

LAPORAN PENELITIAN

**PENERAPAN JARINGAN LONG SHORT TERM MEMORY
(LSTM) DALAM MEMPREDIKSI HARGA CABAI
(STUDI KASUS HARGA CABAI DI PROVINSI JAWA TIMUR)**



Tim Peneliti:

Nurochman, S.Kom., M.Kom.

Fata Nabil Fikri

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2023

DAFTAR ISI

DAFTAR ISI.....	ii
DAFTAR TABEL.....	iv
DAFTAR GAMBAR.....	v
DAFTAR RUMUS.....	vii
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Penelitian.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II.....	5
TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	5
2.1 Tinjauan Pustaka.....	5
2.2 Landasan Teori.....	11
BAB III.....	30
METODE PENELITIAN.....	30
3.1 Metode Penelitian.....	30
3.2 Metode Pengumpulan Data Bahan.....	31
3.3 Alat dan Bahan.....	33
3.4 Langkah-langkah Penelitian.....	34
3.5 Diagram Alir Penelitian.....	45
BAB IV.....	47
HASIL DAN PEMBAHASAN.....	47
4.1 Hasil dan Pembahasan.....	47
4.2 Analisis Skenario Terbaik.....	56
BAB V.....	74
PENUTUP.....	74
5.1 Kesimpulan.....	74
5.2 Saran.....	75
DAFTAR PUSTAKA.....	76

DAFTAR TABEL

Tabel 2.1 Ringkasan Tinjauan Pustaka	9
Tabel 4.1 Hasil Eksperimen Skenario 1 (Jumlah Unit).....	47
Tabel 4.2 Hasil Eksperimen Skenario 2 (Jumlah Lapisan Tersembunyi).....	49
Tabel 4.3 Hasil Eksperimen Skenario 3 (Fungsi Aktivasi).....	50
Tabel 4.4 Hasil Eksperimen Skenario 4 (Rasio Data Latih)	51
Tabel 4.5 Hasil Eksperimen Skenario 5 (Panjang Sekuens)	53
Tabel 4.6 Hasil Eksperimen Skenario 6 (Epoch)	54
Tabel 4.7 Hasil Eksperimen Skenario 7 (Optimasi Model)	55
Tabel 4.8 Skenario Pengujian Parameter Terbaik per Skenario	67
Tabel 4.9 Hasil Pengujian Kombinasi Parameter Terbaik per Skenario pada Kedua Jenis Cabai.....	68
Tabel 4.10 Hasil Pengujian Ulang Skenario 6 Epoch 50 (Skenario Terbaik).....	70
Tabel 4.11 Hasil Pengujian Model yang Sudah Dilatih pada Dataset Lain	72

DAFTAR GAMBAR

Gambar 2.1 Struktur Sel Saraf Otak	12
Gambar 2.2 Model McCulloch dan Pitts.....	13
Gambar 2.3 Model Hebb.....	13
Gambar 2.4 Model MADALINE.....	14
Gambar 2.5 Struktur LSTM (Sumber:Qiu, Wang, & Zhou 2020).....	17
Gambar 2.6 Grafik Fungsi Binary Step	20
Gambar 2.7 Grafik Fungsi Aktivasi Linear.....	21
Gambar 2.8 Grafik Fungsi Aktivasi Sigmoid	22
Gambar 2.9 Grafik Fungsi Aktivasi Tanh.....	23
Gambar 2.10 Grafik Fungsi Aktivasi Relu	23
Gambar 2.11 Grafik Fungsi Aktivasi Leaky Relu	24
Gambar 2.12 Grafik Fungsi Aktivasi SELU.....	25
Gambar 3.1 Beranda PIHPS Nasional	32
Gambar 3.2 Grafik Perubahan Harga.....	32
Gambar 3.3 Data Harga Cabai dari Tahun 2020 hingga Tahun 2023.....	33
Gambar 3.4 Transpos Data Harga Cabai.....	35
Gambar 3.5 Hasil Normalisasi Data Harga Cabai Merah.....	36
Gambar 3.6 Hasil Pembentukan Data menjadi Bentuk Sekuens	37
Gambar 3.7 Contoh Data Label dari Sekuens Data Cabai Merah.....	38
Gambar 3.8 Reshaping Data menjadi 3 Dimensi.....	38
Gambar 3.9 Data Hasil Eksperimen dengan Ekstensi txt	44
Gambar 3.10 Data Hasil Eksperimen dengan Bentuk Tabular	45
Gambar 3.11 Diagram Alir Penelitian	46
Gambar 4.1 Grafik Hasil Pengujian 70 Unit pada Data Cabai Merah.....	58
Gambar 4.2 Grafik Hasil Pengujian 80 Unit pada Data Cabai Rawit.....	58
Gambar 4.3 Grafik Hasil Pengujian 0 Lapisan Tersembunyi pada Data Cabai Merah	59
Gambar 4.4 Grafik Hasil Pengujian 0 Lapisan Tersembunyi pada Data Cabai Rawit	60

Gambar 4.5 Anomali Hasil Pengujian Fungsi Aktivasi Relu pada Data Cabai Merah	61
Gambar 4.6 Anomali Hasil Pengujian Fungsi Aktivasi Relu pada Data Cabai Rawit	61
Gambar 4.7 Grafik Hasil Pengujian Rasio Data Latih 0.8 pada Data Cabai Merah	62
Gambar 4.8 Grafik Hasil Pengujian Rasio Data Latih 0.9 pada Data Cabai Rawit	63
Gambar 4.9 Grafik Hasil Pengujian Panjang Sekuens 7 pada Data Cabai Merah	64
Gambar 4.10 Grafik Hasil Pengujian Panjang Sekuens 7 pada Data Cabai Rawit	64
Gambar 4.11 Grafik Hasil Pengujian Epoch 50 pada Data Cabai Merah.....	65
Gambar 4.12 Grafik Hasil Pengujian Epoch 50 pada Data Cabai Rawit.....	65
Gambar 4.13 Grafik Hasil Pengujian Optimasi RMSprop pada Data Cabai Merah	66
Gambar 4.14 Grafik Hasil Pengujian Optimasi RMSprop pada Data Cabai Rawit	67
Gambar 4.15 Grafik Hasil Pengujian Kombinasi Parameter-Parameter Terbaik per Skenario pada Data Cabai Merah.....	69
Gambar 4.16 Grafik Hasil Pengujian Kombinasi Parameter-Parameter Terbaik per Skenario pada Data Cabai Rawit.....	69
Gambar 4.17 Grafik Hasil Pengujian Ulang Skenario 6 Epoch 50 (Skenario Terbaik) pada Data Cabai Merah	70
Gambar 4.18 Grafik Hasil Pengujian Ulang Skenario 6 Epoch 50 (Skenario Terbaik) pada Data Cabai Rawit	71
Gambar 4.19 Grafik Hasil Pengujian Model yang Sudah Dilatih pada Dataset Cabai Merah Lain.....	73
Gambar 4.20 Grafik Hasil Pengujian Model yang Sudah Dilatih pada Dataset Cabai Rawit Lain.....	73

DAFTAR RUMUS

(2.1) Rumus Bobot Baru dan Perbarui Bobot	14
(2.2) Rumus Bias Baru dan Perbarui Bias	14
(2.3) Rumus Normalisasi Min Max Scaling	18
(2.4) Rumus Denormalisasi Min Max Scaling.....	18
(2.5) Rumus Root Mean Square Error	18
(2.6) Rumus Mean Absolute Percentage Error	19
(2.7) Rumus Fungsi Aktivasi Binary Step	20
(2.8) Rumus Fungsi Aktivasi Linear	20
(2.9) Rumus Fungsi Aktivasi Sigmoid.....	21
(2.10) Rumus Fungsi Aktivasi Tanh	22
(2.11) Rumus Fungsi Aktivasi Relu.....	23
(2.12) Rumus Fungsi Aktivasi Leaky Relu.....	24
(2.13) Rumus Fungsi Aktivai SELU	24

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan bahan pokok di Indonesia menjadi hal yang sangat penting bagi masyarakat Indonesia. Terlebih lagi ketika membahas harga bahan pokok di Indonesia. Harga bahan pokok di Indonesia sering mengalami ketidakstabilan atau fluktuasi. Salah satu bahan pokok yang sering mengalami ketidakstabilan harga yaitu cabai. Yanwardhana dalam CNBC Indonesia menyebutkan harga rata-rata cabai di Indonesia bisa mencapai Rp. 106.764 per kilogramnya. Bahkan dilaporkan di beberapa wilayah, pedagang cabai terpantau mulai menjual cabai dengan harga 150 ribu per kilogram. (Yanwardhana, 2022)

Di Provinsi Jawa Timur harga cabai rawit merah meningkat 241,48 persen dari awalnya Rp. 24.840 per kilogram menjadi Rp. 84.823 per kilogram per 7 Juni 2022. Tidak ketinggalan cabai merah juga meningkat dari harga Rp. 34.798 per kilogram menjadi Rp. 62.144 per kilogramnya atau meningkat sekitar 78,58 persen. Gubernur Jawa Timur, Ibu Khofifah mengisyaratkan untuk segera menanam cabai varietas Bhaskoro dan Dewata.

Beberapa faktor yang dapat menyebabkan ketidakstabilan harga tersebut antara lain seperti cuaca, harga BBM dan juga hari-hari besar yang dapat mempengaruhi harga bahan pokok tersebut. Seperti kabar harga cabai yang mencapai 150 ribu per kilogram, Direktur Jenderal Hortikultura Kementerian Pertanian Prihasto Setyanto menjelaskan bahwa harga cabai yang sering berubah dipengaruhi oleh cuaca, dimana curah hujan tinggi membuat tanaman cabai terserang hama sehingga produksi cabai pun terganggu dan pasokan cabai juga menurun. Selain itu hari-hari besar juga mempengaruhi harga bahan pokok terutama cabai. Seperti yang disebutkan CNN Indonesia, harga cabai menjelang Ramadhan menjadi Rp. 60.000 yang

sebelumnya di kisaran Rp. 55.000 per kilogram. (*Harga Cabai Rawit Di Pasar Wilayah Jatim Naik Jelang Ramadan, 2022*)

Ketidakstabilan harga cabai ini sangat berdampak pada usaha-usaha kuliner terutama usaha kuliner yang membutuhkan cabai. Seperti yang dilansir dari tulisan Sutrisna pada Kompas.com (03/03/2021), seorang pemilik usaha rumah makan memilih untuk mengurangi porsi sambal yang disediakan. Awalnya bisa menghabiskan 2 kilogram menjadi hanya 1 kilogram cabai saja. (Sutrisna, 2021)

Untuk itu, model komputasi tertentu dapat digunakan untuk memprediksi harga bahan pokok tersebut. Model komputasi yang dapat digunakan yaitu Long-Short Term Memory (LSTM). LSTM sendiri merupakan bentuk pengembangan dari jaringan syaraf tiruan Recurrent Neural Network (RNN) yang mana LSTM sendiri melengkapi dan memperbaiki kekurangan dari RNN. (*Apa Itu Long Short Term Memory (LSTM)?*, 2022) Pada awalnya penerapan LSTM banyak diterapkan pada kasus *forecasting* harga saham (*stock forecasting*), cuaca dan lain sebagainya seperti pada penelitian yang dilakukan oleh Chairurrachman I. pada tahun 2022 yang melakukan penelitian tentang *forecasting* saham dari PT Indofood dengan hasil yang cukup memuaskan. (Chairurrachman, 2022)

Untuk itu, dalam penelitian ini akan dibangun arsitektur LSTM dan konfigurasi hyperparameter yang efisien untuk prediksi harga bahan pangan terutama komoditas cabai di Provinsi Jawa Timur.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, penulis dapat merumuskan masalah sebagai berikut.

Bagaimana arsitektur dari model komputasi LSTM serta konfigurasi *hyperparameter* persiapan data dan pelatihan yang efektif untuk prediksi harga cabai di Provinsi Jawa Timur.

1.3 Batasan Penelitian

Agar penelitian lebih terfokus dan tidak menyimpang, penulis merumuskan beberapa batasan masalah sebagai berikut.

1. Data yang digunakan yaitu data harga cabai di Provinsi Jawa Timur
2. Data yang digunakan yaitu data harga cabai dengan rentang waktu kurang lebih 3 (tiga) tahun terakhir, terhitung sejak 1 Januari 2020 sampai 1 Juni 2023.
3. Komoditas cabai yang digunakan sebagai data yaitu Cabai Merah dan Cabai Rawit.
4. Jenis model komputasi yaitu Long Short Term Memory (LSTM)
5. *Hyperparameter* yang diuji dibagi menjadi 2 jenis yaitu *hyperparameter* arsitektur dan *hyperparameter* yang berkaitan dengan data dan pelatihan. *Hyperparameter* arsitektur meliputi jumlah unit, jumlah lapisan tersembunyi, dan fungsi aktivasi. *Hyperparameter* yang berkaitan data dan pelatihan meliputi rasio data latih, panjang sekuens, jumlah epoch, dan optimasi model.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, tujuan dari penelitian ini sebagai berikut.

Membangun arsitektur model komputasi LSTM serta menentukan konfigurasi *hyperparameter* persiapan data dan pelatihan yang efektif untuk prediksi harga cabai di Provinsi Jawa Timur.

1.5 Manfaat Penelitian

Adapun manfaat yang sekiranya dapat diperoleh dari penelitian ini adalah sebagai berikut.

1. Bagi pemerintah setempat, diharapkan hasil penelitian ini dapat digunakan untuk merancang model komputasi yang efisien dan optimal untuk kasus prediksi harga bahan pokok terutama cabai di Provinsi Jawa Timur.

2. Bagi pengembang sistem, diharapkan hasil penelitian ini dapat membantu dalam menentukan arsitektur serta konfigurasi *hyperparameter* dari model komputasi LSTM yang efektif untuk prediksi harga cabai di Provinsi Jawa Timur, sehingga dapat diterapkan pada sistem informasi dan dapat dimanfaatkan oleh masyarakat.
3. Bagi peneliti, peneliti mengetahui arsitektur model komputasi LSTM dan konfigurasi *hyperparameter* yang efektif untuk kasus prediksi harga cabai di Provinsi Jawa Timur.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Pada penelitian tahun 2022 yang dilakukan oleh Chairurrachman I yang berjudul “Penerapan Long Short-Term Memory pada Data Time Series untuk Prediksi Harga Saham Pt. Indofood Cbp Sukses Makmur Tbk (ICBP)” meneliti beberapa arsitektur dari LSTM serta menguji beberapa *hyperparameter* pada LSTM tersebut. Terdapat tiga arsitektur LSTM yang diuji yaitu *vanilla* LSTM, CNN-LSTM, dan *bidirectional* LSTM. Dengan *hyperparameter* yang diuji yaitu jumlah neuron *hidden* dan *learning rate* dengan jumlah epoch konstan yaitu di 50 epoch dan beberapa kombinasi data latih dengan data uji. Pada penelitian ini MAE dan R^2 menjadi metrik penilaian hasil pengujian model komputasi. Hasil terbaik didapatkan oleh arsitektur CNN-LSTM dengan nilai MAE sebesar 74.1365 dan nilai R^2 sebesar 96,6% (Chairurrachman, 2022).

Penelitian yang dilakukan Hardjita P W dan Hidayat R pada tahun 2021 meneliti tentang analisis sentimen dari *tweet* dengan topik Kartu Prakerja menggunakan CNN dan Naïve Bayes. Terdapat tiga jenis arsitektur CNN yang digunakan yaitu CNN-*Bidirectional* LSTM, CNN *GlobalMaxPooling* dan CNN-LSTM. Ketiga arsitektur tersebut dibandingkan dengan *Naïve Bayes Classifier*. *Preprocessing* data juga dilakukan pada data *tweet* meliputi *cleansing*, *stemming*, *tokenizing*, dan *filtering* serta data tersebut dilabeli sebelum di-input pada model komputasi. Hasil dari penelitian tersebut yaitu model *Naïve Bayes Classifier* dengan nilai akurasi 0.762, CNN-*Bidirectional* LSTM dengan nilai akurasi 0.776, CNN *GlobalMaxPooling* dengan nilai akurasi 0.785, dan CNN-LSTM dengan nilai akurasi 0.78. Berdasarkan hasil tersebut, CNN *GlobalMaxPooling* menghasilkan nilai akurasi terbaik (tertinggi) dengan nilai akurasi 0.785 (Hardjita & Hidayat, 2021).

Pada penelitian tahun 2020 yang dilakukan oleh Arfan A dan Lussiana E.T.P. yaitu membandingkan performa antara *Long Short Term Memory*

(LSTM) dengan *Support Vector Regression* (SVR) dengan objek dari perbandingan tersebut adalah data harga saham dari perusahaan PT. Unilever Tbk, PT. Kimia Farma, dan PT. Gudang Garam Tbk. Pada penelitian tersebut dilakukan variasi skenario seperti jumlah lapisan tersembunyi, jumlah memori sel, jumlah sampel data dan jumlah putaran untuk mencari nilai *loss* yang paling kecil. Kemudian dilakukan perbandingan antara LSTM dengan SVR pada 3 rentang waktu yaitu 1 hari, 20 hari dan 60 hari. Hasilnya LSTM memiliki performa lebih baik dibandingkan SVR pada uji rentang data 1 hari dengan nilai MSE 0,0013 selisih 0,0012 dengan SVR dengan nilai MSE 0,0025. Kemudian pada uji rentang data 30 hari LSTM memiliki MSE sebesar 0,0016 sedangkan SVR memiliki nilai MSE sebesar 0,0045. Untuk uji rentang data 60 hari LSTM menghasilkan nilai MSE sebesar 0,0013 sedangkan SVR menghasilkan nilai MSE 0,0065 (Arfan & Lussiana, 2020). Kesimpulan yang dapat diambil bahwa SVR memiliki performa yang lebih rendah jika berhadapan dengan rentang data waktu yang panjang jika dibandingkan dengan LSTM dikarenakan LSTM memiliki memori sehingga data-data sebelumnya memiliki pengaruh pada data selanjutnya.

Pada penelitian yang berjudul “Analisis Prediksi Harga Saham Sektor Perbankan Menggunakan Algoritma *Long-Short Terms Memory* (Lstm)” yang dilakukan oleh Riyantoko, Fahrudin, Hindrayani, dkk pada tahun 2020, membandingkan performa LSTM dengan dua macam skenario yaitu jenis optimasi model yang digunakan dan jumlah epoch saat fase pelatihan dengan objek dari penelitian tersebut yaitu data harga saham dari empat perusahaan perbankan yaitu saham Bank BRI, Bank BNI, Bank BTN dan Bank Mandiri. Pada penelitian tersebut peneliti menggunakan tiga jenis optimasi model komputasi yaitu SGD, Adam dan RMSprop dengan hasil yaitu model optimasi Adam dengan tingkat akurasi disekitar 89% hingga 95% untuk keempat dari bank tersebut. Sedangkan model optimasi RMSprop dengan tingkat akurasi direntang 46% untuk nilai akurasi terendah dan 95% untuk nilai tertingginya. Dan model optimasi SGD memiliki tingkat akurasi di rentang 49% hingga 61%. Kemudian perbandingan jumlah epoch pada saat

fase pelatihan pada penelitian ini membandingkan antara jumlah epoch 25, 50, 75, dan 100. Didapatkan hasil bahwa jumlah epoch mempengaruhi durasi dari komputasi dimana nilai epoch rendah memiliki durasi komputasi yang cepat dan nilai epoch tinggi memiliki durasi komputasi yang cukup lama. Sedangkan untuk nilai RMSE yang dihasilkan dari pengujian tersebut memiliki nilai yang bervariasi dan tidak bergantung pada jumlah epoch, ada yang semakin banyak jumlah epoch semakin kecil RMSE-nya dan ada juga yang semakin banyak jumlah epoch semakin RMSE-nya (Riyantoko et al., 2020). Kesimpulannya model optimasi sangat mempengaruhi pada tingkat *loss* dan akurasi yang sangat bervariasi tergantung pada jenis model optimasinya, sedangkan jumlah epoch akan mempengaruhi pada aspek durasi komputasi pelatihan dan nilai RMSE, untuk durasi komputasi pelatihan cenderung berkaitan dengan jumlah epoch, berbeda dengan nilai RMSE yang lebih bervariasi dan tidak menentu.

Pada penelitian yang dilakukan oleh Syaidah K, Chrisnanto Y, Abdillah G pada tahun 2020 dengan topik “Prediksi Harga Sembako di DKI Jakarta Menggunakan Artificial Neural Network” menggunakan model komputasi *Backpropagation* dengan objek penelitian tersebut adalah harga sembako di provinsi DKI Jakarta dengan tiga komoditas yang dipilih untuk diteliti yaitu beras, cabai dan bawang. Peneliti melakukan beberapa skenario pelatihan seperti mengubah nilai *alpha* (nilai *learning rate*) dan nilai minimum error (nilai *threshold*) dengan metrik penilaian yaitu nilai MSE. Didapatkan hasil pada komoditas beras menghasilkan nilai akurasi 60% hingga 91% pada data latih dan 54% hingga 82% pada data uji, sedangkan pada komoditas cabai dihasilkan nilai akurasi 70% hingga 90% pada data latih dan 63% hingga 80% pada data uji, kemudian pada komoditas bawang dihasilkan nilai akurasi 76% hingga 88% pada data latih dan 68% hingga 78% pada data uji. Selain membandingkan nilai *alpha* dan nilai *threshold*, pada penelitian ini juga membandingkan jumlah *hidden layer* dengan jumlah *hidden layer* satu dan dua. Dari pengujian tersebut didapatkan hasil bahwa jumlah *hidden layer* memiliki pengaruh pada waktu komputasi dan tingkat akurasi yang mana

waktu komputasi akan semakin lama dengan *hidden layer* yang lebih banyak, begitu juga dengan tingkat akurasi yang dihasilkan akan semakin tinggi seiring bertambahnya jumlah *hidden layer* (Syaidah et al., 2020). Kesimpulan yang dapat diambil yaitu nilai *alpha* dan nilai *threshold* akan mempengaruhi pada tingkat akurasi dimana semakin rendah nilai *alpha* dan nilai *threshold* maka tingkat akurasi juga semakin tinggi untuk masing-masing komoditas serta jumlah *hidden layer* juga akan berbanding lurus dengan waktu komputasi dan tingkat akurasi yang dihasilkan, semakin banyak jumlah *hidden layer* yang digunakan maka semakin lama waktu komputasi dan semakin tinggi tingkat akurasi yang dihasilkan pada masing-masing komoditas.

Pada penelitian dengan topik “Peramalan Harga Saham Menggunakan Jaringan Saraf Tiruan Secara Supervised Learning Dengan Algoritma Backpropagation” oleh Eko Riyanto pada tahun 2017 menggunakan model komputasi *Backpropagation* dengan objek dari penelitian tersebut adalah data harga saham dari empat perusahaan antara lain Astra Graphia, Astra International, Indofood Sukses Makmur dan Telekomunikasi Indonesia. Kemudian dilakukan pengujian menggunakan tiga pola yaitu pola pertama dengan jumlah *hidden layer* sebanyak 5, jumlah epoch 10000 dan *learning rate* 0,2, untuk pola kedua yaitu *hidden layer* 5, jumlah epoch 10000 dan *learning rate* 0,5 dan pola yang ketiga dengan jumlah *hidden layer* 5, jumlah epoch 10000, dan *learning rate* 0,15 dengan demikian yang berbeda antar pola satu dengan pola yang lainnya yaitu nilai dari *learning rate* dari masing-masing pola. Melalui pengujian tersebut didapatkan hasil yaitu saham Astra Graphia memiliki tingkat akurasi minimum pada angka 96,62% dan akurasi maksimum pada 98.08%, sedangkan pada saham Astra International dihasilkan tingkat akurasi tertingginya pada angka 99,02% dengan akurasi terendahnya pada angka 98,39%. Kemudian untuk saham Indofood Sukses Makmur menghasilkan tingkat akurasi terbesar pada angka 98,38% dan nilai terendahnya di angka 96,97% dan untuk saham Telekomunikasi Indonesia

menghasilkan tingkat akurasi tertinggi di angka 99,48% dengan akurasi terendahnya di angka 97,79% (Riyanto, 2017).

Tabel 2.1 Ringkasan Tinjauan Pustaka

Peneliti (Tahun)	Topik	Metode	Hasil
Chairurrachman I (2022)	Penerapan Long Short-Term Memory pada Data Time Series untuk Prediksi Harga Saham Pt. Indofood Cbp Sukses Makmur Tbk (ICBP)	<i>Vanilla</i> LSTM, CNN-LSTM, dan <i>Bidirectional</i> LSTM	CNN-LSTM menghasilkan prediksi terbaik dengan nilai MAE sebesar 74.1365 dan nilai R ² sebesar 96,6% (Chairurrachman, 2022).
Hardjita P W, Hidayat R (2021)	<i>Sentiment Analysis of Tweets on Prakerja Card using Convolutional Neural Network and Naive Bayes.</i>	Naïve Bayes Classifier, CNN Bidirectional LSTM, CNN Global MaxPooling, dan CNN-LSTM	CNN Global MaxPooling menghasilkan nilai akurasi tertinggi yaitu 0,785 (Hardjita & Hidayat, 2021).
Arfan A, Lussiana E.T.P. (2020)	Perbandingan algoritma long short-term memory dengan SVR pada prediksi harga saham di Indonesia	Long Short Term Memory dan Support Vector Regression	LSTM menghasilkan performa yang lebih baik dengan metrik penilaian yaitu tingkat <i>loss</i> yang dihasilkan dibandingkan SVR terutama jika

			berhadapan dengan data dengan ketergantungan jangka panjang. (Arfan & Lussiana, 2020)
Riyantoko, Fahrudin, Hindrayani, dkk. (2020)	Analisis Prediksi Harga Saham Sektor Perbankan Menggunakan Algoritma Long-Short Terms Memory (Lstm)	Long Short Term Memory	Variasi epoch dan optimasi model komputasi mempengaruhi beberapa aspek meliputi waktu komputasi, nilai <i>loss</i> dan akurasi serta nilai RMSE. (Riyantoko et al., 2020)
Syaidah K, Chrisnanto Y, Abdillah G (2020)	Prediksi Harga Sembako di DKI Jakarta Menggunakan Artificial Neural Network	Backpropagation	Pelatihan model <i>Neural Network</i> menghasilkan tingkat akurasi 91% untuk data latih dan 82% untuk data uji untuk komoditas beras dan 88% untuk tingkat akurasi data latih dan 78% data uji

			pada komoditas bawang, dan tingkat akurasi 90% pada data latih dan 78% pada data uji untuk komoditas cabai. (Syaidah et al., 2020)
Eko Riyanto (2017)	Peramalan Harga Saham Menggunakan Jaringan Saraf Tiruan Secara Supervised Learning Dengan Algoritma Backpropagation	Backpropagation	Tingkat akurasi dari model <i>Neural Network</i> dengan jenis <i>Backpropagation</i> dengan metode pelatihan <i>supervised learning</i> didapat mencapai 98% jika dibandingkan nilai aktualnya. (Riyanto, 2017)

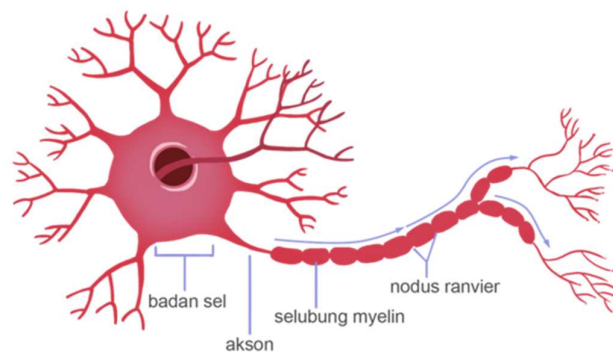
2.2 Landasan Teori

Adapun teori-teori yang menjadi landasan penelitian ini yaitu sebagai berikut.

2.2.1 Jaringan Saraf Tiruan

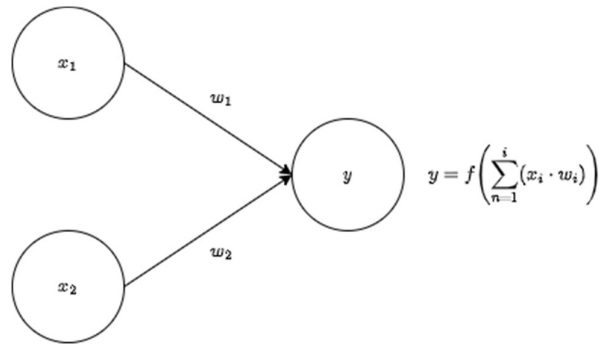
Jaringan saraf tiruan adalah bentuk model komputasi yang terinspirasi oleh sistem saraf pada otak manusia. Pada otak manusia tersusun atas milyaran sel saraf yang sering disebut *neuron*. Neuron ini saling terhubung satu dengan yang lainnya sehingga membentuk jaringan saraf pada otak. Sel-

sel saraf pada otak bekerja dengan mengirimkan impuls dari satu saraf ke saraf lainnya. *Input* dari sel saraf ini dapat berupa impuls dari sensor-sensor pada tubuh seperti sensor penglihatan, penciuman, dan lain sebagainya. Kemudian diproses di otak sehingga menghasilkan *output* yang dapat berupa motorik, ucapan, pemikiran dan lain sebagainya. Sel saraf otak manusia terdiri atas beberapa bagian seperti badan sel, akson, selubung myelin, dan dendrit seperti yang ditunjukkan pada Gambar 2.1 berikut ini.



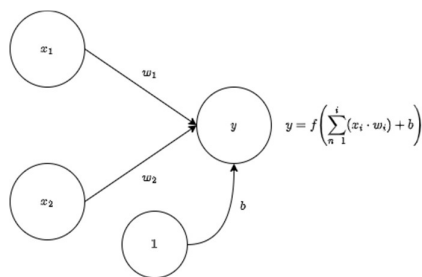
Gambar 2.1 Struktur Sel Saraf Otak

Mekanisme *neuron* tersebut menjadi inspirasi dalam pengembangan jaringan saraf tiruan (*Artificial Neural Network*). Dalam sejarahnya model jaringan saraf tiruan pertama dikenalkan oleh McCulloch dan Pitts pada tahun 1943. Pada model ini terdapat beberapa komponen antara lain yaitu *input* atau masukan yang umumnya dilambangkan dengan x , lalu ada *weight* atau bobot yang dilambangkan dengan w , kemudian ada komponen *output* atau keluaran yang dilambangkan dengan y dan *activation function* atau fungsi aktivasi yang berfungsi untuk memproses nilai *output* dari neuron. Fungsi aktivasi bekerja dengan menjumlahkan semua hasil kali antara *input* dengan *weight*. Gambar 2.2 menunjukkan diagram dari model jaringan saraf tiruan yang dikembangkan oleh McCulloch dan Pitts, sekilas gambar model berikut ini memiliki kemiripan dengan struktur sel saraf otak pada manusia yang ditunjukkan pada Gambar 2.1.



Gambar 2.2 Model McCulloch dan Pitts

Model jaringan saraf tiruan yang dikembangkan McCulloch dan Pitts menjadi dasar pengembangan jaringan saraf tiruan pada masa-masa berikutnya. Pada pengembangan selanjutnya muncul model jaringan saraf tiruan Hebb yang dikembangkan oleh Donald Olding Hebb pada tahun 1949. Pada model ini dikenalkan perubahan nilai bobot yang merupakan bentuk dari pelatihan dari jaringan saraf tiruan. Model ini tidak jauh berbeda dengan model McCulloch dan Pitts dengan penambahan bias pada komponen *output*. Fungsi dari bias sendiri yaitu sebagai parameter tambahan untuk *neuron* agar dapat menyesuaikan *output* dan membantu jaringan untuk mengkomputasi pola-pola yang lebih rumit. Berikut adalah Gambar 2.3 menunjukkan diagram dari jaringan saraf tiruan model Hebb.



Gambar 2.3 Model Hebb

Perubahan bobot pada model ini berfungsi sebagai proses pembelajaran dari model tersebut berdasarkan target dari *output* yang sudah ditetapkan. Selain perubahan bobot, bias juga dihitung perubahannya pada model ini yang ditunjukkan dengan simbol huruf 'b' pada Gambar 2.3. Perubahan bobot

ini dihitung dengan menambahkan bobot sebelumnya dengan hasil kali *input* dengan target *output*. Cara yang sama juga diterapkan pada perubahan bias. Berikut Rumus 2.1 dan Rumus 2.2 menunjukkan rumus dari perubahan dari bobot dan bias secara berurutan, secara konsep kedua rumus tersebut sama yaitu menambahkan variabel lama dengan suatu nilai baru yang didapatkan dari perkalian nilai input dengan nilai target.

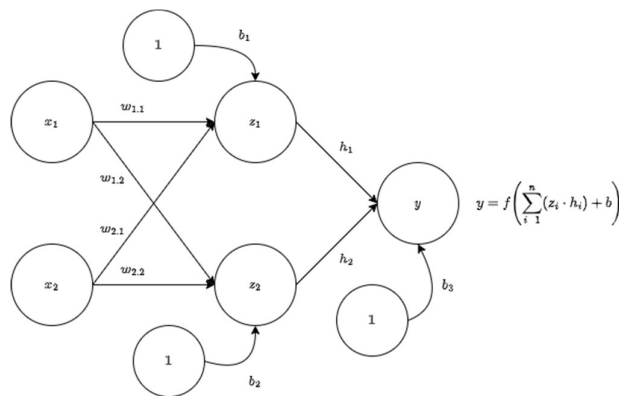
$$w_i(\text{baru}) = w_i(\text{lama}) + \Delta w_i$$

$$\Delta w_i = x_i \cdot t \quad (2.1)$$

$$b(\text{baru}) = b(\text{lama}) + \Delta b$$

$$\Delta b = 1 \cdot t \quad (2.2)$$

Pada tahun-tahun berikutnya pengembangan jaringan saraf tiruan menghasilkan jaringan saraf tiruan dengan lapisan tersembunyi atau *hidden layer*. Model ini dikembangkan oleh Widrow dan Hoff pada tahun 1960 yang diberi nama ADALINE yang dilanjutkan dengan pengembangan MADALINE yang berasal dari kata “Many ADALINE”. MADALINE menggunakan ADALINE sebagai selnya dan dirancang dengan lapisan tersembunyi antara *input* dan *output*-nya. Gambar 2.4 menunjukkan model jaringan saraf tiruan model MADALINE dengan lapisan tersembunyi.



Gambar 2.4 Model MADALINE

Pada masa-masa berikutnya perkembangan jaringan saraf tiruan sempat terhenti pada sekitar tahun 1970-an. Hingga muncullah konsep pelatihan

jaringan saraf tiruan dengan metode Backpropagation. Metode pelatihan ini membuka kesempatan pengembangan dan pengaplikasian jaringan saraf tiruan menjadi lebih maju dan luas. Metode ini pertama kali dikenalkan oleh Paul Werbos pada tahun 1974 dan dikembangkan lebih lanjut oleh David E. Rumelhart dan kawan-kawan pada awal tahun 1980-an.

Metode pelatihan ini berjalan dengan dua tahap yaitu *feedforward* (tahap umpan maju) dan *backpropagation* (tahap propagasi balik). Pada tahap *feedforward*, nilai *input* akan dihitung dengan bobot dan bias secara maju melalui lapisan *input*, lapisan tersembunyi hingga lapisan *output*. Kemudian dilanjutkan dengan tahap *backpropagation*, pada tahap ini terjadi perubahan bobot antara lapisan tersembunyi dengan lapisan *output* dengan menghitung *error* pada lapisan *output* kemudian mengalikan error dengan bobot lama dan *learning rate*.

Perhitungan *error* pada jaringan *backpropagation* berbeda dengan jaringan lainnya. Jika jaringan selain *backpropagation* menghitung error dengan menghitung selisih antara nilai sebenarnya dengan nilai hasil *output*, maka jaringan ini menghitung *error* dengan menggunakan turunan dari fungsi aktivasi yang digunakan dan dikali dengan selisih antara nilai sebenarnya dengan nilai hasil dari lapisan *output*. Nilai yang dibutuhkan untuk turunan fungsi aktivasi tersebut adalah nilai hasil perhitungan dari lapisan *output* pada tahap *feedforward*. Langkah selanjutnya memperbarui bobot antara lapisan tersembunyi dengan lapisan *output* dengan mengalikan bobot lama dengan nilai *error* dan *learning rate*. Langkah tersebut dijalankan hingga pada lapisan *input*. oleh karena itu tahap ini dinamakan *backpropagation* karena melakukan perhitungan *error* dan perubahan bobot dari lapisan *output* berjalan mundur hingga lapisan *input*.

Sejak penemuan *backpropagation* ini pengaplikasian jaringan saraf tiruan menjadi lebih luas dan dapat menjangkau bidang yang lebih kompleks, seperti pengenalan gambar, prediksi, pemrosesan bahasa alami, dan lain sebagainya. Selain itu juga, bermunculan model jaringan saraf tiruan lainnya

seperti CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), LSTM (Long Short Term Memory), dan lain sebagainya.

Fungsi aktivasi pada jaringan saraf tiruan juga tidak ketinggalan dalam hal pengembangannya. Pada awal ditemukannya jaringan saraf tiruan fungsi aktivasi masih terbatas seperti fungsi ambang batas (*Threshold*) dan fungsi sigmoid hingga fungsi tan hiperbola (Tanh). Namun, fungsi aktivasi tersebut sering menghasilkan masalah pada saat pelatihan. Masalah tersebut biasa dikenal dengan *vanishing gradient* (gradien yang menghilang) yang biasa terjadi karena model memiliki layer tersembunyi yang sangat banyak sehingga menghasilkan gradien yang mendekati nol atau bahkan nol. Fungsi aktivasi modern seperti ReLU, Leaky ReLU dapat membantu mengatasi masalah gradien yang hilang tersebut.

2.2.2 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) adalah sebuah arsitektur jaringan saraf tiruan yang sering digunakan pada data yang berbentuk *sequence* atau data yang berurutan seperti *time series* data, sebuah kalimat, video maupun audio. LSTM merupakan bentuk pengembangan Recurrent Neural Network yang ditemukan pada tahun 1986, 11 tahun kemudian LSTM baru ditemukan pada tahun 1997. Satu hal mendasar mengapa LSTM dikembangkan yaitu masalah pada RNN, dimana RNN tidak bisa mengatasi data dengan jangka waktu yang panjang dikarenakan memori pada jaringan saraf tiruan yang terbatas. Hingga LSTM ditemukan dan memiliki performa yang lebih baik dari RNN dalam hal data dengan jangka waktu yang panjang, hal ini diakibatkan LSTM memiliki memori sel yang baik dari RNN.

LSTM memiliki beberapa komponen dalam satu selnya antara lain (Saxena, 2021).

1. Forget Gate

Gerbang ini berfungsi untuk menentukan apakah informasi dari state sebelumnya perlu disimpan atau dilupakan.

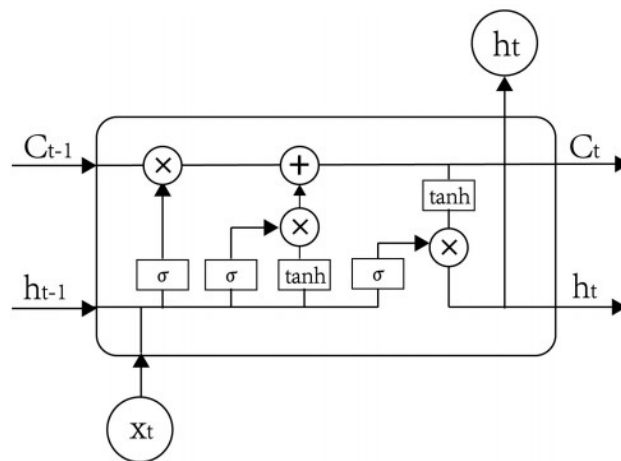
2. Input Gate

Fungsi dari *input gate* yaitu untuk mengukur informasi baru yang dibawa oleh *input* tersebut penting atau tidak. Jika informasi tersebut penting maka informasi tersebut akan ditambahkan pada state yang sedang berjalan jika tidak maka informasi tersebut akan dihilangkan dari state tersebut.

3. Output Gate

Gerbang yang terakhir adalah *output gate*. Gerbang ini berfungsi untuk menentukan nilai keluaran berdasarkan pemrosesan informasi dari *forget gate* maupun *input gate* pada state yang sedang berlangsung.

Berikut Gambar 2.5 menunjukkan struktur dari LSTM dengan beberapa gerbangnya.



Gambar 2.5 Struktur LSTM (Sumber:Qiu, Wang, & Zhou 2020)

2.2.3 MinMaxScaling

MinMaxScaling adalah salah satu proses normalisasi data agar data dapat dibaca oleh sebuah model komputasi. Proses ini bertujuan untuk merubah rentang data dari sebuah dataset menjadi data dengan rentang yang diinginkan, biasanya rentangnya antara 0 hingga 1. Cara kerja dari minmaxscaling yaitu dengan membandingkan suatu nilai dari dataset dengan nilai minimum dan maksimum dari dataset yang sama (Maity, 2021). Rumus dari minmaxscaling dapat dilihat pada Rumus 2.3 berikut.

$$y = \frac{x - x_{min}}{x_{maks} - x_{min}} \quad (2.3)$$

dimana:

y = nilai hasil penskalaan

x = nilai yang akan diskalakan

x_{min} = nilai minimum dari dataset

x_{maks} = nilai maksimum dari dataset

Selain proses normalisasi data, seringkali pada jaringan saraf tiruan membutuhkan proses denormalisasi untuk mengembalikan hasil keluaran dari jaringan saraf tiruan sesuai perbandingan antara nilai minimum dengan nilai maksimum. Rumus dari denormalisasi dari metode minmaxscaling dapat dilihat pada Rumus 2.4.

$$x = y \cdot (x_{maks} - x_{min}) + x_{min} \quad (2.4)$$

2.2.4 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) adalah salah satu cara untuk mengukur perbedaan antara nilai yang diprediksi dengan nilai sebenarnya. RMSE seringkali digunakan pada jaringan saraf tiruan sebagai metrik untuk mengevaluasi hasil dari komputasi jaringan saraf tiruan tersebut. Biasanya RMSE digunakan pada tahap setelah pelatihan jaringan saraf tiruan yakni tahap pengujian. Pada tahap pengujian tersebut akan dihasilkan sebuah nilai keluaran yang kemudian akan dievaluasi menggunakan nilai aktual sebagai data label dari sebuah dataset. Jika selisih antara nilai yang diprediksi dengan nilai aktual akan menghasilkan nilai RMSE yang kecil. Sehingga RMSE dapat digunakan untuk mengukur tingkat hasil pelatihan suatu jaringan saraf tiruan apakah memiliki kualitas yang baik atau tidak. RMSE dapat dihitung menggunakan Rumus 2.5 berikut.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - t_i)^2}{n}} \quad (2.5)$$

Keterangan:

y_i = nilai hasil prediksi ke-i

t_i = nilai aktual ke- i

n = jumlah data

Berdasarkan rumus tersebut langkah perhitungan dari RMSE adalah sebagai berikut.

- a. Hitung selisih antara nilai yang diprediksi dengan nilai aktual.
- b. Kuadratkan hasil perhitungan selisih dua nilai tersebut.
- c. Hitung rata-rata dari hasil kuadrat selisih dari dua nilai.
- d. Akar kuadratkan rata-rata kuadrat selisih dari dua nilai tersebut

2.2.5 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error adalah salah satu indikator statistika yang digunakan untuk mengetahui kualitas sebuah prediksi. Indikator ini menghitung kesalahan rata-rata dari sebuah kumpulan data dan menyajikannya dalam bentuk persentase. Maka semakin kecil nilai MAPE maka dapat dikatakan semakin bagus juga tingkat akurasi suatu hasil prediksi dengan nilai aktualnya (Vivas et al., 2020), Untuk rumus MAPE dapat dilihat pada Rumus 2.6 berikut.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{t_i - y_i}{t_i} \right| \cdot 100 \quad (2.6)$$

Keterangan:

t_i = nilai aktual pada data ke- i

y_i = nilai prediksi pada data ke- i

n = jumlah seluruh data

2.2.6 Fungsi Aktivasi

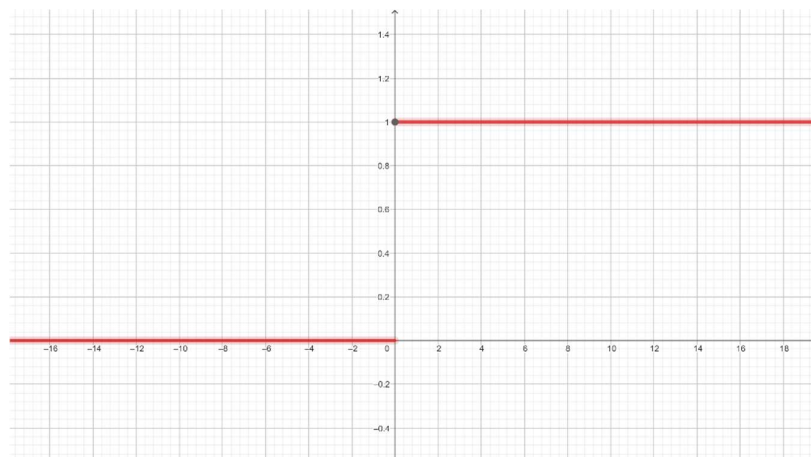
Fungsi aktivasi adalah sebuah fungsi yang digunakan dalam jaringan syaraf tiruan yang berguna untuk menentukan apakah suatu neuron tertentu aktif atau tidak dengan cara menentukan keluaran berdasarkan nilai masukan yang diproses dari lapisan input hingga lapisan output. Selain itu, fungsi aktivasi juga dapat digunakan di setiap node sebagai fungsi yang memutuskan bentuk nilai keluaran dari input sebuah node. Fungsi aktivasi ada berbagai

macam antara lain Binary Step Function, Linear Activation Function, dan Non-Linear Activation Function (Baheti, 2021).

Binary Step Function berfungsi berdasarkan nilai ambang batas yang ditentukan. Jika sebuah nilai kurang dari nilai ambang batas maka fungsi tersebut akan menghasilkan nilai 0 dan jika nilai input lebih dari ambang batas maka fungsi tersebut akan menghasilkan nilai 1. Secara matematis rumus untuk Binary Step Function ditunjukkan dengan Rumus 2.7 berikut.

$$f(x) = \begin{cases} 0 & \text{untuk } x < 0 \\ 1 & \text{untuk } x \geq 0 \end{cases} \quad (2.7)$$

Untuk nilai ambang batas yang digunakan bisa ditentukan sendiri oleh berdasarkan kasus dan data yang diproses. Berikut grafik dari Binary Step Function ditunjukkan oleh Gambar 2.6.

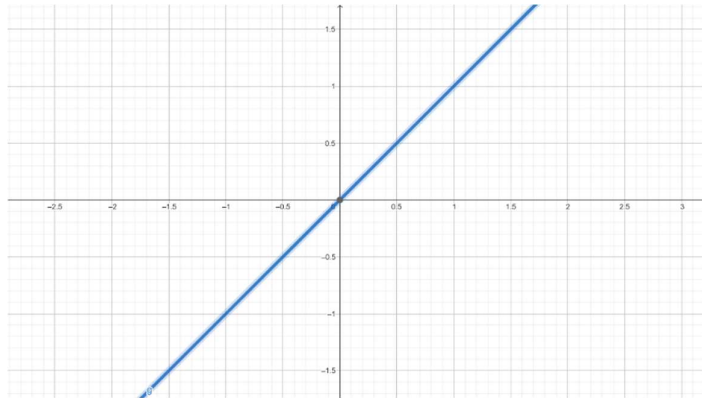


Gambar 2.6 Grafik Fungsi Binary Step

Linear Activation Function dikenal juga dengan fungsi identitas yaitu fungsi yang menghasilkan nilai yang sama dengan nilai inputnya. Sehingga dapat dikatakan fungsi aktivasi linear adalah fungsi tanpa aktivasi. Secara umum fungsi aktivasi linear dapat dituliskan secara matematis menggunakan Rumus 2.8 berikut.

$$f(x) = x \quad (2.8)$$

Dengan grafik ditunjukkan pada Gambar 2.7 berikut.



Gambar 2.7 Grafik Fungsi Aktivasi Linear

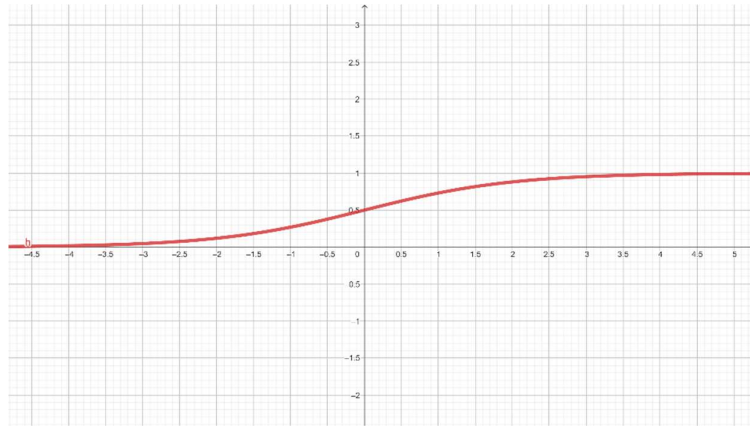
Non-Linear Activation Function adalah fungsi aktivasi selain dua fungsi yang telah disebutkan. Terdapat beberapa macam fungsi aktivasi non-linier sebagai berikut.

1. Fungsi Aktivasi Sigmoid / Logistik

Fungsi sigmoid mengeluarkan nilai direntang 0 hingga 1 berdasarkan besar inputnya, jika inputnya lebih positif maka outputnya akan mendekati angka 1 begitu pun sebaliknya. Secara matematis dapat ditulis dengan Rumsu 2.9 berikut.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.9)$$

Berikut Gambar 2.8 menunjukkan grafik fungsi aktivasi Sigmoid.



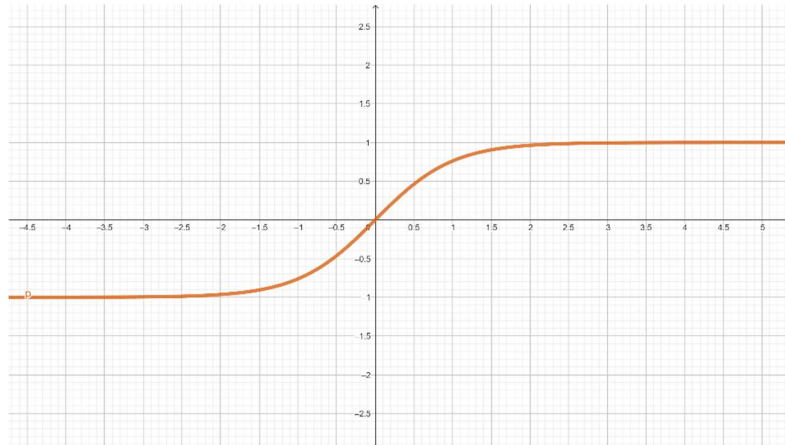
Gambar 2.8 Grafik Fungsi Aktivasi Sigmoid

2. Fungsi Aktivasi Tanh

Fungsi ini mirip dengan fungsi sigmoid perbedaannya yaitu rentang dari fungsi tanh yaitu antara -1 hingga 1. Perilaku dari fungsi tanh juga sama dengan sigmoid yaitu semakin besar input positif maka outputnya akan semakin dekat ke angka 1 begitupun sebaliknya. Berikut rumus dari fungsi tanh yang ditunjukkan oleh Rumus 2.10.

$$f(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (2.10)$$

Grafik fungsi aktivasi Tanh ditunjukkan oleh Gambar 2.9.



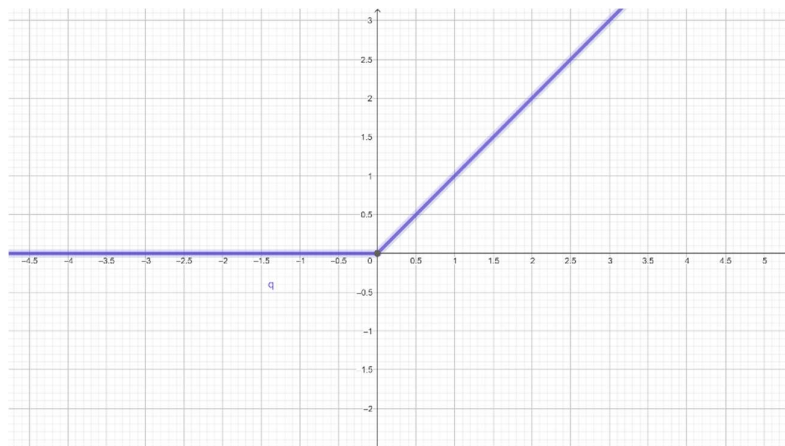
Gambar 2.9 Grafik Fungsi Aktivasi Tanh

3. Fungsi Aktivasi Relu

Fungsi Relu merupakan singkatan dari Rectified Linear Unit yang mana fungsi ini adalah bentuk pengembangan dari fungsi linear. Perbedaan dari relu dengan fungsi linear yaitu fungsi relu akan menghasilkan nilai 0 jika nilai inputnya kurang atau sama dengan 0. Secara rumus dapat dituliskan menggunakan Rumus 2.11 berikut.

$$f(x) = \max(0, x) \quad (2.11)$$

Grafik fungsi aktivasi Relu ditunjukkan oleh Gambar 2.10 berikut.



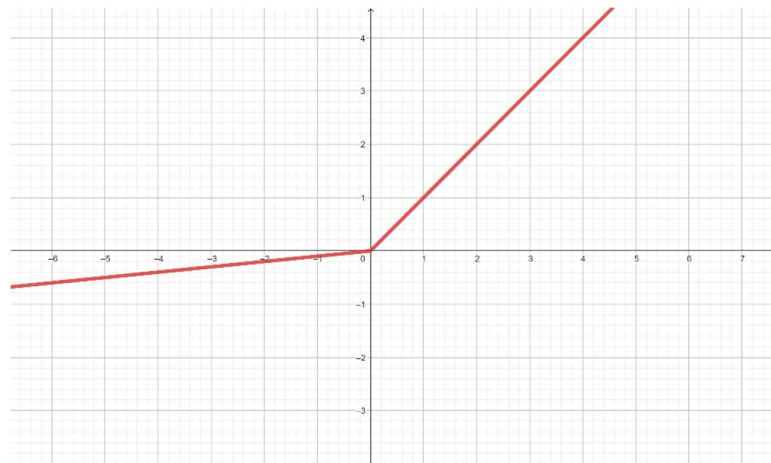
Gambar 2.10 Grafik Fungsi Aktivasi Relu

4. Fungsi Leaky Relu

Fungsi Leaky Relu merupakan versi perbaikan dari fungsi Relu, perbaikan tersebut ditujukan untuk mengatasi masalah yang ditimbulkan Relu ketika nilai input kurang dari 0 yang menghasilkan nilai output 0 juga. Hal tersebut akan mengganggu proses pelatihan model komputasi yang akan berdampak pada hasil pengujian model. Berikut Rumus 2.12 menunjukkan rumus dari fungsi leaky relu.

$$f(x) = \max(0, 1 * x, x) \quad (2.12)$$

Gambar 2.11 menunjukkan grafik fungsi aktivasi Leaky Relu



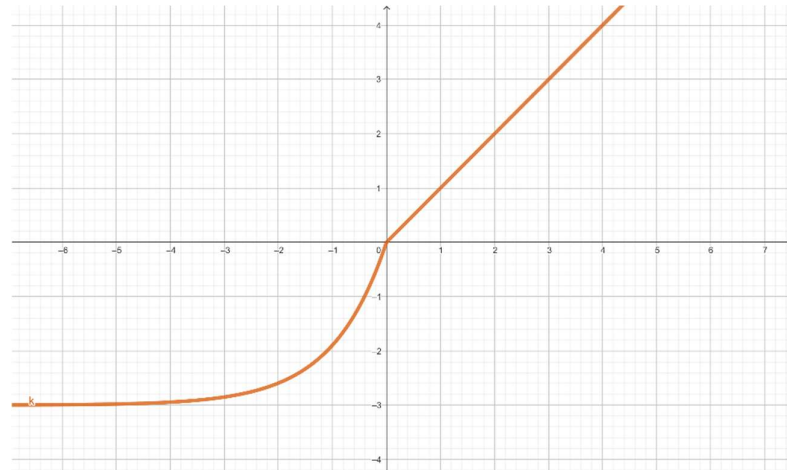
Gambar 2.11 Grafik Fungsi Aktivasi Leaky Relu

5. Fungsi Aktivasi SELU

SELU merupakan singkata dari Scaled Exponential Linear Unit. Fungsi SELU bekerja dengan menormalisasikan diri sendiri dengan menyesuaikan mean atau varians pada lapisan sebelumnya. Fungsi SELU ini masih terbilang baru pada bidang pembelajaran mesin, sehingga perlu adanya penelitian lebih lanjut mengenai perbandingan fungsi SELU dengan fungsi aktivasi lainnya. Berikut rumus dari fungsi SELU ditunjukkan oleh Rumus 2.13.

$$f(\alpha, x) = \lambda \begin{cases} \alpha(e^x - 1) & \text{untuk } x < 0 \\ x & \text{untuk } x \geq 0 \end{cases} \quad (2.13)$$

Fungsi SELU memiliki nilai α dan nilai λ yang telah ditetapkan sebelumnya. Gambar 2.12 menunjukkan grafik fungsi aktivasi SELU dengan nilai $\alpha = 3$ dan nilai $\lambda = 1$.



Gambar 2.12 Grafik Fungsi Aktivasi SELU

2.2.7 Algoritma Optimasi Model

Algoritma Optimasi Model adalah algoritma yang digunakan untuk mengubah beberapa properti pada model komputasi seperti bobot hingga laju pembelajaran. Jadi, bagaimana bobot atau laju pembelajaran tersebut akan diubah atau diperbarui ditentukan oleh algoritma optimasi yang digunakan dengan penggunaan perhitungan atau rumus tertentu. Dengan perubahan bobot atau laju pembelajaran melalui penggunaan algoritma optimasi yang tepat akan mengurangi tingkat loss ketika tahap pelatihan model komputasi dan juga akan berdampak pada tingkat akurasi hasil klasifikasi ataupun prediksi dari model komputasi. Terdapat beberapa macam algoritma optimasi model antara lain (Gupta, 2021).

1. Gradient Descent

Gradient Descent merupakan optimasi model yang paling sederhana namun banyak digunakan pada pengembangan algoritma optimasi berikutnya. Dapat dikatakan semua algoritma optimasi model sekarang berakar dari optimasi model Gradient Descent. Awalnya Gradient Descent sering dipakai pada jaringan saraf tiruan backpropagation,

namun pada saat ini sering digunakan versi pengembangan dari algoritma ini.

2. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent atau sering disebut SGD merupakan pengembangan dari algoritma Gradient Descent. SGD memperbaiki masalah optimasi Gradient Descent dimana performanya akan turun ketika ukuran datasetnya membesar, sehingga membutuhkan memori yang besar dan waktu yang lama. SGD akan memperbarui bobot setiap akhir dari sebuah data dari sebuah dataset per satu kali siklus dataset, alih-alih memperbarui satu kali pada akhir dataset seperti Gradient Descent.

3. Adam

Adaptive Moment Estimation atau disingkat Adam merupakan algoritma optimasi hasil pengembangan dari algoritma Stochastic Gradient Descent (SGD) dengan memperbarui laju pembelajaran dari masing-masing bobot secara individu pada saat tahap pelatihan model komputasi. Sedangkan laju pembelajaran pada SGD tidak diubah selama proses pelatihan. Proses pembaruan laju pembelajaran dalam algoritma Adam yaitu dengan menghitung laju pembelajaran berdasarkan gradien sebelumnya dan kedua momen. Kedua momen tersebut yaitu rata-rata dan varians yang tak terpusat dari gradien. Dengan perhitungan tersebut, algoritma ini dapat membantu mempercepat konvergensi dan meningkatkan performa dari model komputasi. Jadi, perbedaan Adam dengan SGD terletak pada perubahan laju pembelajarannya.

4. RMS Prop

RMS prop adalah singkatan dari Root Mean Square Propagation. Sesuai dengan namanya algoritma optimasi ini menghitung properti dalam model komputasi dengan menghitung rata-rata akar kuadrat dari gradien. Alhasil laju pembelajaran dari sebuah model komputasi dapat disesuaikan dan lebih adaptif terhadap perubahan gradien. Dengan RMS

prop masalah dalam pelatihan model komputasi seperti vanishing gradient dapat diminimalisir.

2.2.8 Python, Tensorflow, dan Keras

Python merupakan salah satu bahasa pemrograman interpretatif, interaktif dan berorientasi objek. Python menyediakan beberapa fitur seperti *high-level structure data*, *dynamic typing*, *classes* dan lain-lain. Python dikembangkan oleh Guido van Rossum pada tahun 1990 yang bersifat gratis dan terbuka bagi umum. Python berjalan dengan menggunakan interpreter sehingga bahasa pemrograman ini dapat dijalankan tanpa perlu proses compile manual yang dilakukan pengguna seperti bahasa pemrograman lainnya (Sanner, 1999).

Kegunaan dari bahasa pemrograman Python sangat bermacam-macam. Kita dapat menggunakan Python untuk membangun situs web, aplikasi atau software, otomatisasi tugas, membangun server, dan melakukan analisis data serta hal-hal yang berkaitan dengan pembelajaran mesin. Hingga saat ini, Python sering digunakan pada bidang pembelajaran mesin dan kecerdasan buatan (Dicoding, 2023).

Beberapa alasan mengapa Python cocok digunakan pada bidang pembelajaran mesin antara lain sederhana dan konsisten. Python memiliki sintaks kode yang ringkas dan mudah untuk dibaca dan dipahami (*good readability*), banyak programmer yang sepakat bahwa Python lebih intuitif dibandingkan bahasa pemrograman lainnya. Alasan selanjutnya yaitu pilihan *library* yang sangat luas sehingga dapat menunjang berbagai macam keperluan seperti pembelajaran mesin. Terdapat *library manager* untuk Python yaitu PIP yang memiliki banyak sekali library yang dapat digunakan pada berbagai macam bidang seperti pembelajaran mesin terdapat Tensorflow, Keras, Scikit-Learn, untuk bidang komputasi terdapat NumPy, dan data analisis terdapat Pandas. Alasan selanjutnya yaitu komunitas yang sangat luas. Python memiliki komunitas yang sangat luar biasa sehingga dengan adanya komunitas tersebut Python akan selalu diperbarui (*update*) terhadap permasalahan yang baru. Selain itu, dengan dukungan komunitas yang luas dapat saling berbagi antar pengembang (Beklemysheva, 2019).

Seperti yang sudah disebutkan sebelumnya, Tensorflow dan Keras merupakan salah satu *library* yang dapat dimanfaatkan pada bidang pembelajaran mesin dan kecerdasan buatan.

Tensorflow adalah sebuah *library* yang dikembangkan oleh Google Inc. yang sangat populer khususnya pada penerapan pembelajaran mesin. Saat ini, Tensorflow dapat diterapkan di berbagai jenis platform seperti desktop, web, hingga mobile. Tensorflow bersifat gratis dan terbuka sehingga semua orang dapat mengakses *library* ini. Tensorflow memiliki komponen inti yaitu tensor dan grafik komputasi. Tensor adalah sebuah objek matematika yang digunakan untuk menggambarkan dimensi dari sebuah data seperti skalar dan vektor. Grafik komputasi menggambarkan grafik yang memiliki arah untuk setiap node dengan operasi matematika tertentu.

Pada bidang pembelajaran mesin Tensorflow dapat digunakan untuk membangun sebuah model komputasi yang digunakan untuk berbagai macam kebutuhan seperti pengenalan gambar, pengenalan pola, prediksi, dan kasus-kasus lainnya. Tentunya model komputasi yang dibuat tersebut harus melewati tahap pelatihan terlebih dahulu untuk kemudian dapat dimanfaatkan.

Tensorflow banyak dimanfaatkan oleh perusahaan-perusahaan besar seperti Lenovo, Twitter, bahkan Google sendiri juga menggunakan Tensorflow di berbagai macam produk mereka. Tensorflow sering dipilih untuk pengembangan pembelajaran mesin karena beberapa alasan seperti abstraksi yang multi-level, pengkodean yang fleksibel, ekosistem yang memadai, dan powerful dalam hal eksperimen dan pengujian konsep (Wedde, 2020).

Selain Tensorflow, terdapat Keras API yang berjalan diatas *library* Tensorflow. Keras adalah sebuah API (Application Programming Interface) yang memudahkan pengembang untuk merancang, melatih, dan menguji model komputasi. Keras berjalan dengan menyederhanakan algoritma-algoritma *deep learning* dan membuat tingkat abstraksi yang tinggi diatas Tensorflow, sehingga Keras lebih berfokus pada pemahaman manusia

terhadap *deep learning* dengan menyediakan API untuk merancang, melatih dan menguji model komputasi (Dhadse, 2021).

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang digunakan pada penelitian ini yaitu eksperimen. Metode eksperimen adalah metode penelitian yang digunakan untuk mengetahui hubungan antara variabel-variabel yang ada. Tujuan dari metode eksperimen ini yaitu mengidentifikasi perubahan sebuah variabel terhadap variabel yang lainnya. Hasil dari metode ini yaitu berupa data-data hasil eksperimen yang dijalankan untuk kemudian dianalisis, diolah, dan diinterpretasikan sehingga dapat menjadi informasi yang dapat dimanfaatkan.

Tahap-tahap dari metode eksperimen pada penelitian ini antara lain.

1. Pengumpulan dan persiapan data bahan
2. Perancangan eksperimen meliputi penentuan variabel-variabel yang diuji
3. Pelaksanaan eksperimen dan pencatatan data hasil eksperimen
4. Analisis dan pembahasan data hasil eksperimen
5. Penarikan kesimpulan dan dokumentasi.

Dalam metode eksperimen dikenal dengan variabel-variabel yang diuji, berikut jenis-jenis variabel-variabel tersebut.

1. Variabel Bebas atau Independen

Variabel bebas adalah variabel yang diuji dengan memanipulasi nilainya pada sebuah skenario untuk mengetahui respons dari variabel dependen atau terikat.

2. Variabel Kontrol

Variabel kontrol adalah variabel yang dibuat tetap atau konstan selama pelaksanaan eksperimen sehingga yang hanya perlu diamati adalah perubahan variabel bebas dan dampaknya terhadap variabel terikat. Penentuan variabel kontrol sangat penting karena berfungsi untuk

menghilangkan pengaruh variabel-variabel yang berpotensi mengganggu jalannya eksperimen yang mana variabel pengganggu tersebut bukan menjadi fokus dari sebuah penelitian atau skenario namun jika tidak dikendalikan akan mempengaruhi hasil penelitian. Selain itu, dengan mengendalikan variabel kontrol perbandingan antar variabel-variabel bebas dapat menjadi akurat karena variabel yang dimanipulasi atau diubah adalah variabel bebasnya.

3. Variabel Terikat

Variabel terikat atau variabel dependen adalah variabel yang diamati dalam sebuah eksperimen atau skenario. Variabel ini merupakan respons atau hasil dari manipulasi variabel-variabel bebas yang diuji. Dengan kata lain variabel terikat ini adalah variabel yang diharapkan akan berubah dengan perubahan variabel bebas. Variabel terikat ini penting karena menjadi sebuah metrik penilaian atau nilai yang akan dianalisis sebab adanya perubahan variabel bebasnya.

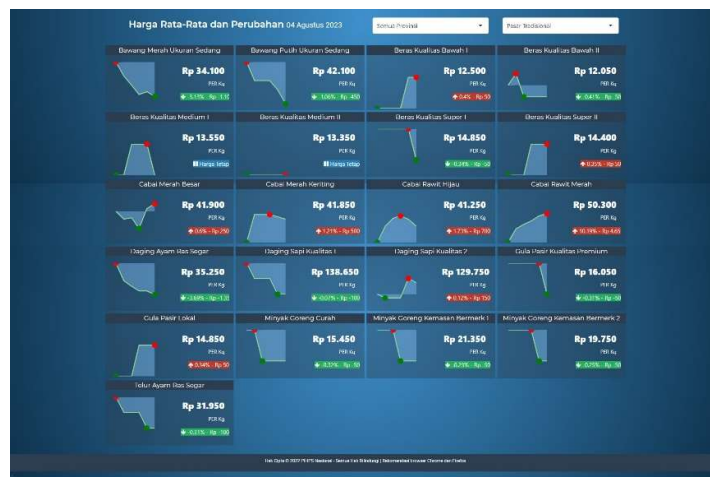
3.2 Metode Pengumpulan Data Bahan

Data yang digunakan pada penelitian ini berupa data harga cabai khususnya di Provinsi Jawa Timur. Terdapat sebuah situs web yang menyediakan data serta informasi dari harga-harga bahan pokok nasional yaitu PIHPS Nasional (Pusat Informasi Harga Pangan Strategis Nasional) dapat diakses pada tautan berikut <https://www.bi.go.id/hargapangan>. Situs web ini dikelola oleh Bank Indonesia dan data akan diperbarui setiap harinya pada hari-hari kerja (Senin-Jumat). Tangkapan layar laman beranda dari situs web PIHPS Nasional ditunjukkan pada Gambar 3.1. Pada Gambar 3.1 terlihat peta Indonesia dengan pembagian perprovinsi dan terdapat skala warna yang menunjukkan harga komoditas pada provinsi tersebut.



Gambar 3.1 Beranda PIHPS Nasional

Terdapat beberapa informasi yang ditampilkan pada situs web tersebut seperti harga rata-rata nasional untuk masing-masing komoditas, grafik perubahan harga per komoditas, dan juga tabel harga. Grafik perubahan harga ditunjukkan pada Gambar 3.2 yang menunjukkan grafik naik turunnya harga dari berbagai macam komoditas.



Gambar 3.2 Grafik Perubahan Harga

Pada menu Tabel Harga, kita akan disuguhkan tabel dari harga nasional untuk masing-masing komoditas. Pengguna dapat menyaring data berdasarkan komoditas dan daerah yang diinginkan. Selain itu, pengguna juga dapat mengatur rentang tanggal untuk melihat data harga pada rentang tanggal tersebut. Peneliti memanfaatkan data yang disediakan oleh PIHPS Nasional sebagai data untuk penelitian ini. Dengan menggunakan filter

komoditas dan daerah peneliti berhasil mendapatkan data harga cabai di Provinsi Jawa Timur. Selanjutnya peneliti mengunduh data tersebut dengan ekstensi xlsx untuk kemudian diolah lebih lanjut untuk kemudian digunakan sebagai data input model komputasi. Peneliti mendapatkan satu (1) file yang berisi data harga cabai terhitung sejak tanggal 31 Desember 2019 hingga tanggal 2 Juni 2023. Gambar 3.3 menunjukkan bentuk data mentah ketika penulis mengunduh file xlsx dari situs web PIHPS Nasional sebagai berikut.

No	Komoditas (Rp)	2019		2020		2021		2022		2023											
		12/2019	01/2020	02/2020	03/2020	04/2020	05/2020	06/2020	07/2020	08/2020	09/2020	10/2020									
1	Cabai Merah	34,750	-	36,800	39,800	47,000	46,900	47,650	50,100	52,300	56,000	58,200	61,450	62,550	61,850	60,200	59,750	60,250	60,300	59,400	59,650
1	Cabai Merah Besar	35,300	-	38,700	41,150	48,300	48,500	49,800	51,400	53,500	58,100	60,150	63,350	64,350	64,050	62,750	61,650	62,300	62,250	63,950	62,450
2	Cabai Merah Kenting	33,650	-	34,450	37,950	44,750	44,650	45,450	49,050	50,950	53,250	55,100	58,000	58,900	57,200	56,450	56,750	56,400	56,200	55,250	55,350
3	Cabai Rawit	27,400	-	28,950	30,150	36,350	37,250	38,400	40,250	41,800	42,950	43,500	46,050	47,300	48,400	51,150	52,100	54,900	54,650	54,200	51,700
4	Cabai Rawit Hijau	22,300	-	24,200	24,400	27,350	28,750	29,050	30,350	31,250	32,000	31,800	33,800	33,900	33,600	35,050	35,150	34,950	35,100	33,550	34,200
7	Cabai Rawit Merah	33,650	-	34,550	36,850	47,200	47,550	49,900	52,300	54,450	56,050	57,450	63,100	65,350	66,900	70,150	72,200	78,050	78,200	77,050	73,350

Gambar 3.3 Data Harga Cabai dari Tahun 2020 hingga Tahun 2023

Gambar 3.3 menunjukkan terdapat baris tanggal, dan beberapa komoditas dan dari file tersebut didapatkan data mentah sebanyak 894 nilai baik itu cabai merah ataupun cabai rawit dan kemudian data tersebut harus dilakukan *preprocessing*.

3.3 Alat dan Bahan

Pada penelitian kali ini alat yang digunakan yaitu sebuah *notebook* dengan spesifikasi yaitu prosesor Intel Core-I5 gen 10, memori 8 GB dan VGA Nvidia GTX 1650. Selain itu terdapat beberapa *library* untuk bahasa pemrograman Python yang peneliti gunakan yaitu Tensorflow dengan API Keras untuk merancang model komputasi LSTM, Pandas untuk membaca file data, Numpy untuk mengubah data menjadi bentuk array, Pyplot untuk memvisualisasikan data, MinMaxScaling dari Scikit Learn untuk normalisasi data selain itu Microsoft Excel digunakan untuk mencatat data hasil eksperimen yang telah dilakukan.

Adapun bahan yang digunakan yaitu data harga cabai di Provinsi Jawa Timur pada tahun 2020 hingga 2023 yang didapat dari situs web PIHPS Nasional yang dikelola oleh Bank Indonesia. Data harga tersebut berupa hasil dari laporan harian dengan metode pengumpulan data dilakukan setiap hari kerja (Senin-Jumat) melalui survei hasil transaksi antara pembeli dan penjual pada pukul 09.00 sampai dengan 10.00 yang kemudian akan dilaporkan pada Bank Indonesia pada pukul 10.00 hingga 12.00 dan dipublikasikan pada pukul 13.00. Data yang didapat yaitu data di Provinsi Jawa Timur yang merupakan harga rata-rata dari seluruh kabupaten atau kota yang disurvei.

3.4 Langkah-langkah Penelitian

Adapun langkah-langkah penelitian yang peneliti tempuh yaitu sebagai berikut.

3.4.1 Pengumpulan dan Persiapan Data Bahan

Pada langkah ini peneliti mengumpulkan data dari situs web PIHPS Nasional (Pusat Informasi Harga Pangan Strategis Nasional) yang dikelola oleh Bank Indonesia. Data disaring berdasarkan komoditas yang diinginkan yaitu cabai merah dan cabai rawit serta disaring berdasarkan wilayahnya yaitu Provinsi Jawa Timur. Data yang berhasil didapat peneliti yaitu sejumlah 894 data baik harga cabai merah maupun cabai rawit. Data tersebut merupakan data yang masih mentah sehingga perlu dilakukan proses persiapan data yang meliputi pembersihan dan normalisasi agar dapat di-*input* pada model komputasi.

Persiapan data pada penelitian ini yaitu meliputi pembersihan dan normalisasi data serta pembagian data menjadi data latih dan data uji. Data yang didapat peneliti masih berupa file dengan format *xlsx* sehingga untuk mempermudah pembacaan data oleh bahasa pemrograman, maka peneliti mengubah format file data dari *xlsx* menjadi *csv* (*Comma-Separated Values*) serta mentranspos dari bentuk mendatar ke bentuk vertikal seperti pada gambar berikut.

	A	B	C	D	E	F	G	H	I	J	K
1	tanggal	cabaiMerah	cabaiRawit								
2	31/12/2019	34750	27400								
3	01/01/2020	-	-								
4	02/01/2020	36800	28950								
5	03/01/2020	39800	30150								
6	06/01/2020	47000	36350								
7	07/01/2020	46900	37250								
8	08/01/2020	47650	38400								
9	09/01/2020	50100	40250								
10	10/01/2020	52300	41800								
11	13/01/2020	56000	42950								
12	14/01/2020	58200	43500								
13	15/01/2020	61450	46050								
14	16/01/2020	62550	47300								

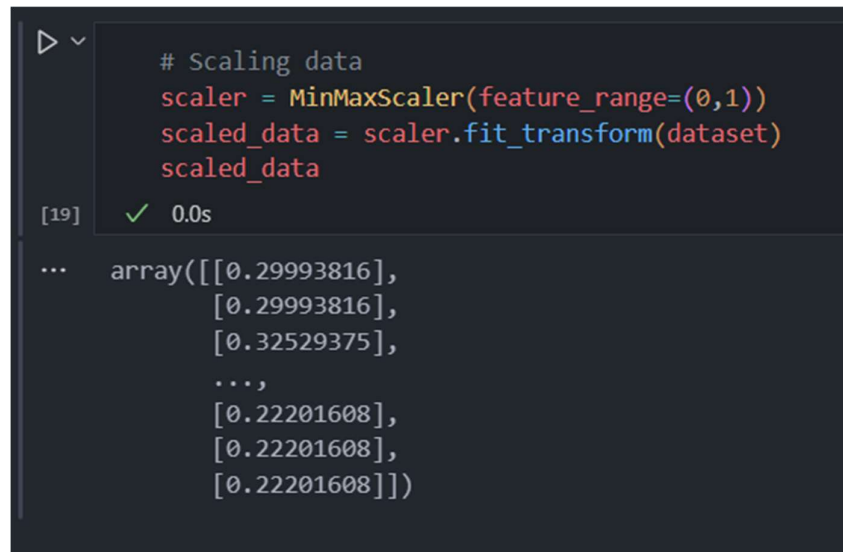
Gambar 3.4 Transpos Data Harga Cabai

Kemudian membersihkan data dari nilai-nilai yang tidak diinginkan contohnya pada kasus ini yaitu data dimana yang memiliki karakter yang tidak diinginkan seperti “-” pada tanggal tertentu. Pada hal ini karena LSTM membutuhkan data yang berupa *time series* dimana setiap data harus memiliki interval yang sama seperti interval satu hari. Untuk data yang kosong atau data yang terdapat karakter yang tidak diinginkan harus diisi dengan sebuah nilai. Seperti data pada hari Sabtu atau Minggu yang kosong itu harus diisi terlihat pada Gambar 3.4 yang mana tanggal 4 – 5 Januari 2020 tidak ada karena pada tanggal tersebut adalah hari Sabtu dan Minggu. Terdapat beberapa cara untuk mengisi suatu data yang kosong seperti rerata nilai sebelum dan sesudah data kosong, menggunakan data pada indeks sebelumnya dan lain sebagainya.

Pada kasus ini, penulis menggunakan metode yang kedua yaitu mengisi data yang kosong atau yang terdapat karakter yang tidak diinginkan dengan data indeks sebelumnya. Cara tersebut dipilih sebab pada kenyataannya data dari sumber juga diperbarui per-harinya, jika menggunakan cara rerata maka membutuhkan data satu indeks setelahnya agar bisa dilakukan perhitungan, namun karena sumber memperbarui data perhari maka untuk tidak mungkin untuk mendapatkan data satu indeks setelahnya sehingga cara dengan mengisi menggunakan data sebelumnya lebih mungkin dilakukan. Setelah dilakukan

pembersihan data dan mengisi data kosong, didapatkan data bersih yaitu sejumlah 1250 nilai.

Normalisasi digunakan ketika rentang data yang ada tidak sesuai dengan rentang data yang menjadi syarat input bagi model komputasi. Umumnya model komputasi hanya bisa menerima input numerik dari 0 hingga 1, untuk proses normalisasi perlu dilakukan agar data dapat diproses oleh model komputasi. Untuk proses normalisasi ini peneliti menggunakan salah satu teknik normalisasi MinMaxScaling. Normalisasi ini akan membandingkan sebuah nilai dari dataset dengan nilai minimum dan nilai maksimum pada dataset tersebut, umumnya rentang data dari metode MinMaxScaling yaitu 0 hingga 1 sehingga metode ini cocok digunakan untuk menormalisasikan data. Gambar 3.5, menunjukkan hasil proses normalisasi data cabai merah menggunakan metode MinMaxScaler yang menghasilkan data dengan rentang 0 hingga 1 berdasarkan nilai minimum dan maksimum dari dataset tersebut.



```
# Scaling data
scaler = MinMaxScaler(feature_range=(0,1))
scaled_data = scaler.fit_transform(dataset)
scaled_data

[19] ✓ 0.0s

... array([[0.29993816],
          [0.29993816],
          [0.32529375],
          ...,
          [0.22201608],
          [0.22201608],
          [0.22201608]])
```

Gambar 3.5 Hasil Normalisasi Data Harga Cabai Merah

Setelah proses normalisasi, langkah selanjutnya peneliti membagi data tersebut menjadi data latih dan data uji dengan rasio tertentu, misalnya 80% untuk data latih dan 20% untuk data uji atau 70% banding 30%. Selanjutnya, data akan dibentuk kedalam bentuk sekuens dengan panjang tertentu. Hal ini

perlu dilakukan mengingat model komputasi yang akan digunakan (LSTM) hanya bisa menerima data dengan bentuk sekuens.

```
# Split train data and validation data
train, validation = train_val_split(scaled_data, sequence_length, train_data_ratio)
# train, validation = train_val_split(dataset, sequence_length, train_data_ratio)

X_train, Y_train = x_y_split_sequence(train, sequence_length)
X_validation, Y_validation = x_y_split_sequence(validation, sequence_length)

print(X_train)

[27] ✓ 0.0s
... [[0.29993816 0.29993816 0.32529375 ... 0.36239951 0.36239951 0.45145331]
      [0.29993816 0.32529375 0.36239951 ... 0.36239951 0.45145331 0.45021645]
      [0.32529375 0.36239951 0.36239951 ... 0.45145331 0.45021645 0.45949289]
      ...
      [0.29808287 0.29560915 0.29560915 ... 0.29560915 0.27520099 0.27210884]
      [0.29560915 0.29560915 0.29560915 ... 0.27520099 0.27210884 0.26035869]
      [0.29560915 0.29560915 0.29560915 ... 0.27210884 0.26035869 0.25602968]]
```

Gambar 3.6 Hasil Pembentukan Data menjadi Bentuk Sekuens

Gambar 3.6 menunjukkan hasil pembentukan sekuens dari dataset cabai merah dengan panjang 7 data per-sekuens-nya. Pada sekuens pertama adalah sekuens yang berisi 7 data pertama kemudian pada sekuens kedua maju 1 indeks data sehingga mulai dari indeks data ke-2 dan 7 data setelahnya menjadi sekuense kedua, begitu seterusnya hingga semua dataset diubah menjadi bentuk sekuens.

Dalam proses pembentukan sekuens, data juga dibagi menjadi dua yaitu data input dan data label. Data input berfungsi untuk data pada lapisan input pada model komputasi sedangkan data label berfungsi sebagai data target atau hasil yang diharapkan dari sebuah model komputasi. Pada jenis data sekuens yang menjadi data label adalah data yang posisinya tepat setelah data-data sepanjang sekuens. Umumnya data input sering diberi nama sebagai X dan data label diberi nama sebagai Y. Contoh dari pembentukan label dapat dilihat pada Gambar 3.7, pada gambar tersebut label untuk sekuens pertama adalah 1 indeks setelah data terakhir pada sekuens pertama yaitu data ke-8, kemudian label untuk sekuens ke-2 adalah data ke-9. Begitupun seterusnya hingga semua dataset habis.

```

# Split train data and validation data
train, validation = train_val_split(scaled_data, sequence_length, train_data_ratio)
# train, validation = train_val_split(dataset, sequence_length, train_data_ratio)

X_train, Y_train = x_y_split_sequence(train, sequence_length)
X_validation, Y_validation = x_y_split_sequence(validation, sequence_length)

print(X_train[0], Y_train[0])
print(X_train[1], Y_train[1])

```

[32] ✓ 0.0s

```

... [0.29993816 0.29993816 0.32529375 0.36239951 0.36239951 0.36239951
0.45145331] 0.4502164502164502
[0.29993816 0.32529375 0.36239951 0.36239951 0.36239951 0.45145331
0.45021645] 0.45949288806431665

```

Gambar 3.7 Contoh Data Label dari Sekuens Data Cabai Merah

Data yang setelah dibentuk menjadi sekuens akan menjadi data dengan 2 dimensi yakni jumlah sekuens yang dibentuk dan panjang dari sekuens. Kemudian data yang sudah dibentuk menjadi sekuens akan ditambah dimensinya menjadi 3 dimensi karena model komputasi yang akan digunakan (LSTM) hanya menerima data input dengan dimensi 3. Dimensi ketiga ini digunakan ketika terdapat lebih dari satu parameter dalam satu sekuens misal data harga saham dengan adanya nilai *open* dan *close* dalam satu sekuens.

```

# Split train data and validation data
train, validation = train_val_split(scaled_data, sequence_length, train_data_ratio)
# train, validation = train_val_split(dataset, sequence_length, train_data_ratio)

X_train, Y_train = x_y_split_sequence(train, sequence_length)
X_validation, Y_validation = x_y_split_sequence(validation, sequence_length)

print("sebelum reshape", X_train.shape)

X_train, X_validation = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1)), np.reshape(X_validation, (X_validation.shape[0],
X_validation.shape[1], 1))

print("setelah reshape", X_train.shape)

```

[36] ✓ 0.0s

```

... sebelum reshape (1118, 7)
setelah reshape (1118, 7, 1)

```

Gambar 3.8 Reshaping Data menjadi 3 Dimensi

Pada Gambar 3.8 menunjukkan proses *reshaping* data yang sebelumnya berbentuk 2 dimensi dengan panjang array 1118 dan setiap elemen array tersebut berisi 7 data. Kemudian dilakukan *reshaping* untuk menambah dimensi pada data tersebut menjadi 3 dimensi dengan panjang dimensi pertama 1118, dimensi kedua 7, dan dimensi ketiga 1. Panjang dimensi ketiga bergantung pada berapa fitur yang akan di-input pada model komputasi.

Pada penelitian ini peneliti hanya akan menggunakan 1 jenis data atau fitur untuk masing-masing pelatihan model komputasi, sehingga ukuran

dimensi ketiga dari dataset hanya 1. Penambahan dimensi pada data dapat dilakukan dengan menggunakan *library* Numpy dengan fungsi *reshape*. Penambahan dimensi ini hanya diterapkan pada semua data input baik pada data latih maupun data uji.

3.4.2 Perancangan eksperimen

Pada tahap ini penulis merancang eksperimen menjadi tujuh (7) skenario dimana setiap skenario akan menguji sebuah variabel bebas. Dari tujuh skenario tersebut terbagi menjadi tiga (3) skenario pengujian arsitektur dari model komputasi dan empat (4) skenario pengujian beberapa parameter yang berhubungan dengan persiapan data, pelatihan model komputasi dan optimasi model komputasi.

Tiga skenario pertama menguji arsitektur dari model komputasi dengan tiga variabel bebas yang diuji yaitu jumlah unit, jumlah lapisan tersembunyi dan fungsi aktivasi yang digunakan. Berikut rincian dari masing-masing skenario.

1. Skenario 1 (Jumlah Unit)

Pada skenario ini jumlah unit menjadi variabel bebas atau variabel yang akan diubah-ubah. Untuk jumlah unit yang diuji yaitu dimulai dari 10 unit terus bertambah dengan beda 10 (10, 20, 30, dst.) hingga nilai RMSE yang dihasilkan minimum yang ditandai dengan pengujian unit selanjutnya menghasilkan nilai RMSE yang kembali naik. Terdapat beberapa variabel kontrolnya meliputi jenis sel yang digunakan yaitu sel dengan jenis LSTM pada lapisan input dan sel jenis Dense pada lapisan output, jumlah lapisan tersembunyi yang digunakan yaitu sebanyak 0 atau tidak menggunakan lapisan tersembunyi sama sekali, fungsi aktivasi yang digunakan yaitu fungsi aktivasi linear.

2. Skenario 2 (Jumlah Lapisan Tersembunyi)

Jumlah lapisan tersembunyi menjadi variabel yang diuji pada skenario ini. Jumlah lapisan tersembunyi yang diuji yaitu 0 lapis, 1 lapis, 2 lapis dan 3 lapis dengan beberapa variabel kontrol antara lain jenis sel yang digunakan pada input maupun lapisan tersembunyi yaitu sel LSTM

dengan sel Dense pada lapisan outputnya, jumlah unit yang digunakan pada sel dengan tipe LSTM yaitu sebanyak 50 unit, dan fungsi aktivasi yang digunakan adalah fungsi aktivasi linear.

3. Skenario 3 (Fungsi Aktivasi yang digunakan)

Skenario 3 menguji beberapa macam fungsi aktivasi antara lain fungsi aktivasi linear, Relu, Leaky Relu, dan SELU. Dengan variabel kontrol pada skenario ini yaitu jenis sel yang digunakan pada lapisan input yaitu LSTM dan sel Dense pada lapisan output, jumlah unit untuk sel LSTM sebanyak 50 units dan sel Dense sebanyak 1 unit, jumlah lapisan tersembunyi yang digunakan yaitu 0 atau tidak menggunakan lapisan tersembunyi.

Untuk parameter yang berkaitan dengan data dan lain-lain juga ditetapkan dan dibuat konstan agar hasil dari pengujian variabel bebas menjadi valid. Adapun beberapa parameter yang ditetapkan yaitu rasio data latih 80%, panjang sekuens 30, epoch yang dijalankan 30, dan optimasi model menggunakan optimasi Adam.

Variabel terikat pada skenario 1, 2, dan 3 sama yaitu RMSE yang dihasilkan dari pengujian model komputasi, MAPE dari data validasi, dan durasi latih dari model komputasi. Tabel 3.1, 3.2, dan 3.3 menunjukkan ringkasan dari rincian skenario 1 hingga 3.

Tabel 3.1 Ringkasan Skenario 1 (Jumlah Unit)

Variabel Bebas	Variabel Kontrol	Variabel Terikat
Jumlah unit sel LSTM: dimulai 10 terus bertambah dengan beda 10	<ul style="list-style-type: none"> Jenis sel input: LSTM Jenis sel output: Dense (1 unit) 	RMSE
	Jumlah lapisan tersembunyi: -	MAPE
	Fungsi aktivasi: fungsi linear	Durasi latih

Tabel 3.2 Ringkasan Skenario 2 (Jumlah Lapisan Tersembunyi)

Variabel Bebas	Variabel Kontrol	Variabel Terikat
Jumlah lapisan tersembunyi: (0, 1, 2, 3)	<ul style="list-style-type: none"> Jenis sel input: LSTM (50 unit) Jenis sel lapisan tersembunyi: LSTM (50 unit) Jenis sel output: Dense (1 unit) 	RMSE
	Fungsi aktivasi: fungsi linear	MAPE
		Durasi latihan

Tabel 3.3 Ringkasan Skenario 3 (Fungsi Aktivasi)

Variabel Bebas	Variabel Kontrol	Variabel Terikat
Fungsi aktivasi: Linear, Relu, Leaky Relu, dan SELU	<ul style="list-style-type: none"> Jenis sel input: LSTM (50 unit) Jenis sel output: Dense (1 unit) 	RMSE
	Jumlah lapisan tersembunyi: -	MAPE
		Durasi latihan

Empat skenario terakhir menguji parameter-parameter yang berkaitan dengan persiapan data, pelatihan model komputasi dan optimasi model komputasi.

1. Skenario 4 (Rasio Data Latih)

Pada skenario ini dilakukan pengujian terhadap rasio data latih dengan data uji. Beberapa rasio yang diuji yaitu 70:30 (70%), 80:20 (80%), dan 90:10 (90%). Dengan beberapa variabel kontrol yaitu panjang sekuens yaitu 30, jumlah epoch 30, dan optimasi model Adam.

2. Skenario 5 (Panjang Sekuens)

Pengujian panjang sekuens dengan beberapa variasi nilai berdasarkan perhitungan mingguan dan bulanan yaitu 7, 14, 21, dan 30. Beberapa variabel kontrol yang ditetapkan yaitu rasio data latih 80%, jumlah epoch 30, dan optimasi model yang digunakan yaitu optimasi Adam.

3. Skenario 6 (Epoch)

Terdapat 3 nilai epoch yang diuji yaitu 10, 20, 30, 40 dan 50 dengan beberapa variabel kontrol yaitu rasio data latih 80%, panjang sekuens 30, dan optimasi model Adam.

4. Skenario 7 (Optimasi Model)

Skenario terakhir menguji beberapa optimasi model yang digunakan pada model komputasi yaitu Adam, RMSprop dan SGD. Dengan variabel kontrol yang dikendalikan yaitu rasio data latih 80%, panjang sekuens 30, dan jumlah epoch 30.

Untuk arsitektur yang digunakan pada skenario 4 hingga 7 yaitu menggunakan arsitektur model yang sama pada skenario 1 dengan penetapan jumlah unit yang digunakan yaitu 50 dengan sel input yaitu LSTM, sel output Dense (1 unit), jumlah lapisan tersembunyi 0, dan fungsi aktivasi yang digunakan yaitu fungsi linear.

Variabel yang diamati atau variabel terikat dari skenario 4 hingga 7 yaitu RMSE yang dihasilkan dari proses pengujian model komputasi, MAPE dari data validasi, dan durasi latih model komputasi. Tabel 3.4, Tabel 3.5, Tabel 3.6 dan Tabel 3.7 menunjukkan ringkasan dari skenario 4 hingga 7.

Tabel 3.4 Ringkasan Skenario 4 (Rasio Data Latih)

Variabel Bebas	Variabel Kontrol	Variabel Terikat
Rasio data latih: (70%, 80%, dan 90%)	Panjang sekuens: 30	RMSE
	Epoch: 30	MAPE
	Optimasi model: Adam	Durasi latih

Tabel 3.5 Ringkasan Skenario 5 (Panjang Sekuens)

Variabel Bebas	Variabel Kontrol	Variabel Terikat
Panjang sekuens: (7, 15, 21, dan 30)	Rasio data latih: 80%	RMSE
	Epoch: 30	MAPE
	Optimasi model: Adam	Durasi latih

Tabel 3.6 Ringkasan Skenario 6 (Jumlah Epoch)

Variabel Bebas	Variabel Kontrol	Variabel Terikat
Epoch: (10, 20, 30, 40, dan 50)	Rasio data latih: 80%	RMSE
	Panjang sekuens: 30	MAPE
	Optimasi model: Adam	Durasi latih

Tabel 3.7 Ringkasan Skenario 7 (Optimasi Model)

Variabel Bebas	Variabel Kontrol	Variabel Terikat
Optimasi model: (Adam, RMSprop, dan SGD)	Rasio data latih: 80%	RMSE
	Panjang sekuens: 30	MAPE
	Epoch: 30	Durasi latih

3.4.3 Pelaksanaan Eksperimen

Eksperimen dilaksanakan dengan menjalankan semua skenario dari 1 hingga 7. Eksperimen dilakukan terhadap dua jenis data harga cabai yaitu cabai merah dan cabai rawit. Dari proses pelaksanaan eksperimen tersebut akan dihasilkan data berupa variabel bebas apa yang diuji, variabel kontrol, dan variabel terikat. Penulis juga merancang agar hasil dari pengujian dari model komputasi tersebut dapat divisualisasikan dalam bentuk grafik garis yang kemudian disimpan untuk bahan analisis lebih lanjut. Setiap data yang dihasilkan dari proses eksperimen tersebut akan tercatat otomatis dalam sebuah file dengan ekstensi txt pada setiap akhir dijalankannya sebuah pengujian variabel bebas. Untuk mendapatkan data yang lebih valid, setiap variabel bebas yang diuji akan dijalankan sebanyak 3 kali sehingga kemudian akan didapatkan hasil dari pengujian 1 variabel bebas sebanyak 3. Kemudian dari 3 data tersebut akan diambil data dengan nilai RMSE terbaik (terendah) untuk kemudian dianalisis dan dibahas pada langkah selanjutnya.

3.4.4 Analisis dan Pembahasan Hasil Penelitian

Setelah semua skenario dijalankan pada tahap pelaksanaan eksperimen, langkah selanjutnya yaitu analisis data yang sudah didapatkan dari hasil eksperimen pada langkah sebelumnya. Untuk mempermudah analisis data, penulis mengubah data dengan ekstensi awal txt menjadi sebuah file xlsx. Berikut gambar dari data mentah yang didapatkan dari hasil eksperimen dengan ekstensi txt.

```

1
2 Scenario 1
3 cabaiMerah,10,0,linear,0.8,30,30,adam,2691.143,6.174,0.792,9.352
4 cabaiMerah,10,0,linear,0.8,30,30,adam,2484.962,5.674,0.822,10.211
5 cabaiMerah,10,0,linear,0.8,30,30,adam,2048.553,4.214,0.879,10.72
6 cabaiMerah,20,0,linear,0.8,30,30,adam,1989.845,4.092,0.886,11.583
7 cabaiMerah,20,0,linear,0.8,30,30,adam,1945.118,4.043,0.891,11.677
8 cabaiMerah,20,0,linear,0.8,30,30,adam,2792.674,6.503,0.776,12.306
9 cabaiMerah,30,0,linear,0.8,30,30,adam,1735.356,3.397,0.913,12.116
10 cabaiMerah,30,0,linear,0.8,30,30,adam,2228.707,4.862,0.857,12.024
11 cabaiMerah,30,0,linear,0.8,30,30,adam,1922.79,3.816,0.894,11.968
12 cabaiMerah,40,0,linear,0.8,30,30,adam,1892.984,3.791,0.897,12.698
13 cabaiMerah,40,0,linear,0.8,30,30,adam,2011.516,4.355,0.884,13.571
14 cabaiMerah,40,0,linear,0.8,30,30,adam,2022.559,4.144,0.882,13.603
15 cabaiMerah,50,0,linear,0.8,30,30,adam,2093.978,4.373,0.874,14.123
16 cabaiMerah,50,0,linear,0.8,30,30,adam,1909.9,4.606,0.895,14.343
17 cabaiMerah,50,0,linear,0.8,30,30,adam,1955.445,4.192,0.89,14.234
18 cabaiMerah,60,0,linear,0.8,30,30,adam,1738.994,3.436,0.913,14.808
19 cabaiMerah,60,0,linear,0.8,30,30,adam,1996.516,4.107,0.885,14.714
20 cabaiMerah,60,0,linear,0.8,30,30,adam,2123.467,5.301,0.87,15.371
21 cabaiMerah,70,0,linear,0.8,30,30,adam,1786.228,3.639,0.908,18.785
22 cabaiMerah,70,0,linear,0.8,30,30,adam,1817.668,3.918,0.905,18.406
23 cabaiMerah,70,0,linear,0.8,30,30,adam,1729.073,3.672,0.914,18.527
24 cabaiMerah,80,0,linear,0.8,30,30,adam,1725.547,3.624,0.914,20.898
25 cabaiMerah,80,0,linear,0.8,30,30,adam,2245.113,5.757,0.855,20.079
26 cabaiMerah,80,0,linear,0.8,30,30,adam,1939.841,4.361,0.892,19.679
27 cabaiMerah,90,0,linear,0.8,30,30,adam,2061.691,4.331,0.878,21.724
28 cabaiMerah,90,0,linear,0.8,30,30,adam,1785.8,3.699,0.908,20.777
29 cabaiMerah,90,0,linear,0.8,30,30,adam,1631.42,3.147,0.923,20.327
30 cabaiMerah,100,0,linear,0.8,30,30,adam,1818.077,3.851,0.905,21.831
31 cabaiMerah,100,0,linear,0.8,30,30,adam,1699.323,3.336,0.917,23.574
32 cabaiMerah,100,0,linear,0.8,30,30,adam,1822.546,3.738,0.904,23.246

```

Gambar 3.9 Data Hasil Eksperimen dengan Ekstensi txt

Pada Gambar 3.9 terdapat beberapa kolom data yang dipisahkan dengan koma (,), hal tersebut digunakan untuk mempermudah pemindahan data dari txt menjadi bentuk tabulasi ke Microsoft Excel. Variabel-variabel yang dicatat pada pelaksanaan eksperimen tersebut adalah jenis cabai, jumlah unit, jumlah lapisan tersembunyi, fungsi aktivasi, rasio data latih, panjang sekuens, epoch, optimasi model, RMSE, MAPE, R2, dan durasi latih model komputasi. Berikut gambar data mentah yang dirubah bentuk menjadi tabular.

Scenario	Units	Hidden Layer	Activation Function	Train Data Ratio	Sequence Length	Epoch	Optimizer	RMSE	MAPE	R2	Training Duration
cabalaMerah	10	0	linear	0.8	30	30	adam	2691.143	6.174 %	0.792	9.352 detik
cabalaMerah	10	0	linear	0.8	30	30	adam	2484.962	5.674 %	0.822	10.211 detik
cabalaMerah	10	0	linear	0.8	30	30	adam	2048.553	4.214 %	0.879	10.720 detik
cabalaMerah	20	0	linear	0.8	30	30	adam	1989.845	4.092 %	0.886	11.583 detik
cabalaMerah	20	0	linear	0.8	30	30	adam	1945.118	4.043 %	0.891	11.677 detik
cabalaMerah	20	0	linear	0.8	30	30	adam	2792.674	6.503 %	0.776	12.206 detik
cabalaMerah	30	0	linear	0.8	30	30	adam	1735.356	3.397 %	0.913	12.116 detik
cabalaMerah	30	0	linear	0.8	30	30	adam	2228.707	4.862 %	0.857	12.024 detik
cabalaMerah	30	0	linear	0.8	30	30	adam	1922.739	3.816 %	0.894	11.968 detik
cabalaMerah	40	0	linear	0.8	30	30	adam	1892.984	3.791 %	0.897	12.698 detik
cabalaMerah	40	0	linear	0.8	30	30	adam	2011.516	4.355 %	0.884	13.571 detik
cabalaMerah	40	0	linear	0.8	30	30	adam	2022.559	4.144 %	0.882	13.603 detik
cabalaMerah	50	0	linear	0.8	30	30	adam	2093.978	4.373 %	0.874	14.123 detik
cabalaMerah	50	0	linear	0.8	30	30	adam	1909.9	4.606 %	0.895	14.343 detik
cabalaMerah	50	0	linear	0.8	30	30	adam	1955.845	4.193 %	0.893	14.234 detik
cabalaMerah	60	0	linear	0.8	30	30	adam	1738.994	3.436 %	0.913	14.808 detik
cabalaMerah	60	0	linear	0.8	30	30	adam	1996.516	4.107 %	0.885	14.714 detik
cabalaMerah	60	0	linear	0.8	30	30	adam	2123.467	5.303 %	0.857	15.211 detik
cabalaMerah	70	0	linear	0.8	30	30	adam	1786.228	3.639 %	0.908	18.785 detik
cabalaMerah	70	0	linear	0.8	30	30	adam	1817.668	3.918 %	0.905	18.406 detik
cabalaMerah	70	0	linear	0.8	30	30	adam	1729.073	3.672 %	0.914	18.537 detik
cabalaMerah	80	0	linear	0.8	30	30	adam	1725.547	3.624 %	0.914	20.898 detik
cabalaMerah	80	0	linear	0.8	30	30	adam	2245.113	5.757 %	0.855	20.079 detik
cabalaMerah	80	0	linear	0.8	30	30	adam	1959.841	4.363 %	0.892	19.679 detik
cabalaMerah	90	0	linear	0.8	30	30	adam	2061.691	4.331 %	0.878	21.724 detik
cabalaMerah	90	0	linear	0.8	30	30	adam	1785.8	3.669 %	0.908	20.777 detik
cabalaMerah	90	0	linear	0.8	30	30	adam	1631.42	3.431 %	0.923	20.337 detik
cabalaMerah	100	0	linear	0.8	30	30	adam	1818.077	3.851 %	0.905	21.831 detik
cabalaMerah	100	0	linear	0.8	30	30	adam	1699.323	3.336 %	0.917	16.993 detik
cabalaMerah	100	0	linear	0.8	30	30	adam	1699.323	3.336 %	0.917	23.574 detik

Gambar 3.10 Data Hasil Eksperimen dengan Bentuk Tabular

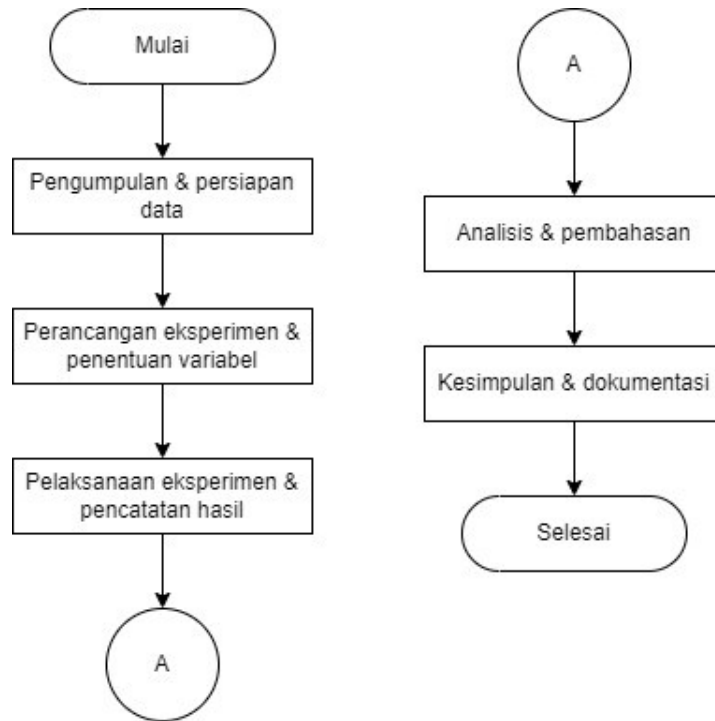
Berdasarkan Gambar 3.10 terdapat 3 data untuk setiap variabel bebas yang diuji. Dari ketiga data tersebut akan diambil data dengan RMSE terkecil dan dilakukan perhitungan rata-rata untuk RMSE dari 3 data yang dihasilkan setiap dijalankannya sebuah skenario. Pengujian variabel dengan nilai rata-rata RMSE terbagus (terkecil) diambil untuk kemudian dilakukan analisa lebih lanjut.

3.4.5 Penarikan Kesimpulan dan Dokumentasi

Langkah terakhir yaitu penarikan kesimpulan berdasarkan analisis dari data hasil eksperimen yang sudah dilakukan. Kesimpulan yang ditarik akan menjadi informasi yang dapat digunakan untuk berbagai macam kepentingan. Dokumentasi dimaksudkan agar apa yang sudah dilakukan dari awal hingga akhir memiliki catatan yang jelas dan terstruktur sehingga dapat menjadi referensi bagi penelitian-penelitian selanjutnya.

3.5 Diagram Alir Penelitian

Adapun diagram alir untuk penelitian ini yaitu sebagai berikut.



Gambar 3.11 Diagram Alir Penelitian

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil dan Pembahasan

Eksperimen dijalankan dengan beberapa skenario. Setiap skenario memiliki variabel bebas dan variabel kontrol masing-masing. Sedangkan variabel terikat sama untuk semua skenario yaitu RMSE hasil pengujian model komputasi, durasi pelatihan, dan selisih antara harga sebenarnya dengan harga yang diprediksi. Setiap variabel bebas yang diuji akan dijalankan tiga kali dan hasil dengan nilai RMSE terendah dari masing-masing variabel bebas yang diuji dipilih menjadi hasil terbaik untuk variabel tersebut. Berikut hasil dan pembahasan dari masing-masing skenario.

Tiga skenario pertama menguji arsitektur model komputasi yang digunakan. Berikut hasil dari tiga skenario pertama dari skenario 1 hingga 3.

Tabel 4.1 Hasil Eksperimen Skenario 1 (Jumlah Unit)

No.	Jenis cabai	Jumlah Unit	RMSE	MAPE	Durasi Latih	Rata-rata RMSE
1.	cabaiMerah	10	2048.553	4.21 %	10.720 detik	2408.219
2.	cabaiMerah	20	1945.118	4.04 %	11.677 detik	2242.546
3.	cabaiMerah	30	1735.356	3.40 %	12.116 detik	1962.284
4.	cabaiMerah	40	1892.984	3.79 %	12.698 detik	1975.686
5.	cabaiMerah	50	1909.9	4.61 %	14.343 detik	1986.441
6.	cabaiMerah	60	1738.994	3.44 %	14.808 detik	1952.992
7.	cabaiMerah	70	1729.07	3.67 %	18.527 detik	1777.656
8.	cabaiMerah	80	1725.547	3.62 %	20.898 detik	1970.167
9.	cabaiMerah	90	1631.42	3.15 %	20.327 detik	1826.304
10.	cabaiMerah	100	1699.323	3.34 %	23.574 detik	1779.982
11.	cabaiRawit	10	1877.101	3.44 %	8.311 detik	2265.177
12.	cabaiRawit	20	1835.567	3.55 %	10.016 detik	1997.059
13.	cabaiRawit	30	1864.518	3.51 %	11.033 detik	2078.377
14.	cabaiRawit	40	1914.028	3.58 %	12.306 detik	2082.391
15.	cabaiRawit	50	1833.49	3.43 %	13.434 detik	2025.368
16.	cabaiRawit	60	2019.487	3.92 %	15.176 detik	2192.937

17.	cabaiRawit	70	1941.592	3.63 %	18.101 detik	2078.805
18.	cabaiRawit	80	1913.82	3.59 %	19.936 detik	1969.866
19.	cabaiRawit	90	1940.052	3.71 %	20.049 detik	2036.716
20.	cabaiRawit	100	1930.288	3.74 %	21.918 detik	1977.832

Tabel 4.1 merupakan hasil eksperimen jumlah unit pada sel LSTM dengan beberapa nilai yaitu 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100 dengan beberapa variabel yang diamati yaitu RMSE, MAPE, dan durasi latih. Pada data cabai merah nilai RMSE yang dihasilkan cenderung menurun walaupun terdapat sedikit fluktuasi pada pengujian jumlah unit 30, 40, dan 50 dimana nilai RMSE sempat mengalami kenaikan dari pengujian jumlah unit 30 ke 40 dan mengalami kenaikan lagi dari pengujian jumlah unit 40 ke 50. Nilai RMSE terkecil yang dapat dihasilkan dari data cabai merah yaitu 1631,42 pada pengujian jumlah unit 90, namun nilai rata-rata RMSE terkecil dihasilkan pada pengujian unit 70 yaitu dengan nilai rata-rata RMSE 1777,656. Pada data cabai rawit tren RMSE cenderung naik walaupun sempat turun pada beberapa pengujian unit. Nilai RMSE terkecil dihasilkan pada pengujian jumlah unit 50 dengan nilai sebesar 1833,49 namun nilai rata-rata RMSE terkecil dihasilkan pada saat pengujian unit 80 yaitu dengan nilai 1969,866. Berdasarkan Tabel 4.1, rata-rata RMSE terkecil dihasilkan pada pengujian unit yang berbeda, pada data cabai merah dihasilkan rata-rata RMSE terkecil pada pengujian unit 70 sedangkan cabai rawit dihasilkan pada pengujian unit 80.

Pada tabel 4.1 terlihat bahwa nilai MAPE yang dihasilkan memiliki tren yang kurang lebih sama seperti tren nilai RMSE. Nilai MAPE yang lebih kecil menunjukkan bahwa hasil prediksi model komputasi memiliki *error* yang kecil. Sama seperti RMSE ketika lebih kecil menunjukkan selisih antara nilai prediksi dan aktualnya tidak jauh berbeda. Pada data cabai rawit nilai MAPE terkecil dihasilkan pada pengujian unit 90 yaitu 3.15% yang mana pada pengujian tersebut juga dihasilkan nilai RMSE terkecil. Sama seperti cabai merah, nilai MAPE terkecil cabai rawit dihasilkan pada saat pengujian 50 dengan nilai 3.43% yang juga menghasilkan nilai RMSE terkecil.

Pada kolom durasi latih, nilai yang dihasilkan memiliki kecenderungan naik seiring bertambahnya jumlah unit yang digunakan. Hal tersebut sesuai dengan ekspektasi dimana bertambahnya unit dalam sebuah lapisan akan membuat model komputasi memiliki kinerja yang lebih berat dalam komputasinya. Pada data cabai merah nilai RMSE memiliki tren menurun seiring bertambahnya unit yang diuji begitupun dengan nilai rata-rata RMSE-nya sedangkan durasi latihnya naik. Berbeda dengan cabai merah, cabai rawit justru menghasilkan nilai RMSE dengan tren yang cenderung naik dan durasi latihnya juga ikut naik seiring bertambahnya jumlah unit yang diuji namun nilai rata-rata RMSE yang dihasilkan cenderung menurun namun tidak terlalu signifikan. Hal tersebut menunjukkan bahwa penambahan jumlah unit akan mempengaruhi durasi latih dari model komputasi. Semakin banyak unit yang digunakan maka durasi latihnya akan semakin lama.

Tabel 4.2 Hasil Eksperimen Skenario 2 (Jumlah Lapisan Tersembunyi)

No.	Jenis cabai	Jumlah Lapisan Tersembunyi	RMSE	MAPE	Durasi Latih	Rata-rata RMSE
1.	Cabai Merah	0	1663.54	3.21 %	13.744 detik	1781.956
2.	Cabai Merah	1	1752.894	3.82 %	25.969 detik	1890.522
3.	Cabai Merah	2	1982.497	3.98 %	38.073 detik	2243.815
4.	Cabai Merah	3	2043.626	4.25 %	50.806 detik	2398.822
5.	Cabai Rawit	0	1929	3.55 %	11.222 detik	2099.155
6.	Cabai Rawit	1	2182.131	4.34 %	22.820 detik	2270.126
7.	Cabai Rawit	2	2165.635	4.34 %	35.736 detik	2241.131
8.	Cabai Rawit	3	2403.779	4.96 %	48.134 detik	2622.612

Pada Tabel 4.2 menunjukkan hasil skenario 2 yang menguji jumlah lapisan tersembunyi yang ditambahkan pada arsitektur model komputasi. Jenis sel yang ditambahkan sebagai lapisan tersembunyi yaitu sel LSTM dengan jumlah unit 50 untuk masing-masing jumlah lapisan tersembunyi yang ditambahkan. Pada data cabai merah, nilai RMSE yang dihasilkan menunjukkan tren naik dari pengujian jumlah lapisan tersembunyi 0 hingga 3 dengan nilai RMSE terkecil yang didapat pada saat pengujian 0 lapisan

tersembunyi yaitu 1663,543. Pada saat pengujian data cabai rawit juga menunjukkan hasil yang sama yakni nilai RMSE terkecil dihasilkan pada saat pengujian 0 lapisan tersembunyi yaitu 1929. Hal yang sama juga ditunjukkan pada kolom MAPE dimana MAPE terkecil dihasilkan pada pengujian lapisan tersembunyi yang menghasilkan nilai RMSE terkecil yaitu pada saat pengujian 0 lapisan tersembunyi dengan nilai MAPE masing-masing 3.21% untuk cabai merah dan 3.55% untuk cabai rawit.

Pada kolom durasi latih untuk kedua jenis data harga cabai menunjukkan tren yang sama yaitu naik seiring bertambahnya jumlah lapisan tersembunyi yang ditambahkan. Hal tersebut sesuai dengan ekspektasi karena semakin kompleks sebuah arsitektur model komputasi maka membutuhkan waktu yang lebih lama dalam tahap pelatihannya. Namun, durasi latih yang panjang tidak selalu menghasilkan nilai RMSE atau rata-rata RMSE yang bagus. Pada skenario 2, nilai RMSE terkecil justru dihasilkan pada saat pengujian 0 lapisan tersembunyi yang membutuhkan waktu latih yang paling singkat yaitu 13,744 detik pada data cabai merah dan 11,222 detik pada data cabai rawit.

Tabel 4.3 Hasil Eksperimen Skenario 3 (Fungsi Aktivasi)

No.	Jenis cabai	Fungsi Aktivasi	RMSE	MAPE	Durasi latih	Rata-rata RMSE
1.	Cabai Merah	linear	1861.34	3.90 %	14.337 detik	1953.674
2.	Cabai Merah	relu	1654.338	3.17 %	15.558 detik	8344.300
3.	Cabai Merah	leaky_relu	1838.108	3.71 %	15.532 detik	2019.435
4.	Cabai Merah	selu	1960.595	4.49 %	11.659 detik	2041.904
5.	Cabai Rawit	linear	2056.25	4.01 %	14.694 detik	2104.328
6.	Cabai Rawit	relu	1886.427	3.58 %	13.839 detik	10344.221
7.	Cabai Rawit	leaky_relu	1982.533	3.90 %	13.757 detik	2135.068
8.	Cabai Rawit	selu	2033.085	3.83 %	14.394 detik	2129.992

Tabel 4.3 menunjukkan hasil eksperimen beberapa fungsi aktivasi yang digunakan pada model komputasi. Terdapat empat fungsi aktivasi yang digunakan yaitu Linear, Relu, Leaky Relu, dan SELU keempat fungsi tersebut merupakan fungsi aktivasi dengan jenis linear. Pada data cabai merah nilai RMSE terkecil dihasilkan pada saat pengujian fungsi aktivasi Relu namun,

pada pengujian tersebut fungsi aktivasi Relu menghasilkan nilai rata-rata RMSE terbesar. Hal tersebut disebabkan adanya nilai RMSE yang sangat besar pada salah satu dari 3 percobaan pengujian fungsi aktivasi Relu yaitu dengan nilai 21252,775 dapat dilihat pada Gambar 4.5 atau Gambar 4.6. Hal yang sama juga terjadi pada pengujian data cabai rawit dimana nilai RMSE terkecil dihasilkan pada saat pengujian fungsi aktivasi Relu namun dengan nilai rata-rata RMSE terbesar. Untuk nilai rata-rata RMSE terkecil dihasilkan pada saat pengujian fungsi aktivasi linear yaitu 1953,674 untuk cabai merah dan 2104,328 untuk cabai rawit. Nilai MAPE yang dihasilkan juga mengikuti tren dari RMSE dimana nilai RMSE terkecil menghasilkan nilai MAPE terkecil juga.

Durasi latih yang dibutuhkan model komputasi untuk kedua jenis data harga cabai pada masing-masing fungsi aktivasi tidak jauh berbeda yaitu pada interval 13 hingga 15 detik sesuai dengan Tabel 4.3. Hal ini menandakan bahwa perbedaan penggunaan fungsi aktivasi tidak terlalu berpengaruh terhadap durasi pelatihan model komputasi. Namun, tren yang dihasilkan durasi latih pada cabai merah dan cabai rawit berbeda, dimana cabai merah menghasilkan tren naik kemudian turun sedangkan cabai rawit menghasilkan tren turun kemudian naik. Durasi latih tercepat pada cabai merah yaitu 11,659 detik saat pengujian fungsi aktivasi SELU, sedangkan cabai rawit membutuhkan durasi latih tercepat pada saat pengujian fungsi aktivasi Leaku Relu yaitu 13,757 detik. Namun kedua fungsi aktivasi tersebut tidak menghasilkan nilai rata-rata RMSE yang lebih baik pada saat pengujian fungsi aktivasi linear dengan durasi latih yang tidak jauh berbeda fungsi aktivasi linear dapat dipilih sebagai fungsi aktivasi untuk model komputasi.

Empat skenario terakhir atau skenario 4 hingga 7 menguji beberapa parameter yang berkaitan dengan persiapan data, pelatihan dan optimasi model. Berikut hasil dari eksperimen skenario 4 hingga 7.

Tabel 4.4 Hasil Eksperimen Skenario 4 (Rasio Data Latih)

No.	Jenis cabai	Rasio data	RMSE	MAPE	Durasi latih	Rata-rata RMSE
-----	-------------	------------	------	------	--------------	----------------

1.	Cabai Merah	0,7	2198.103	3.840 %	12.229 detik	2571.952
2.	Cabai Merah	0,8	1710	3.519 %	10.886 detik	1975.861
3.	Cabai Merah	0,9	2125.779	3.946 %	12.643 detik	2239.434
4.	Cabai Rawit	0,7	2143.658	3.578 %	12.164 detik	2257.019
5.	Cabai Rawit	0,8	2061.814	4.098 %	14.020 detik	2338.824
6.	Cabai Rawit	0,9	1941.93	3.384 %	15.183 detik	2067.900

Pada Tabel 4.4 menunjukkan hasil eksperimen terhadap perbedaan rasio antara data latih dengan data uji. Rasio data yang digunakan yaitu dengan nilai 0,7, 0,8, dan 0,9. Berdasarkan Tabel 4.1 nilai RMSE menurun dari rasio data yang kecil (0,7) menuju rasio data yang lebih besar (0,9) untuk data cabai rawit. Sedangkan hasil RMSE yang didapat pada data cabai merah cenderung bervariasi jika dibandingkan dengan data cabai rawit. Pada data cabai rawit RMSE menurun dengan beda yang hampir sama yakni diangka 100. Rasio data yang lebih besar dapat menghasilkan RMSE yang kecil karena dengan rasio data yang besar artinya porsi data latih dengan data uji lebih besar data latih. Sehingga model komputasi memiliki data yang lebih untuk dipelajari yang akan berpengaruh terhadap tahap pengujian. Model komputasi akan menghasilkan kualitas prediksi yang bagus jika diberi data latih yang lebih banyak, kualitas tersebut ditandai dengan semakin kecilnya nilai RMSE. Nilai RMSE yang kecil menandakan bahwa rata-rata selisih antara nilai yang diprediksi dengan nilai sebenarnya tidak terlalu jauh. Tren yang sama juga ditunjukkan pada kolom rata-rata RMSE untuk masing-masing data cabai. Nilai rata-rata RMSE terkecil untuk masing-masing data cabai yaitu 1975,861 untuk cabai merah dan 2067,900 untuk cabai rawit. Nilai RMSE terendah tersebut didapatkan pada pengujian rasio data latih yang berbeda, untuk data cabai merah menghasilkan RMSE terkecil pada rasio data latih 0,8 dan cabai rawit pada rasio data latih 0,9. Perbedaan tersebut dapat disebabkan oleh kompleksitas data.

Durasi latih untuk kedua jenis data cabai menunjukkan tren yang berbeda yang mana cabai merah menunjukkan durasi latih tercepat pada saat pengujian rasio data latih 0,8 yaitu 10,886 detik dan naik pada saat pengujian

data latih 0,9. Sedangkan cabai rawit menghasilkan durasi latih yang cenderung naik dari pengujian rasio 0,7 hingga 0,9 dengan durasi latih tercepat yaitu 12,164 detik pada saat pengujian rasio 0,7 namun, pada pengujian tersebut nilai rata-rata RMSE yang dihasilkan tidak lebih kecil daripada nilai rata-rata RMSE yang dihasilkan pada saat pengujian rasio data 0,9.

Tabel 4.5 Hasil Eksperimen Skenario 5 (Panjang Sekuens)

No.	Jenis cabai	Panjang sekuens	RMSE	MAPE	Durasi latih	Rata-rata RMSE
1.	Cabai Merah	7	1932.67	3.928 %	5.431 detik	1923.455
2.	Cabai Merah	14	2113.371	4.361 %	7.031 detik	2187.439
3.	Cabai Merah	21	1924.29	4.109 %	9.186 detik	1998.475
4.	Cabai Merah	30	1876.736	3.947 %	11.963 detik	1980.995
5.	Cabai Rawit	7	1936.94	3.485 %	6.531 detik	2061.045
6.	Cabai Rawit	14	2084.739	4.016 %	8.691 detik	2206.045
7.	Cabai Rawit	21	1936.028	3.684 %	10.670 detik	2069.589
8.	Cabai Rawit	30	2009.727	3.800 %	13.764 detik	2121.847

Tabel 4.5 menunjukkan hasil dari eksperimen skenario 5. Pada skenario ini yang menjadi variabel bebasnya yaitu panjang sekuens dengan nilai 7, 14, 21, dan 30. Berdasarkan Tabel 4.5 nilai RMSE yang didapat bervariasi terhadap panjang sekuens yang diberikan. Nilai RMSE terkecil dari data cabai merah yaitu di angka 1876,736 dengan panjang sekuens 30. Pada data cabai rawit nilai RMSE terkecilnya di angka 1936,028 dengan panjang sekuens yang diberikan yaitu 21. Namun, rata-rata RMSE terkecil dihasilkan ketika pengujian panjang sekuens 7 untuk kedua jenis cabai yaitu 1923,455 untuk cabai merah dan 2061,045 untuk cabai rawit. Hal tersebut menunjukkan bahwa pengujian panjang sekuens 7 menghasilkan nilai yang tidak terlalu fluktuatif dan lebih stabil walaupun nilai RMSE terkecilnya tidak lebih kecil ketika pengujian panjang sekuens lainnya. Pola yang sama ditunjukkan oleh MAPE dimana RMSE terkecil maka nilai MAPE-nya juga kecil.

Pada kolom durasi latih Tabel 4.5 menunjukkan tren kenaikan durasi latih seiring bertambahnya panjang sekuens. Durasi latih tercepat pada data

cabai merah yaitu dihasilkan pada saat pengujian panjang sekuens 7 yaitu selama 5,431 detik. Hal yang sama juga ditunjukkan data cabai rawit dimana durasi latih tercepat dihasilkan pada pengujian panjang sekuens 7 yaitu selama 6,531 detik. Hal tersebut dapat terjadi sebab ketika pelatihan panjang sekuens yang lebih kecil membutuhkan proses pengkalian bobot yang sedikit per-epochnya sehingga waktu yang digunakan juga lebih sedikit.

Tabel 4.6 Hasil Eksperimen Skenario 6 (Epoch)

No.	Jenis cabai	Epoch	RMSE	MAPE	Durasi latih	Rata-rata RMSE
1.	Cabai Merah	10	3063.971	7.313 %	4.587 detik	3263.929
2.	Cabai Merah	20	1780.143	3.488 %	7.766 detik	1923.765
3.	Cabai Merah	30	1892.578	3.796 %	11.470 detik	2165.167
4.	Cabai Merah	40	1674.98	3.299 %	19.099 detik	1791.318
5.	Cabai Merah	50	1739.35	3.516 %	19.127 detik	1754.700
6.	Cabai Rawit	10	2537.488	4.855 %	4.941 detik	2793.263
7.	Cabai Rawit	20	2100.658	4.030 %	9.113 detik	2398.242
8.	Cabai Rawit	30	1983.932	3.826 %	14.259 detik	2257.132
9.	Cabai Rawit	40	1767.957	3.396 %	18.709 detik	1905.287
10.	Cabai Rawit	50	1837.71	3.419 %	22.351 detik	1878.329

Pada tabel 4.6 menunjukkan hasil eksperimen skenario 6, dimana pada skenario tersebut yang menjadi variabel bebasnya yaitu jumlah epoch yang dijalankan. Dengan nilai dari epoch yang diuji yaitu 10, 20, 30, 40, dan 50. Jika berdasarkan Tabel 4.6, nilai RMSE menunjukkan tren yang menurun seiring bertambahnya epoch yang dijalankan pada kedua jenis cabai. Sehingga nilai rata-rata RMSE terkecil dihasilkan pada saat pengujian epoch 50 pada kedua jenis cabai, 1754,700 untuk cabai merah dan 1878,329 untuk cabai rawit. Hal tersebut disebabkan oleh banyaknya epoch yang ditentukan untuk model komputasi maka semakin banyak waktu yang dibutuhkan model komputasi untuk mempelajari data latih sehingga dapat memberikan nilai rata-rata RMSE yang kecil pada saat tahap pengujian model komputasi. Tren yang sama ditunjukkan oleh nilai MAPE yang dihasilkan yaitu menurun mengikuti nilai RMSE-nya.

Berdasarkan Tabel 4.6, durasi latih mengalami peningkatan seiring bertambahnya epoch. Hal ini sesuai dengan ekspektasi dimana semakin banyak epoch maka semakin lama durasi latih dari sebuah model komputasi. Pada data cabai merah durasi latih terpanjang yaitu pada epoch 50 dengan durasi latih 19,127 detik namun, menghasilkan nilai rata-rata RMSE terkecil. Sama seperti data cabai rawit dimana durasi latih terpanjang menghasilkan nilai rata-rata RMSE terkecil dengan lama durasi latih 22,351 detik. Hal ini berarti model komputasi membutuhkan durasi latih yang lebih lama untuk menghasilkan prediksi yang lebih akurat.

Tabel 4.7 Hasil Eksperimen Skenario 7 (Optimasi Model)

No.	Jenis cabai	Optimasi model	RMSE	MAPE	Durasi latih	Rata-rata RMSE
1.	Cabai Merah	adam	1854.52	4.119 %	14.505 detik	2090.499
2.	Cabai Merah	rmsprop	1635.07	3.158 %	14.777 detik	1931.978
3.	Cabai Merah	sgd	2949.765	7.227 %	15.165 detik	3064.347
4.	Cabai Rawit	adam	1867.45	3.551 %	16.058 detik	2020.201
5.	Cabai Rawit	rmsprop	1834.37	3.423 %	15.001 detik	1983.437
6.	Cabai Rawit	sgd	3118.195	6.644 %	14.176 detik	3343.540

Pada Tabel 4.7 menunjukkan hasil eksperimen skenario 7 dengan variabel bebas yaitu jenis optimasi model yang digunakan. Terdapat tiga jenis optimasi model yang digunakan yaitu Adam, RMSprop, dan SGD. Pada data cabai merah, optimasi model RMSprop menghasilkan nilai RMSE terkecil dengan nilai 1635,065 dibandingkan optimasi model lain yang diuji. Begitu juga nilai rata-rata RMSE yang dihasilkan pada saat pengujian optimasi model RMSprop menghasilkan nilai terkecil yaitu 1931,97. Pada data cabai rawit nilai RMSE terkecil yang dapat dihasilkan yaitu 1834,365 dengan menggunakan optimasi model RMSprop dan nilai rata-rata RMSE yang dihasilkan juga terkecil dibandingkan pengujian optimasi model lain yang diuji yaitu 1983,43. Data yang dilatih dengan optimasi model SGD memberikan nilai rata-rata RMSE terbesar diantara optimasi model lain yang diuji baik pada data cabai merah maupun pada data cabai rawit dengan nilai RMSE di kisaran 3000-an. Model komputasi dengan optimasi model

RMSprop memiliki performa prediksi yang lebih baik dan akurat ditandai dengan nilai RMSE yang kecil untuk kedua jenis cabai. Sedangkan optimasi model SGD kurang bisa memberikan hasil yang baik dibandingkan dua optimasi model lainnya, sedangkan untuk optimasi Adam menghasilkan nilai rata-rata RMSE yang tidak terlalu jauh dibandingkan nilai rata-rata RMSE yang dihasilkan saat pengujian optimasi model RMSprop, sehingga optimasi model Adam masih mungkin dipakai namun, jika berdasarkan nilai rata-rata terkecil optimasi RMSprop masih unggul dibandingkan optimasi Adam.

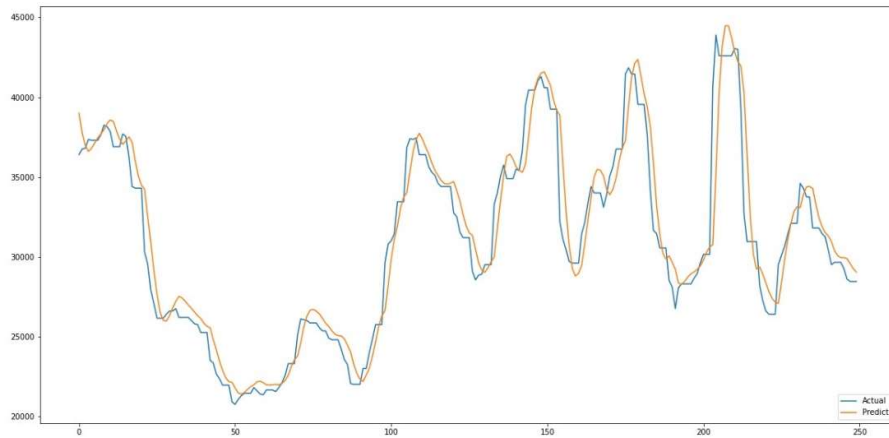
Berdasarkan Tabel 4.7, durasi latih yang ditempuh model komputasi pada data cabai merah memperlihatkan tren yang naik dari pengujian optimasi model Adam, RMSprop, dan SGD dengan durasi latih tercepat yaitu pada saat pengujian optimasi model Adam yaitu 14,505 detik. Namun, perbedaan antara masing-masing durasi latih pada pengujian optimasi model cabai merah tidak terlalu terlihat yaitu disekitar 0,2 hingga 0,4 detik saja. Sedangkan optimasi model Adam menghasilkan nilai rata-rata RMSE yang lebih tinggi dibandingkan pengujian optimasi model RMSprop dengan selisih durasi latih hanya sekitar 0,2 detik. Maka, optimasi model RMSprop lebih bijak digunakan karena dapat menghasilkan nilai RMSE yang lebih kecil meskipun durasi latih lebih lama dibandingkan optimasi model Adam. Pada data cabai rawit, tren durasi latih yang dihasilkan terbalik dibandingkan pada saat pengujian cabai merah yang mana durasi latihnya menurun pada saat pengujian optimasi model Adam, RMSprop, dan SGD. Optimasi model SGD menghasilkan durasi latih tercepat yaitu 14,176 detik yang mana berbeda dengan durasi latih tercepat pada data cabai merah yang dicapai oleh optimasi model Adam. Durasi latih terpanjang dicapai oleh optimasi model Adam yaitu selama 16,058 detik. Namun, nilai rata-rata RMSE terkecil dihasilkan saat pengujian optimasi model RMSprop

4.2 Analisis Skenario Terbaik

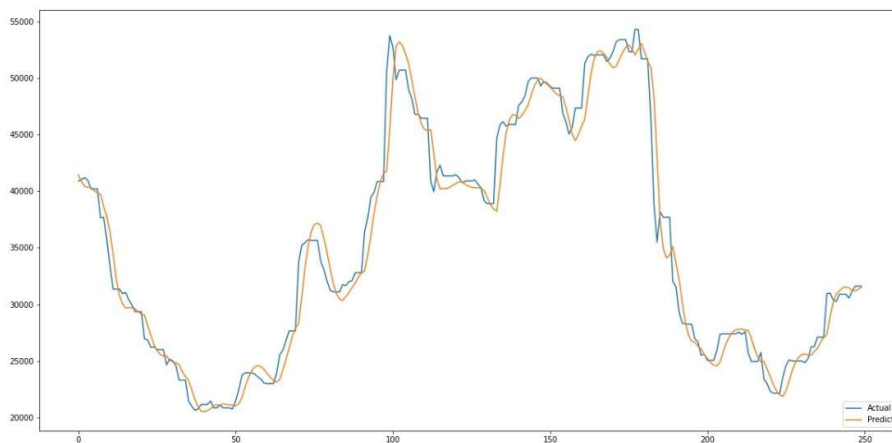
Setelah semua hasil eksperimen dianalisis didapatkan sebuah informasi yang dapat digunakan untuk membuat skenario terbaik berdasarkan beberapa

metrik penilaian seperti nilai RMSE dan durasi latih. Informasi tersebut kemudian akan di analisis kembali dan dilakukan pengujian berdasarkan skenario terbaik yang dapat dihasilkan. Pembahasan pertama akan membahas skenario 1 hingga 3 yang berfokus pada pengujian arsitektur dari model komputasi yang dibuat.

Skenario 1 menguji jumlah unit pada sel LSTM yang digunakan dalam komputasi. Berdasarkan pada Tabel 4.1 hasil eksperimen skenario 1 menunjukkan bahwa jumlah unit dengan nilai 70 menghasilkan nilai rata-rata RMSE terkecil pada data cabai merah yaitu 1777,656 dan jumlah unit 80 menghasilkan nilai rata-rata RMSE tekecil pada data cabai rawit yaitu 1969,866. Namun, durasi latih yang dibutuhkan relatif lebih lama diantara pengujian jumlah unit dibawah 70 dan 80 yaitu membutuhkan durasi 18,527 detik dan 19,936 detik. Pengujian jumlah unit 70 dan 80 memang membutuhkan durasi latih yang lama namun nilai rata-rata RMSE yang dihasilkan adalah yang terkecil diantara pengujian jumlah unit lainnya. Hal tersebut dapat menjadi pertimbangan untuk menggunakan jumlah unit 70 dan 80 meskipun memiliki durasi latih yang lebih lama karena selisih antara durasi latih tercepat hanya sekitar 8 hingga 11 detik, namun pengujian dengan unit 10 menghasilkan nilai rata-rata yang tidak sebgus pegujian unit 70 dan 80 pada cabai merah dan cabai rawit. Jadi, terdapat timbal balik jika kita menggunakan jumlah unit yang banyak akan memperlama durasi namun menghasilkan nilai rata-rata RMSE yang kecil. Namun, seiring bertambahnya unit juga serta merta nilai rata-rata RMSE yang dihasilkan juga akan mengecil, terdapat puncak minimum pada saat pengujian yang dilakukan penulis, berdasarkan Tabel 4.1 puncak minimum untuk data cabai merah yaitu dihasilkan pada saat pengujian jumlah unit 70 sedangkan data cabai rawit pada pengujian jumlah unit 80. Berikut grafik hasil pengujian jumlah unit 70 dan 80 dengan hasil nilai rata-rata RMSE terkecil pada data cabai merah dan cabai rawit secara berurutan.



Gambar 4.1 Grafik Hasil Pengujian 70 Unit pada Data Cabai Merah

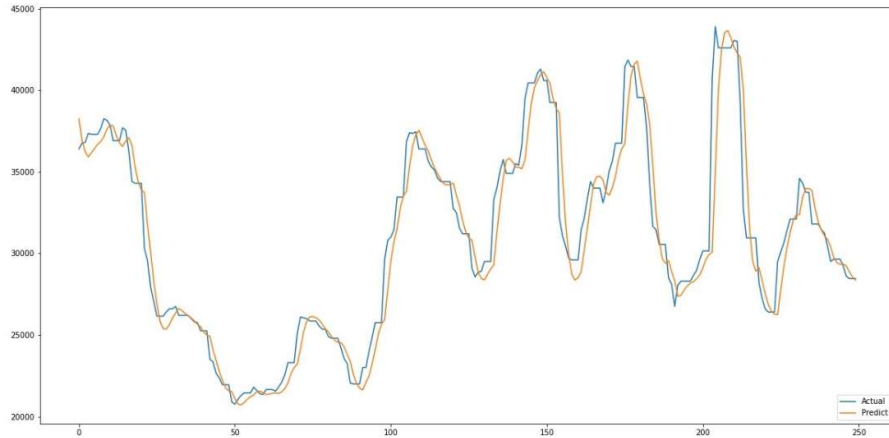


Gambar 4.2 Grafik Hasil Pengujian 80 Unit pada Data Cabai Rawit

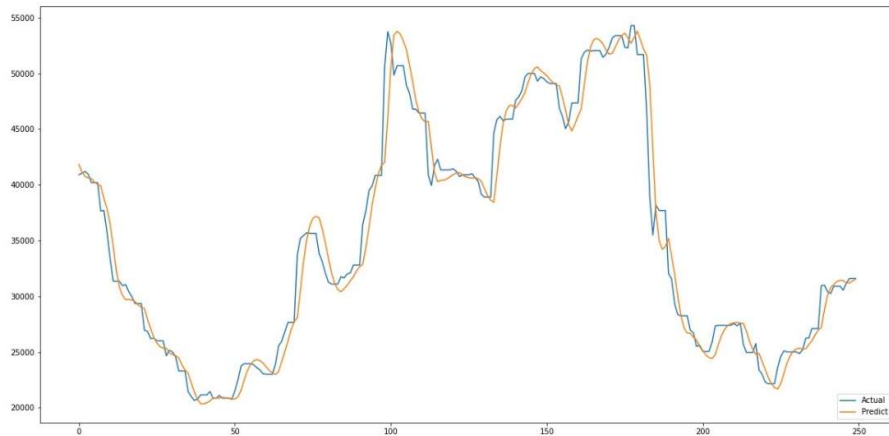
Pada Gambar 4.1 dan 4.2, dapat dilihat terdapat dua kurva dengan warna yang berbeda yaitu kurva biru, dan kurva jingga. Kurva biru menunjukkan grafik data sebenarnya (data aktual) sedangkan kurva jingga menunjukkan grafik data hasil prediksi. Grafik tersebut menunjukkan hubungan antara nilai yang diprediksi model komputasi dengan data uji. Semakin dekat kurva jingga dengan kurva biru menunjukkan hasil prediksi dari model komputasi memiliki kualitas yang bagus. Hal tersebut dapat ditandai dengan nilai RMSE yang semakin kecil ketika kedua kurva tersebut terlihat saling tumpang tindih. Pada Gambar 4.1 dan Gambar 4.2 terlihat bahwa model dapat menghasilkan

prediksi yang cukup baik ditandai dengan minimnya celah antara kurva biru dengan kurva jingga.

Pada skenario 2 menguji jumlah lapisan tersembunyi yang ditambahkan diantara lapisan input dan output. Pada Tabel 4.2 menunjukkan nilai rata-rata RMSE terkecil untuk data cabai merah yaitu 1781,956 pada pengujian 0 lapisan tersembunyi. Pada data cabai rawit, nilai RMSE terkecil juga dihasilkan pada saat pengujian 0 lapisan tersembunyi yaitu 2099,155. Berikut grafik hasil pengujian 0 lapisan tersembunyi untuk kedua jenis data harga cabai.



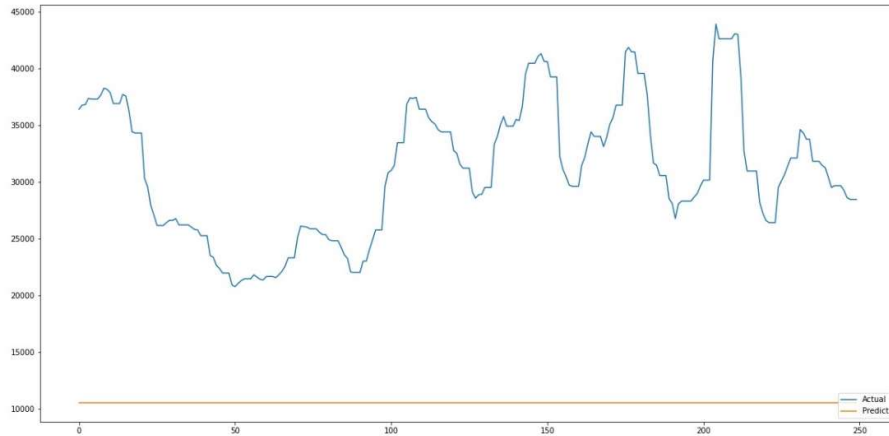
Gambar 4.3 Grafik Hasil Pengujian 0 Lapisan Tersembunyi pada Data Cabai Merah



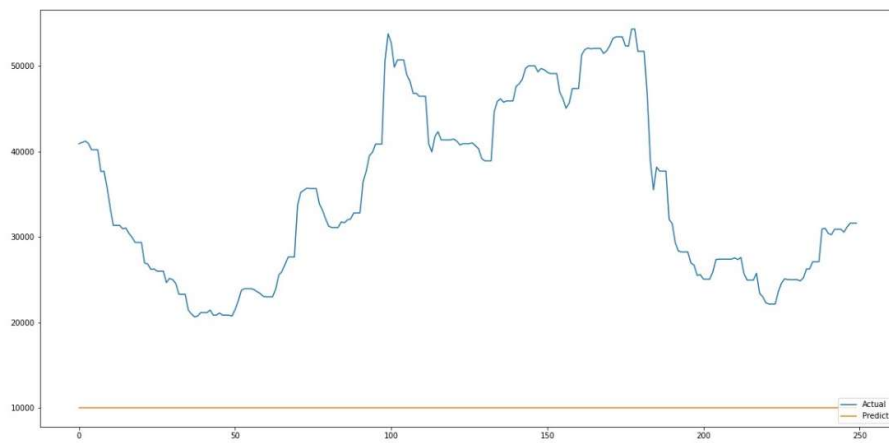
Gambar 4.4 Grafik Hasil Pengujian 0 Lapisan Tersembunyi pada Data Cabai Rawit

Berdasarkan Gambar 4.3 dan 4.4, data cabai merah memiliki hasil yang lebih baik yang dapat dilihat pada grafik yang dihasilkan terlihat kurva jingga dan biru memiliki celah yang lebih sedikit dan tidak terlalu lebar jika dibandingkan hasil data cabai rawit. Hal tersebut dapat ditandai juga dengan nilai rata-rata RMSE cabai merah (1781,956) lebih kecil dibandingkan dengan cabai rawit (2099,155).

Skenario ketiga menguji fungsi aktivasi yang digunakan dalam arsitektur model komputasi. Berdasarkan Tabel 4.3 nilai rata-rata RMSE yang dihasilkan masing-masing fungsi aktivasi tidak jauh berbeda dan memiliki selisih yang kecil satu sama lain baik untuk data cabai merah maupun data cabai rawit. Jadi, penggunaan fungsi aktivasi apapun tidak akan terlalu berpengaruh terhadap hasil dari pengujian model komputasi. Namun, penulis merekomendasikan fungsi aktivasi linier sebab berdasarkan Tabel 4.3, fungsi tersebut menghasilkan nilai RMSE terkecil yakni 1953,674 dan 2104,328. Sebaliknya, penulis tidak merekomendasikan menggunakan fungsi aktivasi Relu karena pada saat eksperimen fungsi aktivasi Relu cenderung menghasilkan prediksi yang sangat jauh dan nilai RMSE yang sangat tinggi. Berikut grafik hasil fungsi aktivasi Relu yang mengalami anomali pada kedua jenis data harga cabai.



Gambar 4.5 Anomali Hasil Pengujian Fungsi Aktivasi Relu pada Data Cabai Merah



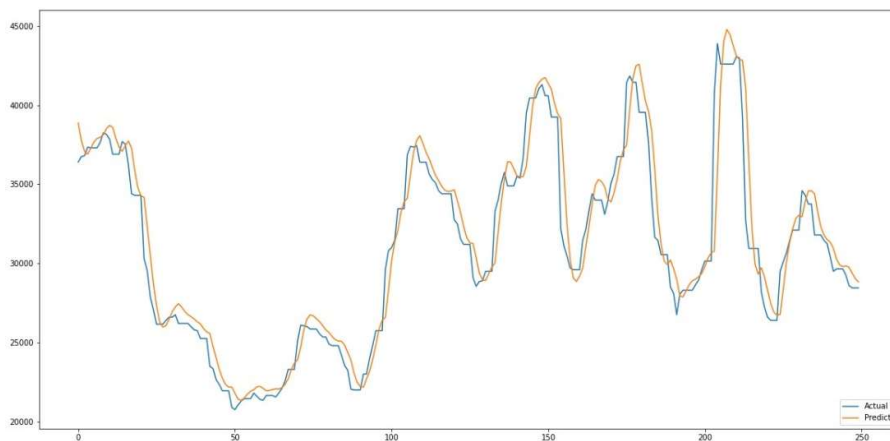
Gambar 4.6 Anomali Hasil Pengujian Fungsi Aktivasi Relu pada Data Cabai Rawit

Pada Gambar 4.5 maupun Gambar 4.6 menunjukkan bahwa kurva jingga berada jauh dibawah kurva biru bahkan berada pada data minimum dari dataset, hal tersebut juga ditandai dengan nilai RMSE yang sangat besar. Dugaan sementara mengenai hal tersebut karena fungsi aktivasi Relu itu sendiri, dimana fungsi tersebut akan mengembalikan nilai yang sama jika nilai tersebut positif dan sebaliknya jika nilai yang masuk tersebut negatif sampai 0 maka yang dikembalikan adalah nilai 0. Jadi, ketika tahap pelatihan model menghasilkan nilai negatif pada epoch pertama maka nilai yang dihasilkan yaitu 0 dan akan terus berlanjut pada epoch-epoch selanjutnya

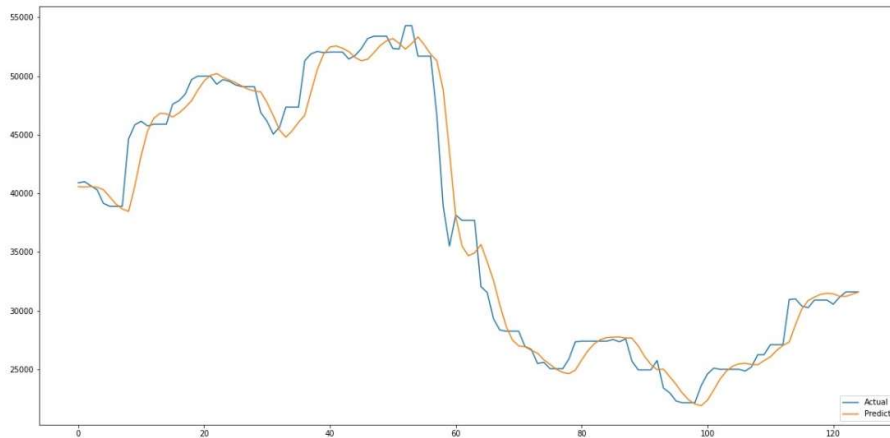
hingga selesai. Sehingga, bobot yang dilatih juga mengalami kekacauan yang akan berdampak pada tahap pengujian dan uji prediksi model komputasi. Meskipun tidak semua hasil pengujian fungsi aktivasi Relu mengalami anomali, alangkah baiknya menghindari penggunaan fungsi tersebut agar model komputasi lebih efisien.

Empat skenario terakhir yaitu menguji beberapa parameter yang berkaitan dengan persiapan data, pelatihan dan optimasi model komputasi. Berikut pembahasan mengenai skenario 4 hingga 7 untuk variabel terbaik yang diuji.

Skenario 4 menguji rasio data latih dengan data uji. Penulis menguji tiga variabel rasio data latih yaitu 70:30 (70%), 80:20 (80%), dan 90:10 (90%). Pada skenario 4, dua jenis data harga cabai menghasilkan nilai rata-rata RMSE terkecilnya pada pengujian variabel yang berbeda. Data cabai merah menghasilkan nilai rata-rata RMSE terkecil pada saat pengujian rasio data latih 80% dengan data uji yaitu diangka 1975,861. Sedangkan data cabai rawit menghasilkan nilai RMSE terbaik pada saat pengujian rasio data latih 90% yaitu 2067,900. Sehingga, pemilihan rasio data latih tidak serta merta tetap untuk semua jenis data. Berikut grafik hasil pengujian rasio data latih pada cabai merah dan cabai rawit.



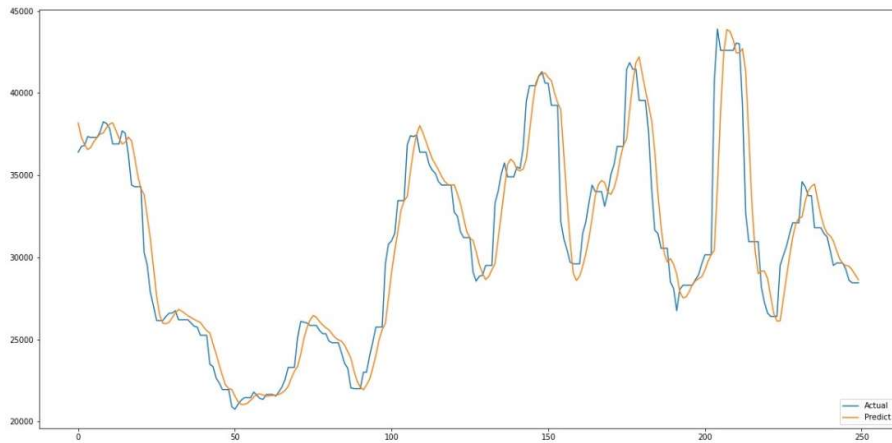
Gambar 4.7 Grafik Hasil Pengujian Rasio Data Latih 0.8 pada Data Cabai Merah



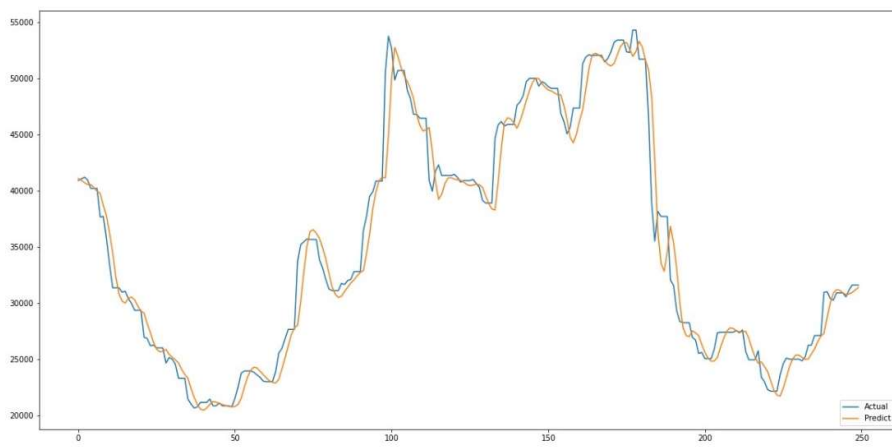
Gambar 4.8 Grafik Hasil Pengujian Rasio Data Latih 0.9 pada Data Cabai Rawit

Hasil yang cukup baik dihasilkan pada kedua jenis cabai. Hal tersebut dapat diamati pada Gambar 4.7 dan Gambar 4.8, namun hasil pengujian pada data cabai merah memiliki hasil yang lebih baik dibandingkan dengan data cabai rawit. Hal tersebut dapat dibuktikan dengan kurva jingga pada data Gambar 4.7 terlihat lebih berimpit dengan kurva biru dibandingkan dengan kurva jingga pada Gambar 4.8. Selain itu, hal tersebut dapat terlihat dari nilai RMSE terkecil yang dihasilkan pada pengujian rasio data latih untuk data cabai merah (1710,004) lebih kecil dibandingkan dengan nilai RMSE terkecil pada data cabai rawit (1941,933).

Skenario 5 menguji panjang sekuens dari setiap sekuens yang dapat dibuat dari dataset. Terdapat empat nilai yang diuji yaitu 7, 14, 21, dan 30. Pada kedua jenis cabai menghasilkan nilai rata-rata RMSE terbaik pada pengujian panjang sekuens 7 dengan nilai rata-rata RMSE yang dihasilkan data cabai merah yaitu 1923,455 dan untuk cabai rawit yaitu 2061,045. Berikut adalah grafik hasil pengujian panjang sekuens dari data cabai merah maupun cabai rawit.



Gambar 4.9 Grafik Hasil Pengujian Panjang Sekuens 7 pada Data Cabai Merah

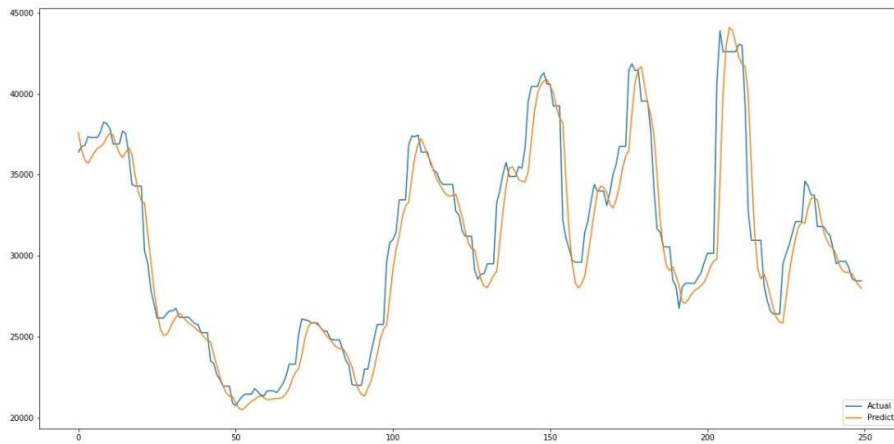


Gambar 4.10 Grafik Hasil Pengujian Panjang Sekuens 7 pada Data Cabai Rawit

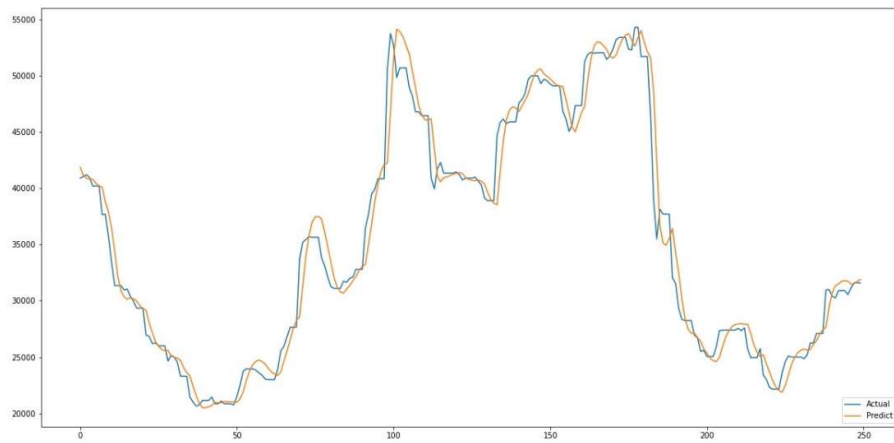
Berdasarkan Gambar 4.9, hasil pengujian panjang sekuens 7 pada cabai merah menghasilkan kurva yang cukup bagus ditandai dengan kurva jingga yang berdekatan dengan kurva biru. Pada pengujian panjang sekuens 7 data cabai rawit menghasilkan grafik yang tidak sebagus pengujian pada data cabai merah. Hal tersebut dapat diamati pada Gambar 4.10, dimana terdapat celah yang terdapat diantara kurva jingga dan kurva biru. Terlebih pada saat ketika kurva biru menurun dan langsung menaik, terlihat dimana kurva jingga turun lebih jauh dibandingkan dengan kurva biru dan terkesan terlambat naik ketika

kurva biru naik. Hal tersebut menunjukkan bahwa harga yang prediksi dari model komputasi lebih rendah dibandingkan harga aslinya dan model kurang bisa beradaptasi dengan adanya perubahan harga secara tiba-tiba.

Pada skenario 6 dilakukan pengujian jumlah epoch pada saat pelatihan model komputasi dengan beberapa nilai antara lain 10, 20, 30, 40, dan 50. Berdasarkan Tabel 4.6, pengujian jumlah epoch 50 menghasilkan nilai rata-rata RMSE terbaik untuk kedua jenis cabai. Pada cabai merah mendapatkan 1754,700 dan untuk cabai rawit yaitu 1878,329. Berikut grafik hasil pengujian jumlah epoch 50 pada kedua jenis cabai.



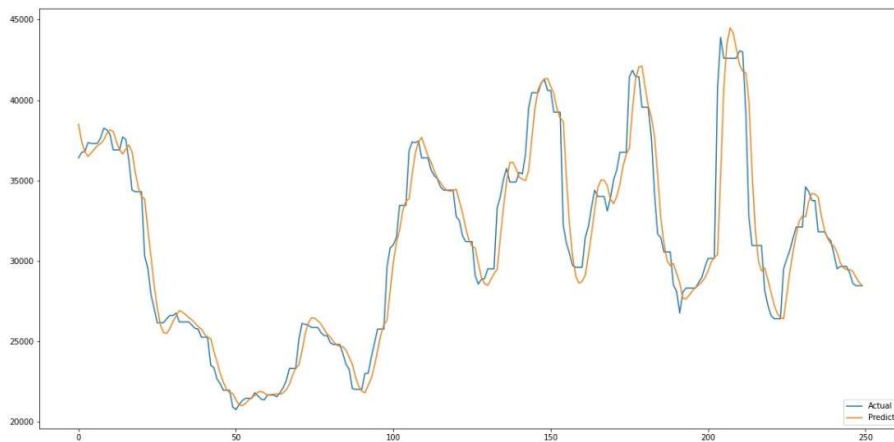
Gambar 4.11 Grafik Hasil Pengujian Epoch 50 pada Data Cabai Merah



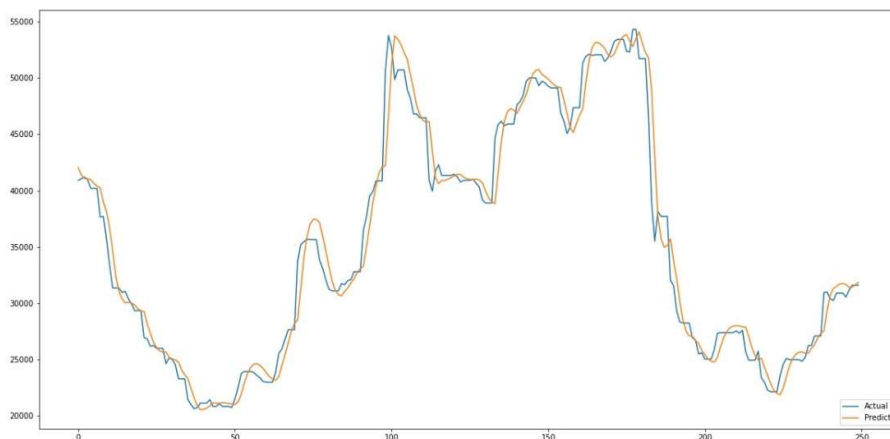
Gambar 4.12 Grafik Hasil Pengujian Epoch 50 pada Data Cabai Rawit

Pengujian epoch terhadap kedua jenis cabai menghasilkan grafik yang bagus dimana kurva jingga terlihat berimpit dengan kurva biru dan terlihat kurva jingga dapat mengikuti alur kurva biru pada Gambar 4.11 maupun pada Gambar 4.12. Namun, pada Gambar 4.12 menunjukkan bahwa terdapat beberapa celah antara kurva jingga dengan kurva biru yang lebih besar jika dibandingkan pada Gambar 4.11 yang berarti hasil prediksi pada Gambar 4.11 lebih baik dibandingkan dengan hasil prediksi pada Gambar 4.12, terlihat juga pada Tabel 4.6 dimana nilai RMSE pada data cabai merah (1739,348) lebih rendah dibandingkan nilai RMSE pada data cabai rawit (1837,712).

Skenario 7 atau skenario terakhir menguji beberapa optimasi model yang digunakan oleh model komputasi. Terdapat tiga optimasi model yang diuji yaitu Adam, RMSprop dan SGD. Pada Tabel 4.7 menunjukkan bahwa optimasi model RMSprop menghasilkan nilai rata-rata RMSE terbaik untuk pengujian pada kedua jenis data harga cabai dengan nilai RMSE pada data cabai merah yaitu 1931,978 dan 1983,436 untuk data cabai rawit. Berikut grafik hasil pengujian optimasi model RMSprop pada kedua jenis data harga cabai.



Gambar 4.13 Grafik Hasil Pengujian Optimasi RMSprop pada Data Cabai Merah



Gambar 4.14 Grafik Hasil Pengujian Optimasi RMSprop pada Data Cabai Rawit

Hasil pengujian optimasi model RMSprop pada data cabai merah sangat bagus. Hal tersebut ditunjukkan pada Gambar 4.13, dimana kurva jingga saling berimpit dengan kurva biru. Pada data cabai rawit, hasil pengujian menunjukkan hasil yang cukup bagus dan celah ketika kurva biru tiba-tiba naik setelah turun yang dihasilkan antara kurva jingga dengan kurva biru juga lebih baik dibandingkan pengujian-pengujian sebelumnya. Hal tersebut menandakan optimasi model RMSprop lebih sensitif terhadap perubahan data yang ada jika dibandingkan dengan optimasi model lainnya.

Berdasarkan eksperimen dari semua skenario yang telah dijalankan didapatkan parameter-parameter terbaik yang menghasilkan nilai rata-rata RMSE terkecil pada setiap skenarionya. Setelah itu dilakukan pengujian model menggunakan kombinasi parameter-parameter terbaik tersebut untuk mengetahui apakah kombinasi parameter-parameter tersebut menghasilkan nilai RMSE yang bagus atau sebaliknya. Berikut rancangan pengujian parameter-parameter terbaik tersebut.

Tabel 4.8 Skenario Pengujian Parameter Terbaik per Skenario

Parameter	Cabai Merah	Cabai Rawit
Jumlah unit	70 unit	80 unit
Jumlah lapisan tersembunyi	0 lapisan tersembunyi	0 lapisan tersembunyi

Fungsi aktivasi	Linier	Linier
Rasio data latih	80% (0,8)	90% (0,9)
Panjang sekuens	7	7
epoch	50	50
Optimasi model	RMSprop	RMSprop

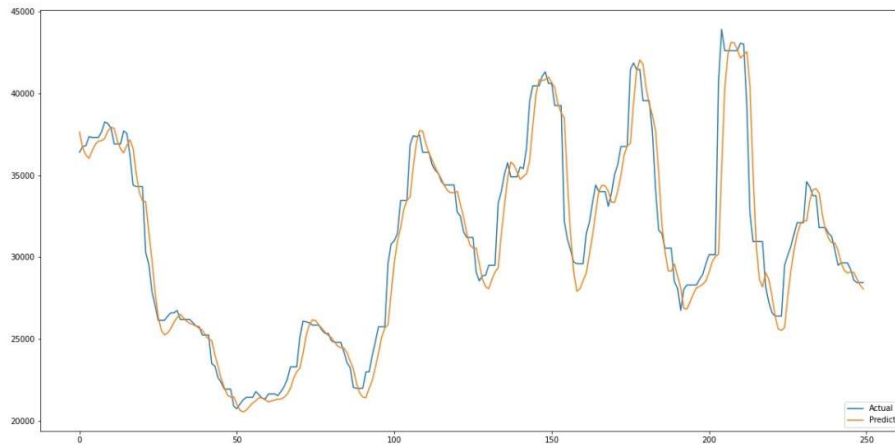
Pada rancangan pengujian diatas, hanya jumlah unit pada lapisan LSTM dan rasio data latih yang digunakan pada masing-masing data cabai yang berbeda dimana data cabai merah akan menggunakan jumlah unit 70 dan rasio data latih 80% sedangkan data cabai rawit akan menggunakan jumlah unit 80 dan rasio data latih 90%. Kemudian dilakukan pengujian kembali berdasarkan rancangan skenario pengujian parameter terbaik pada Tabel 4.8. Setiap data cabai dilakukan pengujian sebanyak lima kali untuk kemudian hasilnya dicatat dan menghitung nilai rata-rata RMSE yang dihasilkan. Setelah dilakukan pengujian parameter-parameter terbaik didapatkan hasil sebagai berikut.

Tabel 4.9 Hasil Pengujian Kombinasi Parameter Terbaik per Skenario pada Kedua Jenis Cabai

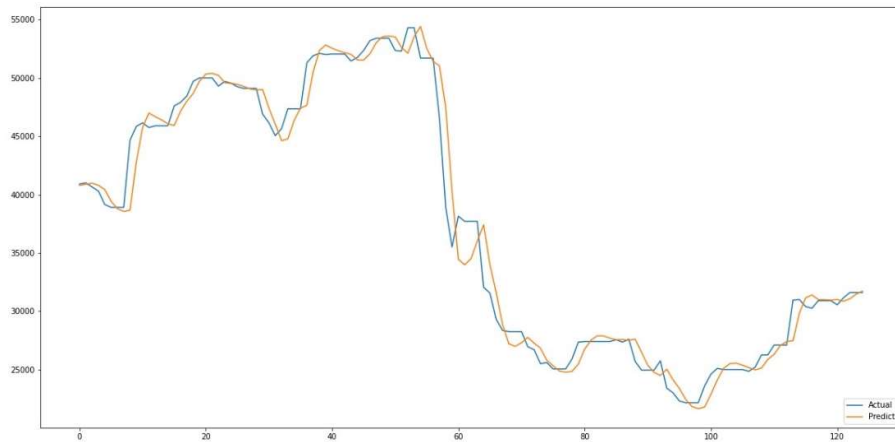
No.	Jenis cabai	Jumlah Unit	Rasio Data Latih	RMSE	MAPE	Durasi Latih	Rata-rata RMSE
1.	Cabai Merah	70	0,8	1645.699	3.196 %	10.833 detik	1828.117
2.	Cabai Rawit	80	0,9	1684.515	2.936 %	19.300 detik	1945.402

Berdasarkan Tabel 4.9, menunjukkan bahwa nilai rata-rata RMSE yang dihasilkan relatif cukup kecil dibandingkan nilai rata-rata RMSE yang dihasilkan pada pengujian-pengujian sebelumnya. Pada pengujian data cabai merah berhasil didapatkan nilai rata-rata RMSE sebesar 1758,578 dengan RMSE terkecil yaitu 1676,373 sedangkan pada data cabai rawit dihasilkan nilai rata-rata RMSE sebesar 1774,142 dan nilai RMSE terkecil yang dapat dihasilkan yaitu 1684,641. Nilai MAPE yang dihasilkan juga cukup kecil yaitu dibawah 4% untuk kedua jenis data cabai yang mana hal tersebut menunjukkan tingkat error antara nilai yang diprediksi dengan nilai yang sebenarnya kecil dan tingkat akurasi cukup tinggi. Durasi latih yang dibutuhkan oleh kedua data cabai tersebut terbilang cukup singkat jika

dibandingkan dengan durasi latih pada pengujian-pengujian sebelumnya yang berkisar antara 4 detik hingga 50 detik. Gambar 4.15 dan Gambar 4.16 menunjukkan grafik hasil pengujian skenario terbaik pada data cabai merah dan cabai rawit secara berurutan.



Gambar 4.15 Grafik Hasil Pengujian Kombinasi Parameter-Parameter Terbaik per Skenario pada Data Cabai Merah



Gambar 4.16 Grafik Hasil Pengujian Kombinasi Parameter-Parameter Terbaik per Skenario pada Data Cabai Rawit

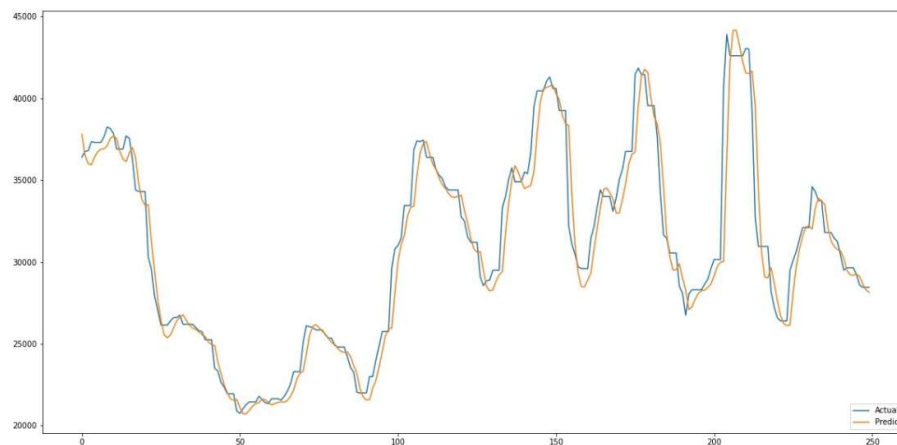
Namun, hasil tersebut ternyata bukan merupakan hasil terbaik jika mengkombinasikan seluruh parameter-parameter terbaik pada setiap skenario menjadi satu. Jika dibandingkan dengan Tabel 4.6 pada kolom rata-rata RMSE yang dihasilkan pada pengujian epoch 50 tersebut menunjukkan hasil yang lebih kecil dibandingkan rata-rata RMSE pada Tabel 4.9 untuk kedua

jenis data cabai. Untuk lebih memastikan jika skenario 6 adalah skenario terbaik, penulis menguji kembali skenario 6 tersebut pada pengujian epoch 50 dengan pengujian sebanyak lima kali untuk setiap data cabai, Berikut hasil dari pengujian ulang skenario 6 epoch 50 untuk masing-masing data cabai.

Tabel 4.10 Hasil Pengujian Ulang Skenario 6 Epoch 50 (Skenario Terbaik)

No.	Jenis cabai	RMSE	MAPE	Durasi Latih	Rata-rata RMSE
1.	Cabai Merah	1540.898	2.923 %	22.270 detik	1751.690
2.	Cabai Rawit	1816.318	3.503 %	28.906 detik	1888.741

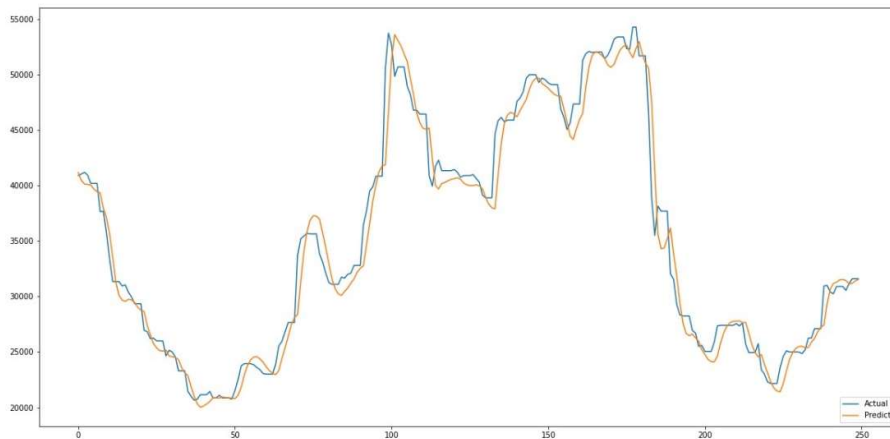
Berdasarkan Tabel 4.10, hasil pengujian ulang skenario 6 epoch 50 menunjukkan nilai rata-rata RMSE yang lebih kecil dibandingkan nilai rata-rata RMSE pada Tabel 4.9, dengan durasi yang sedikit lebih lama dibandingkan dengan Tabel 4.9 namun lebih konsisten dalam menghasilkan nilai RMSE yang kecil. Berikut grafik hasil pengujian ulang skenario ke-6 untuk masing-masing data cabai.



Gambar 4.17 Grafik Hasil Pengujian Ulang Skenario 6 Epoch 50 (Skenario Terbaik) pada Data Cabai Merah

Berdasarkan Gambar 4.17, kurva jingga terlihat berhimpit dengan kurva biru hal tersebut menunjukkan jika nilai prediksi memiliki selisih yang kecil dengan nilai sebenarnya. Namun, nilai RMSE terkecil yang dihasilkan pada Tabel 4.10 tidak lebih baik dari Tabel 4.9 untuk cabai merah. Meskipun begitu, nilai rata-rata RMSE pada Tabel 4.10 lebih rendah dibandingkan

Tabel 4.9. Hal tersebut menunjukkan bahwa hasil pengujian pada Tabel 4.10 lebih konsisten jika dibandingkan dengan hasil pengujian pada Tabel 4.9. Hal yang sama juga terjadi pada data cabai rawit, berikut grafik hasil pengujian ulang skenario 6.



Gambar 4.18 Grafik Hasil Pengujian Ulang Skenario 6 Epoch 50 (Skenario Terbaik) pada Data Cabai Rawit

Hal tersebut juga terlihat pada grafik pada Gambar 4.17 dan Gambar 4.16 yang menunjukkan hasil yang tidak lebih bagus jika dibandingkan dengan grafik pada Gambar 4.15 dan Gambar 4.16. Karena, grafik tersebut berdasarkan nilai RMSE terkecil bukan nilai rata-rata RMSE dari sebuah skenario.

Pada penelitian ini, fokus utama yaitu mencari skenario yang menghasilkan nilai rata-rata RMSE terkecil, hal tersebut menandakan bahwa pengujian skenario tersebut menghasilkan nilai RMSE yang kecil dan konsisten. Sehingga, lebih baik menghasilkan nilai RMSE kecil dan konsisten dibandingkan menghasilkan nilai satu nilai RMSE terkecil namun pengujian selanjutnya nilai RMSE yang dihasilkan tidak konsisten.

Dengan demikian, skenario terbaik untuk membangun sebuah arsitektur LSTM dan konfigurasi *hyperparameter* pelatihan terbaik adalah Skenario 6 yaitu pada saat pengujian jumlah epoch 50.

Pengujian selanjutnya yaitu menguji model yang sudah dilatih menggunakan data lain diluar rentang dataset yang digunakan pada proses pelatihan model. Pengujian ini bertujuan untuk mengetahui apakah model yang dilatih tersebut *overfitting* terhadap dataset pelatihan, yang mana *overfitting* tersebut menandakan model hanya akan menghasilkan hasil terbaik untuk dataset tertentu dengan kata lain model tidak dapat berfungsi terhadap rentang data lainnya.

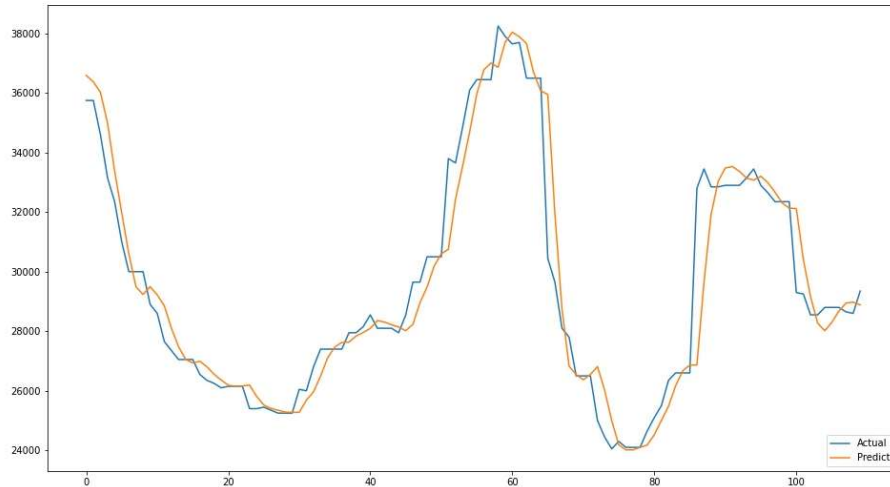
Pada pengujian kali ini penulis menggunakan rentang setelah dataset yaitu dari tanggal 1 Juni 2023 hingga tanggal 18 Oktober 2023 (tanggal Munaqosyah). Dengan menggunakan *preprocessing* yang sama dan panjang sekuens, didapatkan 140 data baik itu data cabai merah ataupun data cabai rawit. Kemudian dilakukan pemuatan model yang sudah dilatih menggunakan fungsi *load_model* dari *Library Keras*. Model yang dilatih yaitu model hasil pelatihan pada skenario 6 yang mana pada skenario tersebut menghasilkan nilai rata-rata RMSE terbaik diantara skenario-skenario lainnya. Untuk hasil pengujian model yang sudah dilatih dengan dataset lain ditunjukkan pada Tabel 4.11 berikut.

Tabel 4.11 Hasil Pengujian Model yang Sudah Dilatih pada Dataset Lain

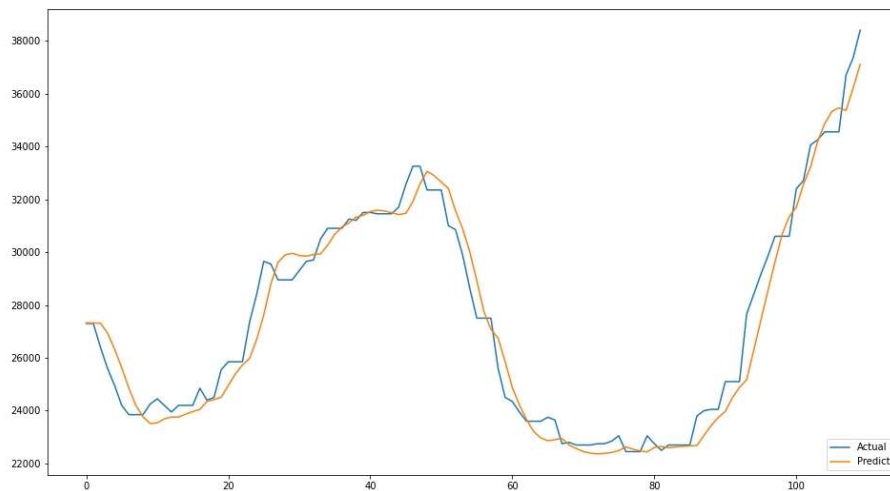
No.	Jenis cabai	RMSE	MAPE
1.	Cabai Merah	1160.695	2.280 %
2.	Cabai Rawit	816.052	2.256 %

Tabel 4.11 menunjukkan bahwa cabai rawit menghasilkan RMSE yang lebih kecil pada pengujian dataset lain dibandingkan cabai merah meskipun pada Tabel 4.10 nilai rata-rata RMSE cabai merah lebih rendah dibandingkan nilai rata-rata RMSE cabai rawit. Hal tersebut menunjukkan bahwa model yang dilatih pada data cabai rawit tidak mengalami *overfitting*. Namun, pengujian untuk kedua jenis cabai tetap menghasilkan nilai RMSE yang sangat rendah yang mana hal tersebut menunjukkan tingkat akurasi prediksi dari model yang sudah dilatih sangat bagus. Hal tersebut juga ditandai dengan indikator *error* yang rendah ditunjukkan oleh nilai MAPE yang rendah yaitu

dibawah 2.5 % untuk kedua jenis cabai. Berikut adalah grafik hasil pengujian model pada dataset baru untuk kedua jenis cabai.



Gambar 4.19 Grafik Hasil Pengujian Model yang Sudah Dilatih pada Dataset Cabai Merah Lain



Gambar 4.20 Grafik Hasil Pengujian Model yang Sudah Dilatih pada Dataset Cabai Rawit Lain

Gambar 4.19 dan Gambar 4.20 menunjukkan kurva jingga berhimpit pada kurva biru dengan celah yang kecil untuk kedua grafik. Dengan demikian, model yang sudah dilatih memiliki kemampuan untuk menghasilkan prediksi yang sangat bagus saat diuji dengan dataset lainnya.

BAB V

PENUTUP

5.1 Kesimpulan

Dengan demikian kesimpulan yang dapat diambil berdasarkan hasil penelitian ini adalah.

1. Penerapan Long Short Term Memory (LSTM) untuk prediksi harga cabai di Jawa Timur dapat dilaksanakan dengan beberapa tahapan seperti pengumpulan dan persiapan data, perancangan model komputasi, pelatihan hingga pengujian model komputasi. Untuk tingkat akurasi yang dihasilkan juga cukup baik yang ditandai dengan nilai RMSE dan rata-rata RMSE yang kecil pada saat pengujian skenario terbaiknya.
2. Tingkat akurasi (nilai RMSE) yang dihasilkan model komputasi dipengaruhi oleh beberapa parameter yang diuji. Adapun beberapa parameter yang sangat signifikan mempengaruhi nilai RMSE seperti jumlah unit, jumlah lapisan tersembunyi dari parameter arsitektur sedangkan dari parameter yang berkaitan dengan data dan pelatihan yaitu rasio data latih, panjang sekuens, epoch, dan optimasi model. Untuk parameter fungsi aktivasi memiliki pengaruh juga namun tidak signifikan parameter yang lainnya.
3. Hasil pengujian kombinasi parameter-parameter terbaik per skenario menunjukkan hasil yang tidak lebih baik dari skenario-skenario pengujian sebelumnya. Hal tersebut ditinjau berdasarkan nilai rata-rata RMSE yang dihasilkan. Adapun skenario yang menghasilkan nilai rata-rata RMSE yang lebih baik dibandingkan pengujian kombinasi parameter terbaik yaitu Skenario 6 yang menguji jumlah epoch pelatihan dan pada saat pengujian epoch 50 menghasilkan nilai rata-rata RMSE terkecil. Dengan demikian, Skenario 6 pada pengujian epoch 50 adalah skenario pengujian terbaik pada penelitian ini dengan konfigurasi

arsitektur serta *hyperparameter* yaitu jumlah unit sebanyak 50, tidak menggunakan lapisan tersembunyi atau 0 lapisan tersembunyi, menggunakan fungsi aktivasi linear, rasio data latih sebesar 80% atau 0,8 dari seluruh dataset, panjang sekuens sebesar 30, jumlah epoch 50 dan menggunakan optimasi model Adam. Berdasarkan pengujian kembali menggunakan konfigurasi tersebut didapatkan nilai rata-rata RMSE sebesar 1751.690 untuk data cabai merah dan 1888.741 untuk cabai rawit.

5.2 Saran

Berdasarkan penelitian yang sudah dijalankan, adapun beberapa saran yang dapat diterapkan untuk penelitian selanjutnya.

1. Melakukan pengujian arsitektur LSTM yang diterapkan pada jenis data yang berbeda seperti data saham, cuaca, dan lain sebagainya yang masih dalam kategori data *time series*.
2. Menentukan variabel-variabel lain yang diuji. Pada penelitian ini variabel-variabel yang diuji adalah jumlah unit, jumlah lapisan tersembunyi, fungsi aktivasi, rasio data latih, panjang sekuens, epoch, dan optimasi model yang menjadi sedikit dari variabel-variabel yang dapat mempengaruhi hasil prediksi dari model komputasi seperti *learning rate*, *batch size* dan lain sebagainya.
3. Untuk model yang dibuat pada penelitian ini hanya efektif untuk memprediksi data tepat satu hari selanjutnya (*day by day*) sehingga perlu dilakukan penelitian dan evaluasi lebih lanjut agar model dapat memprediksi data lebih dari satu hari dengan tingkat akurasi yang lebih baik.
4. Untuk mempermudah penggunaan hasil penelitian ini bagi kalangan yang lebih luas, sebuah sistem informasi ataupun aplikasi dapat dibuat dengan menerapkan hasil penelitian ini untuk menambah kemanfaatan dari hasil penelitian ini.

DAFTAR PUSTAKA

- Apa itu Long Short Term Memory (LSTM)?* (2022). Algoritma. <https://algoritma.blog/lstm-network-adalah-2022/>
- Arfan, A., & Lussiana, E. T. P. (2020). *Perbandingan algoritma long short-term memory dengan SVR pada prediksi harga saham di Indonesia*.
- Baheti, P. (2021). *Activation Functions in Neural Networks*. V7Labs. <https://www.v7labs.com/blog/neural-networks-activation-functions>
- Beklemysheva, A. (2019). *Why Use Python for AI and Machine Learning?* <https://steelkiwi.com/blog/python-for-ai-and-machine-learning/>
- Chairurrachman, I. (2022). *PENERAPAN LONG SHORT-TERM MEMORY PADA DATA TIME SERIES UNTUK PREDIKSI HARGA SAHAM PT. INDOFOOD CBP SUKSES MAKMUR TBK (ICBP)*. UIN SUNAN KALIJAGA YOGYAKARTA.
- Dhadse, A. (2021). *Introduction to TensorFlow with Keras API*. Analytics Vidhya. <https://medium.com/analytics-vidhya/introduction-to-tensorflow-with-keras-api-36cbeeb562d5>
- Dicoding. (2023). *Python: Pengertian, Contoh Penggunaan, dan Manfaat Mempelajarinya*. <https://www.dicoding.com/blog/python-pengertian-contoh-penggunaan-dan-manfaat-mempelajarinya/>
- Gupta, A. (2021). *A Comprehensive Guide on Optimizers in Deep Learning*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/>
- Hardjita, P. W., & Hidayat, R. (2021). Sentiment Analysis of Tweets on Prakerja Card using Convolutional Neural Network and Naive Bayes. *IJID (International Journal on Informatics for Development)*, 10(2), 82–91.
- Harga Cabai Rawit di Pasar Wilayah Jatim Naik Jelang Ramadan*. (2022). CNN Indonesia. <https://www.cnnindonesia.com/ekonomi/20220307190530-92-767987/harga-cabai-rawit-di-pasar-wilayah-jatim-naik-jelang-ramadan>
- Maity, R. (2021). *Min Max Scaler*. <https://medium.com/@ranjitmaity95/min-max-scaler-b2411ab3136d>
- Riyanto, E. (2017). PERAMALAN HARGA SAHAM MENGGUNAKAN JARINGAN SYARAF TIRUAN SECARA SUPERVISED LEARNING DENGAN ALGORITMA BACKPROPAGATION. *Jurnal Informatika Upgris*, 3(2).

<https://doi.org/10.26877/jiu.v3i2.1899>

- Riyantoko, P. A., Fahrudin, T. M., Hindrayani, K. M., & Safitri, E. M. (2020). Analisis Prediksi Harga Saham Sektor Perbankan Menggunakan Algoritma Long-Short Terms Memory (Lstm). *Seminar Nasional Informatika (SEMNASIF)*, 1(1), 427–435.
- Sanner, M. F. (1999). Python: a programming language for software integration and development. *J Mol Graph Model*, 17(1), 57–61.
- Saxena, S. (2021). *What is LSTM? Introduction to Long Short-Term Memory*. <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>
- Sutrisna, T. (2021). *Keluhkan Kenaikan Harga Cabai Rawit Merah, Warga Kurangi Pembelian hingga Ganti Menu Makanan*. Kompas. <https://megapolitan.kompas.com/read/2021/03/03/13110411/keluhkan-kenaikan-harga-cabai-rawit-merah-warga-kurangi-pembelian-hingga?page=all>
- Syaidah, K., Chrisnanto, Y. H., & Abdillah, G. (2020). Prediksi Harga Sembako di DKI Jakarta Menggunakan Artificial Neural Network. *JUMANJI (Jurnal Masyarakat Informatika Unjani)*, 3(02). <https://doi.org/10.26874/jumanji.v3i02.63>
- Vivas, E., Allende-Cid, H., & Salas, R. (2020). A systematic review of statistical and machine learning methods for electrical power forecasting with reported mape score. *Entropy*, 22(12), 1412.
- Wede. (2020). *Belajar Data Science: Apa yang dimaksud dengan Tensorflow dan Bagaimana Penggunaannya?* <https://dqlab.id/belajar-data-science-pahami-tensflow>
- Yanwardhana, E. (2022). *Ternyata Gegara Ini Harga Cabai Terbang Sampai Ratusan Ribu*. CNBC Indonesia. <https://www.cnbcindonesia.com/news/20220615103728-4-347195/ternyata-gegara-ini-harga-cabai-terbang-sampai-ratusan-ribu>