

**PERAMALAN PERMINTAAN EKSPOR NONMIGAS INDONESIA
MENGUNAKAN *LONG SHORT TERM MEMORY***

SKRIPSI

Oleh :
PUJO HASTOWO ARDI
NIM. 17650077



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**PERAMALAN PERMINTAAN EKSPOR NONMIGAS INDONESIA
MENGUNAKAN *LONG SHORT TERM MEMORY***

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
PUJO HASTOWO ARDI
NIM. 17650077

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

HALAMAN PERSETUJUAN

**PERAMALAN PERMINTAAN EKSPOR NONMIGAS INDONESIA
MENGUNAKAN *LONG SHORT TERM MEMORY***


SKRIPSI


**Oleh :
PUJO HASTOWO ARDI
NIM. 17650077**

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 10 Juni 2023

Pembimbing I,

Pembimbing II,


Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004


Fatchurrohman, M.Kom
NIP. 19700731 200501 1 002

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Fachrul Kurniawan ST, M.MT, IPM
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

**PERAMALAN PERMINTAAN EKSPOR NONMIGAS INDONESIA
MENGUNAKAN *LONG SHORT TERM MEMORY***

SKRIPSI

Oleh :
PUJO HASTOWO ARDI
NIM. 17650077

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 19 Juni 2023

Susunan Dewan Penguji

Ketua Penguji	: <u>Dr. M. Amin Hariyadi</u> NIP. 19670018 200501 1 001	()
Anggota Penguji I	: <u>Okta Oomaruddin Aziz, M.Kom</u> NIP. 19911019 201903 1 013	()
Anggota Penguji II	: <u>Dr. Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004	()
Anggota Penguji III	: <u>Fatchurrohman, M.Kom</u> NIP. 19700731 200501 1 002	()

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrul Kurmiawan ST, M.MT, IPM
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Nama Pujo Hastowo Ardi

NIM : 17650077

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Peramalan Permintaan Ekspor Nonmigas Indonesia
Menggunakan *Long Short Term Memory*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 25 Juni 2023

Yang membuat pernyataan,



Pujo Hastowo Ardi
NIM. 17650077

MOTTO

“Kamu tidak bisa mengendalikan takdir, satu satunya yang bisa kamu kendalikan adalah cara kamu menyikapinya”

HALAMAN PERSEMBAHAN

Tiada lembar yang paling indah dalam laporan skripsi ini kecuali lembar persembahan, Skripsi ini saya persembahkan sebagai tanda bukti kepada orang tua tercinta, saudara, sahabat, dan teman-teman yang selalu memberi support untuk menyelesaikan skripsi ini. Serta untuk diri saya sendiri yang telah berusaha untuk menyelesaikan skripsi ini.

KATA PENGANTAR

Syukur alhamdulillah penulis ucapkan kehadiran Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan Skripsi ini dengan baik.

Selanjutnya penulis mengucapkan terima kasih kepada semua pihak yang telah membantu terselesaikannya Skripsi ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Prof. Dr. H. M. Zainuddin MA, selaku rektor UIN Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrul Kurniawan, M.MT., IPM selaku Ketua Program Studi Teknik Informatika yang senantiasa mendorong dan memberikan solusi di setiap permasalahan mahasiswanya.
4. M. Imamudin Lc, MA, selaku dosen wali yang selalu memberikan do'a kesehatan, kesuksesan, dan keberkahan dalam segala hal di setiap awal semester.
5. Seluruh dosen dan staf Teknik Informatika yang telah memberikan ilmu dan pengalaman yang berharga selama masa perkuliahan.

6. Bapak Suyanto dan ibu Sumini selaku orang tua penulis yang telah memberikan dukungan yang berlimpah, baik dari segi ekonomi, moral dan spiritual serta Saudara kandung penulis Dhian Febriyani yang selalu memberi semangat, dukungan dengan do'a sehingga penulis bisa mengerjakan skripsi dengan lancar.
7. Seluruh guru saya yang telah memberikan bimbingan dan doa, semoga Allah senantiasa memberikan kesehatan dan keberkahan.
8. Seluruh sahabat saya, yaitu Muhammad Ali Zamroni, Emha Ahdan Fahmi Elmuna dan Faishal Alfarisi yang tiada henti memberikan motivasi dan dukungannya serta saling support satu sama lain.
9. Seluruh saudara Teknik Informatika UNOCORE yang telah memberikan semangat dan do'a.
10. Seluruh dulur ARJUNA Pesantren Mahasiswa Al Hikam Malang yang telah memberikan semangat dan do'a.
11. Semua pihak yang terlibat dalam penyusunan skripsi ini.

Seiring doa dan harapan penulis, semoga semua yang telah mereka berikan kepada penulis, mendapatkan balasan yang lebih baik dari Allah SWT. Dan terlepas dari segala kekurangan penulis, semoga skripsi ini bermanfaat dan berkah bagi kita semua. Aamiin aamiin ya robbal alamin. Terima kasih.

Malang, 26 Juni 2023



Penulis

DAFTAR ISI

HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
البحث مستخلص	xvi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Batasan Penelitian	5
1.5 Manfaat Penelitian	6
BAB II STUDI PUSTAKA	7
2.1 Penelitian Terkait	7
2.2 Landasan Teori	9
2.2.1 Peramalan Permintaan	9
2.2.2 Ekspor Nonmigas Indonesia	11
2.2.3 <i>Preprocessing Data</i>	12
2.2.4 <i>Recurrent Neural Network (RNN)</i>	13
2.2.5 <i>Long Short Term Memory (LSTM)</i>	15
2.2.6 <i>Mean Absolute Percentage Error (MAPE)</i>	19
2.2.6 Fungsi Sigmoid dan Tanh	20
BAB III DESAIN DAN IMPLEMENTASI SISTEM	22
3.1 Desain Sistem Peramalan Permintaan Ekspor Nonmigas	22
3.1.1 Dataset Ekspor Nonmigas	23
3.1.2 <i>Preprocessing Data</i>	23
3.1.3 Arsitektur Jaringan LSTM	27
3.1.4 Proses <i>Training</i>	31
3.1.5 Proses <i>Testing</i>	35
3.1.6 Evaluasi MAPE	38
3.1.7 Tampilan Sistem	39
3.2 Skenario Pengujian	42
BAB IV HASIL DAN PEMBAHASAN	45
4.1 Hasil	45
4.1.1 Preproccesing	45
4.1.2 Membuat Model LSTM menggunakan Data Latih	45

4.1.3	Proses <i>Training</i>	46
4.1.4	Proses <i>Testing</i>	47
4.1.5	Pengukuran tingkat <i>error</i> menggunakan MAPE	56
4.1.6	Peramalan Bulan Selanjutnya	58
4.2	Pembahasan	59
BAB V PENUTUP		65
5.1	Kesimpulan.....	65
5.2	Saran	65
DAFTAR PUSTAKA		

DAFTAR GAMBAR

Gambar 1.1 Perkembangan Volume Ekspor Non Migas	2
Gambar 2.1 Segmentasi Data	13
Gambar 2.2 Arsitektur RNN	14
Gambar 2.3 Struktur RNN dan LSTM	15
Gambar 2.4 <i>Forget Gate</i> pada LSTM	16
Gambar 2.5 <i>Input Gate</i> pada LSTM	17
Gambar 2.6 Cell State pada LSTM	18
Gambar 2.7 <i>Output Gate</i> pada LSTM	19
Gambar 3.1 Desain Sistem	22
Gambar 3.2 Dataset Volume Ekspor Nonmigas	23
Gambar 3.3 <i>Preprocessing Data</i>	24
Gambar 3.4 Segmentasi data	25
Gambar 3.5 Pembagian Data	26
Gambar 3.6 Arsitektur Jaringan LSTM	27
Gambar 3.7 Pembangunan Model LSTM	34
Gambar 3.8 Proses Peramalan	36
Gambar 3.9 Denormalisasi Data	37
Gambar 3.10 Perhitungan MAPE	38
Gambar 3.11 Proses Peramalan Bulan Selanjutnya	39
Gambar 3.12 Tampilan Sistem	41
Gambar 4.1 Model LSTM yang digunakan	46
Gambar 4.2 Grafik Model 1 dengan Parameter <i>Batch size:2, epochs: 50</i>	48
Gambar 4.3 Grafik Model 2 dengan Parameter <i>Batch size: 3, epochs: 100</i>	49
Gambar 4.4 Grafik Model 3 dengan Parameter <i>Batch size: 6, epochs: 100</i>	49
Gambar 4.5 Grafik Model 4 dengan Parameter <i>Batch size: 3, epochs: 50</i>	50
Gambar 4.6 Grafik Model 5 dengan Parameter <i>Batch size: 3, epochs: 50</i>	51
Gambar 4.7 Grafik Model 6 dengan Parameter <i>Batch size: 12, epochs: 50</i>	51
Gambar 4.8 Grafik Model 7 dengan Parameter <i>Batch size: 12, epochs: 50</i>	52
Gambar 4.9 Grafik Model 8 dengan Parameter <i>Batch size: 12, epochs: 50</i>	53
Gambar 4.10 Grafik Model 9 dengan Parameter <i>Batch size: 6, epochs: 50</i>	53
Gambar 4.11 Grafik Model 10 dengan Parameter <i>Batch size: 3, epochs: 50</i>	54
Gambar 4.12 Grafik Model 11 dengan Parameter <i>Batch size: 4, epochs: 50</i>	55
Gambar 4.13 Grafik Model 12 dengan Parameter <i>Batch size: 12, epochs: 50</i>	55

DAFTAR TABEL

Tabel 2.1 Kriteria nilai MAPE	20
Tabel 3.1 Pembagian dataset.....	42
Tabel 3.2 Parameter pada LSTM	44
Tabel 4.1 Parameter terbaik pada setiap model percobaan	47
Tabel 4.2 Pengukuran tingkat error menggunakan MAPE	56
Tabel 4.3 Data Hasil Peramalan dengan Kombinasi Parameter Terbaik.....	57

ABSTRAK

Ardi, Pujo Hastowo. 2023. **Peramalan Permintaan Ekspor Nonmigas Indonesia Menggunakan Long Short Term Memory** Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Irwan Budi Santoso, M.Kom (II) Fatchurrohman, M.Kom.

Kata Kunci: *Peramalan, Ekspor Nonmigas, Long Short Term Memory*

Ekspor nonmigas memegang peranan penting dalam perekonomian Indonesia. Sektor ini memberikan kontribusi terbesar bagi devisa negara. Terjadinya fluktuasi permintaan ekspor nonmigas di pasar internasional menjadi penyebab dibutuhkannya sebuah sistem peramalan agar bisa mengimbangi antara produksi dan permintaan di pasar internasional. *Long Short Term Memory (LSTM)* digunakan dalam sistem peramalan permintaan ekspor nonmigas. LSTM merupakan salah satu dari banyak jenis *Recurrent Neural Network (RNN)* yang mampu mengatasi masalah kompleks peramalan data *time series*. Penelitian ini menggunakan data sekunder yang diambil dari website www.bps.go.id dengan data *time series* bulanan dari tahun 2012 – 2022. Data tersebut kemudian dibagi menjadi data latih dan data uji. Pengujian model dilakukan dalam beberapa skenario pembagian data, *batch size*, *learning rate*, dan *epoch*. Model terbaik didapatkan dengan nilai perhitungan MAPE dengan nilai terendah 7,87% dengan pembagian data 80% data latih dan 20 % data uji dengan parameter terbaik berupa *batch size* 6, *epoch* 100 dan *learning rate* 0,001.

ABSTRACT

Ardi, Pujo Hastowo. 2023. **Demand Forecasting for Indonesia's Non-Oil and Gas Export Using Long Short Term Memory**. Undergraduate Thesis. Department of Informatics Engineering Faculty of Science and Technology Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Dr. Irwan Budi Santoso, M.Kom (II) Fatchurrohman, M.Kom.

Keywords : *Forecasting, Non-Oil and Gas, Long Short Term Memory*

Non-oil exports play an important role in the Indonesian economy. This sector provides the largest contribution to the country's foreign exchange. The fluctuation in demand for non-oil and gas exports in the international market is the reason for the need for a forecasting system in order to balance production and demand in the international market. Long Short-Term Memory (LSTM) is used in the non-oil and gas export demand forecasting system. LSTM is one of the many types of Recurrent Neural Networks (RNN), which is capable of solving complex problems of forecasting time series data. This study uses secondary data taken from the website of www.bps.go.id with monthly time series data from 2012 – 2022. The data is then divided into training data and test data. Model testing is carried out in several scenarios of data distribution and batch size. The best model is obtained by calculating the MAPE value, with the lowest value being 7.87% with a data division of 80% training data and 20% test data, with the best parameters being a batch size 6, epoch 100 and learning rate 0,001.

البحث مستخلص

أردني فوجو حستووا ٢٠٢٣ توقع الطلب على صادرات إندونيسيا غير النفطية والغازية تخصص في هندسة ذاكرة طويلة و قصيرة باستخدام أطروحة المعلوماتية كلية العلوم والتكنولوجيا الجامعة الإسلامية الحكومية مولان مالك إبراهيم مدينة مالانج مشرف ١ دكتور إروان بودي سانطاسا ماجيستر كومبيوتر مشرف ٢ دكتور فتح الرحمن ماجيستر كومبيوتر

الكلمات الرئيسية : التنبؤ الصادرات غير النفطية والغازية ذاكرة طويلة وقصيرة

أكسبور أن التقلبات في الطلب على الصادرات غير النفطية والغاز في السوق الدولية هي سبب الحاجة إلى نظام تنبؤ حتى يتمكن من موازنة الإنتاج والطلب في السوق الدولية يستخدم ذاكرة طويلة وقصيرة (LSTM) في التنبؤ بالطلب على الصادرات غير النفطية ذاكرة طويلة و قصيرة هي واحدة من أنواع عديدة من الشبكات العصبية (RNN) المتكررة القادرة على حل المشكلات المعقدة www.bps.go.id للتنبؤ ببيانات السلاسل الزمنية والغاز النظام تستخدم هذه الدراسة بيانات ثانوية مأخوذة من موقع الويب المعرف مع بيانات السلاسل الزمنية الشهرية من ٢٠١٢ - ٢٠٢٢ ثم يتم تقسيم البيانات إلى بيانات التدريب وبيانات الإختبار يتم إجراء اختبار النموذج في عدة سيناريوهات لتقسيم البيانات وحجم الدفعة يتم الحصول على أفضل نموذج من خلال حساب قيمة يعني نسبة الخطأ المطلق (MAPE) بأقل قيمة هي ٧,٨٧% مع تقسيم بيانات ٨٠% من بيانات التدريب ٢٠% من بيانات الإختبار مع أفضل المعلمات وهي حجم الحمام ١٢ والعصر ١٥٠

BAB I

PENDAHULUAN

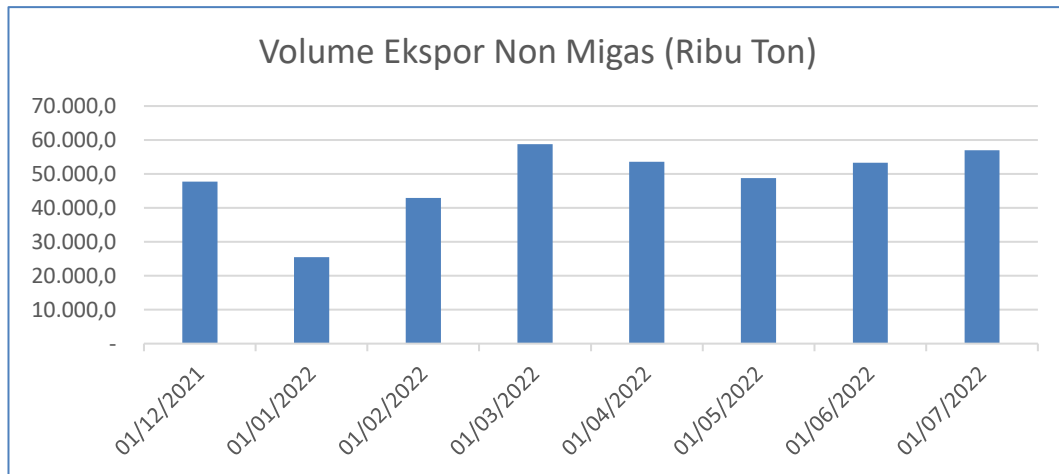
1.1 Latar Belakang

Pertumbuhan ekonomi suatu negara erat kaitannya dengan kegiatan perdagangan internasional yaitu impor dan ekspor. Di negara Indonesia, ekspor ke seluruh negara di dunia kini sedang gencar ditingkatkan. Pemerintah tentunya mencari berbagai aspek ekonomi yang cocok untuk prosedur pengolahan dan implementasi untuk meningkatkan penyerapan tenaga kerja, devisa negara, dan penerimaan pajak (Hidayah, 2017).

Perdagangan internasional juga memainkan peran penting dalam perekonomian setiap negara di dunia. Di sisi lain, perdagangan internasional tidak hanya dapat meningkatkan kemakmuran dan kesejahteraan rakyat, meningkatkan devisa negara, atau meningkatkan hubungan ekonomi yang baik antara negara pengimpor dan pengekspor, tetapi juga mendorong suatu negara untuk dapat berspesialisasi dalam bidang barang. dan jasa di negara lain (Ilah, 2016).

Ekspor nonmigas merupakan sektor yang memegang peranan penting dalam kegiatan perekonomian Indonesia. Perkembangan ekspor Indonesia kini didominasi oleh produk nonmigas yang berasal dari 3 kelompok komoditas yaitu (1) komoditas ekspor hasil pertanian, (2) komoditas ekspor hasil industri pengolahan, dan (3) komoditas ekspor hasil pertambangan dan lainnya (Badan Pusat Statistik, 2022). Menurut Bank Indonesia, sektor nonmigas merupakan sektor yang dapat memberikan kontribusi devisa terbesar bagi Indonesia. Pemilihan komoditas

nonmigas tersebut untuk mengatasi ketergantungan migas terhadap pendapatan nasional, sehingga pemerintah berupaya meningkatkan ekspor komoditas dan jasa nonmigas (Huda, 2006).



Gambar 1.1 Perkembangan Volume Ekspor Non Migas
Sumber : diolah dari (Badan Pusat Statistik, 2021)

Dari grafik tersebut menunjukkan bahwa volume ekspor nonmigas Indonesia yang merupakan representasi dari permintaan ekspor mengalami fluktuasi. Menurut Sihotang (2013) untuk menjaga kestabilan volume ekspor perlu diimbangi dengan strategi peningkatan produksi. Oleh karena itu penting dilakukan peramalan terhadap permintaan ekspor nonmigas Indonesia agar bisa mengimbangi antara produksi dan kebutuhan permintaan nonmigas internasional.

Peramalan dilakukan untuk mengurangi ketergantungan pada hal-hal yang tidak pasti. Dalam melakukan peramalan terdapat dua metode yaitu kualitatif dan kuantitatif. Metode kualitatif didasarkan pada informasi subyektif, sedangkan metode kuantitatif terdiri dari dua jenis: kausal (faktor yang terkait variabel untuk

meramal) dan deret waktu (pengumpulan data masa lalu dan digunakan sebagai referensi untuk prakiraan masa depan) (Munarsih, 2017).

Dalam melakukan peramalan permintaan ekspor nonmigas dibutuhkan suatu metode yang tepat salah satunya adalah ARIMA dan *Double Exponential* yang dilakukan oleh Abdullah dkk (2020), dalam penelitiannya peneliti menggunakan data *timeseries* bulanan nilai ekspor nonmigas tahun 2016 - 2017 namun pada perhitungan *error* metode ini masih menghasilkan nilai error MAPE yang cukup tinggi yakni 37,98629959 pada ARIMA dan 68,33256837 pada *Double Exponential*.

Penelitian peramalan terhadap ekspor non migas juga dilakukan Nurlaily dkk (2022) menggunakan metode ARIMA dan *Tripple Exponential Smoothing*. Pada penelitian ini evaluasi model menghasilkan nilai sMAPE adalah sebesar 1,9344 dan RMSE sebesar 42,68.

Penelitian peramalan *time series* lain dilakukan oleh Son & Kim (2020) melakukan prediksi terhadap data *timeseries* konsumsi listrik bulanan perumahan di Korea Selatan dengan menggunakan metode *Long Short Term Memory*. Pada penelitian ini menghasilkan kinerja peramalan yang sangat akurat dengan nilai MAPE kurang dari satu.

Long Short Term Memory (LSTM) yang merupakan salah satu dari banyak jenis *Recurrent Neural Network* (RNN) yang mampu mengatasi masalah kompleks seperti peramalan *time series*, deteksi ucapan dan pengenalan tulisan tangan (Hammad dkk., 2021). LSTM termasuk ke dalam jaringan saraf tiruan yang meniru cara kerja sistem otak pada manusia. Metode ini meniru cara kerja otak dalam hal

berpikir dan mengingat. Dalam Al-Quran disebutkan manusia berpikir dan mengingat yang dijelaskan pada Q.S Ali Imron ayat 190 – 191 yang berbunyi :

إِنَّ فِي خَلْقِ السَّمٰوٰتِ وَالْاَرْضِ وَاخْتِلَافِ اللَّيْلِ وَالنَّهَارِ لَآيٰتٍ لِّاُولٰٓئِى الْاَلْبَابِ ۙ
 الَّذِيْنَ يَذْكُرُوْنَ اللّٰهَ قِيَامًا وَّفُجُوًا وَّعَلٰٓى جُنُوْبِهِمْ وَيَتَفَكَّرُوْنَ فِيْ خَلْقِ السَّمٰوٰتِ وَالْاَرْضِ ۗ رَبَّنَا مَا خَلَقْتَ هٰذَا بَاطِلًا ؕ
 سُبْحٰنَكَ فَقِنَا عَذَابَ النَّارِ ۙ

“Sesungguhnya dalam penciptaan langit dan bumi, serta pergantian malam dan siang terdapat tanda-tanda (kebesaran Allah) bagi orang-orang yang berakal(190), Orang-orang yang mengingat Allah sambil berdiri, duduk, dan dalam keadaan berbaring, serta memikirkan tentang penciptaan langit dan bumi (seraya berkata), 'Ya Tuhan kami, tiadalah Engkau menciptakan ini dengan sia-sia. Maha Suci Engkau, maka peliharalah kami dari siksa neraka(191)’”.QS. Ali Imron (190-191).

Dalam tafsir Ibnu Katsir ayat tersebut mengandung ajakan untuk merenungkan penciptaan langit dan bumi serta peredaran malam dan siang. Dalam peredaran tersebut terdapat tanda-tanda kebesaran Allah yang harus diambil pelajaran dan makna dari mereka yang berakal. Orang yang berakal yang dimaksud adalah mereka yang selalu ingat kepada Allah baik dalam keadaan berdiri, duduk, ataupun berbaring dan selalu memikirkan tentang penciptaan langit dan bumi serta berdoa memohon perlindungan dari siksa neraka adalah orang-orang yang berakal dan beriman(Katsir, 2014).

Dalam proses mengingat dan berpikir, manusia terkadang bisa lupa karena dalam otak manusia terdapat penyimpanan memori yang ketika menerima tekanan pikiran dapat mengganggu ingatan memori jangka pendek atau yang lebih parah sampai mengganggu ingatan jangka panjang. Hal tersebut yang diadaptasi pada metode *Long Short Term Memory* dengan memiliki memori jangka pendek dan memori jangka panjang yang terdapat dalam blok memori yang tidak dimiliki oleh RNN jenis lain sehingga memiliki keunggulan dalam mengingat data *timeseries*

jangka panjang dan menentukan nilai mana yang akan dipilih sebagai output yang relevan untuk input yang diberikan (Wiranda & Sadikin, 2019). Oleh karena itu diusulkan untuk menggunakan metode LSTM dalam melakukan penelitian ini.

Dari beberapa uraian tersebut memperkuat penggunaan metode LSTM dalam melakukan peramalan time series. Oleh karena itu dilakukan penelitian ini dengan judul “Peramalan Permintaan Ekspor Nonmigas Indonesia Menggunakan *Long Short Term Memory*”.

1.2 Pernyataan Masalah

Berdasarkan penjelasan latar belakang maka didapatkan pernyataan masalah pada penelitian ini adalah seberapa tinggi tingkat *error Long Short Term Memory* (LSTM) yang dihasilkan dalam meramal permintaan ekspor nonmigas Indonesia?

1.3 Tujuan Penelitian

Mengukur tingkat *error Long Short Term Memory* (LSTM) dalam meramal permintaan ekspor nonmigas Indonesia.

1.4 Batasan Penelitian

Batasan penelitian dalam skripsi diperlukan untuk memfokuskan ruang lingkup penelitian dan menghindari penyebaran yang terlalu luas. Dalam sebuah skripsi, batasan masalah mengidentifikasi batas-batas yang jelas mengenai topik yang akan diteliti, parameter dan aspek yang akan diteliti. Berikut merupakan batasan dalam penelitian ini :

1. Dataset yang diambil adalah dataset volume ekspor nonmigas dari website Badan Pusat Statistik.
2. Dataset yang diambil merupakan data bulanan mulai tahun 2012 – 2022.
3. Penelitian ini menggunakan bahasa pemrograman *Python* dengan *Jupyter Notebook* dan *Google Colaboratory*.
4. Pengukuran tingkat *error* pada program menggunakan perhitungan *error Mean Absolute Percentage Error* (MAPE).

1.5 Manfaat Penelitian

Tujuan dari penelitian ini adalah memberikan manfaat yang diharapkan, dan manfaat yang diinginkan termasuk hal-hal berikut ini::

1. Memberikan kontribusi terhadap pengembangan metode dan teknik peramalan menggunakan Long Short Term Memory (LSTM) dalam konteks ekonomi dan perdagangan internasional. Hasil penelitian ini dapat menjadi referensi dan landasan untuk penelitian selanjutnya dalam bidang peramalan ekonomi..
2. Membantu pihak berwenang untuk secara teratur merancang pengembangan potensi pasar komoditas nonmigas, dan mencapai keseimbangan antara pasokan dan permintaan komoditas nonmigas pasar global, sehingga nilai ekspor dapat ditingkatkan.

BAB II

STUDI PUSTAKA

2.1 Penelitian Terkait

Dalam proses penelitian ini membutuhkan beberapa penelitian sebelumnya. Terdapat penelitian-penelitian sebelumnya yang memiliki metode dan topik yang selaras. Penelitian terkait digunakan sebagai referensi untuk memperoleh pemahaman dan referensi terkait topik dan metode.

Penelitian yang dilakukan oleh Sihombing dkk (2021) fokus pada pertumbuhan ekonomi Indonesia antara tahun 2000 hingga 2019 dengan menganalisis dampak dari penanaman modal asing, ekspor migas, dan ekspor non-migas. Hasil penelitian tersebut menunjukkan bahwa ekspor migas dan ekspor non-migas memiliki pengaruh positif dan signifikan terhadap pertumbuhan ekonomi Indonesia. Namun, sektor ekspor non-migas memberikan kontribusi yang lebih besar terhadap penerimaan devisa negara.

Abdullah dkk (2020) melakukan penelitian untuk memprediksi ekspor non migas Indonesia menggunakan metode *ARIMA dan Double Exponential* dengan data bulanan ekspor nonmigas tahun 2016 – 2017 dan digunakan MSE, MAPE, dan MPE untuk mengukur tingkat error metode yang digunakan. Hasil akhir dari penelitian ini adalah perubahan ekspor nonmigas indonesia dapat diprediksi dan namun menghasilkan nilai error yang masih cukup tinggi.

Dalam penelitian yang dilakukan oleh Nurlaily dkk (2022), tujuan utamanya adalah melakukan prediksi terhadap data ekspor non-migas di Kalimantan Timur. Dalam penelitian ini, metode yang digunakan adalah *ARIMA dan Tripple*

Exponential Smoothing. Setelah melakukan evaluasi terhadap model yang dikembangkan, ditemukan bahwa nilai sMAPE (*simetris Mean Absolute Percentage Error*) sebesar 1,9344 dan nilai RMSE (*Root Mean Square Error*) sebesar 42,68.

Chamsukhee (2020) dalam penelitiannya membandingkan metode *Exponential Smoothing*, LSTM, dan ARIMA untuk memprediksi data *timeseries* Langganan E-Database Perpustakaan Universitas Mahidol. Perbandingan dilakukan dengan mengevaluasi model dengan metode *Mean Square Error* (MSE) pada tiga model yaitu ETS, LSTM, dan ARIMA. Hasil MSE untuk masing-masing model percobaan adalah 0,150, 0,127, dan 0,153. Model LSTM merupakan model terbaik untuk mendapatkan nilai error rata-rata minimum dan telah menunjukkan kesesuaian dengan data *time series* dan musiman termasuk jumlah akses ke E-Database dapat menjadi model pelatihan yang tepat.

Penelitian lain yang menggunakan metode LSTM dilakukan oleh Do dkk (2020) yang memprediksi jumlah penumpang pesawat di bandara agar bisa menstabilkan layanan dan memaksimalkan keuntungan, penelitian tersebut membandingkan metode LSTM dengan SARIMA dengan menggunakan matriks pengukuran performa seperti MAPE, MSE, RMSE, dan MAD. Dari penelitian ini dihasilkan bahwa metode LSTM lebih akurat daripada SARIMA.

Ozbek dkk (2021) melakukan penelitian menggunakan metode LSTM tentang prediksi *time series* per-10 menit, per-jam, dan harian terhadap suhu udara atmosfer dengan menggunakan metode LSTM, ANFIS-FCM, dan ARMA. Dalam penelitian ini menunjukkan bahwa LSTM lebih akurat dari pada metode

pembandingnya dengan menghasilkan nilai MAE 0,4275 RMSE 0,6117 MAPE 1,8471 pada interval jam dan MAE 0,9870 RMSE 1,3343 MAPE 4,2673 pada interval harian.

Son & Kim (2020) melakukan penelitian terhadap permintaan listrik perumahan di Korea Selatan dengan menggunakan data *timeseries* bulanan. Metode *Long Short Term Memory* digunakan dalam penelitian. Hasil dari penelitian ini menunjukkan akurasi sangat akurat dengan menghasilkan nilai kurang dari 1 dengan MAPE sebagai metrik pengukur performa tersebut.

Hsieh (2021) melakukan penelitian dengan melakukan peramalan terhadap permintaan pariwisata di Taiwan menggunakan beberapa metode yaitu LSTM, Bi-LSTM dan GRU. Pada penelitian ini digunakan data *timeseries* bulanan kemudian hasil dari penelitian ini menunjukkan LSTM sebagai metode yang paling baik diantara metode yang digunakan dengan menunjukkan nilai RMSE paling kecil.

2.2 Landasan Teori

2.2.1 Peramalan Permintaan

Peramalan permintaan adalah suatu proses yang penting dalam pengambilan keputusan bisnis yang melibatkan peramalan tingkat permintaan suatu produk atau layanan di masa depan. Hal ini memungkinkan perusahaan untuk merencanakan produksi, mengelola persediaan, serta mengatur rantai pasokan dengan lebih efisien. peramalan permintaan membantu perusahaan dalam mengantisipasi fluktuasi pasar, menghindari kekurangan atau kelebihan persediaan, dan meningkatkan kepuasan pelanggan (Heizer dkk., 2017).

Untuk melakukan peramalan permintaan, analisis data historis merupakan

langkah awal yang penting. Data penjualan atau penggunaan produk di masa lalu digunakan untuk mengidentifikasi pola, tren, dan musiman dalam permintaan. Menurut Armstrong dkk (2011), analisis data historis memungkinkan perusahaan untuk memahami pola musiman seperti peningkatan penjualan selama musim liburan atau penurunan permintaan selama musim panas.

Ada berbagai metode yang dapat digunakan untuk meramalkan permintaan. Metode statistik seperti rata-rata bergerak, metode eksponensial, dan regresi linier sering digunakan dalam peramalan permintaan. Sementara itu, metode analisis deret waktu seperti ARIMA (*Autoregressive Integrated Moving Average*) juga digunakan untuk mengidentifikasi pola dan tren dalam data historis. Pemilihan metode peramalan yang tepat sangat penting untuk mencapai akurasi yang tinggi dalam peramalan permintaan.

Tingkat akurasi peramalan permintaan dapat bervariasi tergantung pada berbagai faktor, termasuk kualitas data historis, kompleksitas faktor-faktor yang mempengaruhi permintaan, serta metode peramalan yang digunakan. Evaluasi dan pengujian yang teratur diperlukan untuk memperbaiki dan mengoptimalkan kualitas peramalan. Menurut Mentzer dkk (2019), penggunaan teknik peramalan yang lebih maju seperti model peramalan *ensemble* atau penggunaan kombinasi beberapa metode peramalan dapat membantu meningkatkan akurasi peramalan permintaan.

Peramalan permintaan memiliki peran yang krusial dalam pengambilan keputusan bisnis. Dengan memiliki peramalan yang tepat, perusahaan dapat meningkatkan penggunaan sumber daya secara optimal, menghindari masalah

persediaan yang berlebihan atau kekurangan, dan merencanakan strategi bisnis yang lebih efisien. Menurut Chopra dan Meindl (2016), peramalan permintaan yang baik dapat membantu perusahaan dalam melakukan perencanaan produksi, pengaturan persediaan, serta menghadapi perubahan permintaan dengan lebih adaptif.

2.2.2 Ekspor Nonmigas Indonesia

Ekspor merupakan cara suatu negara untuk menjual komoditas minyak dan gas bumi, serta nonmigas, ke negara lain dengan persyaratan yang ditetapkan oleh pemerintah, dan memperoleh keuntungan berupa peningkatan cadangan devisa.

Indonesia memiliki kekayaan komoditas migas yang besar yang menjadi pijakan utama perekonomian negara ini karena merupakan sumber daya unggulan. Namun, Indonesia menghadapi beberapa tantangan, yaitu keterbatasan dana untuk pembangunan akibat penurunan harga minyak dan gas bumi, serta penurunan kapasitas sumber daya alam seperti minyak dan gas bumi. Untuk mengatasi hal ini, pemerintah Indonesia telah mengambil langkah dengan meningkatkan ekspor nonmigas sebagai sumber pendanaan pembangunan. Melalui peningkatan ekspor nonmigas, diharapkan akan terjadi peningkatan penerimaan devisa, penyerapan tenaga kerja, dan investasi. (Sihombing dkk., 2021).

kelompok nonmigas dibagi menjadi tiga kelompok komoditas ekspor yaitu (1) komoditas hasil pertanian, komoditas hasil industri pengolahan, komoditas hasil pertambangan dan lainnya (Badan Pusat Statistik, 2022).

Indonesia menghadapi masalah keterbatasan pembiayaan untuk pembangunan akibat harga minyak dan gas yang lebih rendah dan berkurangnya

kapasitas sumber daya alam berupa minyak dan gas. Oleh karena itu, solusi yang ditempuh pemerintah adalah meningkatkan ekspor nonmigas untuk membiayai pembangunan. Peningkatan ekspor nonmigas akan meningkatkan devisa negara, demikian pula lapangan kerja dan investasi. Jika jumlah penduduk bertambah maka tingkat produktivitas menurun karena semakin banyak orang yang dikonsumsi masyarakat, pertumbuhan ekonomi menurun, tetapi jika jumlah penduduk sedikit dan kekayaan sumber daya alam melimpah maka tingkat produktivitas masyarakat menurun dan meningkat. sehingga membawa investasi baru dan pertumbuhan ekonomi juga. meningkat terlalu banyak. Ekspor nonmigas berdampak positif terhadap pertumbuhan ekonomi (Sihombing dkk., 2021).

2.2.3 Preprocessing Data

Preprocessing data merupakan langkah persiapan data sebelum memulai proses perancangan sistem peramalan. Salah satu tahap dalam *preprocessing* adalah normalisasi, yang bertujuan untuk menghindari anomali dan inkonsistensi data. Normalisasi dilakukan dengan maksud mengubah skala data menjadi lebih kecil tanpa mengubah nilai aslinya. Salah satu teknik normalisasi yang sering digunakan adalah *min-max scaling*, yang digunakan untuk menangani perbedaan nilai yang besar antara dataset. Teknik ini bekerja dengan mengubah nilai pada data aktual menjadi nilai dengan skala (0,1) tanpa mengubah informasi yang terkandung di dalamnya (Aldi dkk., 2018).

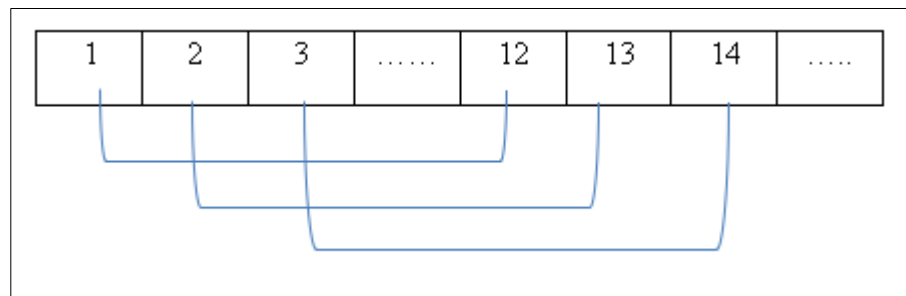
Rumus yang digunakan pada proses ini sebagai berikut :

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (2.1)$$

Keterangan :

- x' = nilai setelah dilakukan normalisasi
 x = nilai sebelum dilakukan normalisasi
 x_{min} = nilai minimum pada data sebelum dinormalisasi
 x_{max} = nilai maximum pada data sebelum dinormalisasi

Tahap kedua dalam preprocessing adalah segmentasi data. Segmentasi data adalah langkah untuk membagi data menjadi segmen-segmen yang lebih kecil, berdasarkan karakteristik atau pola tertentu, sesuai dengan kebutuhan sistem. (Abdul Dwiyanto Suyudi dkk., 2019).



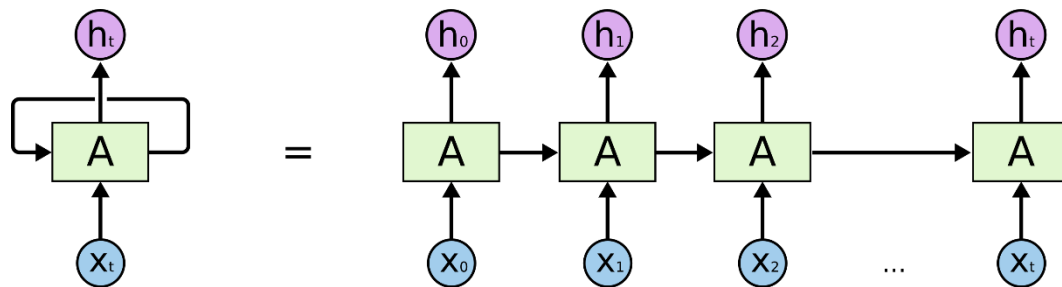
Gambar 2.1 Segmentasi Data

Tahap preprocessing yang ketiga yaitu pembagian data. data tersebut disusun kembali sesuai dengan variabel inputnya dan dalam bentuk yang telah dinormalisasi. Data yang telah disusun dibagi menjadi data latih dan uji yang disebut juga data *training* dan *testing* (Sugiyarto & Abadi, 2019).

2.2.4 Recurrent Neural Network (RNN)

RNN (*Recurrent Neural Network*) pertama kali dikembangkan pada tahun 1980an. Arsitektur RNN terdiri dari lapisan input, satu atau lebih lapisan tersembunyi, dan lapisan *output*. RNN memiliki struktur yang mirip rantai dari modul berulang dengan ide di balik penggunaan modul ini sebagai memori untuk menyimpan informasi penting dari langkah pemrosesan sebelumnya. Tidak seperti jaringan saraf umpan maju (*Feed Forward Neural Network*), RNN menyertakan

loop umpan balik yang memungkinkan jaringan saraf menerima urutan input. Ini berarti *output* dari langkah $t-1$ diumpan kembali ke jaringan untuk mempengaruhi hasil dari langkah t , dan untuk setiap langkah berikutnya. Oleh karena itu, RNN telah berhasil dalam mempelajari urutan.



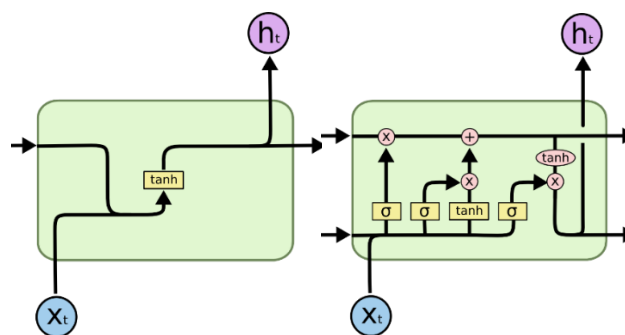
Gambar 2.2 Arsitektur RNN
Sumber : (Le dkk., 2019)

Gambar di atas mengilustrasikan RNN sederhana dengan satu unit input, satu unit *output*, dan satu unit tersembunyi berulang yang diperluas menjadi jaringan penuh, di mana X_t adalah masukan pada langkah waktu t dan h_t adalah kelaran pada langkah waktu t . Selama proses pelatihan, RNN menggunakan algoritma *backpropagation*, algoritma yang umum diterapkan dalam menghitung gradien dan menyesuaikan matriks bobot di JST. Namun, itu akan menyesuaikan dan memperbarui bobot setelah modifikasi proses umpan balik. Oleh karena itu, sering disebut dengan *backpropagation through time* (BPTT). Proses BPTT menggunakan pendekatan *working-backward*, lapis demi lapis, dari keluaran akhir jaringan, menyesuaikan bobot setiap unit sesuai dengan porsi unit yang dihitung dari total kesalahan keluaran. Loop informasi berulang menghasilkan pembaruan besar pada bobot model jaringan saraf dan menyebabkan jaringan tidak stabil karena akumulasi gradien kesalahan selama proses pembaruan. Oleh karena itu, BPTT tidak cukup efisien untuk mempelajari pola dari ketergantungan jangka panjang

karena *gradient vanishing* dan masalah *exploding gradient*. Ini akan menjadi salah satu alasan penting yang menyebabkan kesulitan dalam pelatihan RNN (Le dkk., 2019).

2.2.5 Long Short Term Memory (LSTM)

Long Short-Term Memory (LSTM) adalah pengembangan lebih lanjut dari RNN yang diperkenalkan oleh Hochreiter dan Schmidhuber. Tujuannya adalah untuk mengatasi kelemahan RNN dengan menambahkan interaksi tambahan per modul atau sel. LSTM adalah jenis khusus dari RNN yang memiliki kemampuan untuk mempelajari ketergantungan jangka panjang dan menyimpan informasi dalam jangka waktu yang lama secara default. Menurut Olah (2015), model LSTM memiliki struktur berbentuk rantai, tetapi modul berulang memiliki struktur yang berbeda. Berbeda dengan RNN standar yang menggunakan satu jaringan saraf tunggal, LSTM memiliki empat lapisan yang berinteraksi dengan metode komunikasi yang unik.

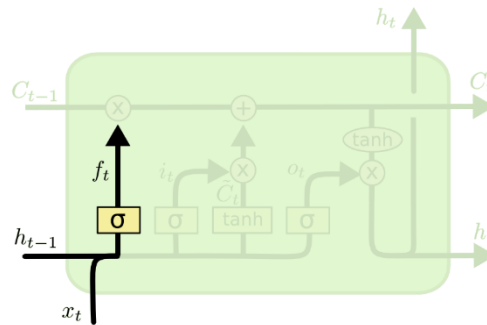


Gambar 2.3 Struktur RNN dan LSTM
Sumber : (Olah, 2015)

Struktur LSTM mempunyai *Cell State* dan 3 *Gates* :

- a. *Forget Gate*

Proses pertama ini berfungsi untuk menentukan informasi mana yang akan digunakan karena penting atau akan dilupakan dengan memproses h_{t-1} dan x_t kemudian dimasukkan ke fungsi aktivasi *sigmoid* sehingga nilai f_t bernilai 0-1 jika mendekati 1 maka informasi disimpan jika mendekati 0 maka informasi dibuang.



Gambar 2.4 *Forget Gate* pada LSTM
Sumber : (Phi, 2018)

Pada *foget gate* menerapkan rumus :

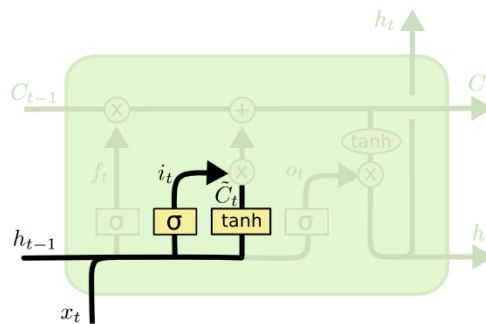
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.2)$$

Keterangan :

- f_t : Nilai gate lupa (*forget gate*) pada *time step t*.
- σ : Fungsi aktivasi *sigmoid*.
- W_f : Bobot untuk *gate* lupa.
- h_{t-1} : Keluaran pada *time step t-1* (*hidden state* sebelumnya).
- x_t : Masukan pada *time step t*.
- b_f : Bias untuk *gate* lupa.

b. *Input Gate*

Input gate memiliki peran penting dalam menentukan informasi baru yang akan disimpan dalam *cell state*. Terdapat dua operasi utama pada *input gate*, yaitu pertama, menggunakan fungsi *sigmoid* untuk menghitung nilai i_t yang menentukan informasi mana yang akan diperbarui. Kedua, menggunakan fungsi *tanh* untuk menghitung nilai C_t yang mengontrol nilai informasi yang akan disimpan dalam *cell state*.



Gambar 2.5 *Input Gate*
Sumber : (Phi, 2018)

Pada *input gate* menerapkan rumus :

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.3)$$

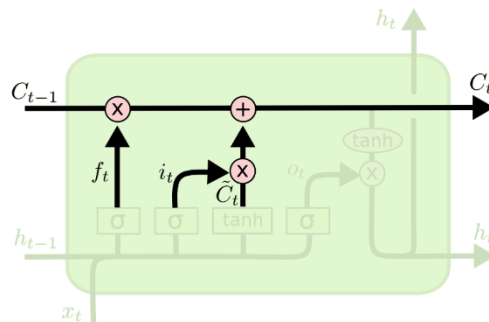
$$C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (2.4)$$

Keterangan :

- i_t : Nilai *gate input* (*input gate*) pada *time step t*.
- σ : Fungsi aktivasi *sigmoid*.
- W_i : Bobot untuk *gate input*.
- h_{t-1} : *Output* pada *time step t-1* (*hidden state* sebelumnya).
- x_t : *Input* pada *time step t*.
- b_i : Bias untuk *gate input*.
- C'_t : Nilai kandidat (*candidate value*) pada *time step t*.
- W_c : Bobot untuk kandidat.
- b_c : Bias untuk kandidat.

c. *Cell State*

Merupakan memori tambahan yang dimiliki oleh LSTM yang tidak dimiliki oleh unit RNN biasa. Pada proses ini terdapat hasil dari *forget gate* (f_t) dan *input gate* (i_t) yang keduanya dihasilkan dari fungsi *sigmoid* serta hasil C'_t , yang dihasilkan dari fungsi *tanh*. f_t dan i_t ini sebagai bobot untuk menentukan informasi lama yang akan disimpan dan informasi baru yang akan ditambahkan.



Gambar 2.6 *Cell State* pada LSTM
Sumber : (Phi, 2018)

Pada *cell state* menerapkan rumus :

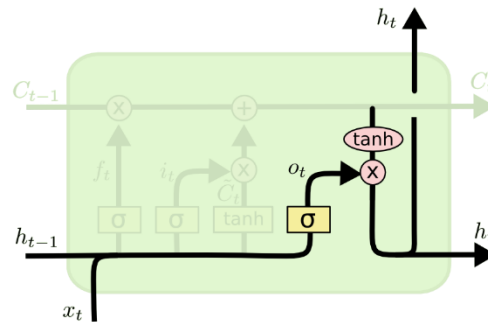
$$C_t = f_t * C_{t-1} + i_t * C'_t \quad (2.5)$$

Keterangan :

- C_t : Nilai *cell state* pada *time step t*.
- f_t : Nilai *gate* lupa (*forget gate*) pada *time step t*.
- C_{t-1} : Nilai *cell state* pada *time step* sebelumnya.
- i_t : Nilai *gate input* (*input gate*) pada *time step t*.
- C'_t : Nilai kandidat (*candidate value*) pada *time step t*.
- * : Operasi perkalian *elemen-wise* (*element-wise multiplication*).

d. *Output Gate*

Kemudian tahap yang terakhir adalah output gate, pada tahap ini berfungsi untuk menentukan hidden state untuk langkah berikutnya. Seperti pada input gate, output gate terdiri dari lapisan sigmoid dan tanh. Dalam menentukan output gate digunakan fungsi sigmoid kemudian hasil tersebut dikalikan dengan C_t yang sudah dilakukan fungsi tanh terlebih dahulu sehingga menghasilkan nilai dari h_t (Phi, 2018).



Gambar 2.7 *Output Gate*
Sumber : (Phi, 2018)

Pada *output gate* menerapkan rumus :

$$o_t = \sigma(W_o \cdot x_t + W_o \cdot h_{t-1} + b_o) \quad (2.6)$$

Keterangan :

- o_t : Nilai *gate output (output gate)* pada *time step t*.
- σ : Fungsi aktivasi *sigmoid*
- W_o : bobot untuk *output gate*
- x_t : Masukan pada *time step t*
- h_{t-1} : Keluaran pada *time step t-1 (hidden state sebelumnya)*
- b_o : bias pada gerbang keluaran

$$h_t = o_t \cdot \tanh(C_t) \quad (2.7)$$

Keterangan :

- h_t : Nilai *output* pada *time step t*
- o_t : bias pada gerbang keluaran
- Tanh : fungsi aktivasi *tanh*
- C_t : *cell state* pada *time step t*

2.2.6 Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error adalah metrik kinerja untuk model regresi, memiliki interpretasi yang sangat intuitif dalam hal kesalahan relatif: karena definisinya, penggunaannya direkomendasikan dalam tugas-tugas di mana lebih penting sensitif terhadap variasi relatif daripada variasi *absolut* (Chicco dkk., 2021).

MAPE mempunyai kriteria nilai yang dapat dijadikan sebagai acuan untuk menarik kesimpulan apakah metode yang digunakan memiliki tingkat *error* yang rendah atau tinggi. (Barus & Ramli, 2013).

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{At - Ft}{At} \right| \times 100\% \quad (2.8)$$

Keterangan :

MAPE = *Mean Absolute Precentage Error*

n = jumlah data

At = nilai aktual

Ft = nilai peramalan

Tabel 2.1 Kriteria nilai MAPE

Nilai	Keterangan
Dibawah 10%	Tingkat <i>error</i> sangat rendah
10% hingga 20%	Tingkat <i>error</i> rendah
20% hingga 50%	Tingkat <i>error</i> cukup
Diatas 50%	Tingkat <i>error</i> tinggi

Semakin kecil nilai MAPE yang dimilikinya maka akan semakin baik untuk peramalannya. kriteria nilai MAPE dapat dilihat pada tabel dibawah (Zulfa dkk., 2021).

2.2.6 Fungsi Sigmoid dan Tanh

Fungsi sigmoid dan fungsi tanh adalah dua jenis fungsi aktivasi yang digunakan dalam jaringan saraf (*neural networks*) dan model pembelajaran mesin (*machine learning*) untuk mengaktifkan output dari neuron atau unit dalam layer jaringan saraf.

Fungsi sigmoid, juga dikenal sebagai fungsi logistik, didefinisikan sebagai:

$$f(x) = 1 / (1 + \exp(-x)) \quad (2.9)$$

dimana x adalah input yang diberikan ke fungsi tersebut, dan $\exp()$ adalah fungsi eksponensial. Fungsi sigmoid menghasilkan output dalam rentang $(0, 1)$, sehingga sering digunakan untuk masalah klasifikasi biner di mana kita ingin menghasilkan probabilitas peramalan terhadap dua kelas yang mungkin.

Fungsi tanh, singkatan dari *hyperbolic tangent* yang merupakan fungsi aktivasi lain yang umum digunakan dalam *neural network* dan model pembelajaran mesin. Fungsi tanh didefinisikan sebagai:

$$f(x) = (\exp(x) - \exp(-x)) / (\exp(x) + \exp(-x)) \quad (2.10)$$

dimana x adalah input yang diberikan ke fungsi tersebut, dan $\exp()$ adalah fungsi eksponensial. Fungsi tanh menghasilkan output dalam rentang $(-1, 1)$, dan memiliki sifat simetris terhadap sumbu y , yaitu $f(-x) = -f(x)$ (Nielsen, 2015).

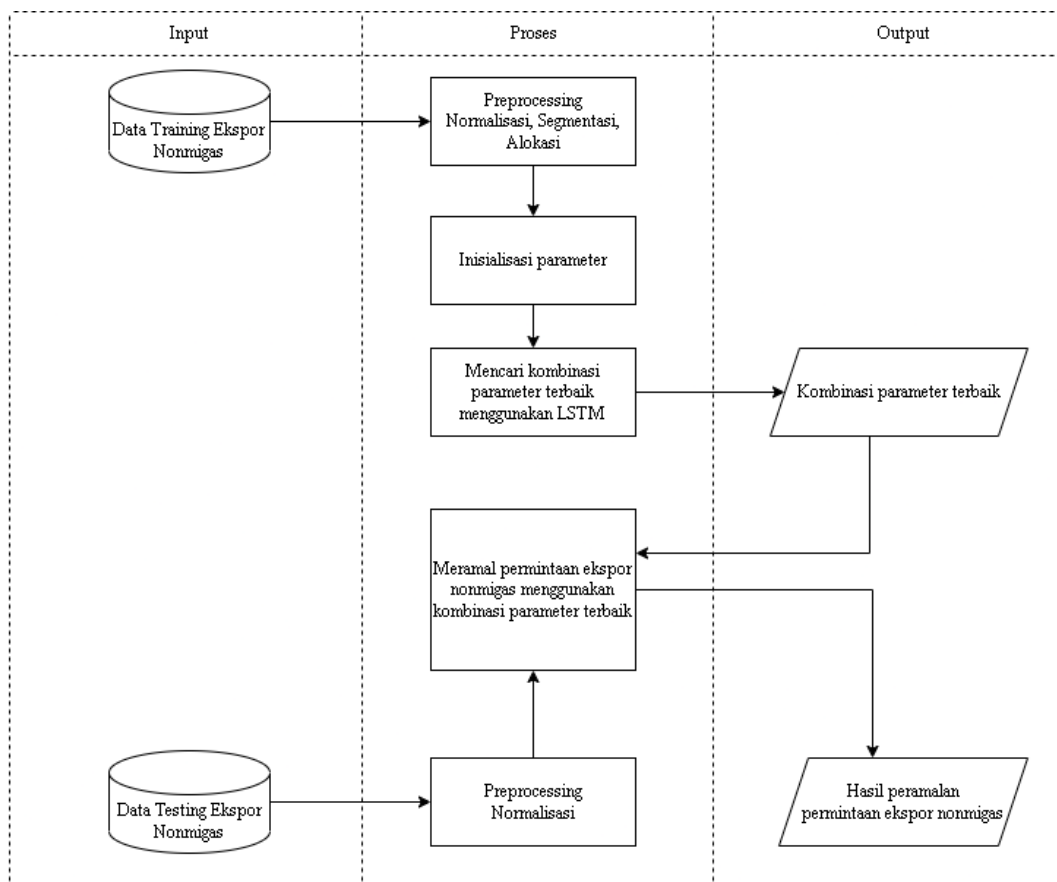
BAB III

DESAIN DAN IMPLEMENTASI SISTEM

Pada bab III, penjelasan diberikan mengenai rancangan dan pelaksanaan sistem peramalan permintaan ekspor nonmigas menggunakan metode long short term memory (LSTM), serta pengujian skenario dari sistem yang telah dikembangkan.

3.1 Desain Sistem Peramalan Permintaan Ekspor Nonmigas

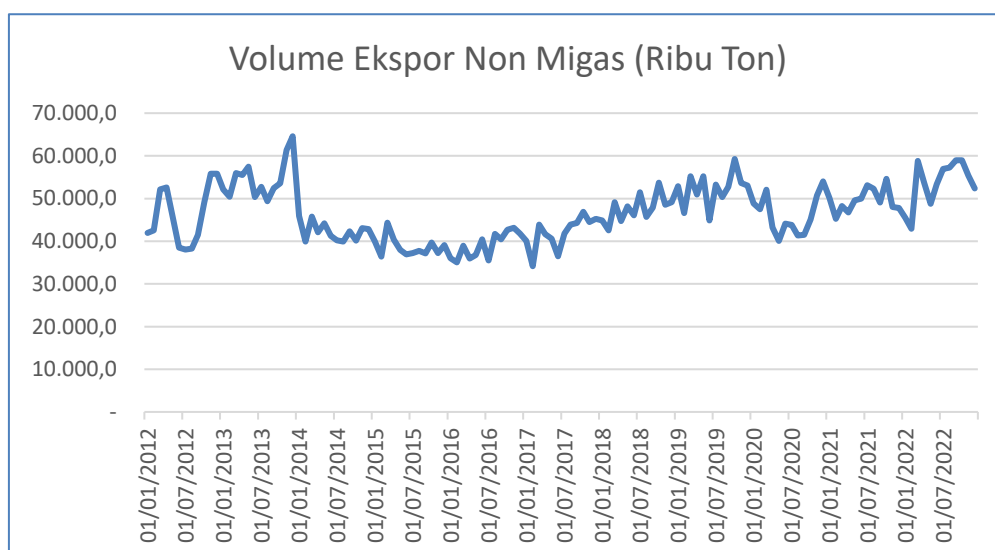
Pada subbab ini menjelaskan alur dalam membuat sistem peramalan permintaan ekspor nonmigas menggunakan metode *Long Short Term Memory*.



Gambar 3.1 Desain Sistem

3.1.1 Dataset Ekspor Nonmigas

Pada langkah awal, data *timeseries* bulanan mengenai volume ekspor nonmigas dipersiapkan dengan menggunakan sumber data dari situs web Badan Pusat Statistik. Peneliti menggunakan data sekunder yang berhubungan dengan volume ekspor nonmigas Indonesia dalam satuan ribu ton, yang diunduh dari situs web Badan Pusat Statistik www.bps.go.id. Dataset yang digunakan terdiri dari data bulanan dalam rentang waktu tahun 2012 hingga 2022. Dataset tersebut terdiri dari 132 baris dengan atribut yang terpilih yaitu "bulan" dan "volume", di mana masing-masing atribut memiliki tipe data *datetime* dan *float*.



Gambar 3.2 Dataset Volume Ekspor Nonmigas

3.1.2 Preprocessing Data

Pada tahap ini terbagi menjadi tiga, Pada *preprocessing* pertama yaitu dengan melakukan normalisasi pada data *timeseries* ekspor nonmigas menggunakan teknik *min-max scaling*.

Rumus yang digunakan pada proses ini adalah sebagai berikut :

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3.1)$$

Keterangan :

x' = volume ekspor nonmigas setelah normalisasi
 x = volume ekspor nonmigas sebelum dinormalisasi
 x_{min} = volume minimum ekspor nonmigas pada data *timeseries*
 x_{max} = volume maximum ekspor nonmigas pada data *timeseries*

Untuk melakukan normalisasi dengan menggunakan teknik *min-max scaling* dituliskan dengan kode dibawah ini :

```
#normalisasi data
min_value = np.min(np_data_unscaled)
max_value = np.max(np_data_unscaled)

np_data_scaled = (np_data_unscaled - min_value) / (max_value -
min_value)
print(np_data_scaled)
```

Gambar 3.3 Normalisasi Data

Pada *code* normalisasi data ditulis menggunakan bahasa pemrograman python dengan menghitung nilai minimum dan maksimum dari array dataset nonmigas yang belum dinormalisasi yang diinisialisasikan sebagai `np_data_unscaled` menggunakan fungsi `np.min()` dan `np.max()`. Selanjutnya, dilakukan normalisasi pada array `np_data_unscaled` dengan rumus $(np_data_unscaled - min_value) / (max_value - min_value)$. Rumus ini mengubah setiap elemen dalam array ke dalam rentang 0 hingga 1. Hasil normalisasi disimpan dalam array `np_data_scaled`. Terakhir, hasil normalisasi dicetak dengan perintah `print(np_data_scaled)`.

Tahap preprocessing yang kedua adalah segmentasi yaitu proses menyiapkan data sesuai format yang diminta sistem yang dalam penelitian ini

format yang diminta adalah *samples*, *time step* dan fitur. Untuk melakukan tahap segmentasi dituliskan dengan *code* berikut :

```
# Segmentasi Data
# Format data RNN [sampel, time steps dan features]
# banyaknya sample, sequence length baris time steps per sample, dan 1 fitur

seq_length = 12

def partition_dataset(seq_length, data):
    x, y = [], []
    data_len = data.shape[0]
    for i in range(seq_length, data_len):
        x.append(data[i-seq_length:i, ])
        y.append(data[i, 0])
    # Convert x dan y dalam bentuk numpy array
    x = np.array(x)
    y = np.array(y)
    return x, y
```

Gambar 3.4 Segmentasi data

Pada *code* segmentasi data mendefinisikan variabel `seq_length` yang merupakan jumlah *time steps* atau panjang urutan dalam satu *sample*. Terdapat fungsi `partition_dataset(seq_length, data)` yang digunakan untuk membagi dataset menjadi *input* (`x`) dan *output* (`y`) yang sesuai dengan format yang dibutuhkan oleh RNN. Fungsi `partition_dataset` mengambil dua parameter yaitu `seq_length` dan `data`. `seq_length` digunakan untuk menentukan panjang urutan dalam satu *sample*, sementara `data` adalah dataset yang akan dipartisi. Variabel `data_len` mengambil panjang dataset `data` dengan menggunakan fungsi `shape[0]`. Melalui perulangan `for`, mulai dari `seq_length` hingga `data_len - 1` untuk membagi dataset menjadi *sample* dan target. Dalam setiap iterasi perulangan, kita menambahkan urutan sebelumnya sepanjang `seq_length` ke dalam list `x`, dan menambahkan target atau nilai berikutnya ke dalam list `y`. Setelah selesai, kita mengubah `x` dan `y` menjadi array NumPy

dengan menggunakan `np.array (x)` dan `np.array (y)`. Fungsi `partition_dataset` mengembalikan `x` dan `y` sebagai *output*.

Tahap *preprocessing* yang ketiga yaitu pembagian data. Data volume ekspor nonmigas yang telah dilakukan normalisasi selanjutnya dilakukan pembagian data menjadi data latih dan data uji hal ini didasarkan pada penelitian (Sugiyarto & Abadi, 2019). Data latih akan diproses melalui proses pelatihan untuk mendapatkan kombinasi parameter *batch size* dan *epoch* terbaik. Sedangkan data uji akan diproses dari kombinasi parameter terbaik yang dihasilkan dari proses pelatihan. Untuk melakukan pembagian data berupa data latih dan uji ditulis dengan kode dibawah ini:

```
# Data latih sebanyak
n_datalatih = 78

# Pembagian data latih dan data uji
train_data = np_data_scaled[0:n_datalatih, ]
test_data = np_data_scaled[n_datalatih - seq_length:, ]

print(train_data.shape)
print(test_data.shape)

x_train, y_train = partition_dataset(seq_length, train_data)
x_test, y_test = partition_dataset(seq_length, test_data)

print(x_train.shape, y_train.shape)
print(x_test.shape, y_test.shape)

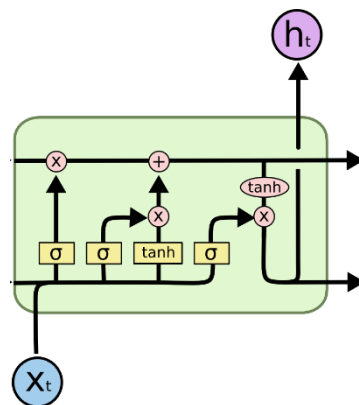
print(x_train)
print(y_train)
```

Gambar 3.5 Pembagian Data

Code pada gambar 3.5 digunakan untuk membagi dataset menjadi data latih dan data uji menggunakan segmentasi yang telah dijelaskan sebelumnya. Variabel `n_datalatih` menyimpan jumlah data latih yang diinginkan. Dalam hal ini, data latih terdiri dari `n_datalatih` sampel pertama dalam array `np_data_scaled`, sementara data

uji terdiri dari sisa sampel setelah data latih dengan panjang urutan `seq_length`. Menggunakan fungsi `print`, kita mencetak bentuk (`shape`) dari `train_data` dan `test_data` untuk memeriksa ukuran dataset yang sudah dibagi. Selanjutnya, memanggil fungsi `partition_dataset(seq_length, train_data)` dan `partition_dataset(seq_length, test_data)` untuk membagi data latih dan data uji menjadi input (`x`) dan output (`y`) sesuai dengan panjang urutan `seq_length`. Kemudian mencetak bentuk (`shape`) dari `x_train`, `y_train`, `x_test`, dan `y_test` untuk memeriksa ukuran masing-masing dataset. Terakhir, mencetak nilai `x_train` dan `y_train` untuk melihat isi dari data latih yang telah di-segmentasi.

3.1.3 Arsitektur Jaringan LSTM



Gambar 3.6 Arsitektur Jaringan LSTM
Sumber (Olah, 2015)

Sistem terdiri dari tiga komponen utama, yakni lapisan masukan (*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Setiap komponen memiliki peran yang berbeda. Lapisan masukan (*input layer*) berfungsi sebagai tempat penyimpanan nilai masukan di setiap *nodenya*. Lapisan tersembunyi (*hidden layer*) digunakan untuk mengurangi kesalahan (*error*) yang muncul pada keluaran (*output*) dengan cara memperkecilnya melalui adanya lapisan tambahan. Fungsi lapisan keluaran

(*output layer*) adalah untuk menampilkan hasil perhitungan sistem yang telah dilakukan. Pada lapisan tersembunyi, fungsi aktivasi dihasilkan dari *input* yang diterima dan diterapkan pada lapisan tersebut (Lecun dkk., 2015). Pada penelitian ini simpul *hidden layer* digantikan oleh LSTM.

Metode Long Short Term Memory yang diterapkan pada lapisan kedua dari struktur jaringan yang telah dikonstruksi. Data input yang telah disesuaikan dengan skala diproses menggunakan fungsi-fungsi yang ada dalam unit LSTM. Output dari unit-unit dalam lapisan LSTM kemudian diproses di lapisan output. Pada lapisan output, terdapat kaitan antara fungsi aktivasi sigmoid dengan bobot dan bias.

Berikut merupakan proses yang terjadi pada *layer* LSTM:

1. Setelah data dinormalisasi pada *layer input*, bobot dan bias di *layer* kedua perlu diinisialisasi secara acak untuk menjalankan fungsi-fungsi yang ada.
2. Setelah melakukan inisialisasi bobot dan bias, langkah selanjutnya adalah melakukan perhitungan nilai pada proses *forget gate* (f_t). *Forget gate* memutuskan seberapa banyak informasi lama harus dihapus dari memori jangka panjang. Ini mengambil input saat ini dan hidden state sebelumnya, melewatkannya melalui fungsi aktivasi *sigmoid*, dan menghasilkan nilai antara 0 dan 1 untuk setiap elemen memori. Nilai 0 menunjukkan informasi yang harus dihapus, sedangkan nilai 1 menunjukkan informasi yang harus dipertahankan. Berikut merupakan formula yang digunakan untuk menghitung nilai pada *forget gate* (f_t):

$$f_t = \sigma(W_f \cdot h_{t-1} + W_f \cdot x_t + b_f) \quad (3.2)$$

Keterangan :

- f_t : Hasil gate lupa (*forget gate*) pada *time step t*.
 σ : Fungsi aktivasi *sigmoid*.
 W_f : Bobot untuk *gate* lupa.
 h_{t-1} : Hasil keluaran pada *time step t-1* (*hidden state* sebelumnya).
 x_t : Masukan pada *time step* setelah dilakukan normalisasi.
 b_f : Bias untuk *gate* lupa.

3. Setelah memperoleh nilai dari *forget gate*, langkah berikutnya adalah menghitung nilai pada input gate dengan menggunakan data input yang telah dinormalisasi dan menerapkan bobot dan bias yang telah diinisialisasi sebelumnya.. *Input gate* mengendalikan seberapa banyak informasi baru yang harus ditambahkan ke memori jangka panjang. Ini mengambil input saat ini dan *hidden state* sebelumnya, melewatkannya melalui fungsi aktivasi *sigmoid*, dan menghasilkan nilai antara 0 dan 1 untuk setiap elemen memori. Nilai ini menunjukkan sejauh mana informasi baru harus dimasukkan ke dalam memori. Berikut merupakan formula yang digunakan untuk menghitung nilai pada input gate :

$$i_t = \sigma(W_i \cdot h_{t-1} + W_i \cdot x_t + b_i) \quad (3.3)$$

Keterangan :

- i_t : Hasil *gate input* (*input gate*) pada *time step t*.
 σ : Fungsi aktivasi *sigmoid*.
 W_i : Bobot untuk *gate input*.
 h_{t-1} : Keluaran pada *time step t-1* (*hidden state* sebelumnya).
 x_t : Masukan pada *time step t* setelah dilakukan normalisasi.
 b_i : Bias untuk *gate input*.

4. Selanjutnya, dilakukan perhitungan pada proses *cell gate*. Berikut merupakan formula yang digunakan untuk menghitung nilai pada *cell gate*:

$$C'_t = \tanh(W_c \cdot h_{t-1} + W_c \cdot x_t + b_c) \quad (3.4)$$

Keterangan :

$C't$: Hasil kandidat (*candidate value*) pada *time step t*.
 Wc : Bobot untuk kandidat.
 bc : Bias untuk kandidat.

Hasil dari perhitungan pada *cell gate* digunakan untuk menghitung nilai pada *cell state*, yang juga melibatkan hasil dari *input gate*. *Cell gate* menggabungkan informasi dari *forget gate* dan *input gate* untuk menghitung pembaruan nilai memori jangka panjang. *Forget gate* mengendalikan seberapa banyak informasi lama yang akan dihapus, sementara *input gate* mengendalikan seberapa banyak informasi baru yang akan ditambahkan. Informasi ini digabungkan untuk menghasilkan nilai baru untuk memori jangka panjang. Berikut merupakan formula yang digunakan untuk menghitung nilai pada *cell state*:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \quad (3.5)$$

Keterangan :

C_t : Hasil *cell state* pada *time step t*.
 f_t : Hasil *gate* lupa (*forget gate*) pada *time step t*.
 C_{t-1} : Hasil *cell state* pada *time step* sebelumnya.
 i_t : Hasil *gate* input (*input gate*) pada *time step t*.
 $C't$: Hasil kandidat (*candidate value*) pada *time step t*.
 $*$: Operasi perkalian *elemen-wise* (*element-wise multiplication*).

- Selanjutnya dilakukan penghitungan *output gate* (o_t). *Output gate* mengendalikan seberapa banyak informasi dari memori jangka panjang yang akan dikeluarkan. Ini mengambil input saat ini dan hidden state sebelumnya, melewatkannya melalui fungsi aktivasi *sigmoid*, dan menghasilkan nilai antara 0 dan 1 untuk setiap elemen memori jangka panjang. Nilai ini menunjukkan sejauh mana informasi yang akan

dikeluarkan. Berikut merupakan formula yang digunakan untuk menghitung nilai pada *output gate*:

$$o_t = \sigma(W_o \cdot x_t + W_o \cdot h_{t-1} + b_o) \quad (3.6)$$

Keterangan :

- o_t : Nilai *gate output (output gate)* pada *time step t*.
- σ : Fungsi aktivasi *sigmoid*
- W_o : bobot untuk *output gate*
- x_t : *Input* pada *time step t*
- h_{t-1} : Keluaran pada *time step t-1 (hidden state sebelumnya)*
- b_o : bias pada gerbang keluaran

6. Setelah setiap gerbang menghasilkan nilai, selanjutnya adalah dilakukan penghitungan pada *hidden state (h_t)*. *Hidden state* dihasilkan dengan mengalikan *output gate* dengan nilai yang telah diolah dari memori jangka panjang menggunakan fungsi aktivasi tertentu. Nilai ini kemudian akan digunakan sebagai *hidden state* pada langkah waktu selanjutnya. Berikut merupakan formula yang digunakan untuk menghitung nilai pada *hidden state*:

$$h_t = o_t \cdot \tanh(C_t) \quad (3.7)$$

Keterangan :

- h_t : Nilai *output* pada *time step t*
- o_t : bias pada gerbang keluaran
- Tanh : fungsi aktivasi *tanh*
- C_t : *cell state* pada *time step t*

3.1.4 Proses Training

Sebelum memulai pelatihan (*training*), perlu dilakukan pemodelan LSTM dengan 3 *layer*, menentukan ukuran *batch (batch size)*, memilih jenis *optimizer*,

menentukan jumlah *epoch*, dan menggunakan *Mean Squared Error* sebagai *fungsi loss*. Proses pelatihan akan dilakukan dengan melatih model menggunakan data latih. Bobot (*weights*) dan bias akan diperbarui secara terus-menerus untuk mendapatkan model yang sesuai.

Batch size mengacu pada jumlah sampel data yang diproses dalam satu iterasi ketika melatih model. Ukuran *batch* dapat bervariasi, dengan batch besar menghasilkan pembaruan parameter yang lebih jarang tetapi lebih efisien dalam komputasi, sementara batch kecil menghasilkan pembaruan yang lebih sering tetapi membutuhkan waktu komputasi yang lebih lama. Sementara itu, *epoch* merujuk pada satu siklus lengkap ketika seluruh data pelatihan telah dilalui oleh model. Setiap epoch melibatkan pemrosesan semua sampel data, termasuk semua *batch*, dan digunakan untuk mengukur kinerja, menghitung *gradien*, dan memperbarui parameter-model. Memilih ukuran *batch* dan jumlah *epoch* yang tepat adalah penting dalam pelatihan model untuk mencapai tingkat akurasi yang baik dan mencegah *overfitting* (Brownlee, 2018).

Selain bobot, proses RNN LSTM juga melibatkan inisialisasi parameter pembelajaran. Parameter pembelajaran dalam RNN LSTM pada penelitian ini mencakup nilai *learning rate*, jumlah *epoch*, dan fungsi aktivasi. Proses pembelajaran berhenti ketika nilai kesalahan mencapai target atau mencapai jumlah iterasi maksimum yang ditentukan.

Learning rate adalah parameter yang mengontrol seberapa besar pembaruan parameter-model selama pelatihan. Ini menentukan sejauh mana langkah yang diambil dalam arah *gradien* saat mengoptimalkan model. *Learning rate* yang terlalu

besar dapat menyebabkan ketidakstabilan, sementara yang terlalu kecil bisa memperlambat konvergensi. Pemilihan learning rate yang tepat penting untuk mencapai konvergensi yang baik dan efisiensi dalam pelatihan model (Brownlee, 2020).

Proses *training* yang terjadi pada *Long Short Term Memory* adalah sebagai berikut:

- a. Setelah mendapatkan data *training* yang telah dinormalisasi, langkah selanjutnya adalah melakukan inisialisasi bobot dan bias.
- b. Lakukan perhitungan fungsi *gates* pada setiap *neuron* secara berurutan. Langkah-langkahnya dimulai dengan menghitung *forget gates*, kemudian *input gates*, dan terakhir *output gates*, menggunakan persamaan yang telah diketahui sebelumnya.
- c. Jika jumlah perulangan telah mencapai jumlah *epoch* yang telah ditentukan, proses dihentikan. Jika belum, akan dilakukan optimisasi menggunakan metode optimisasi Adam dan dilakukan pembaruan pada bobot dan bias dalam sistem. Kemudian, proses kembali ke langkah dua untuk melanjutkan iterasi selanjutnya.

Setelah satu putaran pelatihan, langkah validasi dilakukan. Dalam validasi, setiap kelompok data akan diuji menggunakan dataset validasi untuk memeriksa kecocokan hasil keluaran dengan nilai yang sebenarnya. Proses validasi ini menghasilkan suatu nilai kesalahan yang disebut "loss", yang digunakan sebagai parameter untuk menentukan bobot yang akan diberikan pada kelompok data berikutnya. Untuk melakukan penentuan parameter guna membangun arsitektur LSTM serta melakukan training ditulis dengan code pada gambar 3.7

```

# Konstruksi Model LSTM
def build_model(optimizer):
    grid_model = Sequential()
    grid_model.add(LSTM(16, activation='tanh',
        recurrent_activation='sigmoid',
        return_sequences=True, input_shape=(x_train.shape[1], x_train
        .shape[2])))
    grid_model.add(LSTM(16, activation='tanh',
        recurrent_activation='sigmoid'))
    grid_model.add(Dense(1))

    grid_model.compile(loss = 'mse', optimizer=tf.keras
    .optimizers.Adam(learning_rate=0.001))
    return grid_model
grid_model = KerasRegressor(build_fn=build_model, verbose=1)
parameters = {'batch_size' : [2,3,6,13],
    'epochs' : [50,100,150,200],
    'optimizer' : ['adam'] }

grid_search = GridSearchCV(estimator = grid_model,
    param_grid = parameters, cv = 3)

# Menyesuaikan model dengan data
grid_search = grid_search.fit(x_train,y_train)
print(grid_search.best_params_)
model_terbaik = grid_search.best_estimator_.model

```

Gambar 3.7 Pembangunan Model LSTM

Code di atas merupakan pembangunan model LSTM menggunakan Keras untuk melakukan optimisasi dengan menggunakan teknik Grid Search.

1. Fungsi `build_model(optimizer)` digunakan untuk membangun arsitektur model LSTM. Model ini memiliki dua lapisan LSTM dengan masing-masing memiliki 16 unit dan menggunakan fungsi aktivasi tangen hiperbolik (*tanh*) dan aktivasi *rekuren sigmoid*. *Input shape* dari model ini adalah bentuk dari `x_train`, yaitu (jumlah *sample*, panjang urutan, fitur).
2. Selanjutnya, menambahkan lapisan *Dense* dengan satu unit pada model untuk menghasilkan output.

3. Model tersebut dikompilasi dengan menggunakan fungsi `compile()`. *Loss function* yang digunakan adalah *mean squared error (mse)*, dan *optimizer* yang digunakan adalah Adam dengan *learning rate* 0.001.
4. Membuat objek `grid_model` menggunakan `KerasRegressor` yang menerima fungsi `build_model` sebagai argumen. `verbose=1` digunakan untuk menampilkan detail proses *training* model.
5. Variabel *parameters* mengandung kombinasi parameter yang ingin dioptimasi menggunakan *Grid Search*. Dalam hal ini, mencari kombinasi terbaik dari *batch_size* (2, 3, 6, 13), *epochs* (50, 100, 150 dan 200) dan menggunakan *optimizer adam*.
6. Objek `GridSearchCV` digunakan untuk melakukan *grid search* dengan *estimator* `grid_model` dan parameter *grid parameters* menggunakan 3-fold cross-validation (`cv=3`) untuk mengevaluasi performa model.
7. Melalui perintah `grid_search.fit(x_train, y_train)`, model akan disesuaikan dengan data latih dan mencari parameter terbaik menggunakan *grid search*.
8. Setelah proses *grid search* selesai, kita mencetak parameter terbaik yang ditemukan dengan `grid_search.best.params_`.
9. `grid_search.best_estimator_.model` mengambil model terbaik dari hasil *grid search* dan menyimpannya dalam variabel `model_terbaik`. Model ini dapat digunakan untuk melakukan peramalan pada data uji.

3.1.5 Proses *Testing*

Kombinasi parameter model terbaik hasil proses pelatihan kemudian digunakan pada proses *testing*. Tujuan dilakukannya proses *testing* adalah untuk

mendapatkan nilai *error* pada model yang dihasilkan oleh proses *testing*. Untuk melakukan proses *testing* ditulis dengan kode dibawah ini:

```
# nilai peramalan (normalisasi)
y_ramal_scaled = model_terbaik.predict(x_test)
print(y_ramal_scaled)
```

Gambar 3.8 Proses Peramalan

Code pada gambar 3.8 digunakan untuk melakukan peramalan pada data uji menggunakan model terbaik yang telah ditemukan. `model_terbaik.predict(x_test)` digunakan untuk melakukan peramalan pada data uji `x_test` dengan menggunakan model terbaik yang telah ditemukan melalui proses *grid search* sebelumnya. Hasil peramalan ini masih dalam bentuk nilai yang telah dinormalisasi. Hasil peramalan tersebut disimpan dalam variabel `y_ramal_scaled`. Terakhir, mencetak nilai peramalan dalam bentuk normalisasi dengan perintah `print(y_ramal_scaled)`.

Kemudian digunakan formula berikut untuk menghasilkan nilai yang sudah di denormalisasi :

$$x_t = x(x'max - x'min) + x'min \quad (3.8)$$

Keterangan :

x_t	= nilai peramalan ekspor sebenarnya
x	= nilai hasil peramalan ekspor
$x'max$	= nilai maximum pada data hasil peramalan ekspor
$x'min$	= nilai maximum pada data hasil peramalan ekspor

```
min_value = np.min(np_data_unscaled)
max_value = np.max(np_data_unscaled)

print("data peramalan")
print("=====")

# Penskalaan peramalan
y_ramal_scaled_copies_array = np.repeat(y_ramal_scaled, 1, axis=-1)
y_ramal = ((y_ramal_scaled_copies_array * (max_value - min_value)) + min_value)
print (y_ramal)
```

Gambar 3.9 Denormalisasi Data

Code pada gambar 3.9 bertujuan untuk melakukan denormalisasi pada hasil peramalan yang telah dilakukan sebelumnya dan mengembalikannya ke dalam bentuk semula yang tidak ternormalisasi. `min_value` dan `max_value` dihitung dengan menggunakan fungsi `np.min()` dan `np.max()` pada `np_data_unscaled`. `np_data_unscaled` merupakan data awal sebelum dilakukan normalisasi. Kemudian, dilakukan pencetakan judul "data peramalan" dan pemisah garis untuk memisahkan output dengan bagian sebelumnya. Selanjutnya, dilakukan penskalaan peramalan. Pada baris ini, terdapat proses membuat salinan hasil peramalan yang dinormalisasi dengan menggunakan fungsi `np.repeat()`. Salinan ini digunakan untuk memastikan bahwa dimensi hasil peramalan sesuai dengan dimensi data awal sebelum normalisasi. Hasil peramalan yang dinormalisasi (`y_ramal_scaled`) dikalikan dengan selisih nilai maksimum dan minimum dari data awal (`max_value - min_value`), kemudian ditambahkan nilai minimum dari data awal (`min_value`). Hal ini dilakukan untuk mengembalikan hasil peramalan ke dalam bentuk semula yang tidak ternormalisasi. Hasil peramalan yang telah dikembalikan ke dalam bentuk semula disimpan dalam variabel `y_ramal`. Terakhir, dilakukan pencetakan nilai

peramalan yang telah dikembalikan ke dalam bentuk semula dengan menggunakan perintah `print(y_ramal)`.

3.1.6 Evaluasi MAPE

Selanjutnya setelah mendapatkan nilai hasil peramalan kemudian dilakukan perhitungan *error* dengan menggunakan *Mean Absolute Percentage Error*.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{At - Ft}{At} \right| \times 100\% \quad (3.9)$$

Keterangan :

MAPE = hasil perhitungan *Mean Absolute Percentage Error*

n = jumlah data volume ekspor nonmigas

At = nilai aktual volume ekspor nonmigas

Ft = nilai peramalan volume ekspor nonmigas

```
# Mean Absolute Percentage Error (MAPE)
MAPE = np.mean((np.abs(np.subtract(y_test_unscaled, y_ramal)/
y_test_unscaled))) * 100
print(f'Mean Absolute Percentage Error (MAPE): {np.round(MAPE,2)} %')
```

Gambar 3.10 Perhitungan MAPE

Koding di atas bertujuan untuk menghitung *Mean Absolute Percentage Error (MAPE)* atau Kesalahan Persentase Rata-rata Mutlak pada hasil peramalan. Pada baris pertama, MAPE dihitung menggunakan rumus yang digunakan untuk menghitung persentase kesalahan antara nilai aktual (`y_test_unscaled`) dengan nilai peramalan yang telah dikembalikan ke dalam bentuk semula (`y_ramal`). `np.mean()` digunakan untuk menghitung rata-rata dari persentase kesalahan tersebut. Terakhir, hasilnya dikalikan dengan 100 untuk mendapatkan persentase. Pada baris kedua,

dilakukan pencetakan hasil MAPE. Hasil MAPE akan ditampilkan dengan 2 angka di belakang koma.

3.1.7 Tampilan Sistem

Peramalan pada bulan selanjutnya dibuat dalam bentuk *Graphical User Interface* (GUI) menggunakan *library tkinter*. Pengguna dapat memasukkan jumlah bulan kedepan yang ingin dilakukan peramalan kemudian akan menghasilkan volume hasil peramalan dalam satuan ribu ton.

```

forecast_steps = 0 # Variabel global untuk menyimpan jumlah bulan yang ingin diramalkan
forecast_data = [] # List kosong untuk menyimpan nilai hasil peramalan

def lakukan_peramalan():
    global forecast_steps, forecast_data

    # Dapatkan nilai input dari widget entry
    forecast_steps = int(entry.get())

    # Ambil `seq_length` data terakhir dari data uji sebagai masukan awal untuk peramalan
    input_data = np_data_scaled[-seq_length:, ]

    # Lakukan iterasi sebanyak langkah peramalan yang diinginkan
    for _ in range(forecast_steps):
        # Ubah bentuk data masukan agar sesuai dengan bentuk masukan model
        x_input = np.reshape(input_data, (1, seq_length, np_data_scaled.shape[1]))

        # Lakukan peramalan menggunakan model yang telah dilatih
        y_pred_scaled = model_terbaik.predict(x_input)

        # Kembalikan skala nilai hasil peramalan
        y_pred = ((y_pred_scaled * (max_value - min_value)) + min_value)[0, 0]

        # Tambahkan nilai hasil peramalan ke dalam list hasil
        forecast_data.append(y_pred)

        # Perbarui data masukan untuk langkah peramalan berikutnya
        input_data = np.concatenate((input_data[1:, ], np.reshape(y_pred_scaled, (1, 1))), axis=0)

    # Tampilkan volume hasil peramalan dalam label
    label_hasil.config(text="Volume Peramalan: " + str(forecast_data))

```

Gambar 3.11 Proses Peramalan Bulan Selanjutnya

Fungsi `lakukan_peramalan()` adalah implementasi dari algoritme peramalan. Fungsi ini menerima input dari pengguna yang merupakan jumlah bulan yang ingin diramalkan. Kemudian, fungsi ini menggunakan data terakhir dari data uji sebagai masukan awal untuk peramalan.

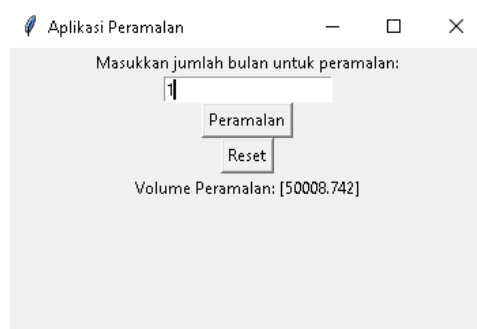
Selanjutnya, fungsi melakukan iterasi sebanyak langkah peramalan yang diinginkan. Pada setiap iterasi, data masukan diubah menjadi bentuk yang sesuai dengan masukan model peramalan. Kemudian, peramalan dilakukan menggunakan model yang telah dilatih sebelumnya. Hasil peramalan yang diperoleh awalnya dalam bentuk yang telah diubah skala.

Selanjutnya, hasil peramalan dikembalikan ke dalam skala aslinya menggunakan rumus yang diberikan. Nilai hasil peramalan tersebut ditambahkan ke dalam list `forecast_data`, yang berfungsi untuk menyimpan hasil peramalan secara berurutan.

Data masukan diperbarui untuk langkah peramalan berikutnya dengan menghapus data pertama dan menambahkan nilai peramalan terbaru. Hal ini dilakukan untuk memperbarui konteks dan memberikan informasi terbaru pada setiap iterasi peramalan.

Hasil peramalan yang telah dikonversi ke dalam skala aslinya kemudian ditampilkan dalam label dengan menggunakan metode `label_hasil.config()`. Teks label akan menampilkan volume hasil peramalan dalam format yang diinginkan.

Dengan demikian, fungsi `lakukan_peramalan()` memberikan kemampuan untuk melakukan peramalan berdasarkan input pengguna, dan menampilkan hasil peramalan secara berurutan dalam label.



Gambar 3.12 Tampilan Sistem

Berikut adalah penjelasan dari tampilan sistem :

1. Masukkan jumlah bulan untuk peramalan : Ini adalah label yang memberikan instruksi kepada pengguna untuk memasukkan jumlah bulan yang ingin diperamalkan.
2. *Entry Widget* : Ini adalah tempat di mana pengguna dapat memasukkan jumlah bulan yang ingin diperamalkan. Pengguna dapat mengetik angka bulan di sini
3. *Peramalan Button* : Ini adalah tombol yang akan memicu proses peramalan ketika diklik. Ketika tombol ini diklik, algoritma peramalan akan dijalankan dan hasil peramalan akan ditampilkan.
4. *Reset Button*: Ini adalah tombol yang digunakan untuk mereset input pengguna dan menghapus hasil peramalan yang ditampilkan. Ketika tombol ini diklik, input pengguna akan dikosongkan, dan hasil peramalan akan dihapus.
5. *Volume Peramalan* : Ini adalah label yang menampilkan hasil peramalan dalam bentuk volume. Setelah tombol "Peramalan" diklik, hasil peramalan akan ditampilkan di sini. Jumlah bulan peramalan yang ditampilkan akan sesuai dengan input pengguna.

Setelah pengguna memasukkan jumlah bulan untuk peramalan dan mengklik tombol "Peramalan", hasil peramalan akan ditampilkan dalam label "Volume Peramalan". Jika pengguna ingin mereset input dan hasil peramalan, mereka dapat mengklik tombol "Reset" untuk mengosongkan input dan menghapus hasil peramalan yang ditampilkan.

3.2 Skenario Pengujian

Skenario pengujian adalah rangkaian prosedur yang dirancang untuk menguji dan mengevaluasi kinerja serta efektivitas metode peramalan permintaan ekspor nonmigas Indonesia menggunakan *Long Short Term Memory (LSTM)*. Skenario ini bertujuan untuk memvalidasi kemampuan LSTM dalam memprediksi permintaan ekspor nonmigas dengan nilai *error* yang rendah.

Tabel 3. 1 Pembagian dataset

Dataset	Training	Testing	Batch Size	Learning Rate
Model 1	60% (78 Data)	40% (54 Data)	2,3,6,13	0,001
Model 2	70% (96 Data)	30% (36 Data)	3,4,6,12	0,001
Model 3	80% (108 Data)	20% (24 Data)	3,4,6,12	0,001
Model 4	90% (120 Data)	10 % (12 Data)	3,4,6,12	0,001
Model 5	60% (78 Data)	40% (54 Data)	2,3,6,13	0,01
Model 6	70% (96 Data)	30% (36 Data)	3,4,6,12	0,01
Model 7	80% (108 Data)	20% (24 Data)	3,4,6,12	0,01
Model 8	90% (120 Data)	10 % (12 Data)	3,4,6,12	0,01

Penentuan *batch size* didasarkan pada nilai faktorisasi dari jumlah data latih dan data uji. Hal tersebut dilakukan karena dapat memberikan beberapa keuntungan berupa efisiensi memori. Jika *batch size* merupakan faktorisasi dari jumlah dataset, maka akan memastikan bahwa setiap *batch* akan menggunakan seluruh data latih atau uji secara optimal. Dalam kasus ini, tidak akan ada "sisipan" atau "sisa" data yang tidak digunakan dalam proses pelatihan atau pengujian.

Kecepatan pelatihan dalam beberapa kasus, menggunakan *batch size* yang lebih kecil pada tahap pelatihan dapat mempercepat proses pelatihan model. Dengan *batch size* kecil, update parameter dapat dilakukan lebih sering dan secara lebih cepat, karena waktu komputasi yang dibutuhkan untuk memproses setiap langkah pembelajaran lebih singkat. Ini dapat menghasilkan konvergensi yang lebih cepat dan memungkinkan model untuk mencapai hasil yang baik dengan waktu pelatihan yang lebih sedikit.

Evaluasi yang stabil dengan menggunakan *batch size* yang lebih besar pada tahap pengujian, kita dapat memperoleh perkiraan kinerja model yang lebih stabil dan andal. Dalam beberapa kasus, menggunakan *batch size* yang kecil pada tahap pengujian dapat menghasilkan perkiraan yang bervariasi secara signifikan karena fluktuasi dalam sampel data yang digunakan untuk evaluasi. Dengan menggunakan *batch size* yang lebih besar, model akan mengambil rata-rata prediksi pada *batch* yang lebih besar, yang dapat menghasilkan estimasi yang lebih stabil dan mewakili performa model secara keseluruhan dengan lebih baik.

Melalui skenario-skenario pengujian ini, diharapkan dapat ditemukan kombinasi yang optimal dari pembagian data, *learning rate*, *batch size*, dan jumlah

epoch untuk mencapai kinerja dan akurasi peramalan permintaan ekspor nonmigas Indonesia yang optimal menggunakan model LSTM.

Dalam penelitian ini, model LSTM yang dibangun terdiri dari tiga layer dengan rincian dua layer LSTM dan satu layer dense. Pada layer input, terdapat 16 neuron, diikuti oleh layer LSTM dengan jumlah neuron yang disesuaikan. Pada layer dense, terdapat satu neuron pada output layer.

Tabel 3.2 Parameter pada LSTM

No	Parameter	Jumlah/Keterangan
1	<i>Activation</i>	<i>Tanh & Sigmoid</i>
2	<i>Optimizer</i>	<i>Adam</i>
3	<i>Epoch</i> maksimal	50, 100, 150,200
4	<i>Learning Rate</i>	0.001 , 0.01 , dan 0.1

Beberapa parameter yang digunakan mengikuti *default keras*, seperti fungsi aktivasi *tanh* dan *sigmoid*, serta *optimizer* biasa. Variabel lainnya dipilih setelah melalui beberapa percobaan.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini dijelaskan tentang implementasi serta pembahasan dari skenario pengujian untuk membuat sistem peramalan permintaan ekspor nonmigas. Dalam bab ini juga dijelaskan integrasi antara penelitian dengan islam.

4.1 Hasil

4.1.1 Preprocessing

Pada tahap *preprocessing* yang pertama adalah dengan melakukan normalisasi pada data dengan menggunakan metode *min-max scaling*. Selanjutnya dilakukan segmentasi data yaitu dengan membuat data sesuai dengan format yang dibutuhkan oleh metode yakni *long short term memory*. Setelah dilakukan normalisasi kemudian dilakukan pembagian data untuk mendapatkan data *training* dan data *testing* dengan beberapa rasio yaitu 60%:40 %, 70%:30%, 80%:20%, 90%:10% yang diambil dari tabel 3.1. Hasil *preprocessing* data terlampir.

4.1.2 Membuat Model LSTM menggunakan Data Latih

Untuk membuat model LSTM menggunakan data latih memerlukan beberapa parameter pendukung. Beberapa parameter yang dimaksud adalah sebagai berikut :

1. Jumlah *layer* : 3
2. *Neuron* : 16
3. *Epoch Maksimum* : 50,100,150,200
4. *Optimizer* : *Adam*
5. Fungsi aktivasi : *Sigmoid dan tanh*
6. *Learning rate* : 0.001 , 0.01 , 0.1

Parameter-parameter yang terpilih tidak memiliki aturan yang pasti, sehingga dapat disesuaikan dengan kebutuhan sistem. Model LSTM dipengaruhi

oleh jumlah masukan dan arsitektur yang dibuat berdasarkan parameter-parameter yang telah disebutkan sebelumnya.

Layer (type)	Output Shape	Param #
lstm_12 (LSTM)	(None, 12, 16)	1152
lstm_13 (LSTM)	(None, 16)	2112
dense_6 (Dense)	(None, 1)	17
Total params: 3,281		
Trainable params: 3,281		
Non-trainable params: 0		

Gambar 4.1 Model LSTM yang digunakan

Gambar 4.1 mengilustrasikan bahwa model tersebut terdiri dari 3 layer tersembunyi. Pada lapisan awal, terdapat lapisan masukan (input) yang menerima masukan dalam format tiga dimensi yang terdiri dari sampel (jumlah baris data), fitur dan *timestep*. Kemudian, ada lapisan kedua dengan 16 neuron yang menghasilkan total 2112 parameter. Terakhir, terdapat lapisan *dense* yang menunjukkan bahwa lapisan tersebut memiliki satu keluaran, yaitu nilai target (y). Model yang telah dibangun memiliki total 3281 parameter, termasuk bobot dan bias.

4.1.3 Proses *Training*

Pada proses *training* atau pelatihan menggunakan data latih yang diperoleh dari pembagian data pada proses *preprocessing* dengan beberapa macam skenario pembagian data yang sudah ditentukan. model yang telah ditentukan dilakukan proses *training* menghasilkan output model terbaik sesuai parameter yang dibentuk. Hasil parameter terbaik pada masing-masing model terdapat pada tabel 4.1.

Tabel 4.1 Parameter terbaik pada setiap model percobaan

Model	Learning rate	Parameter Terbaik
Model 1	0,001	<i>Batch size: 2, epochs: 50</i>
Model 2	0,001	<i>Batch size: 3, epochs: 100</i>
Model 3	0,001	<i>Batch size: 6, epochs: 100</i>
Model 4	0,001	<i>Batch size: 3, epochs: 50</i>
Model 5	0,01	<i>Batch size: 3, epochs: 50</i>
Model 6	0,01	<i>Batch size: 12, epochs: 50</i>
Model 7	0,01	<i>Batch size: 12, epochs: 50</i>
Model 8	0,01	<i>Batch size: 12, epochs: 50</i>
Model 9	0,1	<i>Batch size: 6, epochs: 50</i>
Model 10	0,1	<i>Batch size: 3, epochs: 50</i>
Model 11	0,1	<i>Batch size: 4, epochs: 50</i>
Model 12	0,1	<i>Batch size: 12, epochs: 50</i>

Proses mencari model terbaik pada kode sistem peramalan dilakukan dengan menggunakan *GridSearchCV*, yang merupakan bagian dari *scikit-learn*. *GridSearchCV* adalah sebuah metode untuk melakukan pencarian parameter terbaik pada model yang diberikan dengan mencoba semua kombinasi parameter yang mungkin.

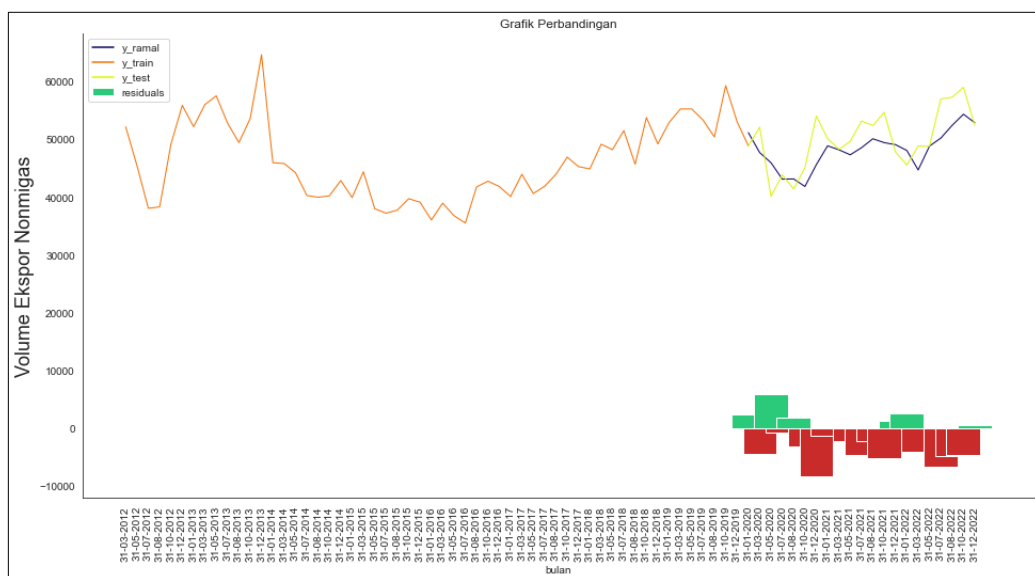
4.1.4 Proses *Testing*

Parameter terbaik menunjukkan akurasi terbaik yang didapatkan dari hasil training model percobaan. Untuk menggambarkan volume hasil peramalan dan aktual ditunjukkan dengan grafik. Pada grafik tersebut menampilkan tiga warna grafik yaitu *y_ramal*, *y_train*, dan *y_test*. Warna jingga menampilkan data latih, sedangkan warna kuning menampilkan data uji sedangkan warna biru menampilkan

data volume ekspor nonmigas hasil peramalan.

Pada grafik juga menampilkan grafik residual. Dalam konteks peramalan, residual yang bernilai positif menunjukkan bahwa model meramal nilai yang terlalu rendah, sedangkan residual yang bernilai negatif menunjukkan bahwa model meramal nilai yang terlalu tinggi. Residual yang ideal adalah nol, yang berarti model peramalan dapat meramal dengan sempurna. Namun, dalam kenyataannya, residual selalu ada. Gambar grafik diambil dari parameter terbaik masing-masing model percobaan.

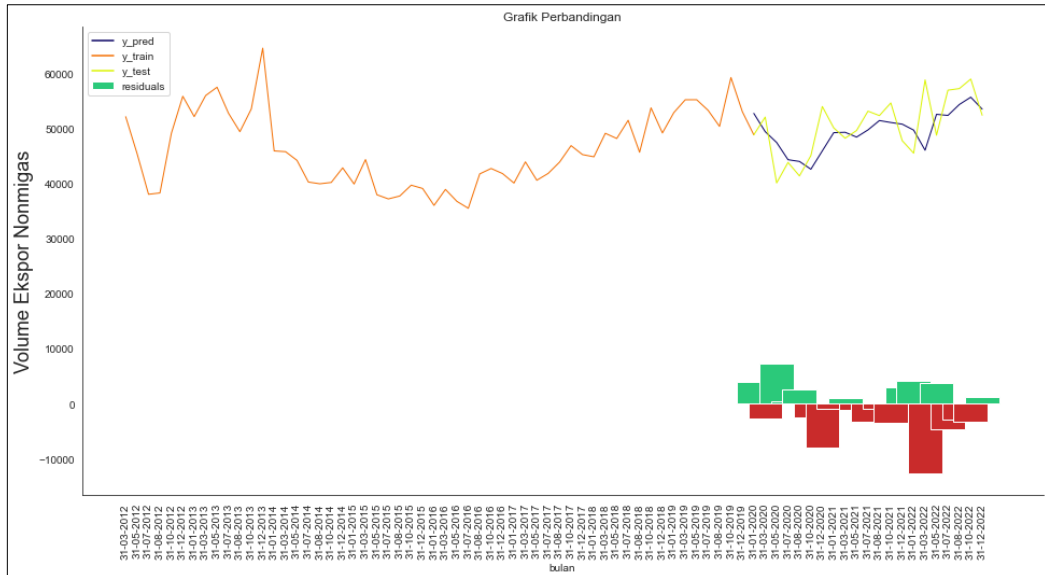
4.1.4.1 Pengujian Model 1



Gambar 4.2 Grafik Model 1 dengan Parameter *Batch size:2, epochs: 50*

Pengujian model 1 menggunakan *learning rate* 0,001 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 2 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.2.

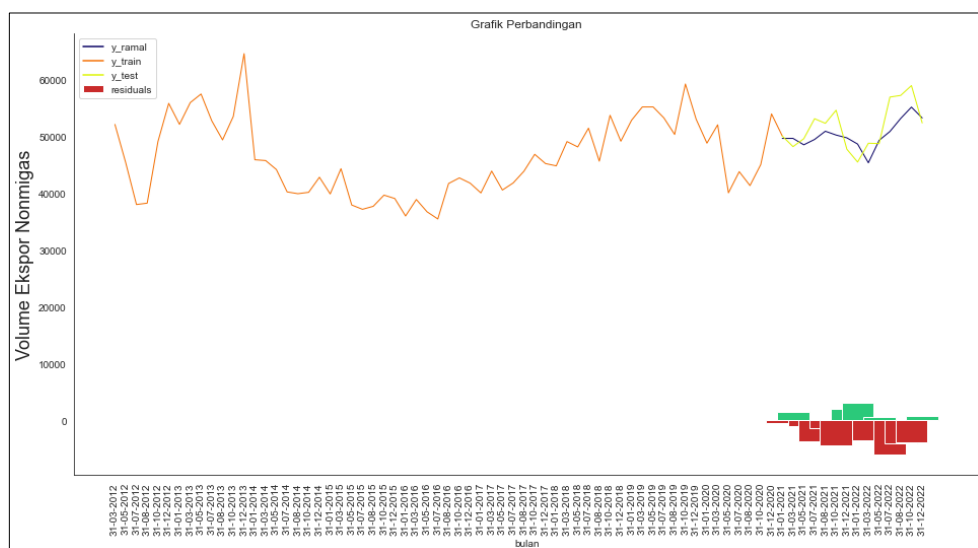
4.1.4.2 Pengujian Model 2



Gambar 4.3 Grafik Model 2 dengan Parameter *Batch size: 3, epochs: 100*

Pengujian model 2 menggunakan *learning rate* 0,001 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 3 dan *epoch* 100. Hasil pengujian digambarkan dalam diagram pada gambar 4.3.

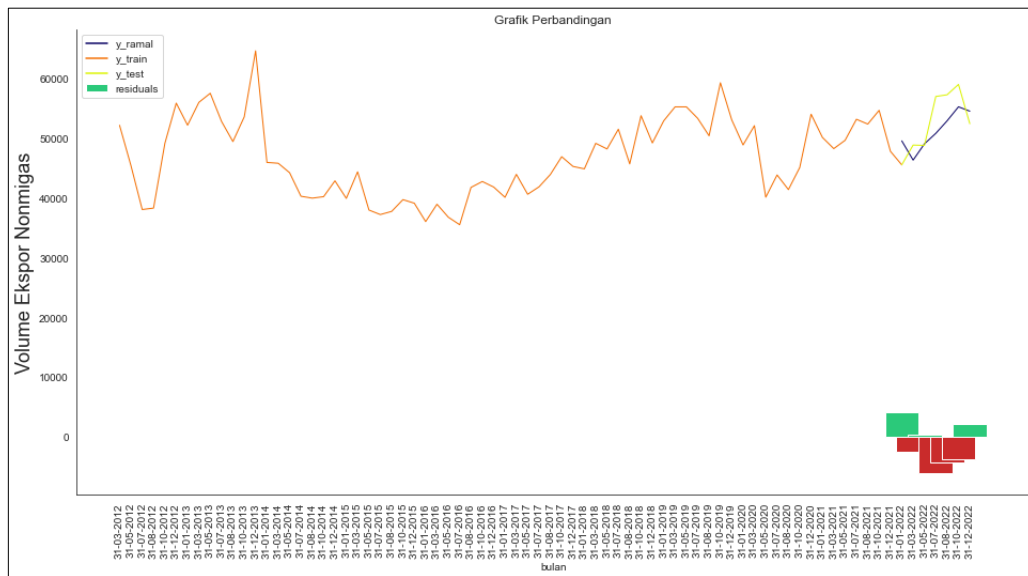
4.1.4.3 Pengujian Model 3



Gambar 4.4 Grafik Model 3 dengan Parameter *Batch size: 6, epochs: 100*

Pengujian model 3 menggunakan *learning rate* 0,001 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 6 dan *epoch* 100. Hasil pengujian digambarkan dalam diagram pada gambar 4.4. Hasil Pengujian Model 3 dalam bentuk tabel terdapat pada tabel 4.5.

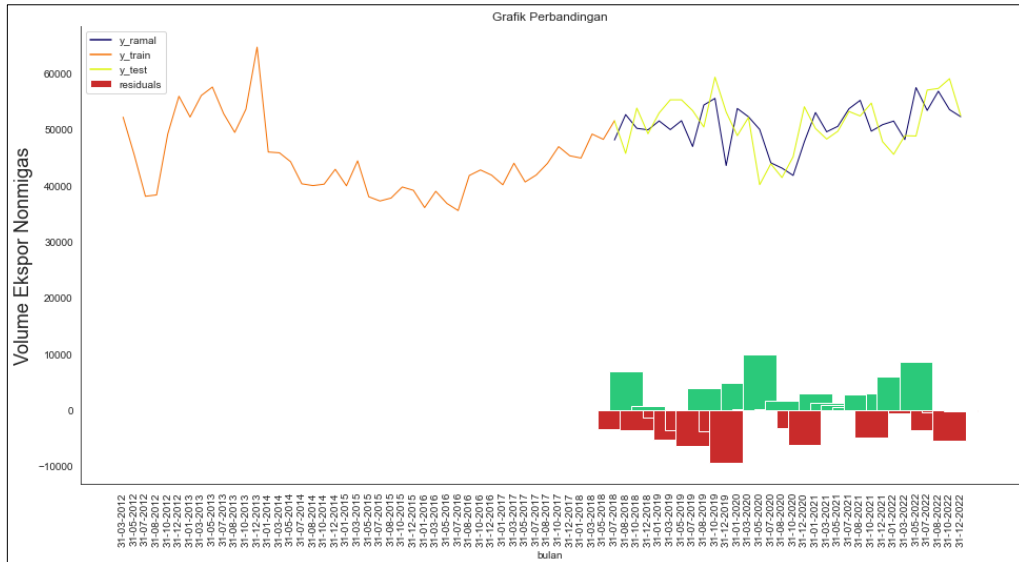
4.1.4.4 Pengujian Model 4



Gambar 4.5 Grafik Model 4 dengan Parameter *Batch size*: 3, *epochs*: 50

Pengujian model 4 menggunakan *learning rate* 0,001 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 3 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.5.

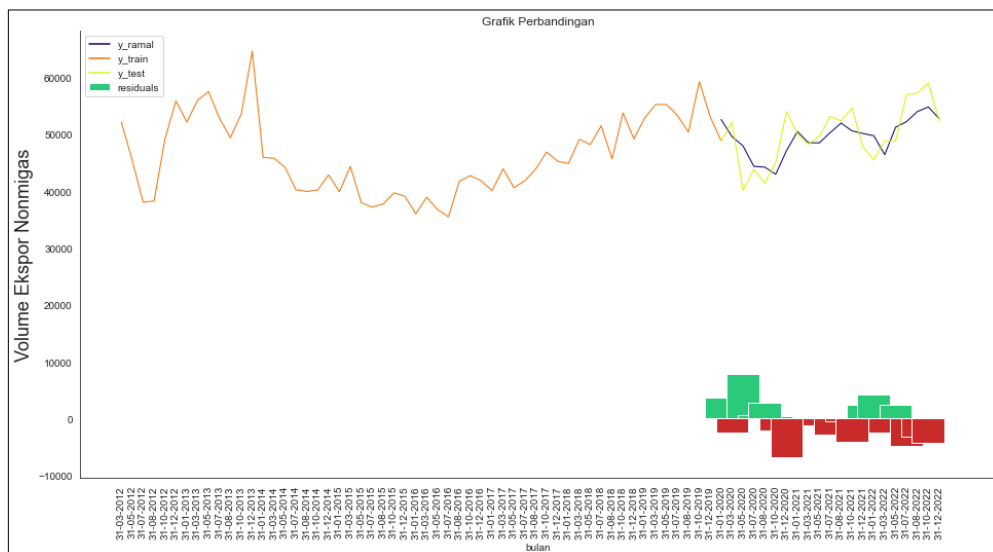
4.1.4.5 Pengujian Model 5



Gambar 4.6 Grafik Model 5 dengan Parameter *Batch size: 3, epochs: 50*

Pengujian model 5 menggunakan *learning rate* 0,01 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 3 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.6.

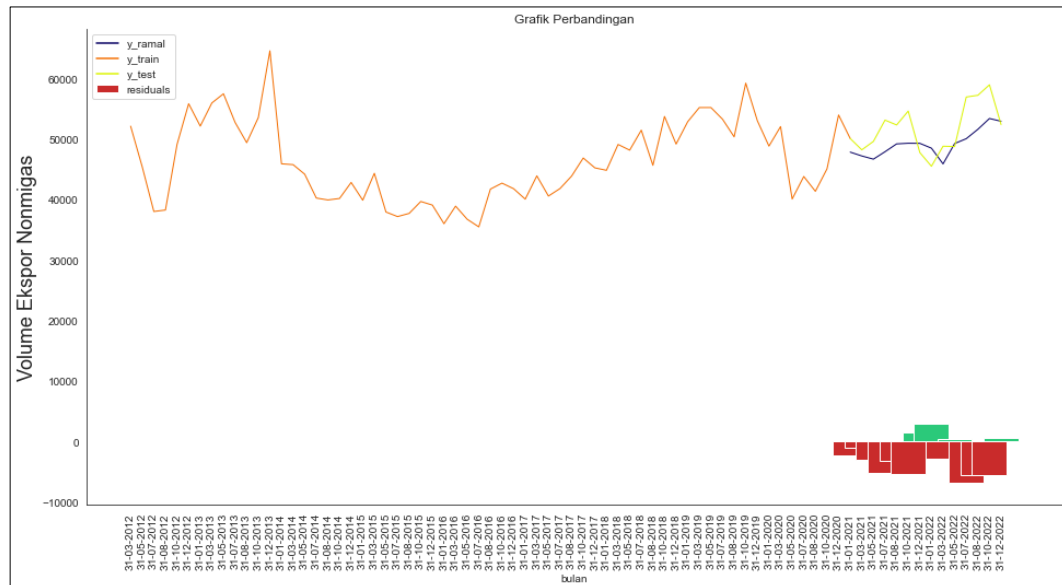
4.1.4.6 Pengujian Model 6



Gambar 4.7 Grafik Model 6 dengan Parameter *Batch size: 12, epochs: 50*

Pengujian model 6 menggunakan *learning rate* 0,01 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 12 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.7.

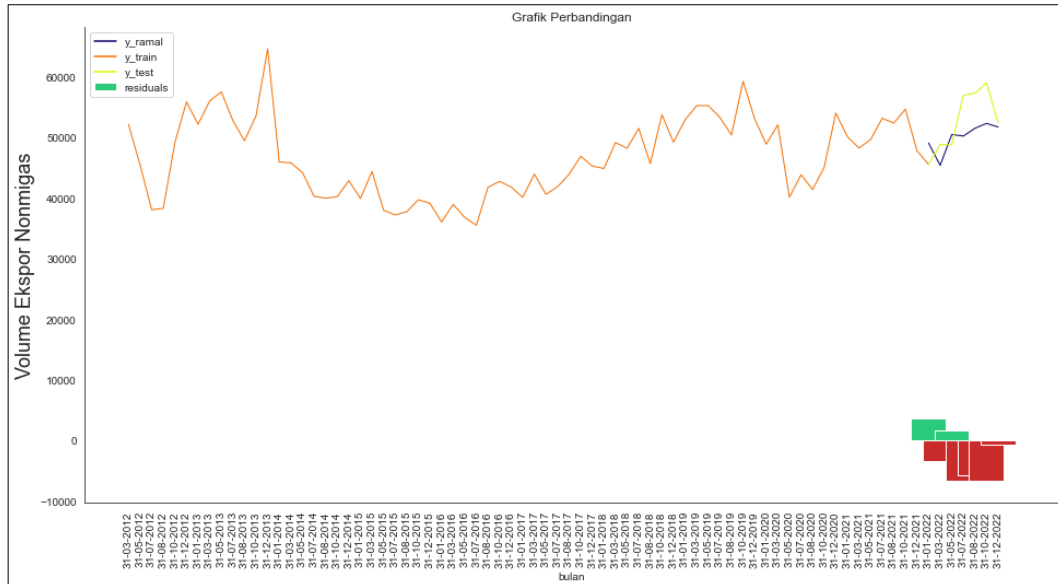
4.1.4.7 Pengujian Model 7



Gambar 4.8 Grafik Model 7 dengan Parameter *Batch size*: 12, *epochs*: 50

Pengujian model 7 menggunakan *learning rate* 0,01 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 12 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.8.

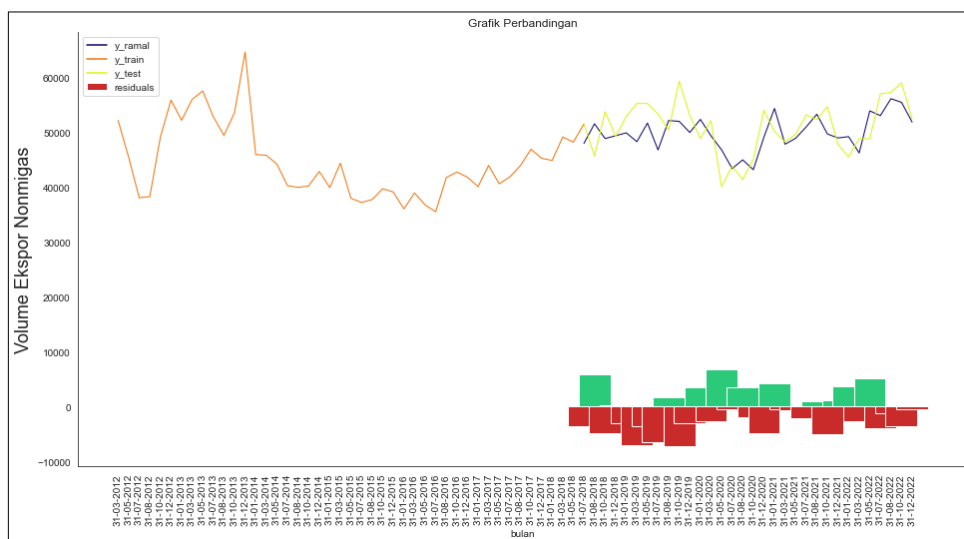
4.1.4.8 Pengujian Model 8



Gambar 4.9 Grafik Model 8 dengan Parameter *Batch size: 12, epochs: 50*

Pengujian model 8 menggunakan *learning rate* 0,01 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 12 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.9.

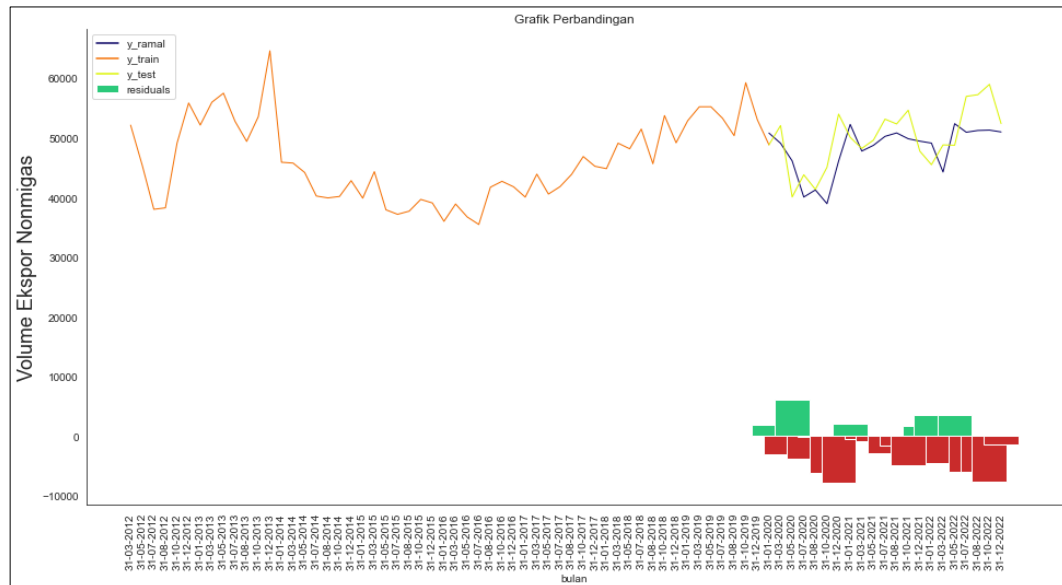
4.1.4.9 Pengujian Model 9



Gambar 4.10 Grafik Model 9 dengan Parameter *Batch size: 6, epochs: 50*

Pengujian model 9 menggunakan *learning rate* 0,1 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 6 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.10.

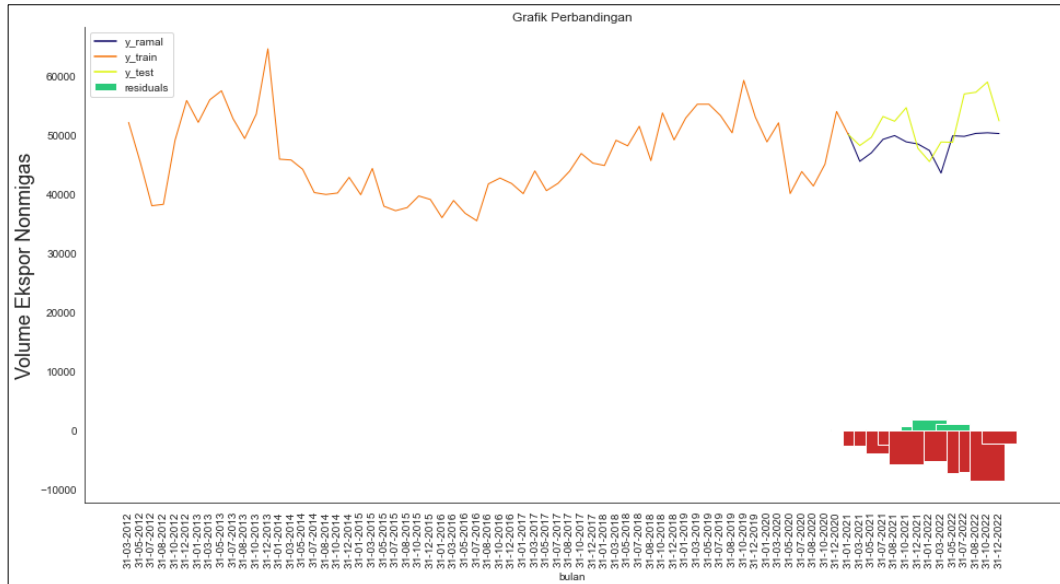
4.1.4.10 Pengujian Model 10



Gambar 4.11 Grafik Model 10 dengan Parameter *Batch size*: 3, *epochs*: 50

Pengujian model 10 menggunakan *learning rate* 0,1 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 3 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.11.

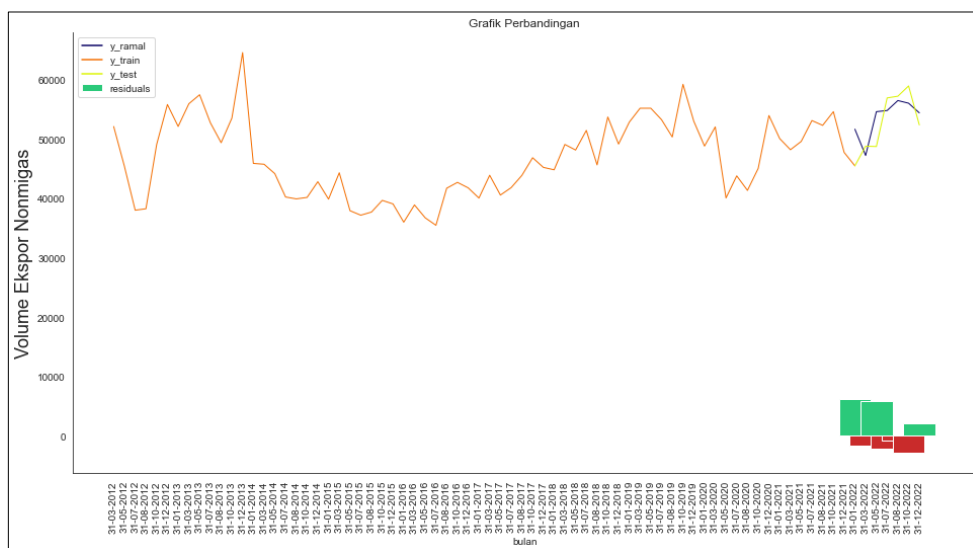
4.1.4.11 Pengujian Model 11



Gambar 4.12 Grafik Model 11 dengan Parameter *Batch size: 4, epochs: 50*

Pengujian model 11 menggunakan *learning rate* 0,1 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 4 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.12.

4.1.4.12 Pengujian Model 12



Gambar 4.13 Grafik Model 12 dengan Parameter *Batch size: 12, epochs: 50*

Pengujian model 12 menggunakan *learning rate* 0,1 parameter terbaik yang dihasilkan dari proses pelatihan yaitu *batch size* 12 dan *epoch* 50. Hasil pengujian digambarkan dalam diagram pada gambar 4.13.

4.1.5 Pengukuran tingkat *error* menggunakan MAPE

Berikut merupakan hasil pengukuran tingkat *error* menggunakan MAPE dari skenario model percobaan yang telah diuji coba :

Tabel 4.2 Pengukuran tingkat error menggunakan MAPE

Model Percobaan	Learning Rate	Parameter Terbaik	Nilai MAPE
Model 1	0,001	<i>Batch size: 2, epochs: 50</i>	9,25 %
Model 2	0,001	<i>Batch size: 3, epochs: 100</i>	9,89 %
Model 3	0,001	<i>Batch size: 6, epochs: 100</i>	7,87 %
Model 4	0,001	<i>Batch size: 3, epochs: 50</i>	9,86 %
Model 5	0,01	<i>Batch size: 3, epochs: 50</i>	10,36 %
Model 6	0,01	<i>Batch size: 12, epochs: 50</i>	9,84 %
Model 7	0,01	<i>Batch size: 12, epochs: 50</i>	7,90 %
Model 8	0,01	<i>Batch size: 12, epochs: 50</i>	10,01 %
Model 0	0,1	<i>Batch size: 6, epochs: 50</i>	9,34 %
Model 10	0,1	<i>Batch size: 3, epochs: 50</i>	10,53 %
Model 11	0,1	<i>Batch size: 4, epochs: 50</i>	8,00 %
Model 12	0,1	<i>Batch size: 12, epochs: 50</i>	9,50 %

Pengukuran menggunakan MAPE dengan tingkat *error* yang sangat rendah menunjukkan kemampuan peramalan sangat baik ketika nilainya berada dibawah 10%, pada sembilan model menunjukkan nilai dibawah 10%, kemampuan peramalan akan sangat lebih baik ketika nilai MAPE mendekati nol. Dalam penelitian ini

model percobaan 3 menunjukkan nilai paling rendah dan mendekati nol dengan begitu dapat disimpulkan percobaan dengan model 3 dengan parameter *batch size*: 6, *epochs*: 100, *learning rate* 0,001 lebih baik dari pada model percobaan yang lain.

Tabel 4.3 Data Hasil Peramalan dengan Kombinasi Parameter Terbaik

Bulan	Data Aktual	Data Peramalan	Residual
31/01/2021	50.092,20	49.663,41	428,79
28/02/2021	45.219,10	50.849,25	5630,15
31/03/2021	48.216,80	49.667,65	1450,85
30/04/2021	46.737,20	49.263,84	2526,64
31/05/2021	49.610,80	48.538,39	1072,41
30/06/2021	49.999,40	48.923,15	1076,26
31/07/2021	53.130,50	49.479,93	3650,57
31/08/2021	52.324,00	50.928,95	1395,06
30/09/2021	49.038,90	51.477,88	2438,98
31/10/2021	54.627,10	50.257,79	4369,31
30/11/2021	48.018,20	51.449,14	3430,94
31/12/2021	47.763,30	49.765,75	2002,45
31/01/2022	45.498,80	48.667,50	3168,70
28/02/2022	42.938,00	47.191,85	4253,85
31/03/2022	48.814,30	45.374,24	3440,06
30/04/2022	53.604,80	46.576,97	7027,83
31/05/2022	48.756,10	49.329,37	573,27
30/06/2022	53.310,00	49.375,12	3934,88
31/07/2022	56.945,00	50.884,33	6060,67
31/08/2022	57.233,30	53.177,00	4056,30
30/09/2022	58.938,30	54.402,41	4535,89
31/10/2022	58.966,10	55.173,05	3793,05
30/11/2022	55.363,90	55.277,82	86,08
31/12/2022	52.386,50	53.258,14	871,64
Rata-Rata Absolute Residual			2969,78

Tabel 4.3 menunjukkan hasil peramalan dengan kombinasi parameter terbaik yang membandingkan data aktual dan data hasil peramalan.

4.1.6 Peramalan Bulan Selanjutnya

Peramalan pada bulan selanjutnya dibuat dalam bentuk *Graphical User Interface* (GUI) menggunakan *library tkinter*. Pengguna dapat memasukan jumlah bulan ke depan yang ingin dilakukan peramalan kemudian akan menghasilkan volume hasil peramalan dalam satuan ribu ton.

Tabel 4.4 Peramalan Bulan Selanjutnya

Bulan	Volume Aktual	Volume Peramalan
31/01/2022	45.498,80	48.667,50
28/02/2022	42.938,00	47.191,85
31/03/2022	48.814,30	45.374,24
30/04/2022	53.604,80	46.576,97
31/05/2022	48.756,10	49.329,37
30/06/2022	53.310,00	49.375,12
31/07/2022	56.945,00	50.884,33
31/08/2022	57.233,30	53.177,00
30/09/2022	58.938,30	54.402,41
31/10/2022	58.966,10	55.173,05
30/11/2022	55.363,90	55.277,82
31/12/2022	52.386,50	53.258,14
31/01/2023		50.008,75

Hasil peramalan bulan berikutnya didapatkan dengan menggunakan model terbaik sebelumnya dengan menginisialisasi *forecast_steps* dengan jumlah langkah peramalan yang diinginkan. Dalam kasus ini, meramalkan 1 bulan ke depan, jadi *forecast_steps* = 1. Kemudian mengambil *seq_length* data terakhir dari data uji sebagai masukan awal untuk peramalan. Data ini akan digunakan sebagai input awal untuk peramalan, di mana model akan menggunakan data ini untuk meramal langkah-langkah berikutnya. Iterasi dilakukan sebanyak *forecast_steps*. Pada setiap iterasi, langkah-langkah berikutnya akan diramalkan berdasarkan data masukan saat ini.

4.2 Pembahasan

Perhitungan hasil nilai MAPE (*Mean Absolute Percentage Error*) dipengaruhi oleh perbandingan antara nilai aktual dan nilai peramalan. MAPE merupakan salah satu metrik evaluasi kinerja yang digunakan untuk mengukur tingkat kesalahan relatif dalam peramalan.

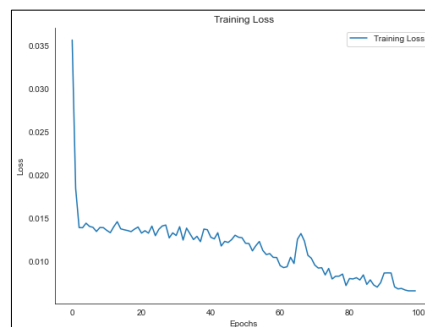
MAPE dihitung dengan mengambil selisih absolut antara nilai aktual dan nilai peramalan, kemudian membaginya dengan nilai aktual, dan mengalikannya dengan 100 untuk menghasilkan persentase.

Dari beberapa pengujian dengan bermacam kombinasi parameter yang di uji coba, dari gambar 4.4 menunjukkan model yang memiliki nilai residual yang merupakan selisih antara nilai aktual dan nilai peramalan tidak terlalu jauh dibandingkan dengan residual model lainnya. Hasil pengukuran tingkat *error* dari metode LSTM menggunakan *Mean Absolute Percentage Error* pada tabel 4.4 menunjukkan nilai MAPE terendah adalah 7,87 % dengan kombinasi parameter *batch size* 6 dan *epochs* 100 serta *learning rate* 0,001. Nilai tersebut mengindikasikan bahwa tingkat error pada model sangat rendah.

Batch size yang sangat kecil seperti ini dapat mengakibatkan variasi yang tinggi dalam update parameter dan membuat model lebih rentan terhadap pengaruh dari setiap sampel individu. Akibatnya, model dapat mengalami fluktuasi yang signifikan dalam pembelajaran dan kurang mampu menangkap pola dan tren yang konsisten dalam dataset. *batch size* yang lebih besar, model memproses lebih banyak sampel data pada setiap langkah pembelajaran. Ini dapat membantu dalam mengurangi fluktuasi dan meningkatkan kestabilan model selama pelatihan.

Namun, *batch size* yang terlalu besar juga dapat menyebabkan peningkatan konsumsi memori dan waktu pelatihan yang lebih lama (Brownlee, 2018). Dari hasil penelitian *batch size* 6 dianggap sebagai nilai yang optimal. *Batch size* ini memberikan keseimbangan yang baik antara variasi dalam pembelajaran model, kemampuan model dalam menangkap pola dan tren yang konsisten, serta efisiensi dan penggunaan memori yang cukup efektif.

Penggunaan nilai *epoch* yang terlalu kecil membuat model hanya dilatih dengan jumlah langkah yang terbatas. Hal ini dapat menyebabkan model tidak memiliki waktu yang cukup untuk menyesuaikan parameter dan mempelajari pola yang kompleks dalam dataset. Sebagai hasilnya, kinerja model kurang optimal dan tidak mampu menangkap variasi yang lebih halus dalam data. Di sisi lain, menggunakan jumlah *epoch* yang terlalu besar berarti melatih model dengan lebih banyak langkah. Hal ini memberikan model waktu yang lebih lama untuk menyesuaikan parameter dan mencari pola yang lebih kompleks dalam dataset. Namun, penggunaan jumlah *epoch* yang terlalu besar juga dapat mengakibatkan overfitting, di mana model terlalu terfokus pada data pelatihan dan tidak dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya (Brownlee, 2018).



Gambar 4.14 Grafik Training Loss Model Terbaik

Hasil eksperimen menunjukkan bahwa epoch 100 merupakan jumlah *epoch* yang optimal. Setelah mencapai jumlah *epoch* ini, peningkatan kinerja model tidak lagi signifikan. Model sudah mencapai tingkat konvergensi yang baik dan mencapai batas optimal dalam kemampuan peramalan.

Nilai *learning rate* yang terlalu tinggi dapat menyebabkan model Anda melompati titik optimal atau bahkan menyebabkan divergensi, sementara nilai *learning rate* yang terlalu rendah dapat mengakibatkan model bergerak terlalu lambat dan membutuhkan waktu yang lama untuk mencapai konvergensi (Brownlee, 2020). *Learning rate* 0,001 terpilih berdasarkan evaluasi skenario ujicoba. *Learning rate* ini memungkinkan model untuk mengoptimalkan parameter dengan tingkat pembelajaran yang tepat, tidak terlalu cepat sehingga melompati titik optimal, maupun terlalu lambat sehingga memperlambat konvergensi.

Pembagian data membantu dalam mengidentifikasi apakah model mengalami *overfitting*, yaitu ketika model terlalu terfokus pada data pelatihan dan tidak mampu menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Dengan memisahkan data uji yang independen, dapat dievaluasi sejauh mana model dapat melakukan peramalan yang baik pada data yang tidak terlibat dalam pelatihan. Hasil dari penelitian ini menunjukkan penggunaan pembagian data sebesar 80% data latih dan 20% data uji sebagai pembagian yang optimal. Hal ini memberikan keseimbangan antara melatih model dengan cukup banyak data dan menguji model dengan dataset yang independen untuk mendapatkan estimasi kinerja yang baik.

Pada penelitian sebelumnya meramal permintaan ekspor nonmigas di

provinsi Aceh dengan dataset bulanan dari tahun 2016 hingga 2017 menggunakan metode ARIMA dan Double Exponential. Pada penelitian tersebut menghasilkan nilai MAPE sebesar 37,98 dan 68,33 (Abdullah dkk., 2020). Hal ini menunjukkan bahwa penggunaan metode LSTM lebih baik dari pada metode pembandingan sebelumnya dalam meramal permintaan ekspor nonmigas.

Pengukuran tingkat *error* metode LSTM menunjukkan kemampuan model dalam melakukan peramalan pada kemungkinan banyaknya permintaan ekspor nonmigas yang akan terjadi di bulan selanjutnya. Nilai error yang rendah menunjukkan penggunaan metode *Long Short Term Memory* bisa untuk melakukan peramalan permintaan ekspor nonmigas. Hasil peramalan permintaan ekspor nonmigas dibulan selanjutnya dapat membantu pemerintah dalam merencanakan pengambilan keputusan produksi dan pemanfaatan sumber daya alam yang baik sehingga dapat meningkatkan perekonomian di Indonesia.

Meningkatkan ekonomi suatu negara dapat dipandang sebagai bentuk saling tolong menolong karena prosesnya melibatkan kerja sama antara berbagai pihak dalam masyarakat, termasuk pemerintah, sektor swasta, dan masyarakat umum. Hal tersebut sangat dianjurkan sebagaimana sabda Rosulullah dalam hadis yang diriwayatkan oleh Imam Muslim dalam kitab shahih muslim :

مَنْ نَفَسَ عَنْ مُؤْمِنٍ كُرْبَةً مِنْ كُرْبِ الدُّنْيَا نَفَسَ اللَّهُ عَنْهُ كُرْبَةً مِنْ كُرْبِ يَوْمِ الْقِيَامَةِ، وَمَنْ يَسَّرَ عَلَى مُعْسِرٍ يَسَّرَ اللَّهُ عَلَيْهِ فِي الدُّنْيَا وَالْآخِرَةِ، وَمَنْ سَتَرَ مُسْلِمًا سَتَرَهُ اللَّهُ فِي الدُّنْيَا وَالْآخِرَةِ، وَاللَّهُ فِي عَوْنِ الْعَبْدِ مَا كَانَ الْعَبْدُ فِي عَوْنِ أَخِيهِ

" *Barangsiapa yang meringankan kesedihan seorang mukmin, Allah akan meringankan kesedihannya di dunia dan di hari kiamat. Barangsiapa yang memudahkan keadaan bagi orang yang kesulitan, Allah akan memudahkan urusannya di dunia dan di akhirat. Barangsiapa yang menutupi aib seorang muslim, Allah akan menutupi aibnya di dunia dan di akhirat. Allah senantiasa*

membantu hamba-Nya selama hamba tersebut senantiasa membantu saudaranya." (HR. Muslim).

Dalam hadis tersebut menjelaskan agar saling membantu antar sesama manusia karena Allah senantiasa menolong hambanya yang saling tolong menolong saudaranya. Selain itu pemanfaatan sumber daya hasil bumi dijelaskan dalam Al Quran QS. Al Baqarah ayat 267 :

يَا أَيُّهَا الَّذِينَ ءَامَنُوا أَنْفِقُوا مِنْ طَيِّبَاتِ مَا كَسَبْتُمْ وَمِمَّا أَخْرَجْنَا لَكُمْ مِنَ الْأَرْضِ ۖ وَلَا تَيَمَّمُوا الْخَبِيثَ مِنْهُ تُنْفِقُونَ وَلَسْتُمْ بِآخِذِيهِ إِلَّا أَنْ تُغْمِضُوا فِيهِ ۚ وَاعْلَمُوا أَنَّ اللَّهَ عَزِيزٌ حَمِيدٌ

"Hai orang-orang yang beriman, infakkanlah (harta) yang baik-baik dari apa yang kamu peroleh dan dari apa yang Kami keluarkan untuk kamu dari bumi. Dan janganlah kamu memilih yang buruk-buruk dari padanya untuk kamu infakkan, padahal kamu sendiri tidak akan mau mengambilnya melainkan dengan memejamkan mata terhadapnya. Dan ketahuilah, bahwa Allah Maha Kaya lagi Maha Terpuji."

Dalam tafsir al azhar ayat tersebut menjelaskan bahwa orang beriman suka melakukan usaha atau bisnis. usaha yang halal tersebut diantar lain bertani atau berladang. Maka hasil dari usaha halal tersebut (pertanian) dibelanjakan atau diinfakkan (Amrullah, 1990). Dalam konteks ini, dapat diinterpretasikan sebagai pentingnya pengelolaan sumber daya alam yang baik dan adil, serta berkontribusi pada kesejahteraan umat manusia secara umum.

Penelitian ini diharapkan merencanakan kebutuhan pasar dan mengatur produksi untuk mencapai keseimbangan antara penawaran dan permintaan sehingga Indonesia meningkatkan ekspor nonmigas pada perdagangan internasional.

Perdagangan internasional juga memainkan peran penting dalam

perekonomian setiap negara di dunia. Di sisi lain, perdagangan internasional tidak hanya dapat meningkatkan devisa negara, atau meningkatkan hubungan ekonomi yang baik antara negara pengimpor dan pengeksport tapi juga meningkatkan kemakmuran dan kesejahteraan rakyat yang hal tersebut merupakan salah satu bentuk saling tolong menolong antar sesama manusia.

BAB V

PENUTUP

Pada bab v penutup berisi kesimpulan hasil penelitian dari penggunaan metode *long short term memory* dalam meramal permintaan ekspor nonmigas Indonesia dan saran bagi peneliti selanjutnya dengan harapan penelitian ini dapat dikembangkan sehingga menghasilkan nilai error yang lebih rendah.

5.1 Kesimpulan

Pad bab sebelumnya dijelaskan bahwa penelitian ini bertujuan untuk mengukur tingkat *error Long Short Term Memory (LSTM)* dalam meramal permintaan ekspor nonmigas Indonesia. Perhitungan *error* menggunakan *Mean Absolute Percentage Error* menghasilkan nilai terendah 7,87% dengan skenario pembagian data 80% data latih dan 20% data uji dengan kombinasi parameter *learning rate* 0,001, *batch size* 6 dan *epoch* 100. Berdasarkan tabel 2.1 kriteria nilai *MAPE*, nilai tersebut tergolong memiliki tingkat *error* yang sangat rendah sehingga metode *Long Short Term Memory* memiliki kemampuan peramalan yang sangat baik untuk melakukan peramalan permintaan ekspor nonmigas.

5.2 Saran

Penelitian ini diharapkan bisa menjadi referensi untuk pengembangan metode *Long Short Term Memory* yang diimplementasikan untuk meramal permintaan ekspor nonmigas Indonesia pada penelitian ini untuk penelitian selanjutnya. Penelitian ini masih memiliki kekurangan yang perlu dievaluasi oleh

karena itu terdapat beberapa saran agar penelitian sejenis selanjutnya dapat lebih baik sebagai berikut :

1. Melakukan penyesuaian skenario pembagian data dan penentuan nilai parameter *batch size*, *epoch*, jumlah *hidden layer* ataupun menambahkan *dropout* pada *hidden layer* agar mencapai nilai error yang mendekati nol.
2. Menambahkan metode optimasi seperti *Particle Swarm Optimization (PSO)*, *Fly Fruit Optimization (FOA)* atau yang lainnya.

DAFTAR PUSTAKA

- Abdul Dwiyanto Suyudi, M., Djamal, E. C., Maspupah Jurusan Informatika, A., & Sains dan Informatika Universitas Jenderal Achmad Yani Cimahi, F. (2019). Prediksi Harga Saham menggunakan Metode Recurrent Neural Network. *Seminar Nasional Aplikasi Teknologi Informasi (SNATi)*, 1907–5022.
- Abdullah, R., Hasyim, S., & Husaini, H. (2020). *Modeling Of Non-Oil And Gas Export Forecasting Aceh Province Using Arima And Double Exponential Methods*. <https://doi.org/10.4108/eai.17-10-2018.2294214>
- Aldi, M. W. P., Jondri, & Aditsania, A. (2018). Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. *Jurnal Informatika*, 5, No(2), 3548. <http://openlibrarypublications.telkomniversity.ac.id>
- Amrullah, A. A. (1990). Tafsir Al-Azhar Jilid 1. *Pustaka Nasional PTE LTD Singapura*, 1–710.
- Armstrong, J. S. (2011). Methods to Elicit Forecasts from Groups: Delphi and Prediction Markets Compared. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.1153124>
- Badan Pusat Statistik. (2022). *BULETIN STATISTIK PERDAGANGAN LUAR NEGERI EKSPOR MENURUT KELOMPOK KOMODITI DAN NEGARA AGUSTUS 2022*.
- Barus, J., & Ramli, R. (2013). Analisis Peramalan Ekspor Indonesia Pasca Krisis Keuangan Eropa Dan Global Tahun 2008 Dengan Metode Dekomposisi. *Jurnal Ekonomi dan Keuangan*, 1(3), 14880.
- Brownlee, J. (2018, July 20). Difference Between a Batch and an Epoch in a Neural Network. <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/#:~:text=The%20batch%20size%20is%20a%20number%20of%20samples%20processed%20before,samples%20in%20the%20training%20dataset>.
- Brownlee, J. (2020, September 12). Understand the Impact of Learning Rate on Neural Network Performance. <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>
- Chamsukhee, V. (2020). Time Series Forecasting of E-Databases Subscription Mahidol University Library with Exponential Smoothing, LSTM, and ARIMA Models. *InCIT 2020 - 5th International Conference on Information*

Technology, 202–207. <https://doi.org/10.1109/InCIT50588.2020.9310785>

- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7, 1–24. <https://doi.org/10.7717/PEERJ-CS.623>
- Do, Q. H., Lo, S. K., Chen, J. F., Le, C. L., & Anh, L. H. (2020). Forecasting air passenger demand: A comparison of LSTM and SARIMA. *Journal of Computer Science*, 16(7), 1063–1084. <https://doi.org/10.3844/JCSSP.2020.1063.1084>
- Hammad, M., Shoaib, M., Salahudin, H., Baig, M. A. I., Khan, M. M., & Ullah, M. K. (2021). Rainfall forecasting in upper Indus basin using various artificial intelligence techniques. *Stochastic Environmental Research and Risk Assessment*, 35(11), 2213–2235. <https://doi.org/10.1007/s00477-021-02013-0>
- Heizer, J., Barry, R., & Chuck, M. (2017). *Operations Management: Sustainability and Supply Chain Management*. In Edinburgh: Pearson Education Limited.
- Hidayah, N. E. (2017). Analisis Ekspor Kopi Indonesia ke Amerika Serikat dengan Error Correction Model (ECM) Tahun 1990-2014. *Jurnal Universitas Islam Indonesia, Fakultas Ekonomi, Yogyakarta*, 1–10.
- Hsieh, S. C. (2021). Tourism demand forecasting based on an lstm network and its variants. *Algorithms*, 14(8). <https://doi.org/10.3390/a14080243>
- Huda, S. (2006). Analisis Beberapa Faktor Yang Mempengaruhi Ekspor Non Migas Indonesia Ke Jepang. *Jurnal Ilmu-Ilmu Ekonomi*, 6, 7.
- Ilah, M. (2016). *PADA KELOMPOK KOMODITI EKSPOR UDANG SEGAR / BEKU DAN TONGKOL / TUNA DENGAN METODE ARIMA BOX-JENKINS PADA KELOMPOK KOMODITI EKSPOR UDANG SEGAR / BEKU DAN TONKOL / TUNA DENGAN METODE ARIMA BOX-JENKINS*.
- Katsir, I. (2014). *Tafsir Ibnu Katsir Juz 4*. 1–570.
- Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of Long Short-Term Memory (LSTM) neural network for flood forecasting. *Water (Switzerland)*, 11(7). <https://doi.org/10.3390/w11071387>
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Munarsih, E. (2017). Penerapan Metode Penghalusan Eksponensial Untuk Meramal Ekspor Kopi Indonesia. *Kntia*, 1–4. <http://seminar.ilkom.unsri.ac.id/index.php/kntia/article/download/1140/511>
- Nielsen, M. (2015). Neural Networks and Deep Learning. *Neural Networks and Deep Learning: A Textbook*. Determination Press., 83.

- Nurlaily, D., Silfiani, M., & Hayati, N. F. (2022). Perbandingan Metode Arima, Dan Triple Exponential Smoothing Pada Studi Kasus Data Ekspor Non Migas Di Kalimantan Timur. *Jurnal SNATI*, 1.
- Olah, C. (2015, August 27). Understanding LSTM Networks. Understanding LSTM Networks. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Olah, C. (2015, August 27). Understanding LSTM Networks. Understanding LSTM Networks. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Ozbek, A., Sekertekin, A., Bilgili, M., & Arslan, N. (2021). Prediction of 10-min, hourly, and daily atmospheric air temperature: comparison of LSTM, ANFIS-FCM, and ARMA. *Arabian Journal of Geosciences*, 14(7). <https://doi.org/10.1007/s12517-021-06982-y>
- Sihombing, M., Sihotang, J., & Purba, M. L. (2021). Analisis Pengaruh Ekspor Migas, Ekspor Non Migas Dan Penanaman Modal Asing Terhadap Pertumbuhan Ekonomi Indonesia Tahun 2000-2019. *Journal of Economics and Business*, 02(02), 40–51.
- Sihotang, J. (2013). Analisis Faktor-Faktor yang Memengaruhi Ekspor Kopi Indonesia di Pasar Internasional. *Jurnal Ekonomi dan Bisnis Nommensen*, 4, 9–18.
- Son, H., & Kim, C. (2020). A deep learning approach to forecasting monthly demand for residential-sector electricity. *Sustainability (Switzerland)*, 12(8), 3103. <https://doi.org/10.3390/SU12083103>
- Sugiyarto, A. W., & Abadi, A. M. (2019). Prediction of Indonesian palm oil production using long short-term memory recurrent neural network (LSTM-RNN). *Proceedings - 2019 1st International Conference on Artificial Intelligence and Data Sciences, AiDAS 2019*, 53–57. <https://doi.org/10.1109/AiDAS47888.2019.8970735>
- Wiranda, L., & Sadikin, M. (2019). Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk Pt. Metiska Farma. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 8(3), 184–196.
- Zulfa, I. I., Novitasari, D. C. R., Setiawan, F., Fanani, A., & Hafiyusholeh, M. (2021). Prediction of Sea Surface Current Velocity and Direction Using LSTM. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 11(1), 93. <https://doi.org/10.22146/ijeis.63669>

LAMPIRAN

LAMPIRAN

Dataset volume ekspor nonmigas 20122022

NO	bulan	volume
1	31/01/2012	41.923,5
2	29/02/2012	42.572,5
3	31/03/2012	52.125,5
4	30/04/2012	52.601,8
5	31/05/2012	45.405,3
6	30/06/2012	38.525,9
7	31/07/2012	38.023,5
8	31/08/2012	38.265,9
9	30/09/2012	41.472,5
10	31/10/2012	49.114,2
11	30/11/2012	55.823,7
12	31/12/2012	55.836,4
13	31/01/2013	52.137,7
14	28/02/2013	50.427,6
15	31/03/2013	55.966,6
16	30/04/2013	55.523,8
17	31/05/2013	57.483,3
18	30/06/2013	50.333,9
19	31/07/2013	52.753,1
20	31/08/2013	49.401,3
21	30/09/2013	52.469,4
22	31/10/2013	53.538,3
23	30/11/2013	61.343,2
24	31/12/2013	64.584,9
25	31/01/2014	45.906,9
26	28/02/2014	39.950,9
27	31/03/2014	45.767,8
28	30/04/2014	42.085,5
29	31/05/2014	44.185,6
30	30/06/2014	41.238,6
31	31/07/2014	40.247,4
32	31/08/2014	39.931,0
33	30/09/2014	42.346,0
34	31/10/2014	40.177,4
35	30/11/2014	43.056,9
36	31/12/2014	42.828,2
37	31/01/2015	39.889,4
38	28/02/2015	36.401,5

39	31/03/2015	44.335,5
40	30/04/2015	40.371,0
41	31/05/2015	37.937,9
42	30/06/2015	36.943,1
43	31/07/2015	37.182,8
44	31/08/2015	37.709,9
45	30/09/2015	37.124,8
46	31/10/2015	39.679,9
47	30/11/2015	37.204,1
48	31/12/2015	39.082,6
49	31/01/2016	36.004,2
50	29/02/2016	35.022,1
51	31/03/2016	38.909,3
52	30/04/2016	35.962,2
53	31/05/2016	36.740,8
54	30/06/2016	40.464,0
55	31/07/2016	35.480,2
56	31/08/2016	41.726,6
57	30/09/2016	40.451,8
58	31/10/2016	42.713,2
59	30/11/2016	43.136,8
60	31/12/2016	41.788,0
61	31/01/2017	40.066,9
62	28/02/2017	34.138,2
63	31/03/2017	43.912,8
64	30/04/2017	41.613,8
65	31/05/2017	40.576,0
66	30/06/2017	36.478,9
67	31/07/2017	41.822,8
68	31/08/2017	43.887,0
69	30/09/2017	44.274,1
70	31/10/2017	46.865,8
71	30/11/2017	44.468,4
72	31/12/2017	45.236,9
73	31/01/2018	44.833,3
74	28/02/2018	42.574,8
75	31/03/2018	49.106,0
76	30/04/2018	44.709,7
77	31/05/2018	48.163,3
78	30/06/2018	46.063,1
79	31/07/2018	51.469,9
80	31/08/2018	45.659,4
81	30/09/2018	47.859,6
82	31/10/2018	53.732,1

83	30/11/2018	48.510,9
84	31/12/2018	49.169,9
85	31/01/2019	52.869,6
86	28/02/2019	46.576,1
87	31/03/2019	55.204,9
88	30/04/2019	50.922,9
89	31/05/2019	55.207,7
90	30/06/2019	44.884,4
91	31/07/2019	53.279,0
92	31/08/2019	50.361,9
93	30/09/2019	52.709,9
94	31/10/2019	59.240,0
95	30/11/2019	53.659,4
96	31/12/2019	53.030,3
97	31/01/2020	48.830,4
98	29/02/2020	47.504,3
99	31/03/2020	52.061,4
100	30/04/2020	43.242,9
101	31/05/2020	40.082,7
102	30/06/2020	44.098,9
103	31/07/2020	43.809,3
104	31/08/2020	41.360,3
105	30/09/2020	41.454,9
106	31/10/2020	45.063,5
107	30/11/2020	50.692,3
108	31/12/2020	53.979,4
109	31/01/2021	50.092,2
110	28/02/2021	45.219,1
111	31/03/2021	48.216,8
112	30/04/2021	46.737,2
113	31/05/2021	49.610,8
114	30/06/2021	49.999,4
115	31/07/2021	53.130,5
116	31/08/2021	52.324,0
117	30/09/2021	49.038,9
118	31/10/2021	54.627,1
119	30/11/2021	48.018,2
120	31/12/2021	47.763,3
121	31/01/2022	45.498,8
122	28/02/2022	42.938,0
123	31/03/2022	58.814,3
124	30/04/2022	53.604,8
125	31/05/2022	48.756,1
126	30/06/2022	53.310,0

127	31/07/2022	56.945,0
128	31/08/2022	57.233,3
129	30/09/2022	58.938,3
130	31/10/2022	58.966,1
131	30/11/2022	55.363,9
132	31/12/2022	52.386,5

Dataset Volume Ekspor Nonmigas Hasil Normalisasi

bulan	volume	normalisasi
31/01/2012	41.923,5	0,25570259
29/02/2012	42.572,5	0,27701853
31/03/2012	52.125,5	0,59077995
30/04/2012	52.601,8	0,60642368
31/05/2012	45.405,3	0,37005981
30/06/2012	38.525,9	0,14411086
31/07/2012	38.023,5	0,12760989
31/08/2012	38.265,9	0,13557134
30/09/2012	41.472,5	0,24088982
31/10/2012	49.114,2	0,49187597
30/11/2012	55.823,7	0,71224468
31/12/2012	55.836,4	0,71266180
31/01/2013	52.137,7	0,59118065
28/02/2013	50.427,6	0,53501365
31/03/2013	55.966,6	0,71693812
30/04/2013	55.523,8	0,70239468
31/05/2013	57.483,3	0,76675305
30/06/2013	50.333,9	0,53193614
31/07/2013	52.753,1	0,61139302
31/08/2013	49.401,3	0,50130556
30/09/2013	52.469,4	0,60207510
31/10/2013	53.538,3	0,63718235
30/11/2013	61.343,2	0,89352869
31/12/2013	64.584,9	1
31/01/2014	45.906,9	0,38653450
28/02/2014	39.950,9	0,19091396
31/03/2014	45.767,8	0,38196586
30/04/2014	42.085,5	0,26102336
31/05/2014	44.185,6	0,32999964
30/06/2014	41.238,6	0,23320754
31/07/2014	40.247,4	0,20065229
31/08/2014	39.931,0	0,19026036
30/09/2014	42.346,0	0,26957930
31/10/2014	40.177,4	0,19835319
30/11/2014	43.056,9	0,29292830
31/12/2014	42.828,2	0,28541681
31/01/2015	39.889,4	0,18889403
28/02/2015	36.401,5	0,07433646

31/03/2015	44.335,5	0,33492300
30/04/2015	40.371,0	0,20471184
31/05/2015	37.937,9	0,12479842
30/06/2015	36.943,1	0,09212493
31/07/2015	37.182,8	0,09999770
31/08/2015	37.709,9	0,11730992
30/09/2015	37.124,8	0,09809273
31/10/2015	39.679,9	0,18201316
30/11/2015	37.204,1	0,10069728
31/12/2015	39.082,6	0,16239527
31/01/2016	36.004,2	0,06128743
29/02/2016	35.022,1	0,02903106
31/03/2016	38.909,3	0,15670335
30/04/2016	35.962,2	0,05990797
31/05/2016	36.740,8	0,08548053
30/06/2016	40.464,0	0,20776636
31/07/2016	35.480,2	0,04407703
31/08/2016	41.726,6	0,24923555
30/09/2016	40.451,8	0,20736566
31/10/2016	42.713,2	0,28163972
30/11/2016	43.136,8	0,29555256
31/12/2016	41.788,0	0,25125219
31/01/2017	40.066,9	0,19472389
28/02/2017	34.138,2	0
31/03/2017	43.912,8	0,32103972
30/04/2017	41.613,8	0,24553071
31/05/2017	40.576,0	0,21144492
30/06/2017	36.478,9	0,07687861
31/07/2017	41.822,8	0,25239517
31/08/2017	43.887,0	0,32019234
30/09/2017	44.274,1	0,33290636
31/10/2017	46.865,8	0,41802888
30/11/2017	44.468,4	0,33928800
31/12/2017	45.236,9	0,36452883
31/01/2018	44.833,3	0,35127288
28/02/2018	42.574,8	0,27709407
31/03/2018	49.106,0	0,49160664
30/04/2018	44.709,7	0,34721333
31/05/2018	48.163,3	0,46064434
30/06/2018	46.063,1	0,39166478
31/07/2018	51.469,9	0,56924724

31/08/2018	45.659,4	0,37840554
30/09/2018	47.859,6	0,45066953
31/10/2018	53.732,1	0,64354758
30/11/2018	48.510,9	0,47206101
31/12/2018	49.169,9	0,49370539
31/01/2019	52.869,6	0,61521938
28/02/2019	46.576,1	0,40851389
31/03/2019	55.204,9	0,69192064
30/04/2019	50.922,9	0,55128142
31/05/2019	55.207,7	0,69201260
30/06/2019	44.884,4	0,35295122
31/07/2019	53.279,0	0,62866583
31/08/2019	50.361,9	0,53285578
30/09/2019	52.709,9	0,60997415
31/10/2019	59.240,0	0,82445060
30/11/2019	53.659,4	0,64115980
31/12/2019	53.030,3	0,62049746
31/01/2020	48.830,4	0,48255476
29/02/2020	47.504,3	0,43899996
31/03/2020	52.061,4	0,58867463
30/04/2020	43.242,9	0,29903733
31/05/2020	40.082,7	0,19524283
30/06/2020	44.098,9	0,32715204
31/07/2020	43.809,3	0,31764034
31/08/2020	41.360,3	0,23720469
30/09/2020	41.454,9	0,24031176
31/10/2020	45.063,5	0,35883363
30/11/2020	50.692,3	0,54370753
31/12/2020	53.979,4	0,65166997
31/01/2021	50.092,2	0,52399767
28/02/2021	45.219,1	0,36394420
31/03/2021	48.216,8	0,46240151
30/04/2021	46.737,2	0,41380511
31/05/2021	49.610,8	0,50818644
30/06/2021	49.999,4	0,52094973
31/07/2021	53.130,5	0,62378846
31/08/2021	52.324,0	0,59729954
30/09/2021	49.038,9	0,48940279
31/10/2021	54.627,1	0,67294321
30/11/2021	48.018,2	0,45587863
31/12/2021	47.763,3	0,44750663

31/01/2022	45.498,8	0,37313075
28/02/2022	42.938,0	0,28902311
31/03/2022	58.814,3	0,81046879
30/04/2022	53.604,8	0,63936650
31/05/2022	48.756,1	0,48011443
30/06/2022	53.310,0	0,62968401
31/07/2022	56.945,0	0,74907297
31/08/2022	57.233,3	0,75854198
30/09/2022	58.938,3	0,81454148
31/10/2022	58.966,1	0,81545455
30/11/2022	55.363,9	0,69714288
31/12/2022	52.386,5	0,59935231