

🔍 Hasil Pencarian Daftar hasil pencarian

978-623-02-4913-6



Judul

Kepengarangan

Penerbit

ISBN

Hasil pencarian '978-623-02-4913-6' berdasarkan kategori 'ISBN'

Judul	Seri	Kepengarangan	Penerbit	ISBN
— Algoritma long short term memory dengan hyperparameter tuning : prediksi penjualan produk		Ali Khumaidi, S.Kom., M.Kom., Ika Ayu Nirmala, S.Kom. ; editor, Tiara Azhari	Deepublish	978-623-02-4913-6

Judul: Algoritma long short term memory dengan hyperparameter tuning : prediksi penjualan produk

Penerbit: Deepublish

Pengarang: Ali Khumaidi, S.Kom., M.Kom., Ika Ayu Nirmala, S.Kom. ; editor, Tiara Azhari

Tahun: 2022

Received: -

Seri: -

ISBN: 978-623-02-4913-6

Link: -

Website: www.deepublish.co.id

Email: deepublish@gmail.com

Menampilkan 1 sampai 1 dari 1 baris

Jam Layanan

Permohonan ISBN dan KDT
dilayani setiap hari kerja :

Senin - Jumat

🕒 09.00 – 15.00(WIB)

Kontak Kami

☎ 021 3812 136

Konfirmasi dan pengaduan:

✉ isbn@mail.perpusnas.go.id

🌐 isbn.perpusnas.go.id

Link Terkait

Perpusnas

Deposit perpusnas

Bibliografi Nasional RI



International Standard Book Number

PERPUSTAKAAN NASIONAL REPUBLIK INDONESIA
Pusat Bibliografi dan Pengolahan
Bahan Perpustakaan

Jl. Salemba Raya 28A
Daerah khusus Ibukota Jakarta 10430 - Indonesia

Tim ISBN/KDT ©2019.

Persaingan bisnis yang semakin kompetitif menuntut para pelaku bisnis untuk mendesain ulang strategi bisnisnya, salah satunya dengan menerapkan metode prediksi. Beberapa teknik prediksi telah digunakan dalam penelitian prediksi penjualan dengan akurasi yang cukup baik dalam waktu yang singkat. Buku ini akan fokus pada optimasi algoritma *Long Short Term Memory* (LSTM) untuk meningkatkan kinerja model yang terbentuk selama 60 hari ke depan. Studi kasus menggunakan data primer dari hasil observasi dan wawancara dan data sekunder menggunakan laporan transaksi penjualan produk pada Ciwo Pet Shop pada bulan Januari 2018 sampai dengan Maret 2020. Metode yang digunakan dalam penelitian ini adalah *Cross Industry Standard Process for Data Mining* (CRISP-DM), dalam *preprocessing*, *data cleaning*, *labeling*, *summary*, dan transformasi data. Tahap pemahaman data menggunakan metode Analisis Data Eksplorasi. Pengembangan model LSTM menggunakan beberapa parameter yaitu partisi data, jumlah *hidden layer*, skenario *dropout* untuk mencegah *overfitting*, jumlah neuron, *epoch* yang menggambarkan jumlah iterasi pelatihan, ukuran *batch* adalah jumlah data pelatihan yang harus diperhatikan dalam setiap proses pemutakhiran bobot.

Algoritma *Long Short Term Memory* dengan Hyperparameter Tuning: Prediksi Penjualan Produk

Penerbit Deepublish (CV BUDI UTAMA)
Jl. Kalurang Km 9.3 Yogyakarta 55581
Telp/Fax : (0274) 4533427
Anggota IKAPI (076/DIY/2012)
cs@deepublish.co.id
Penerbit Deepublish
@penerbitbuku_deepublish
www.penerbitdeepublish.com



Kategori :

Algoritma *Long Short Term Memory* dengan Hyperparameter Tuning: Prediksi Penjualan Produk



Ali Khumaidi, S.Kom., M.Kom.
Ika Ayu Nirmala, S.Kom.

Algoritma *Long Short Term Memory* dengan Hyperparameter Tuning: Prediksi Penjualan Produk

Ali Khumaidi, S.Kom., M.Kom. | Ika Ayu Nirmala, S.Kom.



**ALGORITMA LONG SHORT
TERM MEMORY DENGAN
HYPERPARAMETER TUNING:
PREDIKSI PENJUALAN PRODUK**

deepublish / publisher

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. Penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. Penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

**ALGORITMA LONG SHORT
TERM MEMORY DENGAN
HYPERPARAMETER TUNING:
PREDIKSI PENJUALAN PRODUK**

Ali Khumaidi, S.Kom., M.Kom.

Ika Ayu Nirmala, S.Kom.



Cerdas, Bahagia, Mulia, Lintas Generasi.

**ALGORITMA LONG SHORT TERM MEMORY DENGAN HYPERPARAMETER
TUNING: PREDIKSI PENJUALAN PRODUK**

Ali Khumaidi & Ika Ayu Nirmala

Editor :
Tiara Azhari

Desain Cover :
Rulie Gunadi

Sumber :
www.shutterstock.com

Tata Letak :
Amira Dzatin Nabila

Ukuran :
viii, 53 hlm, Uk: 15.5x23 cm

ISBN :
No ISBN

Cetakan Pertama :
Bulan 2022

Hak Cipta 2022, Pada Penulis

Isi diluar tanggung jawab percetakan

Copyright © 2022 by Deepublish Publisher
All Right Reserved

Hak cipta dilindungi undang-undang
Dilarang keras menerjemahkan, memfotokopi, atau
memperbanyak sebagian atau seluruh isi buku ini
tanpa izin tertulis dari Penerbit.

PENERBIT DEEPUBLISH
(Grup Penerbitan CV BUDI UTAMA)
Anggota IKAPI (076/DIY/2012)

Jl.Rajawali, G. Elang 6, No 3, Drono, Sardonoharjo, Ngaglik, Sleman
Jl.Kaliurang Km.9,3 – Yogyakarta 55581
Telp/Faks: (0274) 4533427
Website: www.deepublish.co.id
www.penerbitdeepublish.com
E-mail: cs@deepublish.co.id



KATA PENGANTAR PENERBIT

Membaca adalah sarana ekspresi diri dalam berkomunitas serta untuk terus maju menuju pencerdasan dan pencerahan. Ini menjadi sebuah motivasi dan dorongan bagi kami di Penerbit Deepublish untuk ikut berikhtiar dalam mencerdaskan dan memuliakan umat manusia, serta memanfaatkan ilmu pengetahuan dan teknologi untuk menciptakan industri *processing* berbasis sumber daya alam (SDA) Indonesia. Berdasarkan pandangan, sikap dasar, tujuan itu, maka buku yang berjudul *Algoritma Long Short Term Memory dengan Hyperparameter Tuning: Prediksi Penjualan Produk* ini diterbitkan.

Kami sadar masih terdapat berbagai kekurangan dalam buku ini. Namun, kami mencoba untuk terus mengembangkan diri, dan mencoba memperkecil kesalahan-kesalahan.

Kami mengucapkan terima kasih kepada penulis yang telah memberikan perhatian, kepercayaan, dan kontribusi demi kesempurnaan buku ini. Dan kepada pihak-pihak lainnya yang terus menjadi inspirasi dan memberikan semangat dalam menerbitkan buku yang berkualitas dan bermanfaat.

Dengan dukungan dari pembaca, kami dapat terus memberikan kontribusi bagi upaya mencerdaskan dan memuliakan umat manusia, serta memanfaatkan ilmu pengetahuan dan teknologi. Semoga buku ini dapat memperkaya khazanah dan memberi manfaat bagi para pembaca.

Hormat Kami,
Penerbit Deepublish



DAFTAR ISI

KATA PENGANTAR PENERBIT	v
DAFTAR ISI	vi
BAB 1. PERTUMBUHAN EKONOMI INDONESIA.....	1
BAB 2. GAGASAN DASAR IMPLEMENTASI ALGORITME LONG SHORT TERM MEMORY	5
2.1. Gagasan Penulis	5
2.2. Teori yang Memadai	6
2.2.1. Peramalan (<i>Forecasting</i>)	6
2.2.2. Metode CRIPS <i>Data Mining</i>	6
2.2.3. <i>Algoritme Long Short Term Memory</i> (LSTM)	8
2.2.4. Metode Optimasi	10
2.2.5. Fungsi Aktivasi	10
2.3. Korelasi Penelitian	17
BAB 3. SKEMA RISET	18
3.1. Bahan	18
3.2. Perangkat	18
3.3. Tahapan Riset	19
3.3.1. Identifikasi Masalah	20
3.3.2. Pemahaman Bisnis (<i>Business</i> <i>Understanding</i>)	21
3.3.3. Pemahaman Data (<i>Data</i> <i>Understanding</i>)	22
3.3.4. Persiapan Data	22
3.3.5. Pemodelan (<i>Modelling</i>)	24

3.3.6. Penyebaran (<i>Deployment</i>)	27
BAB 4. EFEKTIVITAS DATA	28
4.1. Pengumpulan Data	28
4.2. <i>Preprocessing Data</i>	29
4.3. Pemahaman Data.....	34
4.4. Pemodelan.....	38
4.4.1. Hasil Pengujian	40
4.4.2. Hasil Pengaplikasian	45
4.4.3. Evaluasi Model.....	47
4.4.4. Prediksi 60 Hari Ke Depan.....	48
BAB 5. KONKLUSI.....	52
TENTANG PENULIS	53

deepublish / publisher



BAB 1.

PERTUMBUHAN EKONOMI INDONESIA

UMKM (Usaha Mikro Kecil dan Menengah) merupakan salah satu pemain penting dalam pertumbuhan ekonomi Indonesia. UMKM dapat meningkatkan pendapatan dan menyerap tenaga kerja sekaligus mengurangi pengangguran. UMKM merupakan penggerak utama sektor riil dan berdampak langsung terhadap pertumbuhan ekonomi nasional (Zahrah dan Wijaya, 2019: 111). Statistik Kementerian Koperasi dan Usaha Kecil Menengah (UKM) pada tahun 2019 menunjukkan bahwa pada tahun 2018, kontribusi usaha kecil dan menengah terhadap PDB (Produk Domestik Bruto) mencapai 60,34 dari PDB nasional, yang jumlah terbesarnya usaha adalah 5.550, dan jumlah usaha besar dan menengah adalah 60.702. Ada 783.132 usaha kecil dan 63,5 juta usaha kecil dan mikro. Menurut Sensus Ekonomi 2016, UMKM juga mampu menyerap 97% tenaga kerja Indonesia. Selain itu, dibandingkan dengan industri lainnya, industri perdagangan memberikan kontribusi terbesar yaitu sebesar 31,54% (Kementerian Koperasi dan UKM, 2019: 5).

Saat ini banyak orang yang mendirikan usaha makro dan mikro untuk menyemarakkan persaingan di dunia wirausaha. Ada berbagai bidang usaha, salah satunya adalah perdagangan (Virrayyani & Sutikno, 2016: 57). Pelaku usaha saat ini menghadapi tantangan persaingan usaha yang semakin kompetitif, yang menuntut setiap pelaku usaha untuk mendesain ulang strategi usahanya sendiri untuk memenuhi tuntutan pasar. Dalam proses

bisnis, penjualan menjadi salah satu metrik terpenting. Jika tingkat penjualan yang dihasilkan tinggi, maka keuntungan yang dihasilkan juga besar. Hanya dengan cara ini perusahaan dapat bertahan dalam persaingan bisnis dan mengembangkan bisnisnya sendiri (Pakaja dan Naba, 2015: 23).

Ciwo Pet Shop merupakan salah satu perusahaan yang bergerak di bidang industri retail. Ciwo Pet Shop menyediakan berbagai kebutuhan hewan seperti makanan, obat-obatan, vitamin, dan aksesoris hewan peliharaan. Ciwo Pet Shop berdiri sejak tahun 2016. Saat ini Ciwo Pet Shop telah menerapkan metode prediksi untuk memperkirakan penjualan, namun prediksi yang digunakan sering kali tidak akurat sehingga kurang efektif. Hal ini mempengaruhi akumulasi produk ketika permintaan konsumen dan frekuensi penjualan rendah, yang mengakibatkan peningkatan biaya penyimpanan. Juga, ketika permintaan produk meningkat tetapi kehabisan stok untuk waktu yang cukup lama, toko kehilangan penjualan (*lost sales*). Untuk meminimalisir kerugian akibat kesalahan dalam melakukan peramalan penjualan, maka perlu dilakukan peramalan dengan menggunakan data historis transaksi penjualan dengan menggunakan suatu metode untuk mendapatkan akurasi yang terbaik dalam hasil peramalan.

Kajian sebelumnya terkait prediksi penjualan telah dilakukan oleh (Virrayyani & Sutikno, 2016). Kajian ini membahas tentang prediksi penjualan barang menggunakan algoritme *Adaptive Neuro-Fuzzy Inference System* (ANFIS). Hasil kajian ini dapat menghasilkan prediksi yang cukup baik dengan kesalahan MAPE kurang dari 20%. Studi lanjutan yang dilakukan oleh (Wiranda *et al.*, 2019) membahas tentang prediksi penjualan produk “X” untuk PT. Mestika Farma menggunakan algoritme *Long Short Term Memory* (LSTM). Hasil penelitian ini menghasilkan prediksi yang baik dengan kesalahan MAPE sebesar 12%. Studi lanjutan oleh (Helmini *et al.*, 2019) membahas peramalan penjualan di 1.150 toko di Jerman

dengan membandingkan algoritme *Extreme Gradient Boosting* (XGB) dan *Random Forest* (RFR) dengan *Long Short Term Memory* (LSTM). Hasil penelitian menunjukkan bahwa, akurasi LSTM adalah 12-14% lebih tinggi dari XGB dan RFR. Penelitian–penelitian yang telah dilakukan menunjukkan bahwa algoritme *Long Short Term Memory* (LSTM) memiliki akurasi yang cukup baik untuk melakukan sebuah prediksi.

Berdasarkan penelitian sebelumnya di bidang peramalan penjualan, banyak pekerjaan telah dilakukan dengan tujuan menggunakan algoritme tertentu atau membandingkan algoritme tertentu untuk menentukan produk terlaris. Penelitian ini akan fokus pada peningkatan kinerja model yang dihasilkan dengan mengoptimalkan *hyperparameter* dari algoritme *Long Short Term Memory* untuk memprediksi jumlah produk yang terjual di masa mendatang.

Berdasarkan pengantar di atas, maka beberapa topik dalam kajian ini adalah bagaimana mengimplementasikan konfigurasi parameter algoritme *Long Short Term Memory* untuk memprediksi jumlah produk yang terjual dalam 60 hari ke depan. Kemudian tujuan utama kajian ini adalah untuk menemukan konfigurasi model yang optimal dengan menentukan parameter yang optimal untuk menghasilkan prakiraan penjualan menggunakan algoritme *Long Short Term Memory* (LSTM), sehingga dapat meningkatkan kinerja model-model yang terbentuk. Harapan yang muncul yaitu dapat menghasilkan konfigurasi model yang optimal yang memprediksi jumlah penjualan dengan akurasi terbaik, sehingga membantu dalam merencanakan strategi penjualan dan mengatur persediaan produk.

Berdasarkan pertanyaan di atas, ruang lingkup buku ini dilakukan di Ciwo Pet Shop Bekasi. Data yang digunakan adalah data transaksi penjualan Ciwo Pet Shop dari tahun 2018 sampai dengan tahun 2020. Pengolahan data menggunakan algoritme *Long Short Term Memory* (LSTM). Bahasa pemrograman yang digunakan

adalah Python 3,7, dan editornya menggunakan Google Colab dengan library seperti Pandas, Sklearn, Matplotlib, Keras, Tensor Flow. Penelitian ini mengimplementasikan algoritme *Long Term Memory* untuk menghasilkan perkiraan penjualan hanya dengan menentukan *hyperparameter* yang optimal.



BAB 2.

GAGASAN DASAR IMPLEMENTASI ALGORITME LONG SHORT TERM MEMORY

2.1. Gagasan Penulis

Adanya buku ini dikarenakan penulis terinspirasi oleh beberapa kajian yang telah dilakukan peneliti lain. Kajian yang telah menginspirasi penulis yang pertama berasal dari kajian yang dilakukan oleh Ahmad Ashril Rizal dan Siti Soraya yang berjudul “*Multi Time Steps Prediction dengan Reccurent Neural Network Long Short Term Memory*”. Kajian tersebut berisi mengenai prediksi kunjungan wisatawan. Variabel yang digunakan dalam penelitian ini adalah jumlah kunjungan wisatawan dari Januari 2009 sampai dengan Desember 2014. Hasil prediksi tersebut menghasilkan nilai RMSE sebesar 6888,37 pada data *training* dan 14684,33 pada data *testing*.

Kajian yang kedua berasal dari Muhammad Wildan Putra Aldi, Jondri & Annisa Aditsania yang berjudul “Analisis dan Implementasi *Long Short Term Memory Neural Network* untuk Harga Bitcoin”. Kajian tersebut berisi mengenai prediksi pergerakan kurs nilai tukar Bitcoin. *Dateset* yang digunakan berupa data jual beli *index* harga Bitcoin terhadap US Dollar. Hasil analisis menunjukkan bahwa sistem mampu menghasilkan harga Bitcoin dengan rata-rata tingkat akurasi sebesar 93.5% untuk data *testing*.

Kajian yang ketiga berasal dari Laras Wiranda & Mujiono Sadikin yang berjudul “Penerapan *Long Short Term Memory* pada Data *Time Series* untuk Memprediksi Penjualan Produk PT. Metiska

Farma”. Membahas mengenai prediksi penjualan produk “X”. Atribut yang digunakan dalam kajian ini adalah waktu (tanggal penjualan) dan nilai (*value* penjualan harian yang didapat dalam bentuk rupiah). Hasil dari prediksi tersebut menghasilkan nilai RMSE sebesar 13,762,154.00 dalam nilai Rupiah dan MAPE sebesar 12%.

2.2. Teori yang Memadai

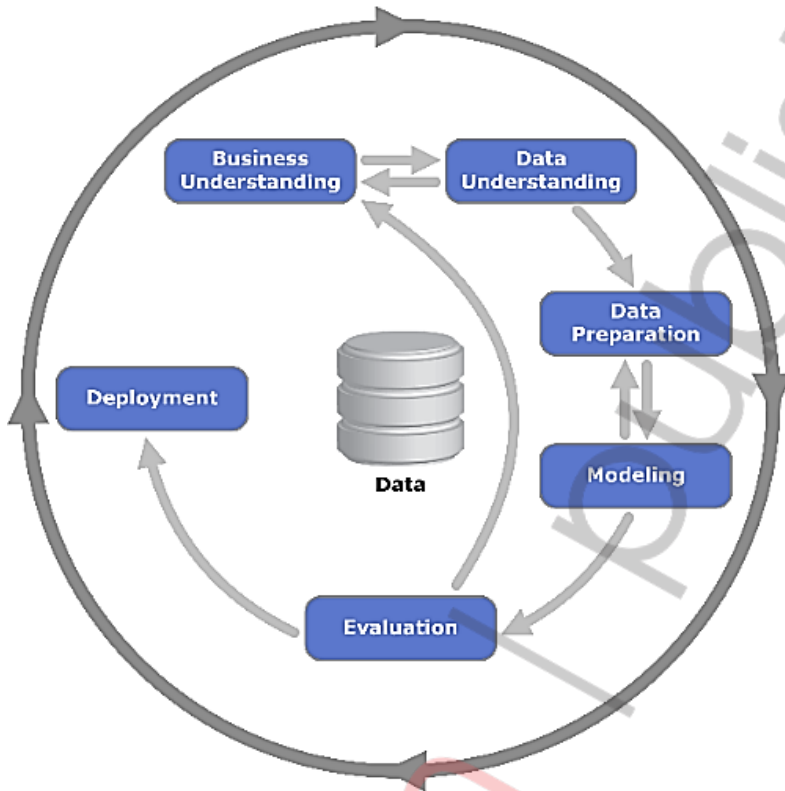
2.2.1. Peramalan (*Forecasting*)

Peramalan (*forecasting*) adalah seni dan ilmu untuk memprediksi kemungkinan kejadian di masa depan. Hal tersebut dapat dilakukan dengan mengolah data historis dan memproyeksikannya ke masa depan. Dengan membentuk model matematis atau prediksi intuisi subjektif yang disesuaikan dengan berbagai pertimbangan (Wiranda *et al.*, 2019:185).

Prediksi penjualan (*sales forecasting*) adalah suatu metode prediksi penjualan di masa yang akan datang, sehingga dapat diperoleh suatu perkiraan yang mendekati keadaan yang sebenarnya. Dengan mengetahui perkiraan penjualan yang mendekati kenyataan, maka dapat mengurangi risiko pengambilan keputusan yang kurang tepat. (Erni, Widajanti & Suprayitno, 2020: 443)

2.2.2. Metode CRIPS *Data Mining*

CRIPS *Data Mining* adalah singkatan dari *Cross Industry Standard Process for Data Mining* yang bertujuan untuk menemukan pola menarik dan bermakna dalam suatu data. Dalam penerapannya, CRIPS *Data Mining* beberapa disiplin ilmu seperti Statistika, *Machine learning*, *Artificial Intelligence*, *Pattern Recognition*, dan *Data Mining*. (Palupi *et al.*, 2020: 135) CRIPS *Data Mining* merupakan deskripsi rinci dari rencana proyek *data mining*.



Gambar 2.1 Metode CRISP Data Mining

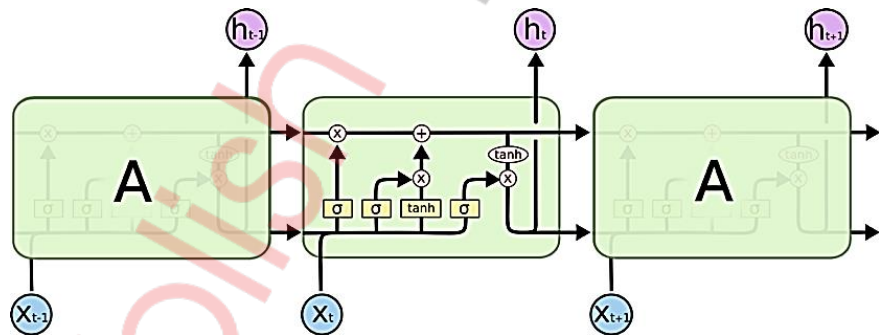
Proses dalam *data mining* dapat dibagi menjadi enam tahap, yakni *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, *Deployment*. (Khumaidi, A., 2020: 4) masing-masing tahapan dijelaskan sebagai berikut:

1. Pemahaman Bisnis (*Bussiness Understanding*)
2. Pemahaman Data (*Data Understanding*)
3. Pengolahan Data (*Data Preparation*)
4. Pemodelan (*Modelling*)
5. Evaluasi (*Evaluation*)
6. Penyebaran (*Deployment*)

2.2.3. Algoritme Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) adalah salah satu jenis *Recurrent Neural Network* (RNN). (Wildan *et al.*, 2018: 3548) RNN mampu memproses data sekuensial yang panjang dan kompleks. RNN biasanya menggunakan model *backpropagation* yang mirip dengan jaringan saraf, dan rantai yang terlalu panjang dalam topologi RNN cenderung mengalami penurunan gradien (*vanishing gradient*) (Kristian *et al.*, 2018:311).

Long Short Term Memory (LSTM) dikembangkan untuk mengatasi *vanishing gradient* pada RNN. (Kristian *et al.*, 2018: 311) LSTM telah dimodifikasi dengan menambahkan *memory cell* sehingga dapat menyimpan informasi dalam jangka waktu yang lebih lama. Keuntungan dari LSTM adalah adanya *memory block* yang menentukan nilai mana yang akan dipilih sebagai *output* yang relevan untuk *input* yang diberikan. (Wiranda *et al.*, 2019: 186) Contoh arsitektur LSTM dapat dilihat pada Gambar 2.1.



Gambar 2.2 Arsitektur *Long Short Term Memory* (LSTM)

Arsitektur LSTM terdiri dari lapisan *input*, lapisan *output*, dan lapisan tersembunyi. Lapisan tersembunyi terdiri dari *memory cell*, masing-masing memori terdiri dari gerbang masukan (*input gate*), gerbang lupa (*forget gate*) dan gerbang keluaran (*output gate*) (Wiranda *et al.*, 2019:189).

Masing-masing gerbang yang termasuk pada LSTM dijelaskan sebagai berikut:

a. Forget Gate (f_t)

Forget gate berfungsi untuk mengontrol seberapa baik nilai dipertahankan dalam *memory cell* (Wiranda *et al.*, 2019: 189). *Forget gate* adalah lapisan sigmoid yang mengambil nilai *output* pada waktu $t-1$ dan nilai *input* pada waktu t , yang kemudian digabungkan dan diterapkan pada fungsi aktivasi sigmoid. Di mana hasil *output* sigmoid bernilai 0 dan 1 (Wildan, Aldi, *et al.*, 2018: 3550). Jika nilai $f_t = 1$ maka *state* sebelumnya tidak akan berubah dan data akan disimpan, sedangkan jika $f_t = 0$ maka *state* sebelumnya akan dilupakan. (Wiranda *et al.*, 2019: 190) Persamaan dari f_t yaitu:

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

b. Input Gate (I_t)

Peran *input gate* adalah untuk mengontrol sejauh mana nilai baru mengalir di dalam *cell*. Ini dirancang untuk menghindari penyimpanan data yang tidak perlu. *Input gate* mengambil nilai *output* sebelumnya dan nilai *input* baru serta melewati lapisan sigmoid. Nilai *gate* ini dikembalikan sebagai menjadi 0 atau 1 (Wiranda *et al.*, 2019: 189) Persamaan dari I_t yaitu:

$$I_t = \sigma (W_i S_{t-1} + W_i X_t) \dots\dots\dots (2)$$

Kemudian kalikan nilai *input gate* dengan nilai *output gate* dari lapisan kandidat \hat{C} . (Wiranda *et al.*, 2019: 190)

Persamaan dari \hat{C} yaitu:

$$\hat{C} = \tanh (W_c S_{t-1} + W_c X_t) \dots\dots\dots (3)$$

$$c_t = (i_t * \hat{C}_t + f_t * c_{t-1}) \dots\dots\dots (4)$$

c. **Output Gate (O_t)**

Output Gate digunakan untuk mengendalikan seberapa banyak nilai yang berada di dalam *memory cell* yang berfungsi untuk menghitung nilai *output* (Wiranda *et al.*, 2019: 190).

Rumus dari O_t yaitu:

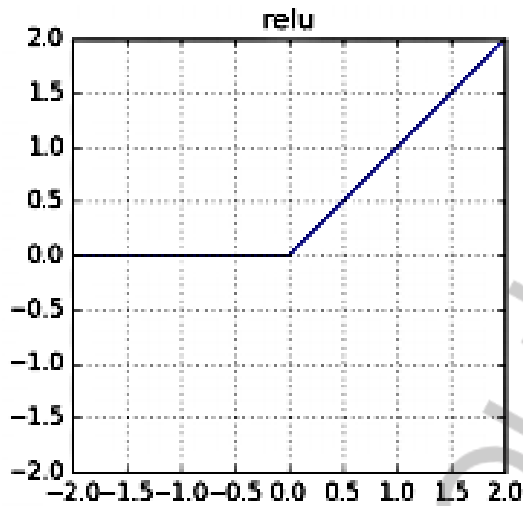
$$O_t = \sigma (W_t S_{t-1} + W_t X_t) \dots\dots\dots (5)$$

2.2.4. Metode Optimasi

Adaptive Moment Estimation (Adam) adalah metode optimasi yang secara adaptif menghitung tingkat pembelajaran untuk setiap parameter. Seperti metode optimasi lainnya seperti Adadelta dan RMSProp, Adam menyimpan rata-rata *gradient* proses sebelumnya meningkat secara eksponensial. (Zahara *et al.*, 2017: 360) Optimizer Adam dapat mengendalikan masalah *sparse gradient*. Hal tersebut merupakan perluasan untuk keturunan gradien stokastik yang saat ini telah melihat adopsi yang lebih luas untuk menerapkan *deep learning* seperti *Natural Language Processig* (Winda Kurnia Sari *et al.*, 2020: 278).

2.2.5. Fungsi Aktivasi

Rectified Linear Unit (ReLU) adalah fungsi aktivasi yang sederhana secara komputasi. Proses *forward* dan *backward* melalui ReLU hanya menggunakan kondisi *if*. Jika elemen bernilai negatif maka nilainya di *set* menjadi 0, dan tidak ada operasi eksponensial, perkalian atau pembagian. Dengan karakteristik seperti itu, keunggulan ReLU terwujud ketika berhadapan dengan jaringan dengan jumlah *neuron* yang besar, yang secara signifikan mengurangi waktu *training* dan *testing* (Zahara *et al.*, 2017).



Gambar 2.3 Diagram Fungsi Aktivasi ReLU

2.2.6. Hyperparameter Tuning

Hyperparameter adalah nilai parameter yang dapat dikonfigurasi dalam proses pelatihan model, nilai parameter ini dapat disesuaikan secara berbeda selama proses pembelajaran. Contohnya, pada jaringan neural network, dapat menentukan jumlah epoch, hidden layer, jumlah simpul tiap layer, dan iterasi. Pada algoritme Random Forest, dapat menentukan kedalaman maksimum (max_kedalaman), jumlah estimator (n_estimator), dan kriteria. Hyperparameter dapat mempengaruhi kinerja dari model.

Hyperparameter tuning atau penyetelan hyperparameter adalah proses menemukan nilai hyperparameter terbaik untuk model dengan mencari lebih dari satu set nilai yang ditentukan untuk masing-masing hyperparameter. Pemilihan parameter yang optimal merupakan langkah signifikan yang akan mempengaruhi kinerja model. Proses penentuan nilai parameter tidak ada panduan khusus namun dilakukan secara trial dan error untuk mendapatkan nilai optimal.

Pada machine learning kegiatan hyperparameter tuning merupakan tantangan dalam memilih kumpulan hyperparameter yang sesuai untuk algoritme pembelajaran. Hyperparameter tuning adalah penentuan nilai untuk parameter yang digunakan dalam mempengaruhi proses pembelajaran. Selain itu, faktor-faktor lain, seperti bobot simpul juga dipelajari. Untuk menggeneralisasi pola data yang beragam, model machine learning yang sama akan memerlukan batasan, bobot, atau kecepatan pembelajaran yang berbeda. Nilai-nilai ini dikenal sebagai hyperparameter, dan nilai tersebut harus disesuaikan agar model dapat melakukan tugas machine learning secara optimal. Hyperparameter tuning adalah proses mengidentifikasi tuple hyperparameter yang menghasilkan model optimal yang meminimalkan fungsi kerugian yang telah ditentukan pada data independen yang disediakan.

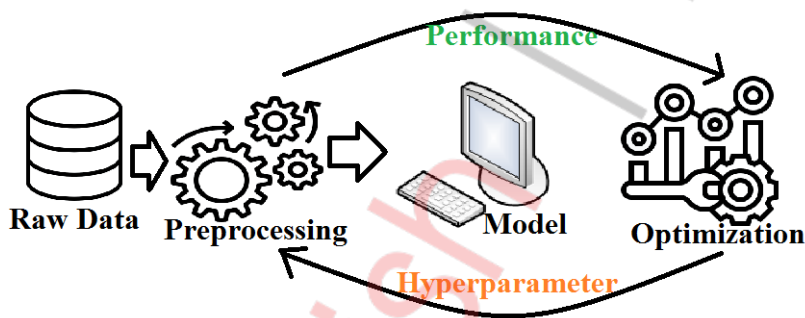
Hyperparameter model diatur secara manual dan digunakan dalam proses untuk membantu memperkirakan model parameter. Model hyperparameter sering disebut sebagai parameter karena merupakan bagian dari machine learning yang harus diatur dan disetel secara manual. Model hyperparameter adalah pengaturan eksternal yang nilainya tidak dapat ditebak dari data. Nilai model hyperparameter ditentukan sebelum model memulai pelatihan. Mereka tidak dapat ditemukan dengan memasang model ke data. Kecepatan dan ketepatan proses pembelajaran model dipengaruhi oleh hyperparameter. Jumlah hyperparameter yang dibutuhkan oleh berbagai sistem bervariasi.

Pengoptimalan hyperparameter adalah proses menemukan kombinasi nilai hyperparameter yang tepat untuk mencapai performa maksimum pada data dalam waktu yang wajar. Ini memainkan peran penting dalam keakuratan prediksi algoritme pembelajaran mesin. Oleh karena itu, pengoptimalan Hyperparameter dianggap sebagai paling sulit bagian dari membangun model pembelajaran mesin. Gambar 2.4 menjelaskan bagaimana proses pemodelan dimulai dari

data mentah, praproses hingga kegiatan penyetelan hyperparameter untuk peningkatan kinerja model.

Sebagian besar algoritme pembelajaran mesin ini dilengkapi dengan nilai hyperparameter default. Nilai default tidak selalu berfungsi dengan baik pada berbagai jenis project Machine Learning yang Anda miliki, itulah mengapa Anda perlu mengoptimalkannya untuk mendapatkan kombinasi yang tepat yang akan memberikan performa terbaik.

Pada hyperparameter tuning dapat memanfaatkan infrastruktur pemrosesan Google Cloud untuk menguji konfigurasi hyperparameter yang berbeda saat melatih model. Ini dapat memberi nilai yang dioptimalkan untuk hyperparameter, yang memaksimalkan akurasi prediksi modelmu.



Gambar 2.4. Optimasi Model dengan Hyperparameter

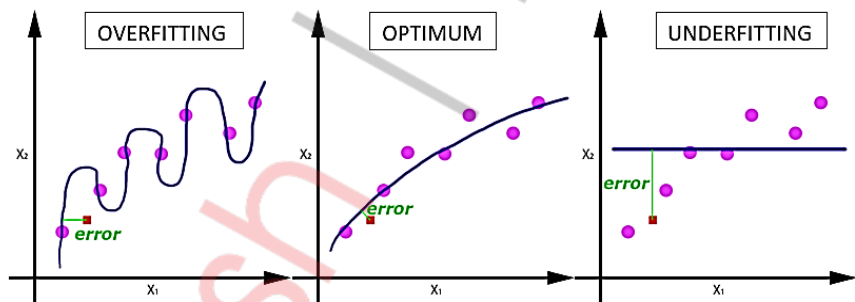
Paramater yang digunakan dalam hyperparameter tuning diantaranya:

a. Epoch

Epoch adalah ketika seluruh dataset sudah melalui proses training pada Neural Netwok sampai dikembalikan ke awal untuk sekali putaran, karena satu Epoch terlalu besar untuk dimasukkan (feeding) ke dalam komputer maka dari itu diperlukan pembagian dalam satuan kecil (batches).

Tapi, mengapa diperlukan lebih dari satu Epoch? Karena tidak masuk akal di awal bahwa melewati seluruh dataset melalui jaringan saraf tidak cukup dan diperlukan melewati dataset penuh beberapa kali ke jaringan saraf yang sama. Namun perlu diingat bahwa penggunaan dataset yang terbatas dan untuk mengoptimalkan pembelajaran dan grafik yang digunakan adalah Gradient Descent yang merupakan proses iteratif. Jadi, mengupdate weight (bobot) dengan satu epoch saja tidak cukup. Satu epoch mengarah pada underfitting seperti pada grafik di bawah ini.

Seiring bertambahnya jumlah epoch, semakin banyak pula weight (bobot) yang berubah dalam Neural Network dan kurvanya melengkung dari kurva yang kurang sesuai hingga selaras dengan kurva yang overfitting.



Gambar 2.5. Grafik Overfitting, Optimum, Underfitting dari Jumlah Epoch

Lalu berapakah jumlah epoch yang harus ditentukan? Tidak ada jawaban yang benar untuk pertanyaan ini. Jawabannya berbeda untuk dataset yang berbeda tapi bisa dikatakan bahwa jumlah epoch terkait dengan beragamnya data, jadi jumlah epoch tergantung dataset yang dimiliki.

b. Iterasi

Iterasi adalah jumlah batch yang diperlukan untuk menyelesaikan satu epoch. Tapi, untuk memahami iterasi sebenarnya hanya perlu mengetahui tabel perkalian atau memiliki kalkulator. Katakanlah memiliki 2000 contoh training yang akan digunakan, maka dapat membagi dataset dari 2000 contoh tersebut menjadi batch dari 500 maka akan dibutuhkan 4 iterasi untuk menyelesaikan 1 epoch.

c. Batch Size

Batch Size adalah jumlah sampel data yang disebarkan ke Neural Network. Contoh: jika mempunyai 100 dataset dan batch size adalah 5 maka algoritme ini akan menggunakan 5 sampel data pertama dari 100 data yang dimiliki (ke1, ke2, ke3, ke4, dan ke5) lalu disebarkan atau ditraining oleh Neural Network sampai selesai kemudian mengambil kembali 5 sampel data kedua dari 100 data (ke6, ke7, ke8, ke9, dan ke10), dan begitu seterusnya sampai 5 sampel data ke 20 ($100/5=20$).

Tapi, apa itu Batch? Seperti yang telah dikatakan sebelumnya, tidak bisa melewati seluruh dataset ke dalam jaring saraf sekaligus. Jadi, perlu membagi dataset menjadi sejumlah atau satu set atau bagian.

d. Hidden Layer

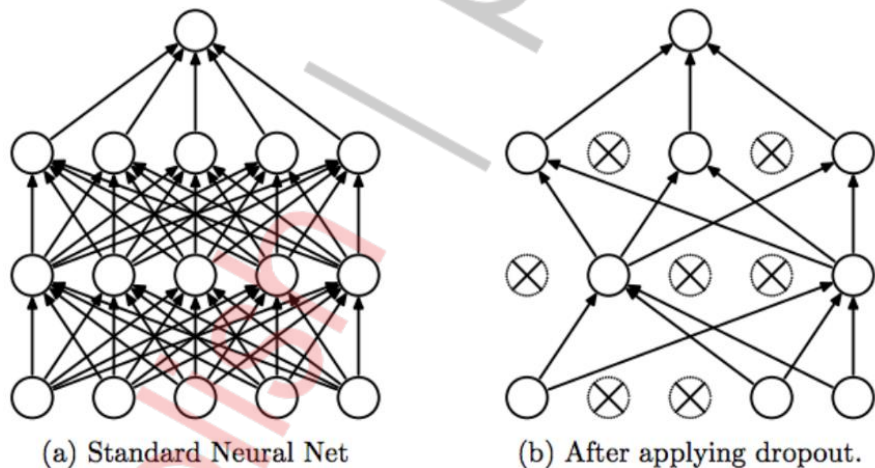
Hidden layer adalah lapisan antara input layer dan output layer, dimana artificial neuron yang memiliki sekumpulan input pembobot 'weight' dan prosedur untuk menghasilkan output neuron melalui activation function.

Pada hidden layer dapat dibuat banyak lapis atau berlapis-lapis untuk menemukan komposisi algoritme yang tepat agar dapat meminimalisir error pada output. Semakin banyak layer yang

dihasilkan, maka akan semakin kecil error yang dihasilkan, sehingga dapat menghasilkan akurasi yang lebih bagus atau lebih tinggi.

e. Dropout

Dropout merupakan sebuah teknik regulasi jaringan syaraf dengan tujuan memilih beberapa neuron secara acak dan tidak akan dipakai selama proses pelatihan, dengan kata lain neuron-neuron tersebut dibuang secara acak. Hal ini berarti bahwa kontribusi neuron yang dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak diterapkan pada neuron pada saat melakukan backpropagation. Gambar 2.6 merupakan proses dropout pada suatu jaringan.



Gambar 2.6. Proses Dropout Pada Suatu Jaringan

Berdasarkan gambar tersebut, pada bagian a merupakan jaringan syaraf biasa yang memiliki dua hidden layer. Sedangkan pada bagian b merupakan jaringan syaraf dengan menggunakan dropout. Dari gambar tersebut terlihat terdapat beberapa neuron aktivasi yang tidak dipakai lagi.

2.3. Korelasi Penelitian

Berdasarkan tinjauan pustaka di atas penggunaan penelitian-penelitian terdahulu mengenai *forecasting* menggunakan algoritme *long short term memory* telah diterapkan di berbagai bidang dan menghasilkan nilai akurasi yang baik. Permasalahan yang diajukan oleh penelitian-penelitian tersebut pun berbeda-beda. Permasalahan ini menimbulkan masalah yang berbeda dari kajian sebelumnya, namun menggunakan solusi yang sama dan menambahkan hasil tambahan berupa prediksi penjualan selama 60 hari ke depan berdasarkan konfigurasi model terbaik yang dihasilkan dengan parameter kinerja RMSE yang dioptimalkan sehingga dapat memperkecil nilai *error* yang dihasilkan. Landasan teori yang dipaparkan berkaitan dengan pembahasan penelitian dimaksudkan untuk memberikan referensi dan acuan untuk mempertajam pokok pembahasan penelitian agar lebih terarah ke tujuan dan berfokus pada implementasi algoritme *long short term memory*.



BAB 3.

SKEMA RISET

3.1. Bahan

Sumber data yang digunakan dalam kajian ini bersumber dari Ciwo Pet Shop, salah satu usaha yang bergerak di bidang retail yang menjual berbagai kebutuhan hewan peliharaan. Sumber data yang digunakan dalam penelitian ini terbagi menjadi dua bagian, antara lain:

1. Data Primer

Data mentah (primer) yang diperoleh dari *setting* kajian, diperoleh dalam bentuk observasi dan wawancara dari sumber-sumber yang relevan, seperti pemilik usaha, karyawan, dan pengelola Ciwo Pet Shop.

2. Data Sekunder

Data sekunder yang digunakan adalah laporan penjualan produk Ciwo Pet Shop periode Januari 2018 sampai Maret 2020. Data tersebut meliputi data penjualan dari salah satu *marketplace* (Tokopedia) dan data penjualan *offline* dari toko.

3.2. Perangkat

Untuk memenuhi kebutuhan penelitian “Prediksi Penjualan Produk Menggunakan Algoritme *Long Short Term Memory* (LSTM) (Studi Kasus: Ciwo Pet Shop)”).

Digunakan beberapa perangkat pendukung seperti terlihat pada Tabel 3.1 dan Tabel 3.2 di bawah ini:

Tabel 3.1 Daftar Kebutuhan Perangkat Keras

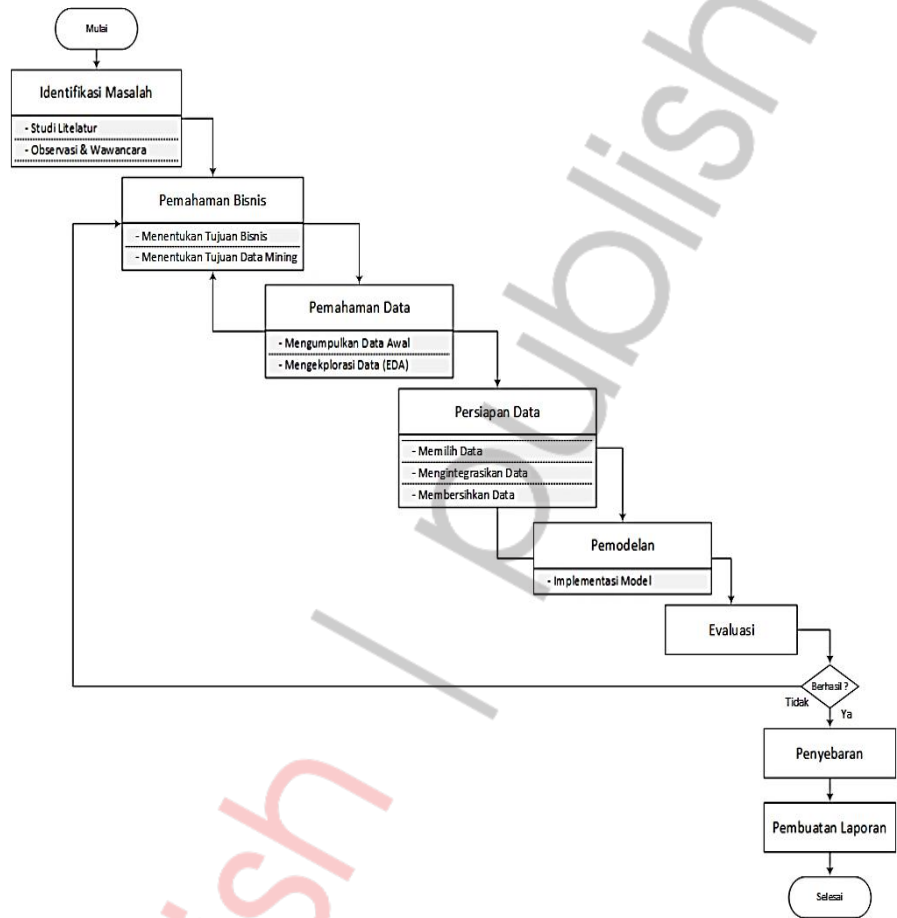
No.	Perangkat Keras (<i>Hardware</i>)	Spesifikasi
1	Laptop	Asus VivoBook Max
2	<i>Processor</i>	AMD A6-9220 Radeon R4
3	<i>Installed Memory</i> (RAM)	4 GB
4	<i>Hardisk</i>	1 TeraByte
5	Grafis	AMD Radeon R4 Graphics

Tabel 3.2 Daftar Kebutuhan Perangkat Lunak

No.	Spesifikasi	Keterangan
1	Windows 10 Pro 64-Bit	Sistem Operasi
2	Python 3.7	Bahasa Pemrograman
3	Jupiter Notebook atau Google Colab	Text Editor Python
4	Tensorflow	Pustaka Perangkat Lunak
5	Pandas	Pustaka Perangkat Lunak
6	Matplotlib	Pustaka Perangkat Lunak
7	Numpy	Pustaka Perangkat Lunak
8	Math	Pustaka Perangkat Lunak
9	Sklearn.metrics	Pustaka Perangkat Lunak
10	Tensorflow Keras	Pustaka Perangkat Lunak
11	Anaconda	<i>Management Tools</i>
12	Microsoft Excel 2019	<i>Tools</i> untuk membaca <i>dataset</i>

3.3. Tahapan Riset

Metode yang digunakan dalam kajian ini adalah CRISP-DM (*Cross Industry Standard Process for Data Mining*) yang merupakan standar proses untuk pemecahan masalah pada unit kajian di bidang *data mining*. Tujuan skema riset adalah melakukan kajian untuk mendapatkan hasil yang sesuai dengan tujuan yang ingin dicapai. Seperti serangkaian tahapan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.3.1. Identifikasi Masalah

Pada tahap ini, beberapa metode diterapkan bertujuan agar analisis lebih terfokus pada identifikasi masalah. Pengumpulan data yang dihasilkan dari setiap metode akan menghasilkan rangkuman *state of the art* untuk pemecahan masalah.

Metode yang digunakan untuk mengidentifikasi masalah adalah sebagai berikut:

1. Studi Literatur

Studi literatur atau penelitian kepustakaan adalah metode pengumpulan data dengan cara membaca literatur dalam buku, jurnal, dan makalah untuk memperoleh konsep teoretis tentang suatu pertanyaan penelitian. Penulis juga melihat penelitian sebelumnya yang terkait dengan apa yang penulis pelajari. Tahap ini menjadi acuan untuk menentukan solusi dari masalah tersebut. Penyelesaian setiap masalah harus didasarkan pada pendekatan yang dianggap relevan dengan masalah yang dihadapi.

2. Observasi

Observasi dan wawancara adalah kegiatan observasi langsung dan sistematis, berupa informasi dan sumber tempat penelitian dan data dikumpulkan. Teknik wawancara yang digunakan dalam kajian ini adalah wawancara semi terstruktur yang disampaikan kepada narasumber yang relevan seperti pemilik perusahaan dan karyawan terkait. Kegiatan ini dirancang untuk menjadikan lokasi penelitian relevan dengan topik kajian yang dilakukan.

3.3.2. Pemahaman Bisnis (*Business Understanding*)

Tahap pemahaman bisnis berfokus pada pemahaman dari perspektif bisnis. Pengetahuan ini kemudian diterjemahkan ke dalam definisi masalah. Selanjutnya akan ditentukan rencana dan strategi untuk mencapai tujuan tersebut. Pada tahap ini perlu dipahami konteks dan tujuan dari proses bisnis yang terkait dengan penjualan produk. Sedangkan tujuan dari data mining dalam kajian ini adalah untuk menggali pengetahuan (*discovering knowledge*) tentang pola penjualan (*pattern*) untuk memprediksi penjualan di masa yang akan datang.

3.3.3. Pemahaman Data (*Data Understanding*)

Pada tahap pemahaman data, memberikan dasar analisis untuk penelitian dengan meringkas dan mengidentifikasi potensi masalah yang terkandung dalam data. Saat memahami data, beberapa proses perlu dilakukan untuk mendeteksi himpunan bagian (*subset*) sebagai referensi untuk membentuk hipotesis informasi tersembunyi. Selama proses ini, eksplorasi data dilakukan dengan menggunakan metode *Exploratory Data Analysis* (EDA), yang merangkum data observasional dengan menerapkan teknik aritmatika dan grafik sederhana untuk memudahkan pemahaman data.

3.3.4. Persiapan Data

Tahap persiapan data, atau pra-pemrosesan data, mencakup semua aktivitas membangun kumpulan data akhir (data yang akan diproses dalam tahap pemodelan) dari data mentah (*primer*) awal yang digunakan dalam pengaturan penambangan data (*set up data mining*). Tahap ini dapat diulang untuk mendapatkan data yang sesuai. Tahap ini juga mencakup langkah-langkah berikut:

1. Membersihkan Data (*Clean Data*)

Sesuaikan format data yang ada sesuai kebutuhan untuk menghilangkan data yang hilang (*missing value*) dan tidak diinginkan.

2. Membangun Data (*Construct Data*)

Mempersiapkan dataset dilakukan, pada tahap ini catatan atau atribut baru dapat diperbarui atau dibuat. Karena data dalam penelitian ini sangat beragam, maka dilakukan proses *labelling* untuk memberikan ID dan nama pembeli secara acak untuk setiap jenis produk.

3. Transformasi Data (*Data Transformation*)

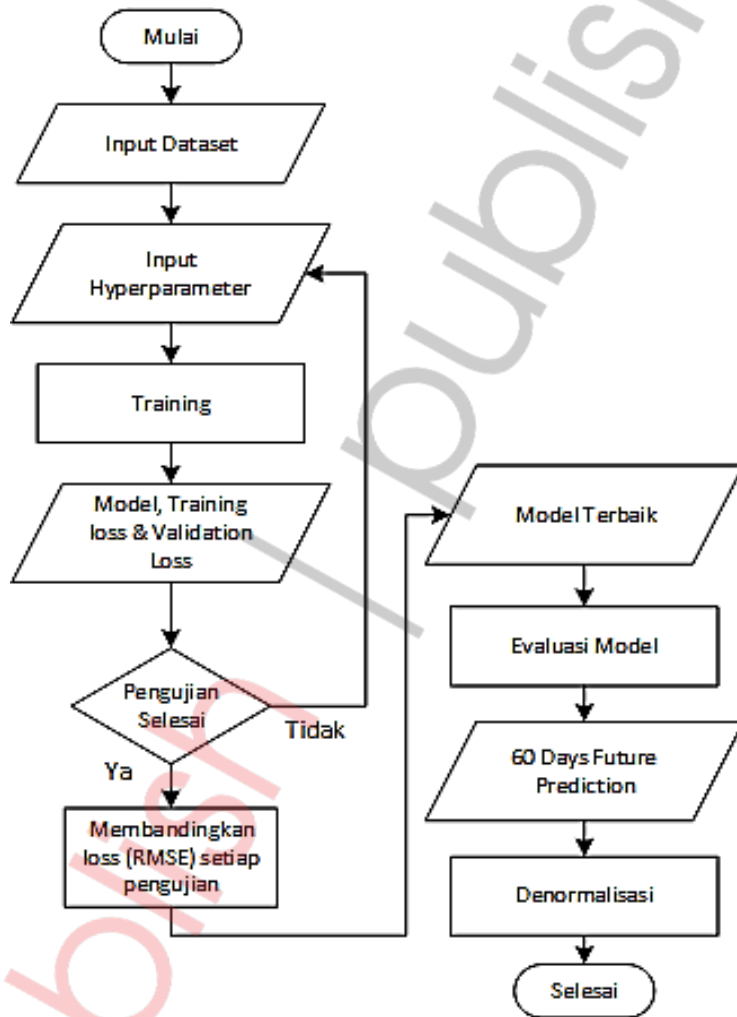
Tahapan ini merupakan cara normalisasi data untuk menyamakan format, berupa skala umum mulai dari 0 sampai 1. Dalam penelitian ini akan digunakan metode Min-MaxScaler dan persamaan sebagai berikut:

$$x' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \dots\dots\dots (6)$$

Keterangan:

- x' = Nilai hasil normalisasi
- X = Nilai data aktual yang akan dinormalisasi
- X_{\min} = Nilai minimum dari data aktual
- X_{\max} = Nilai maksimum dari data actual

3.3.5. Pemodelan (*Modelling*)



Gambar 3.2 Diagram Alir Pemodelan

1. Input Data

Tahap *input* data merupakan tahap di mana data yang telah diproses sebelumnya dimasukkan ke dalam skenario model yang akan diuji.

2. Input Hyperparameter

Tahapan ini merupakan proses *setting scene* parameter agar menghasilkan parameter yang benar sehingga algoritme *Long Term Memory* dapat mempelajari pola-pola yang terdapat dalam data dengan benar. Tahapan melakukan *input hyperparameter* adalah sebagai berikut:

Tabel 3.3 Tahapan Uji Coba Parameter

Tahap Ke	Parameter
1	Komposisi Data
2	<i>Hidden Layer</i>
3	Skenario <i>Dropout</i>
4	<i>Batch Size</i>
5	<i>Epoch</i>

Pada tahap awal komposisi *training* dan *testing* yang menghasilkan *loss* terkecil perlu diuji. Setelah mendapatkan komposisi yang sesuai, gunakan hasil eksperimen untuk menemukan *hidden layer* yang terbaik. Jika banyaknya *hidden layer* mengakibatkan *loss* terkecil, maka tahap selanjutnya adalah membuat skenario *dropout*. *Dropout* digunakan untuk mencegah data mengalami *overfitting*. Jika jumlah *dropout* menyebabkan *error* yang kecil maka akan didapatkan partisi data dengan *layer* yang telah sesuai. Selanjutnya adalah penentuan *batch size* dan *epoch* yang terbaik berdasarkan komposisi data dan *layer* yang telah ditentukan.

3. Training

Pada tahap ini, penerapan uji coba prediksi berbasis *Machine Learning* yaitu algoritme *Long Short Term Memory* dapat diterapkan pada *data training*. Proses *training* dirancang untuk memberikan petunjuk pada data melalui algoritme sehingga mereka dapat belajar dari pola data yang diberikan dan mencari korelasinya.

4. Model, Training Loss dan Validation Loss

Pada tahap ini akan ditampilkan hasil konfigurasi *hyperparameter tuning* berupa nilai *training loss* dan *validation loss*. Hasil dari *training* dan *validation loss* merupakan representasi seberapa baik model yang dihasilkan dengan parameter uji berupa RMSE.

5. Evaluasi Model

Setelah melalui serangkaian prosedur, selanjutnya perlu dilakukan evaluasi kinerja model dengan menggunakan perhitungan *Root Mean Square Error* (RMSE). *Root Mean Square Error* (RMSE) merupakan metode alternatif yang digunakan untuk mengukur tingkat akurasi teknik peramalan yang digunakan. RMSE adalah *mean diferensiasi absolute* antara nilai prediksi dan nilai aktual yang dinyatakan sebagai bentuk persentase. Untuk dapat menghitung nilai RMSE dapat menggunakan persamaan berikut:

- *Root Mean Square Error* (RMSE)

$$\sqrt{\frac{1}{n} \sum_i^n (\tilde{y}_i - y_i)^2} \dots \dots \dots (7)$$

Keterangan:

\tilde{y}_i = Nilai hasil peramalan

y_i = Nilai aktual/Nilai sebenarnya

n = Jumlah data

6. Prediksi 60 Hari Ke Depan

Tahap prediksi adalah proses memprediksi nilai masa depan berdasarkan pola tertentu dalam data. Dalam kajian ini, peramalan digunakan untuk memperkirakan penjualan tahun depan. Hasil dari proses peramalan tersebut kemudian direkomendasikan untuk digunakan sebagai alat pengambilan keputusan.

7. Denormalisasi

Setelah diperoleh hasil prediksi dari proses prediksi, data didenormalisasi, yaitu data diubah kembali ke nilai sebenarnya. Karena data tersebut masih berupa rangkaian interval, yang sebelumnya merupakan data yang dinormalisasi. Tujuan dari denormalisasi adalah untuk mempermudah pembacaan nilai keluaran yang dihasilkan.

Berikut adalah rumus denormalisasi:

$$d = d'(\max - \min) + \min \dots\dots\dots (8)$$

Keterangan:

d = Nilai hasil denormalisasi

d' = Nilai hasil normalisasi

max = Nilai maksimum dari data aktual

min = Nilai minimum dari data aktual

3.3.6. Penyebaran (Deployment)

Pada tahap ini, pengetahuan atau informasi yang telah diperoleh akan dipublikasikan sehingga pemilik dapat menggunakannya sebagai acuan untuk menentukan persediaan. Pada tahap ini, hasil pengolahan data dapat diimplementasikan dalam bentuk prakiraan penjualan untuk digunakan perusahaan sebagai acuan dalam pengambilan keputusan.

BAB 4.

EFEKTIVITAS DATA

4.1. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data transaksi penjualan dari bulan Januari 2018 sampai dengan Maret 2020, dari Ciwo Pet Shop, sebuah bisnis retail yang menjual berbagai kebutuhan hewan peliharaan. Data yang diperoleh kemudian disajikan dalam bentuk .xlsx untuk memudahkan proses pengolahan data tersebut. Data yang diperoleh adalah 7 kolom dan 140.953 baris.

Tabel transaksi penjualan Ciwo Pet Shop yang dinyatakan pada Gambar 4.1 berikut:

Tabel 4.1 Tabel Transaksi Penjualan Ciwo Pet Shop

	Date	Customer	Invoice	City	Product	Quantity	Price
0	01/01/2018	Tilar Handayani	INV/20180101/XXVIII/126179649	Jakarta Timur	MAKANAN KUCING WHISKAS SACHET - WHISKAS BASAH ...	35.0	4350.0
1	01/01/2018	Tilar Handayani	INV/20180101/XXVIII/126180557	Jakarta Timur	MAKANAN KUCING WHISKAS SACHET - WHISKAS BASAH ...	35.0	4350.0
2	01/01/2018	Ika Kurniawati	INV/20180101/XXVIII/126163178	Kota Bekasi	PET CARGO KUCING ANJING KELINCI MUSANG LANDAK ...	1.0	160000.0
3	01/01/2018	Agung Damawan	INV/20180101/XXVIII/126180935	Jakarta Selatan	PASIR KUCING WANGI GUMPAL 25 LITER	1.0	83000.0
4	01/01/2018	Asli	INV/20180101/XXVIII/126214321	Kota Bekasi	PASIR KUCING GUMPAL WANGI KAWAN 10L 10 L 10 LITER	1.0	48000.0
...
140948	31/03/2020	Duta Wiryanto	INV/20200331/XXIII/512250603	Kota Yogyakarta	PASIR KUCING WANGI GUMPAL 25 LITER	3.0	67000.0
140949	31/03/2020	Teguh Triono	INV/20200331/XXIII/512231140	Kota Bekasi	PASIR KUCING WANGI GUMPAL 25 LITER	1.0	67000.0
140950	31/03/2020	Marcellus Adrian	INV/20200331/XXIII/511823299	Kota Bekasi	PASIR KUCING WANGI GUMPAL 25 LITER	1.0	67000.0
140951	31/03/2020	Tjahjadi Wijaja	INV/20200331/XXIII/511370148	Jakarta Barat	REPACK ORI CAT FOOD ORICAT 1KG SEJENIS BOLT MA...	3.0	17000.0
140952	31/03/2020	Ditto Ardyanto	INV/20200331/XXIII/511892475	Kab. Tangerang	GUNTING KUKU HEWAN ANJING KUCING SUGAR GLIDER ...	1.0	15000.0

140953 rows x 7 columns

4.2. Preprocessing Data

Pada tahap *pre-processing* data, ada beberapa tahapan sebagai berikut:

- a. *Data Cleaning* adalah suatu metode untuk menghilangkan data yang tidak diinginkan dan mengisi data yang hilang (*missing values*). Dalam kajian ini, data yang digunakan pertama kali diidentifikasi sebagai kolom yang hilang (*missing*) pada tabel. Setelah itu, periksa korelasi antara tabel yang hilang.

```
1 #Identifikasi Missing Value
2
3 # memeriksa persentase nilai nan yang ada di setiap fitur
4 # membuat daftar fitur yang memiliki missing value
5 features_with_na=[features for features in df.columns if df[features].isnull().sum(>)1]
6
7 ## cetak nama fitur dan persentase missing value
8 for feature in features_with_na:
9 | | print(feature, np.round(df[feature].isnull().mean(), 4), '% missing values')
```

Date 0.4247 % missing values
Customer 0.4251 % missing values
Invoice 0.425 % missing values
City 0.4258 % missing values

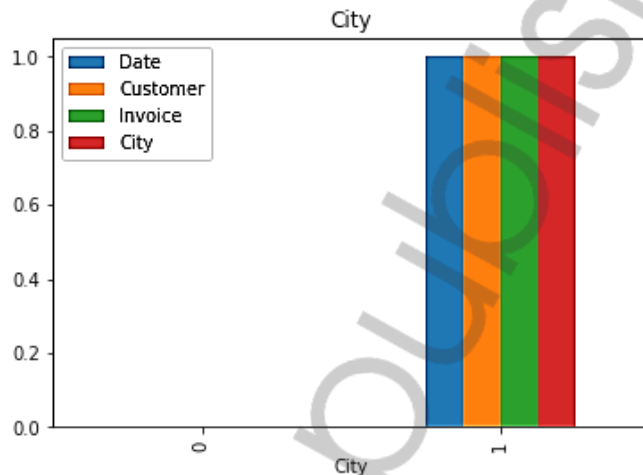
Gambar 4.1 Sebelum Dilakukan Proses *Cleaning*

Berdasarkan hasil presentase "*missing value*" terdapat *missing value* yang cukup besar. Oleh sebab itu, perlu dilihat hubungan atau korelasi antara *missing value* tersebut.

```
1 for feature in features_with_na:
2     df = df.copy()
3
4     # let's make a variable that indicates 1 if the observation was missing or zero otherwise
5     df[feature] = np.where(df[feature].isnull(), 1, 0)
6
7     # let's calculate the mean SalePrice where the information is missing or present
8     df.groupby(feature)['Date', 'Customer', 'Invoice', 'City'].median().plot.bar()
9     plt.title(feature)
10    plt.show()
```

Gambar 4.2. Perintah untuk Melihat Korelasi Antar *Missing Value*

Adapun hasil tampilan diagram dapat dilihat pada Gambar 4.3.



Gambar 4.3 Diagram Korelasi *Missing Value*

Berdasarkan hasil visualisasi bar plot menunjukkan bahwa antar kolom yang terdapat *missing value* saling memiliki keterkaitan satu sama lain. Setelah ditinjau kembali pada dataset yang tersedia perlu dilakukan *forward fill* pada baris yang Nan (*Not a Number*) dan menghapus transaksi tanpa *no invoice*, yang perintahnya dapat dilihat pada Gambar 4.4.

```
1 #forward fill missing value
2 df = df.ffill()
3
4 # Menghapus baris yang tidak memiliki no invoice
5 df.dropna(axis = 0, subset = ['Invoice'], inplace = True)
6 df['Invoice'] = df['Invoice'].astype('str')
```

Gambar 4.4. Perintah untuk *Forward Fill* Pada Baris yang NaN dan Hapus Transaksi Tanpa *No Invoice*

Gambar 4.5 merupakan perintah untuk cleaning dan gambar 4.6 adalah hasil proses cleaning yang menunjukkan tidak ada data yang missing value.

```
1 #check missing value
2 df.isnull().sum()
3
```

Gambar 4.5. Perintah untuk Cek *Missing Value*

```
➤ Date      0
  Customer  0
  Invoice    0
  City      0
  Product   0
  Quantity  0
  Price     0
  dtype: int64
```

Gambar 4.6 Hasil Setelah Tahap *Cleaning*

- b. *Labelling* merupakan cara memberikan ID secara random pada kategori produk dan pelanggan untuk mempermudah pendefinisian dalam bentuk numerik. Perintah untuk *labelling* dapat dilihat pada Gambar 4.7. dan hasil *labelling* dapat lihat pada tabel 4.2.

```
1 df.columns
Index(['Date', 'Customer', 'Invoice', 'City', 'Product', 'Quantity ', 'Price'], dtype='object')

1 df["Product"] = df["Product"].astype('category')
2 df["Customer"] = df["Customer"].astype('category')
3 df["ProductID"] = df["Product"].cat.codes
4 df["CustomerID"] = df["Customer"].cat.codes
5
6 #membuat kolom baru (Quantity*Price) dengan nama total_amount
7 df['Total'] = df['Quantity ']* df['Price']
8
9 # Ubah jadi tipe data tanggal
10 df["Date"] = pd.to_datetime(df["Date"], dayfirst=True)
```

Gambar 4.7. Perintah *Labelling*

Tabel 4.2 Hasil Setelah Proses Labelling

	Date	Customer	Invoice	City	Product	Quantity	Price	ProductID	CustomerID	Total
0	2018-01-01	Tilar Handayani	INV/20180101XXVIII/126179649	Jakarta Timur	MAKANAN KUCING WHISKAS SACHET - WHISKAS BASAH ...	35.0	4350.0	660	23137	152250.0
1	2018-01-01	Tilar Handayani	INV/20180101XXVIII/126180557	Jakarta Timur	MAKANAN KUCING WHISKAS SACHET - WHISKAS BASAH ...	35.0	4350.0	660	23137	152250.0
2	2018-01-01	Ika Kurniawati	INV/20180101XXVIII/126163178	Kota Bekasi	PET CARGO KUCING ANJING KELINCI MUSANG LANDAK ...	1.0	160000.0	942	11390	160000.0
3	2018-01-01	Agung Darmawan	INV/20180101XXVIII/126180935	Jakarta Selatan	PASIR KUCING WANGI GUMPAL 25 LITER	1.0	83000.0	891	760	83000.0
4	2018-01-01	Asri	INV/20180101XXVIII/126214321	Kota Bekasi	PASIR KUCING GUMPAL WANGI KAWAN 10L 10 L 10 LITER	1.0	48000.0	878	3203	48000.0
...
140948	2020-03-31	Duta Wiryanto	INV/20200331XXVIII/512250603	Kota Yogyakarta	PASIR KUCING WANGI GUMPAL 25 LITER	3.0	67000.0	891	6940	201000.0
140949	2020-03-31	Teguh Triono	INV/20200331XXVIII/512231140	Kota Bekasi	PASIR KUCING WANGI GUMPAL 25 LITER	1.0	67000.0	891	22939	67000.0
140950	2020-03-31	Marcellus Adrian	INV/20200331XXVIII/511823299	Kota Bekasi	PASIR KUCING WANGI GUMPAL 25 LITER	1.0	67000.0	891	14701	67000.0
140951	2020-03-31	Tjahjadi Widjaja	INV/20200331XXVIII/511370148	Jakarta Barat	REPACK ORI CAT FOOD ORICAT 1KG SEJENIS BOLT MA...	3.0	17000.0	1074	23254	51000.0
140952	2020-03-31	Ditto Ardianto	INV/20200331XXVIII/511892475	Kab. Tangerang	GUNTING KUKU HEWAN ANJING KUCING SUGAR GLIDER ...	1.0	15000.0	292	6697	15000.0

140953 rows * 10 columns

c. *Summary* jumlah penjualan dalam format harian untuk mempermudah proses *modelling* agar algoritme yang digunakan dapat mempelajari pola dengan lebih baik. Perintah untuk melihat jumlah penjualan harian dapat dilihat pada Gambar 4.8 dan hasil *summary* penjualan harian dapat dilihat pada tabel 4.3.

```
1 df.columns
Index(['Date', 'Customer', 'Invoice', 'City', 'Product', 'Quantity', 'Price',
      'ProductID', 'CustomerID', 'Total'],
      dtype='object')

1 # Jumlah Transaksi Perhari
2 df2 = df['Total'].groupby(df['Date']).sum().reset_index().round()
3 df3 = df['Quantity'].groupby(df['Date']).sum().reset_index().round()
4 df4 = df['Price'].groupby(df['Date']).sum().reset_index().round()
5 df_fix = df4.join(df3['Quantity'])
6 df_fix = df_fix.join(df2['Total'])
7
8 # Ubah nama kolom
9 df_fix = df_fix.rename(columns={"Quantity": "Quantity"})
```

Gambar 4.8 Perintah untuk Summary Jumlah Penjualan Harian

Tabel 4.3 Hasil Summary Total Penjualan per Hari

	Date	Price	Quantity	Total
0	2018-01-01	1190200.0	92.0	1781500.0
1	2018-01-02	1830100.0	126.0	2949200.0
2	2018-01-03	4405100.0	165.0	5513300.0
3	2018-01-04	5931450.0	327.0	8161250.0
4	2018-01-05	3536400.0	273.0	5148500.0
...
814	2020-03-27	14485200.0	760.0	23278600.0
815	2020-03-28	16073000.0	667.0	25065300.0
816	2020-03-29	15537100.0	577.0	23371700.0
817	2020-03-30	11885683.0	746.0	21249710.0
818	2020-03-31	12882950.0	627.0	22781750.0

819 rows × 4 columns

- d. *Data transformation* ialah kaidah menormalkan data untuk menyamakan format pada skala umum (0-1). Dalam kajian ini, kaidah Min-MaxScaler digunakan untuk pemodelan data. Proses normalisasi menggunakan Persamaan 6. Contoh keputusan ternormal berikut dibentangkan dalam bentuk tabel. Keputusan normalisasi keseluruhan dapat didapati di Lampiran 1.

Tabel 4.4 Contoh Hasil Normalisasi

Date	Quantity	X'
01/01/2018	92	0,00613079
02/01/2018	126	0,01771117
03/01/2018	165	0,03099455
04/01/2018	327	0,08617166
05/01/2018	273	0,06777929
06/01/2018	488	0,14100817
07/01/2018	372	0,10149863
08/01/2018	575	0,17064032

Date	Quantity	X'
09/01/2018	426	0,11989109
10/01/2018	327	0,08617166

Contoh perhitungan yang dapat dilakukan secara manual menggunakan persamaan *MinMax-Scalar* adalah sebagai berikut:

$$X_n = \frac{X_0 - X_{\min}}{X_{\max} - X_{\min}}$$

$$X_1 = \frac{92 - 74}{3010 - 74}$$

$$X_1 = \frac{18}{2936}$$

$$X_1 = 0,00613079$$

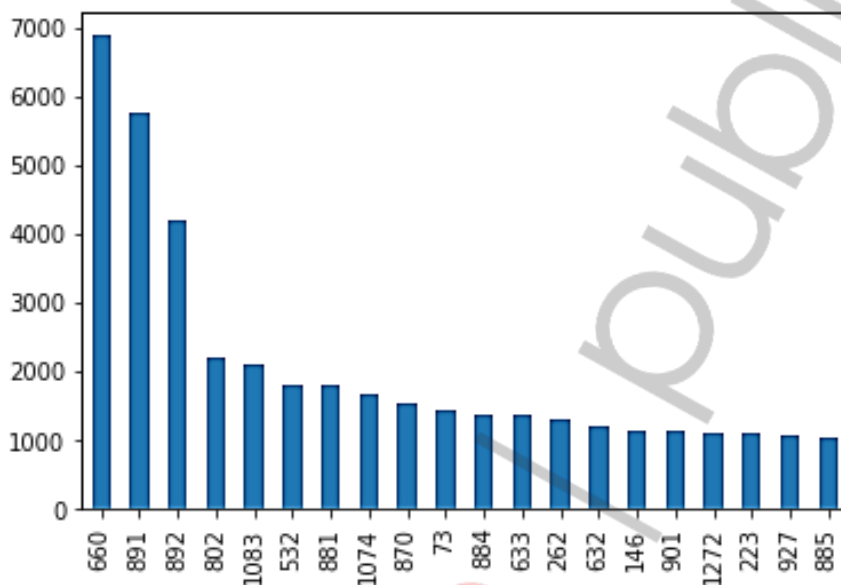
4.3. Pemahaman Data

Tahap pemahaman data dapat memberikan landasan analitis untuk penelitian dengan meringkas dan mengidentifikasi potensi masalah yang terkandung dalam data sebelum pemodelan. Tahap pemahaman data menggunakan metode *Exploratory Data Analysis* (EDA), teknik eksplorasi data yang menggunakan teknik aritmatika sederhana dan teknik grafis untuk meringkas data observasi untuk memudahkan pemahaman data. Hasil analisis data eksplorasi adalah sebagai berikut:

Tabel 4.5 Distribusi Data Transaksi

	Quantity	Price	ProductID	CustomerID	Total
count	140953.000000	1.409530e+05	140953.000000	140953.000000	1.409530e+05
mean	4.491980	8.250755e+04	735.295198	13093.044575	1.260052e+05
std	15.581414	1.505750e+05	386.411018	7450.191321	2.350766e+05
min	1.000000	1.000000e+02	0.000000	0.000000	1.000000e+02
25%	1.000000	1.400000e+04	483.000000	6675.000000	2.500000e+04
50%	1.000000	2.500000e+04	731.000000	12959.000000	6.750000e+04
75%	3.000000	7.800000e+04	927.000000	19464.000000	1.360000e+05
max	3000.000000	8.550000e+06	1522.000000	26554.000000	1.800000e+07

Hasil distribusi data menunjukkan bahwa terdapat perbedaan yang signifikan antara median (50%) dan mean, sehingga dapat disimpulkan bahwa jumlah transaksi antar pembeli sangat beragam.



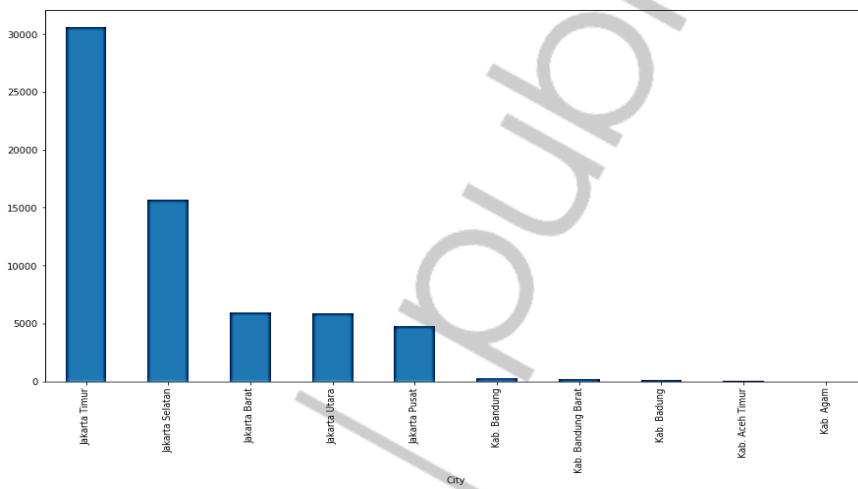
Gambar 4.9 Diagram Produk Terlaris Berdasarkan Produk ID

Dilihat dari hasil eksplorasi, ragam produk Ciwo Pet Shop sangat kaya. Ciwo Pet Shop memiliki 1.523 jenis produk. Grafik menunjukkan 10 buku terlaris teratas dengan jumlah produk terjual tertinggi. Deskripsi di bawah untuk setiap ID Produk adalah sebagai berikut:

Tabel 4.6 Deskripsi Produk

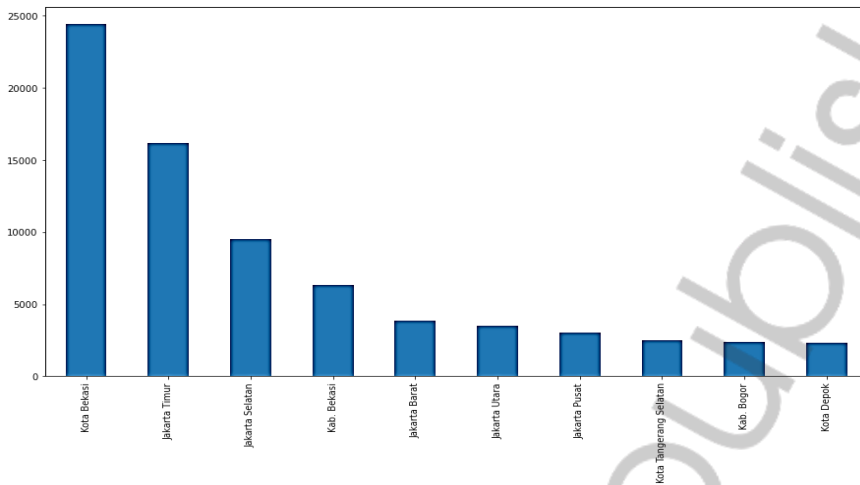
660	WHISKAS BASAH 85GR 85 GR-JUNIOR
891	PASIR KUCING WANGI GUMPAL 25 LITER
892	PASIR CUB AND KIT 20 KG
802	OBAT KUTU TETES DETICK DETIK DETIC
1083	REPACK PROPLAN KITTEN 500 GR

532	MAKANAN KUCING BOLT CAT TUNA 1KG
881	PASIR WANGI MARKOTOP 25LT-CATSAND KUCING
1074	REPACK ORICAT 1 KG
870	PASIR BENTONIE SEJENIS MAXI
73	CAT CHOIZE REPACK



Gambar 4.10 Diagram Distribusi Kota Berdasarkan Frekuensi Pembelian

Menurut hasil eksplorasi data, kota Jakarta Timur memiliki frekuensi pembelian tertinggi dibandingkan dengan 266 kota atau wilayah lainnya.



Gambar 4.11 Diagram Distribusi Kota Berdasarkan Banyaknya Variasi Produk

Berdasarkan hasil eksplorasi data, Kota Bekasi memiliki tingkat pembelian dengan variasi produk tertinggi dibandingkan dengan 266 kota atau wilayah lainnya.

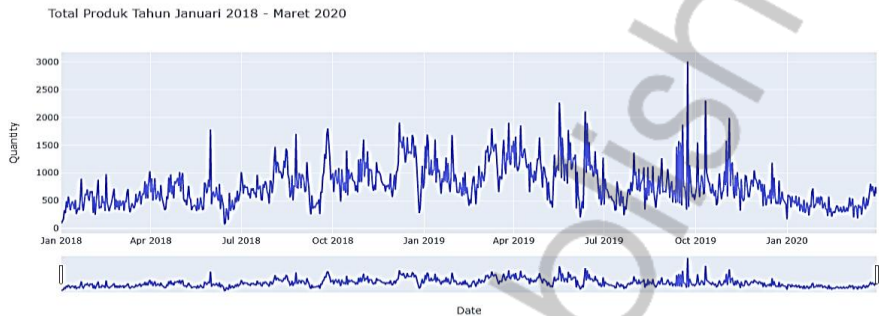
Untuk mengetahui total penjualan produk periode Januari 2018 hingga Maret 2020 menggunakan perintah pada Gambar 4.12. Gambar 4.13 merupakan visualisasi grafik penjualan produk periode Januari 2018 hingga Maret 2020.

```

1 # Total Produk Terjual Tahun Januari 2018 - Maret 2020
2
3 import plotly.graph_objects as go
4 import plotly.express as px
5 import pandas as pd
6
7 fig = px.line(df_fix, x="Date", y="Quantity", title='Total Produk Tahun Januari 2018 - Maret 2020')
8
9
10 fig.update_xaxes(rangeslider_visible=True)
11 fig.show()

```

Gambar 4.12 Perintah untuk menampilkan grafik penjualan produk: Januari 2018-Maret 2020

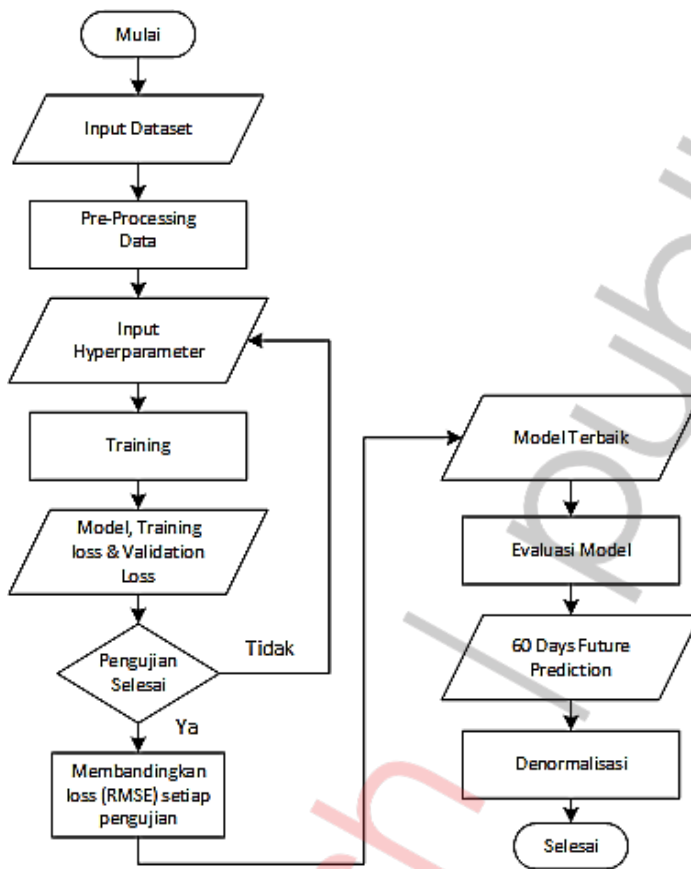


Gambar 4.13 Grafik Penjualan Januari 2018-Maret 2020

Berdasarkan hasil eksplorasi data, diasumsikan volume penjualan tertinggi berada di Kota Bekasi dan Jakarta Timur, dengan pertumbuhan signifikan pada Oktober 2019, dan penjualan akan berfluktuasi.

4.4. Pemodelan

Saat membuat model LSTM, beberapa parameter yang digunakan antara lain partisi data, jumlah lapisan tersembunyi, skenario *dropout* untuk mencegah *overfitting*, jumlah neuron, *epoch* yang menggambarkan jumlah iterasi *training*, ukuran *batch* adalah jumlah data *training* yang harus diperhatikan untuk setiap proses pemutakhiran bobot.



Gambar 4.14 Diagram Alir Pemodelan

Setiap kerugian dan nilai RMSE yang dihasilkan dari setiap percobaan dibandingkan dan dianalisis dari beberapa model yang dihasilkan. Nilai parameter yang digunakan untuk membuat model LSTM ditentukan sebagai berikut:

Tabel 4.7 Parameter Pengujian LSTM

Parameter	Jumlah	Keterangan
<i>Input</i>	3	Data Total Penjualan Harian
<i>Hidden Layer</i>	<i>Trial Error</i>	30-120 <i>neuron</i>

Parameter	Jumlah	Keterangan
<i>Dropout</i>	<i>Trial Error</i>	1-3 layer
<i>Epoch</i>	<i>Trial Error</i>	50-200
<i>Batch Size</i>	<i>Trial Error</i>	10-400
Optimizer	1	Adam
Output	1	Jumlah penjualan produk per hari

Setiap parameter akan diuji, dan hasil pengujian parameter yang menghasilkan kerugian (*loss*) yang cukup baik akan digunakan untuk pengujian berikutnya guna menghasilkan model terbaik. Berikut adalah tahapan pengujian parametrik:

Tabel 4.8 Tahapan Pengujian Parameter

Tahap Ke	Parameter
1	Komposisi Data
2	<i>Hidden Layer</i>
3	Skenario <i>Dropout</i>
4	<i>Batch Size</i>
5	<i>Epoch</i>

4.4.1. Hasil Pengujian

Pada percobaan pertama, percobaan dilakukan untuk mendapatkan komposisi data yang optimal. *Hyperparameter* yang digunakan untuk mencari jumlah kombinasi data adalah 30 *batch size* dan 100 *epoch*, serta menggunakan Adam optimizer. Menurut hasil percobaan, hasil akhir ditunjukkan pada Tabel 4.9.

Tabel 4.9 Hasil Pengujian Partisi Data

Partisi Data					
Training	Testing	<i>Loss</i>	<i>Val Loss</i>	Total RMSE	Total RMSE
		(MSE)	(MSE)	Training	Testing
50	50	0,0078	0,0120	0,0802	0,1094
60	40	0,0077	0,0115	0,0844	0,1070

Partisi Data					
Training	Testing	Loss	Val Loss	Total RMSE	Total RMSE
		(MSE)	(MSE)	Training	Testing
65	35	0,0092	0,0074	0,0913	0,0858
70	30	0,0093	0,0051	0,9638	0,0714
80	20	0,0106	0,0058	0,1028	0,0760

Hasil eksperimen pencarian partisi data ditunjukkan pada Tabel 4.9. Berdasarkan hasil percobaan, kerugian (*loss*) yang diperoleh pada pengujian 1 dan 2 tergolong *overfitting* karena kerugian pelatihan lebih kecil dari kerugian validasi (*validation loss*). Pada percobaan 3 sampai 5, diklasifikasikan sebagai *underfitting* karena kerugian pelatihan (*training loss*) lebih besar dari kerugian validasi. Untuk mendapatkan model terbaik dari hasil percobaan pertama diambil dari selisih terkecil antara nilai latih RMSE dengan nilai uji RMSE. Berdasarkan pengujian pertama, jumlah kombinasi data yang optimal dari dataset yang dilatih terletak pada percobaan ketiga, dengan nilai 0,0913 untuk pelatihan RMSE dan 0,0858 untuk pengujian RMSE. Sedangkan untuk jumlah kombinasi data yang belum optimal, pada percobaan keempat nilai RMSE Training sebesar 0,9638 dan nilai RMSE Testing sebesar 0,0714.

Tabel 4.10 Partisi Dataset

Dataset	Partisi	Jumlah Data
Training	01 Januari 2018 s/d 17 Juni 2019	532
Testing	18 Juni 2019 s/d 31Maret 2020	287

Setelah hasil pengujian pertama diperoleh, maka dilakukan pengujian tahap kedua untuk mendapatkan lapisan tersembunyi yang optimal. Pengujian tahap 2 dengan ukuran *batch size* 30, *epoch* 100, dan kombinasi data 65:35, hasil dari pengujian pertama.

Tabel 4.11 Hasil Pengujian *Hidden Layer*

Komposisi <i>Hidden Layer</i>				
<i>Hidden Layer</i>	<i>Loss</i>	<i>Val Loss</i>	Total RMSE	Total RMSE
	(MSE)	(MSE)	Training	Testing
3	0,0089	0,0079	0,0910	0,0889
4	0,0083	0,0090	0,0900	0,0949
5	0,0134	0,0142	0,1145	0,1193
6	0,0132	0,0173	0,1140	0,1314
7	0,0133	0,0199	0,1141	0,1409

Hasil pengujian untuk menemukan lapisan tersembunyi (*hidden layer*) ditunjukkan pada Tabel 4.11. Berdasarkan hasil percobaan, nilai *loss* yang diperoleh pada pengujian 2 sampai dengan 5 cenderung *overfitting* karena lebih kecil dari nilai *validation loss*. Model yang optimal didapatkan dari hasil eksperimen tersebut maka dapat dilihat dari nilai *loss* dan *validation loss* yang memiliki selisih kecil, didukung oleh nilai RMSE Train dan RMSE Testing yang juga memiliki selisih yang paling kecil.

Berdasarkan hasil pengujian, pada penelitian ini penambahan jumlah *hidden layer* tidak berpengaruh nyata terhadap penurunan nilai parameter pengujian. Hal ini dibuktikan dengan peningkatan nilai *loss* yang dihasilkan pada percobaan 2 menjadi 5 pada tabel di atas. Dari hasil pengujian ini, jumlah *hidden layer* yang optimal pada dataset yang dilatih adalah pada *hidden 3* neuron, dengan nilai pelatihan RMSE sebesar 0,0910 dan nilai pengujian RMSE sebesar 0,0889. Sedangkan untuk jumlah *hidden layer* yang kurang optimal terletak pada percobaan ke-5 dengan jumlah *hidden layer 7*.

Setelah pengujian tahap kedua, dilanjutkan ke pengujian tahap ketiga untuk mendapatkan skenario putus sekolah terbaik. *Dropout* digunakan untuk mencegah *overfitting* menjadi terlalu besar. Pengujian tahap ketiga dilakukan dengan menggunakan *batch size 50*, *epoch 100*, kombinasi data 60:35 berdasarkan hasil

pengujian pertama, dan jumlah *hidden layer* 3 berdasarkan hasil pengujian kedua.

Tabel 4.12 Hasil Pengujian Skenario *Dropout*

Skenario <i>Dropout</i>				
<i>Dropout</i>	<i>Loss</i>	<i>Val Loss</i>	Total RMSE	Total RMSE
	(MSE)	(MSE)	Training	Testing
1	0,0097	0,0073	0,0857	0,0941
2	0,0102	0,0067	0,1021	0,0821
3	0,0091	0,0080	0,0906	0,0895

Hasil percobaan ketiga ditunjukkan pada Tabel 4.12 di atas. Berdasarkan hasil pengujian, semakin banyak jumlah *dropout* maka nilai hasil memiliki cenderung naik turun, sehingga dapat disimpulkan bahwa jika jumlah *hidden layer* yang telah ditentukan terlampaui maka peningkatan nilai *dropout* tidak berpengaruh signifikan. Pada percobaan kedua nilai *loss training* cenderung meningkat, namun nilai validasi cenderung menurun. Sedangkan pada pengujian ketiga nilai *training loss* cenderung menurun dan nilai *validation loss* cenderung meningkat.

Pada tahap ini ditentukan nilai *dropout* yang optimal dengan cara membandingkan nilai RMSE Training dan hasil RMSE Testing dengan selisih terkecil pada masing-masing tes. Berdasarkan hasil tahapan saat ini, nilai *dropout* terbaik ada pada tes ketiga, nilai RMSE Training 0,0906, dan nilai RMSE Testing 0,0895.

Setelah pengujian tahap ketiga selesai, pengujian tahap keempat dilakukan untuk mendapatkan nilai *batch size* terbaik. Pengujian tahap keempat menggunakan *batch size* 50, *epoch* 100, kombinasi data 60:35 berdasarkan hasil pengujian pertama dan jumlah *hidden layer* 3 berdasarkan hasil pengujian kedua dan 3 jumlah putus sekolah berdasarkan tes ketiga.

Tabel 4.13 Hasil Pengujian *Batch Size*

Jumlah <i>Batch Size</i>				
<i>Batch Size</i>	<i>Loss</i>	<i>Val Loss</i>	Total RMSE	Total RMSE
	(MSE)	(MSE)	Training	Testing
10	0,0088	0,0073	0,9031	0,0856
15	0,0086	0,0069	0,0898	0,0829
20	0,0079	0,0064	0,0899	0,0801
25	0,0092	0,0070	0,0911	0,0838
30	0,0097	0,0077	0,0914	0,0879
35	0,0087	0,0068	0,0905	0,0824
40	0,0094	0,0069	0,0926	0,0832

Hasil percobaan keempat ditunjukkan pada Tabel 4.13 di atas. Menurut hasil pengujian, penambahan nilai *batch size* cenderung menghasilkan nilai kerugian yang fluktuatif. Sehingga dapat disimpulkan bahwa tidak ada regulasi khusus untuk nilai *batch size* yang optimal. Hal ini dikarenakan setiap data memiliki cara pengolahan tersendiri dalam menentukan *batch size* yang benar.

Pada tahap ini, nilai *batch size* yang optimal ditentukan dengan membandingkan nilai RMSE Training dan hasil pengujian RMSE Training dengan selisih terkecil pada setiap pengujian. Berdasarkan hasil pada tahap ini, nilai *batch size* yang optimal terdapat pada pengujian kelima, dengan nilai RMSE Training 0,0914 dan nilai RMSE Training test 0,0879.

Setelah pengujian tahap ketiga selesai dilakukan pengujian tahap kelima untuk mendapatkan nilai *epoch* terbaik. Untuk penggunaan tes tahap kelima, komposisi datanya adalah 60:35 berdasarkan hasil tes pertama dan jumlah *hidden layer* 3 berdasarkan hasil tes kedua dan 3 berdasarkan jumlah *dropout* berdasarkan hasil pengujian tahap keempat, Pengujian ketiga dijalankan dengan *batch size* 30.

Tabel 4.14 Hasil Pengujian *Epoch*

Jumlah <i>Epoch</i>				
<i>Epoch</i>	<i>Loss</i>	Val <i>Loss</i>	Total RMSE	Total RMSE
	(MSE)	(MSE)	Training	Testing
50	0,0111	0,0077	0,0977	0,0878
80	0,0096	0,0067	0,0997	0,0816
100	0,0087	0,0074	0,0881	0,0860
120	0,0079	0,0089	0,0856	0,0943
150	0,0076	0,0071	0,0855	0,0844
180	0,0087	0,0090	0,0898	0,0949
200	0,0076	0,0095	0,0837	0,0974

Hasil percobaan kelima ditunjukkan pada Tabel 4.14 di atas. Berdasarkan hasil pengujian, semakin bertambahnya jumlah *epoch* maka nilai hasil cenderung berfluktuasi, sehingga dapat disimpulkan bahwa penentuan jumlah *epoch* yang optimal untuk mengurangi jumlah *loss* didasarkan pada percobaan. Hal ini dikarenakan setiap data memiliki pola dan kompleksitas yang berbeda. Pada tahap ini, nilai *epoch* optimal ditentukan dengan membandingkan nilai hasil RMSE Training dan RMSE Testing dengan selisih terkecil pada setiap pengujian. Berdasarkan hasil pada tahap ini, nilai *batch size* terbaik terdapat pada pengujian ketiga, dengan nilai RMSE Training 0,0855 dan nilai RMSE Testing 0,0844.

4.4.2. Hasil Pengaplikasian

Prediksi dilakukan dengan menentukan *hyperparameter* dari model terbaik yang diperoleh selama pengujian menggunakan algoritme *Long Short Term Memory* (LSTM) dengan arsitektur yang dimodelkan pada Gambar 4.15, dan hasil peramalan penjualan menggunakan algoritme LSTM dapat dilihat pada Gambar 4.16. Hasilnya berupa perbandingan antara data aktual dengan hasil prediksi untuk training dan testing.

Model: "sequential"

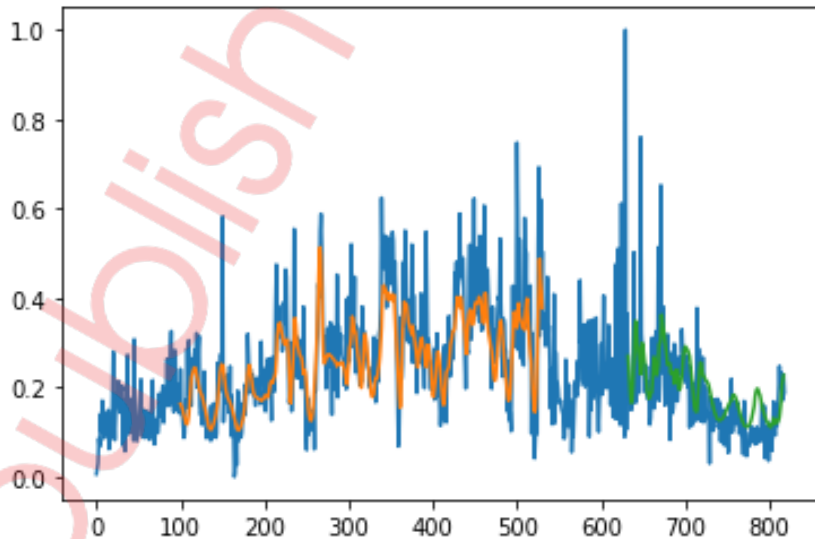
Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 100, 100)	40800
dropout (Dropout)	(None, 100, 100)	0
lstm_1 (LSTM)	(None, 100, 100)	80400
dropout_1 (Dropout)	(None, 100, 100)	0
lstm_2 (LSTM)	(None, 100)	80400
dropout_2 (Dropout)	(None, 100)	0
dense (Dense)	(None, 1)	101

Total params: 201,701

Trainable params: 201,701

Non-trainable params: 0

Gambar 4.15 Arsitektur *Layer* LSTM



Gambar 4.16 Grafik Hasil Prediksi

4.4.3. Evaluasi Model

Setelah hasil prediksi penjualan diperoleh, RMSE digunakan untuk menghitung akurasi model. Hasil untuk prediksi training dan testing disajikan dalam tabel yang disertakan dalam Lampiran 2 dan 3. Di bawah ini adalah contoh tabel hasil prediksi:

Tabel 4.15 Contoh Hasil Data Training

y_train	train_predict	Error	Absolute Value of Error	Square of Error
		Y-Y'	Y-Y'	(Y-Y')^2
0,24114441	0,16321075	0,07793366	0,07793366	0,006073655
0,16859673	0,1588028	0,00979393	0,00979393	9,59211E-05
0,14339237	0,15410239	-0,01071002	0,01071002	0,000114705
0,11580381	0,14753246	-0,03172865	0,03172865	0,001006707
0,08753406	0,13931462	-0,05178056	0,05178056	0,002681226
0,09604905	0,12974457	-0,03369552	0,03369552	0,001135388
0,17132153	0,12256317	0,04875836	0,04875836	0,002377378
0,15497275	0,11816353	0,03680922	0,03680922	0,001354919
0,28746594	0,11797047	0,16949547	0,16949547	0,028728714
0,19414169	0,12738954	0,06675215	0,06675215	0,00445585
0,30551771	0,14062479	0,16489292	0,16489292	0,027189675

Tabel 4.16 Contoh Hasil Prediksi Data Testing

y_test	test_predict	Error	Absolute Value of Error	Square of Error
		Y-Y'	Y-Y'	(Y-Y')^2
0,27316076	0,2704018	0,00275896	0,00275896	7,61186E-06
0,19277929	0,23946959	-0,0466903	0,0466903	0,002179984
0,16519074	0,20686641	-0,04167567	0,04167567	0,001736861
0,1863079	0,17960873	0,00669917	0,00669917	4,48789E-05
0,14952316	0,16540805	-0,01588489	0,01588489	0,00025233
0,18017711	0,17102489	0,00915222	0,00915222	8,37631E-05
0,22513624	0,20790431	0,01723193	0,01723193	0,000296939
0,5023842	0,2564119	0,2459723	0,2459723	0,060502372
0,25953678	0,32995042	-0,07041364	0,07041364	0,004958081
0,16553134	0,34898752	-0,18345618	0,18345618	0,03365617

- *Root Mean Square Error (RMSE) Training*

$$\text{RMSE} = \sqrt{\frac{1}{431} \sum_i^{431} (y' - y_i)^2}$$

$$\text{RMSE} = \sqrt{3,147983731}$$

$$\text{RMSE} = 0,0855$$

- *Root Mean Square Error (RMSE) Testing*

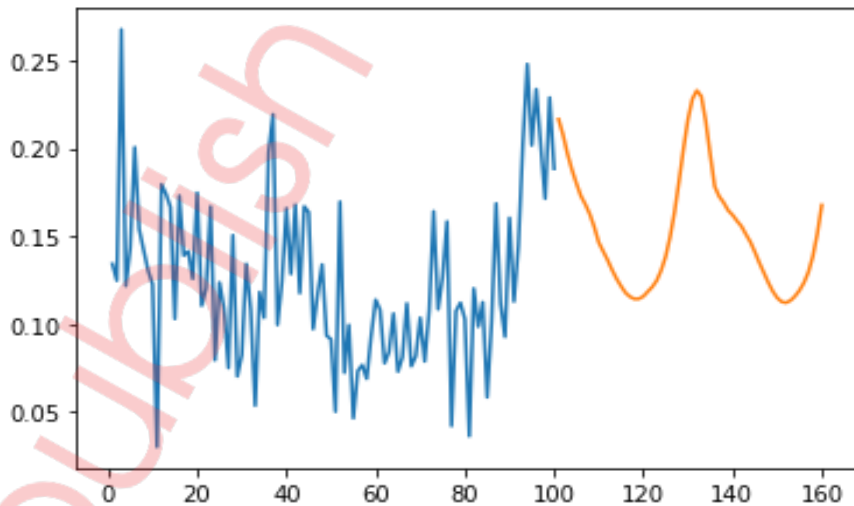
$$\text{RMSE} = \sqrt{\frac{1}{185} \sum_i^{185} (y' - y_i)^2}$$

$$\text{RMSE} = \sqrt{1,323776708}$$

$$\text{RMSE} = 0,0846$$

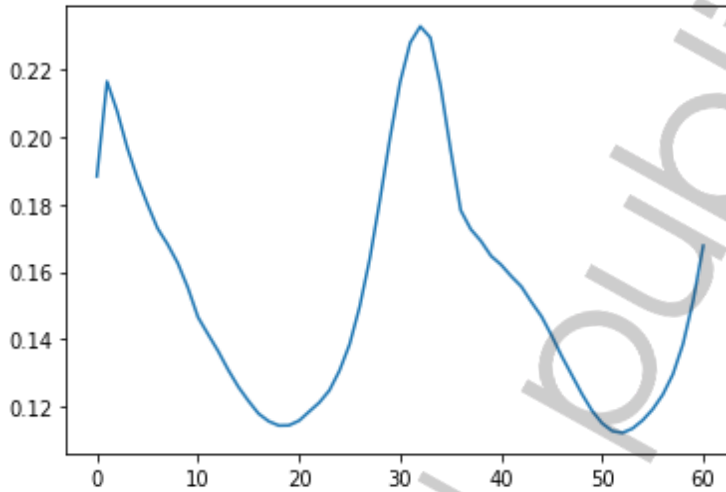
4.4.4. Prediksi 60 Hari Ke Depan

Menggunakan kombinasi *hyperparameter* yang sudah dipasang, para peneliti menggunakan *lookback* 100 hari dari data uji untuk memprediksi kemungkinan penjualan 60 hari ke depan. Hasilnya disajikan dalam bentuk grafik dan tabel sebagai berikut:

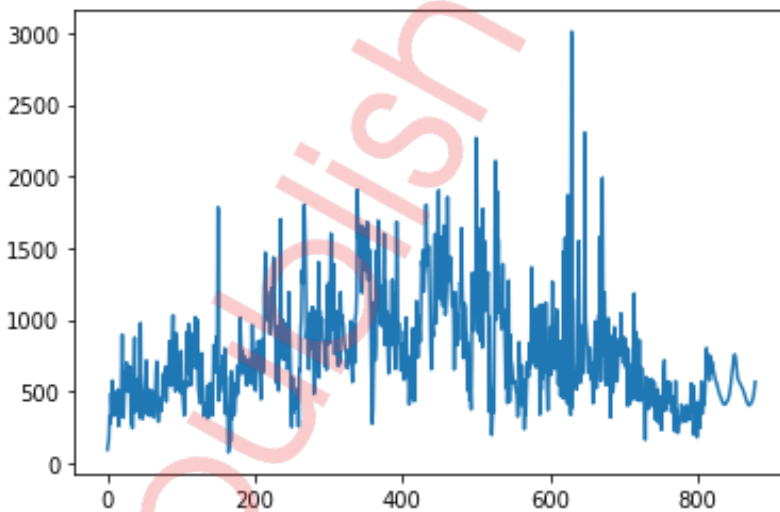


Gambar 4.17 Hasil Prediksi Penjualan 60 Hari Mendatang

Gambar di atas menunjukkan grafik dari hari ke 781 (biru) hingga prediksi ditampilkan sebagai garis grafik oranye.



Gambar 4.18 Grafik Hasil Prediksi 60 Hari



Gambar 4.19 Grafik Penggabungan Aktual dan Hasil Prediksi

Prediksi dihasilkan sebagai data, dengan *range interval* 0 sampai dengan 1, jadi untuk membuat prediksi lebih mudah dibaca, lakukan denormalisasi menggunakan persamaan 8. Berikut adalah hasil prediksi penjualan untuk 60 hari ke depan:

Tabel 4.17 Hasil Prediksi Penjualan Selama 60 Hari Ke Depan

Tanggal	Sebelum di Denormalisasi	Setelah di Denormalisasi
01/04/2020	0,216552	709,7967
02/04/2020	0,207718	683,8608
03/04/2020	0,196872	652,0167
04/04/2020	0,187789	625,3489
05/04/2020	0,180030	602,5673
06/04/2020	0,172879	581,5717
07/04/2020	0,168107	567,5609
08/04/2020	0,162540	551,2181
09/04/2020	0,155221	529,7274
10/04/2020	0,146479	504,0633
11/04/2020	0,141372	489,0683
12/04/2020	0,136363	474,3621
13/04/2020	0,130754	457,8939
14/04/2020	0,125733	443,1516
15/04/2020	0,121549	430,8681
16/04/2020	0,117776	419,7899
17/04/2020	0,115507	413,1275
18/04/2020	0,114244	409,4218
19/04/2020	0,114300	409,5860
20/04/2020	0,115683	413,6443
21/04/2020	0,118383	421,5734
22/04/2020	0,121072	429,4676
23/04/2020	0,124669	440,0286
24/04/2020	0,130526	457,2230
25/04/2020	0,138339	480,1635
26/04/2020	0,149569	513,1344
27/04/2020	0,163793	554,8971
28/04/2020	0,181184	605,9563

Tanggal	Sebelum di Denormalisasi	Setelah di Denormalisasi
29/04/2020	0,199652	660,1774
30/04/2020	0,216345	709,1891
01/05/2020	0,228074	743,6244
02/05/2020	0,232923	757,8619
03/05/2020	0,229565	748,0040
04/05/2020	0,215251	705,9777
05/05/2020	0,196280	650,2773
06/05/2020	0,178239	597,3099
07/05/2020	0,172594	580,7359
08/05/2020	0,169118	570,5293
09/05/2020	0,164683	557,5087
10/05/2020	0,162036	549,7376
11/05/2020	0,158653	539,8038
12/05/2020	0,155582	530,7893
13/05/2020	0,150981	517,2791
14/05/2020	0,146667	504,6140
15/05/2020	0,140997	487,9664
16/05/2020	0,134859	469,9448
17/05/2020	0,129246	453,4667
18/05/2020	0,123636	436,9940
19/05/2020	0,118651	422,3595
20/05/2020	0,114899	411,3435
21/05/2020	0,112626	404,6704
22/05/2020	0,112058	403,0029
23/05/2020	0,113365	406,8385
24/05/2020	0,115806	414,0075
25/05/2020	0,119093	423,6583
26/05/2020	0,123446	436,4372
27/05/2020	0,129486	454,1708
28/05/2020	0,138503	480,6462
29/05/2020	0,151356	518,3798
30/05/2020	0,167701	566,3695



BAB 5.

KONKLUSI

Dari penelitian yang telah dilakukan dapat disimpulkan bahwa model terbaik yang diperoleh dari *Long Short Term Memory* (LSTM) dengan *hyperparameter batch size 30, epoch 150, 3 hidden layer* dan *3 dropout* menghasilkan RMSE Training sebesar 0,0855 dan RMSE Testing 0,0846. Selama pelatihan, itu diklasifikasikan sebagai *overfitting*, yaitu nilai kerugian validasi (*validation loss*) lebih besar dari *training loss*. *Overfitting* terjadi karena data *training* yang dilatih lebih mudah dipelajari daripada data testing. Selanjutnya, besar kecilnya nilai kerugian (*loss*) sangat dipengaruhi oleh konfigurasi *hyperparameter tuning* seperti partisi data, *hidden layer, batch size, dan epoch*.



TENTANG PENULIS



Ali Khumaidi adalah dosen tetap pada program studi Teknik Informatika, Fakultas Teknik, Universitas Krisnadwipayana. Bidang penelitiannya yaitu machine learning, data science, optimasi, Internet of Things (IoT) dan software engineering. Beberapa penelitiannya telah dipublikasikan dalam jurnal dan konferensi nasional dan internasional. Telah memperoleh beberapa hibah dari Kementerian Pendidikan dan Kementerian Keuangan Republik Indonesia. Saat ini cukup aktif dalam penelitian non-destruktif berbasis machine learning. Sebagai dosen senior telah meluluskan mahasiswa dengan topik penelitian komputasi dan optimasi, data science, software engineering, IoT dan machine learning. ORCID: 0000-0002-4319-5018.



Ika Ayu Nirmala adalah mahasiswa program studi Teknik Informatika, Fakultas Teknik, Universitas Krisnadwipayana. Aktif pada kegiatan kemahasiswaan yaitu Himpunan Mahasiswa Teknik Informatika (HIMATIF) dan Badan Perwakilan Mahasiswa (BPM) Fakultas Teknik. Kegiatan lain yaitu sebagai asisten dosen dan asisten praktikum algoritme dan pemrograman, basis data, pemrograman berbasis objek dan pemrograman web. Selain aktivitas di kampus, juga aktif membangun usaha berupa jasa.