b_f, b_c, b_o]):
    print(f'{gate} Gate Weights:')
    print(f'{weight_labels[0]}: {W}')
    print(f'{weight_labels[1]}: {U}')
    print(f'{weight_labels[2]}: {b}')
    # Menampilkan bobot LSTM
    print(f'{gate} Gate Bias:')
    print(f'{bias_labels[0]}: {W}')
    print(f'{bias_labels[1]}: {U}')
    print(f'{bias_labels[2]}: {b}')
    print("\n")
# Hitung rata-rata akurasi untuk data latih
train_accuracy_percentage = (1 - (train_mape / 100)) * 100
# Hitung rata-rata akurasi untuk data uji
test_accuracy_percentage = (1 - (test_mape / 100)) * 100
# Hitung rata-rata akurasi keseluruhan
overall_accuracy_percentage = ((train_accuracy_percentage +
test_accuracy_percentage) / 2)
# Tampilkan metrik akurasi untuk data latih
print("Akurasi Data Latih: {:.2f}%".format(train_accuracy_percentage))
# Tampilkan metrik akurasi untuk data uji
print("Akurasi Data Testing: {:.2f}%".format(test_accuracy_percentage))
# Tampilkan metrik akurasi keseluruhan
print("Akurasi Keseluruhan: {:.2f}%".format(overall_accuracy_percentage))
# shift train predictions for plotting
look_back = time_step
trainPredictPlot = np.empty_like(closedf)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(train_predict) + look_back, :] = train_predict
# shift test predictions for plotting
testPredictPlot = np.empty_like(closedf)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(train_predict) + (look_back * 2) + 1:len(closedf) - 1, :] =
test_predict
# Create DataFrame for plotting
plotdf = pd.DataFrame({
    'date': df['Date'].values,
    'original_close': df['Close'].values,
    'train_predicted_close': trainPredictPlot.reshape(1, -1)[0][:len(df)],
    'test_predicted_close': testPredictPlot.reshape(1, -1)[0][:len(df)]
})
# Plot the data
fig = px.line(plotdf, x='date', y=['original_close', 'train_predicted_close',
'test_predicted_close'],
              labels={'value': 'Stock price', 'date': 'Date'})
fig.update_layout(title_text='Perbandingan antara harga penutupan asli vs harga
penutupan yang diprediksi',
                  plot_bgcolor='white', font_size=15, font_color='black',
                  legend_title_text='Close Price')
fig.show()

# Display the results in a DataFrame

```

```

result_df = pd.DataFrame({
    'Date': df['Date'][time_step + 1:time_step + 1 + len(train_predict) +
len(test_predict)],
    'Original_Close': np.concatenate([original_ytrain.flatten(),
original_ytest.flatten()]),
    'Predicted_Close': np.concatenate([train_predict.flatten(), test_predict.flatten()])
})
# Save DataFrame to Excel
result_df.to_excel('result_dataframe.xlsx', index=False)
# Print the DataFrame
print(result_df)

```

## Lampiran 2

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas_datareader as data
from keras.models import load_model
from requests.exceptions import HTTPError
import streamlit as st
import plotly.graph_objects as go
import datetime as dt
import yfinance as yf
import pandas_ta as ta
from plotly.subplots import make_subplots
from datetime import timedelta
from datetime import date
start = '2020-01-01'
end = date.today()
st.title('Prediksi Harga Bitcoin')
user_input = st.text_input('Masukan Nama Coin', 'BTC-USD')
# Mendapatkan informasi saham menggunakan yfinance
stock_info = yf.Ticker(user_input).info
company_name = stock_info['shortName']
st.subheader(company_name)
market_price = stock_info.get('regularMarketPrice', 'Data tidak tersedia')
previous_close_price = stock_info['regularMarketPreviousClose']
st.write(f'Harga Pasar: {market_price}')
st.write('Harga penutupan sebelumnya : ', previous_close_price)
#try
try:
    ticker = yf.Ticker(user_input)
    info = ticker.info
except HTTPError as http_err:
    print(f'HTTP error occurred: {http_err}')
except Exception as err:
    print(f'Other error occurred: {err}')
# Mendapatkan data historis menggunakan yfinance
df = yf.download(user_input, start, end)
# describing data
st.subheader('Data dari 2020-2023')
#df= df.reset_index()
st.write(df.tail(10))
st.write(df.describe())
# Force lowercase (optional)

```



```

        showlegend=False ), row=1, col=1)
    # Fast Signal (%k)
    fig2.append_trace(go.Scatter(x=df.index,
                                y=df['%k'],
line=dict(color='#ff9900', width=2), name='macd',
                                # showlegend=False,
                                legendgroup='2'), row=2, col=1)
    # Slow signal (%d)
    fig2.append_trace(go.Scatter(x=df.index,
                                y=df['%d'],
line=dict(color='#000000', width=2),
                                # showlegend=False,
                                legendgroup='2', name='signal'), row=2, col=1)
    # warna histogram
    colors = np.where(df['macd'] < 0, '#000', '#ff9900')
    # Plot histogram
    fig2.append_trace(go.Bar(x=df.index, y=df['macd'], name='histogram',
marker_color=colors, ), row=2, col=1)
    # layout
    layout = go.Layout(autosize=False,
width=1000,
height=1000, plot_bgcolor='#efefef',
    # font
    font_family='Monospace',font_color='#000000', font_size=20,
    xaxis=dict(
        rangelslider=dict(visible=True) ))
    # Opsi pembaruan dan tampilkan plot
    fig2.update_layout(layout)
    st.plotly_chart(fig2)
elif infoType == 'RSI & CCI':
    st.subheader('Relative Strength Index (RSI) & Comodity Channel Index
(CCI)')
    df["RSI(2)"] = ta.rsi(df['close'], length= 2)
    df["RSI(7)"] = ta.rsi(df['close'], length= 7)
    df["RSI(14)"] = ta.rsi(df['close'], length= 14)
    df["CCI(30)"] = ta.cci(close=df['close'],length=30, high= df["high"], low =
df["low"])
    df["CCI(50)"] = ta.cci(close= df['close'],length= 50, high= df["high"], low =
df["low"])
    df["CCI(100)"] = ta.cci(close= df['close'],length= 100, high= df["high"], low =
df["low"])
    fig3=plt.figure(figsize=(15,15))
    ax1 = plt.subplot2grid((10,1), (0,0), rowspan = 4, colspan = 1)
    ax2 = plt.subplot2grid((10,1), (5,0), rowspan = 4, colspan = 1)
    ax1.plot( df['close'], linewidth = 2.5)
    ax1.set_title('CLOSE PRICE')
    ax2.plot(df['RSI(14)'], color = 'orange', linewidth = 2.5)
    ax2.axhline(30, linestyle = '--', linewidth = 1.5, color = 'grey')
    ax2.axhline(70, linestyle = '--', linewidth = 1.5, color = 'grey')
    ax2.set_title('RELATIVE STRENGTH INDEX')
    st.pyplot(fig3)
    fig4= plt.figure(figsize=(15,15))
    ax1 = plt.subplot2grid((10,1), (0,0), rowspan = 4, colspan = 1)
    ax2 = plt.subplot2grid((10,1), (5,0), rowspan = 4, colspan = 1)
    ax1.plot(df['close'], linewidth = 2.5)
    ax1.set_title('CLOSE PRICE')
    ax2.plot(df['CCI(30)'], color = 'orange', linewidth = 2.5)
    ax2.axhline(-100, linestyle = '--', linewidth = 1.5, color = 'grey')
    ax2.axhline(100, linestyle = '--', linewidth = 1.5, color = 'grey')

```

```

        ax2.set_title('COMMODITY CHANNEL INDEX')
        st.pyplot(fig4)
elif infoType == 'Williams %R':
    st.subheader('Williams %R')
    def get_wr(high, low, close, lookback):
        highh = high.rolling(lookback).max()
        lowl = low.rolling(lookback).min()
        wr = -100 * ((highh - close) / (highh - lowl))
        return wr
    df['wr_14'] = get_wr(df['high'], df['low'], df['close'], 14)
    fig5 = plt.figure(figsize=(15,12))
    ax1 = plt.subplot2grid((1,1), (0,0), rowspan = 5, colspan = 1)
    ax2 = plt.subplot2grid((1,1), (6,0), rowspan = 5, colspan = 1)
    ax1.plot(df['close'], linewidth = 2)
    ax1.set_title('HARGA PENUTUPAN')
    ax2.plot(df['wr_14'], color = 'orange', linewidth = 2)
    ax2.axhline(-20, linewidth = 1.5, linestyle = '--', color = 'grey')
    ax2.axhline(-50, linewidth = 1.5, linestyle = '--', color = 'green')
    ax2.axhline(-80, linewidth = 1.5, linestyle = '--', color = 'grey')
    ax2.set_title('WILLIAMS %R 14')
    st.pyplot(fig5)
else:
    start = dt.datetime.today() - dt.timedelta(2 * 365)
    end = dt.datetime.today()
    #df = yf.download(user_input, start, end)
    df = df.reset_index()
    fig = go.Figure(
        data=go.Scatter(x=df.index, y=df['adj close'])
    )
    fig.update_layout(
        title={
            'text': "Harga Selama Dua Tahun Terakhir",
            'y': 0.9,
            'x': 0.5,
            'xanchor': 'center',
            'yanchor': 'top'})
    st.plotly_chart(fig, use_container_width=True)
st.subheader("")
# Membagi tanggal menjadi pelatihan dan pengujian
data_training= pd.DataFrame(df['close'][0:int(len(df)*0.90)])
data_testing = pd.DataFrame(df['close'][int(len(df)*0.90): int(len(df))])
print("Data Latih: ",data_training.shape)
print("Data Uji: ", data_testing.shape)
# Penskalaan beban data menggunakan penskalaan maksimum min (0,1)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
data_training_array = scaler.fit_transform(data_training)
#Memuat model
model = load_model("model_baru.h5")
#Tes
past_100_days = data_training.tail(100)
final_df = pd.concat([past_100_days, data_testing], ignore_index=True)
input_data = scaler.fit_transform(final_df)
x_test = []
y_test = []
for i in range (100, input_data.shape[0]):
    x_test.append(input_data[i-100 : i])

```

```

    y_test.append(input_data[i, 0])
x_test, y_test = np.array(x_test), np.array(y_test)
y_predicted = model.predict(x_test)
scaler = scaler.scale_
scale_factor = 1/scaler[0]
y_predicted = y_predicted * scale_factor
y_test = y_test* scale_factor
# Hasil Akhir
st.subheader("Prediksi vs Original")
fig2= plt.figure(figsize = (12,6))
plt.plot(y_test, 'b', label = 'Harga Asli')
plt.plot(y_predicted, 'r', label = 'Harga Prediksi')
plt.xlabel('Waktu')
plt.ylabel('Harga')
plt.legend()
st.pyplot(fig2)
st.subheader('Prediksi Harga Berdasarkan Tanggal')
df1=df.reset_index()['close']
scaler=MinMaxScaler(feature_range=(0,1))
df1=scaler.fit_transform(np.array(df1).reshape(-1,1))
#datemax="24/06/2022"
datemax=dt.datetime.strptime(dt.datetime.now() - timedelta(1), "%d/%m/%Y")
datemax =dt.datetime.strptime(datemax,"%d/%m/%Y")
x_input=df1[:].reshape(1,-1)
temp_input=list(x_input)
temp_input=temp_input[0].tolist()
date1 = st.date_input("Masukan Tanggal Dalam Format ini yyyy-mm-dd")
result = st.button("Prediksi")
#st.write(result)
if result:
    from datetime import datetime
    my_time = datetime.min.time()
    date1 = datetime.combine(date1, my_time)
    #date1=str(date1)
    #date1=dt.datetime.strptime(time_str,"%Y-%m-%d")
    nDay=date1-datemax
    nDay=nDay.days
    date_rng = pd.date_range(start=datemax, end=date1, freq='D')
    date_rng=date_rng[1:date_rng.size]
    lst_output=[]
    n_steps=x_input.shape[1]
    i=0
    while(i<=nDay):
        if(len(temp_input)>n_steps):
            #print(temp_input)
            x_input=np.array(temp_input[1:])
            print("{} day input {}".format(i,x_input))
            x_input=x_input.reshape(1,-1)
            x_input = x_input.reshape((1, n_steps, 1))
            #print(x_input)
            yhat = model.predict(x_input, verbose=0)
            print("{} day output {}".format(i,yhat))
            temp_input.extend(yhat[0].tolist())
            temp_input=temp_input[1:]
            #print(temp_input)
            lst_output.extend(yhat.tolist())
            i=i+1

```

```

else:
    x_input = x_input.reshape((1, n_steps, 1))
    yhat = model.predict(x_input, verbose=0)
    print(yhat[0])
    temp_input.extend(yhat[0].tolist())
    print(len(temp_input))
    lst_output.extend(yhat.tolist())
    i=i+1
res = scaler.inverse_transform(lst_output)
#output = res[nDay-1]
output = res[nDay]
st.write("*Prediksi Harga Pertanggal :*", date1, "*Adalah*",
np.round(output[0], 2))
st.success('Harga {}'.format(np.round(output[0], 2)))
#st.write("Prediksi Harga : ",output)
predictions=res[res.size-nDay:res.size]
print(predictions.shape)
predictions=predictions.ravel()
print(type(predictions))
print(date_rng)
print(predictions)
print(date_rng.shape)
@st.cache
def convert_df(df):
    return df.to_csv().encode('utf-8')
df = pd.DataFrame(data = date_rng)
df['Prediksi'] = predictions.tolist()
df.columns = ['Tanggal', 'Harga']
st.write(df)
csv = convert_df(df)
st.download_button(
    "Download Data",
    csv,
    "file.csv",
    "text/csv",
    key='download-csv'
)
#visualisasi
fig =plt.figure(figsize=(10,6))
xpoints = date_rng
ypoints =predictions
plt.xticks(rotation = 90)
plt.plot(xpoints, ypoints)
st.pyplot(fig)

```