

**TESIS**

**IMPLEMENTASI REGRESI PADA PREDIKSI JUMLAH PENDAFTAR  
MAHASISWA BARU  
(Studi Kasus : IAIN Madura)**



Disusun oleh:

**Nama : Atin Hasanah**  
**NIM : 21.55.2153**  
**Konsentrasi : Business Intelligence**

**PROGRAM STUDI S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA  
2023**

**TESIS**

**IMPLEMENTASI REGRESI PADA PREDIKSI JUMLAH PENDAFTAR  
MAHASISWA BARU  
(Studi Kasus : IAIN Madura)**

**THE IMPLEMENTATION OF REGRESSION TO FORECAST THE  
NUMBERS OF THE REGISTERED NEW STUDENTS  
(Case Study : IAIN Madura)**

Diajukan untuk memenuhi salah satu syarat memperoleh derajat Magister



Disusun oleh:

**Nama : Atfn Hasanah**  
**NIM : 21.55.2153**  
**Konsentrasi : Business Intelligence**

**PROGRAM STUDI S2 TEKNIK INFORMATIKA  
PROGRAM PASCASARJANA UNIVERSITAS AMIKOM YOGYAKARTA  
YOGYAKARTA  
2023**

**HALAMAN PENGESAHAN**

**IMPLEMENTASI REGRESI PADA PREDIKSI JUMLAH PENDAFTAR  
MAHASISWA BARU  
(Studi Kasus : IAIN Madura)**

**THE IMPLEMENTATION OF REGRESSION TO FORECAST THE NUMBERS OF  
THE REGISTERED NEW STUDENTS  
(Case Study : IAIN Madura)**

Diperstapkan dan Disusun oleh

**Atlu Hasanah  
21.55.2153**

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Rabu, 01 November 2023

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 01 November 2023  
**Rektor**

**Prof. Dr. M. Suyanto, M.M.**  
**NIK. 190302001**

## HALAMAN PERSETUJUAN

IMPLEMENTASI REGRESI PADA PREDIKSI JUMLAH PENDAFTAR

MAHASISWA BARU

(Studi Kasus : IAIN Madura)

THE IMPLEMENTATION OF REGRESSION TO FORECAST THE NUMBERS OF  
THE REGISTERED NEW STUDENTS

(Case Study : IAIN Madura)

Dipersiapkan dan Disusun oleh

Atin Hasanah

21.55.2153

Telah Diujikan dan Dipertahankan dalam Sidang Ujian Tesis  
Program Studi S2 Teknik Informatika  
Program Pascasarjana Universitas AMIKOM Yogyakarta  
pada hari Rabu, 01 November 2023

Pembimbing Utama

Anggota Tim Penguji

Prof. Dr. Kusriani, M.Kom.  
NIK. 190302106

Dhani Ariatmanto, M.Kom., Ph.D.  
NIK. 190302197

Pembimbing Pendamping

Dr. Ferry Wahyu Wibowo, S.Si., M.Cs.  
NIK. 190302235

Kusnawi, S.Kom., M.Eng.  
NIK. 190302112

Prof. Dr. Kusriani, M.Kom.  
NIK. 190302106

Tesis ini telah diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Magister Komputer

Yogyakarta, 01 November 2023  
Direktur Program Pascasarjana

Prof. Dr. Kusriani, M.Kom.  
NIK. 190302106

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

Nama mahasiswa : Atin Hasanah  
NIM : 21.55.2153  
Konsentrasi : Business Intelligence

Menyatakan bahwa Tesis dengan judul berikut:  
**Implementasi Regresi Pada Prediksi Jumlah Pendaftar Mahasiswa Baru  
(Studi Kasus : IAIN Madura)**

Dosen Pembimbing Utama : Prof. Dr. Kusriani, M.Kom.  
Dosen Pembimbing Pendamping : Kusnawi, S.Kom., M.Eng.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Tim Dosen Pembimbing
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi

Yogyakarta, 01 November 2023

Yang Menyatakan,



Atin Hasanah

## HALAMAN PERSEMBAHAN

Alhamdulillah, penulis panjatkan puji dan syukur kehadirat Allah SWT karena berkat Rahmat dan Hidayah-Nya penulis ingin mempersembahkan haturan terimakasih yang sangat khusus dan permintaan maaf:

1. Bapak saya yang tercinta, terrindu dan terkasih (Alm) Mohammad Rasjid Sukram mohon maaf karena baru sekarang saya melanjutkan studi ini. Tetap doakan kami dari sisiNya.
2. Ibu saya tercinta Masrurah, terimakasih sudah memberikan dukungan maupun doa demi kelancaran studi.
3. Adik saya tersayang Utari Sanaba, yang sama-sama berjuang pada studi lanjutan ke Magister tetap semangat dan terimakasih sudah memberikan dukungan semangat.

Mari kita tetap menjadi pribadi yang kuat dan saling menguatkan karena harus menjadi wonder woman dalam segala hal urusan.

Yogyakarta, 01 November 2023

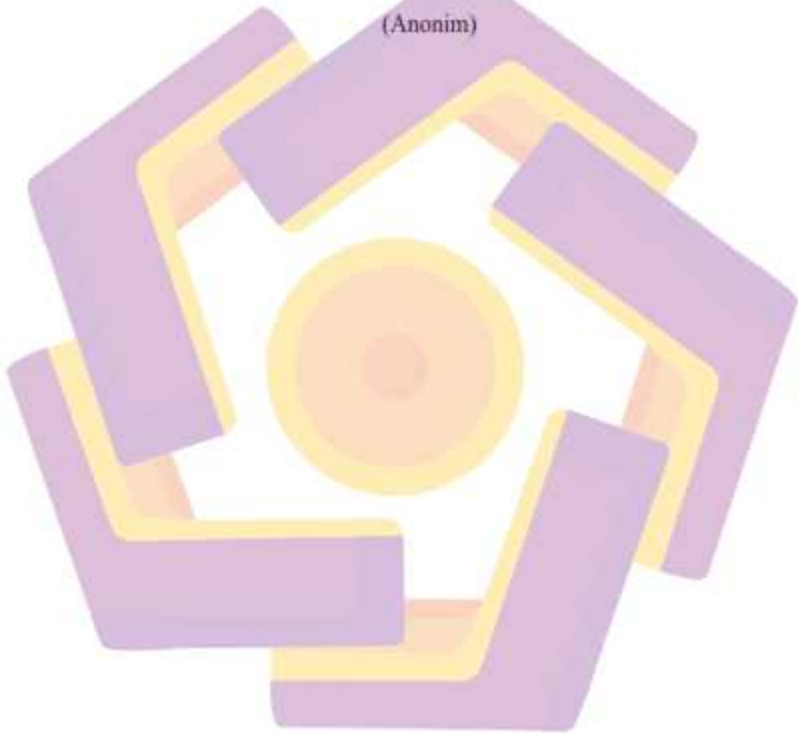
Penulis

## HALAMAN MOTTO

“Mumpung masih ada, maka berbaiklah dan saling minta maaf. Jangan sampai saling acuh atau menundanya, sehingga ketika sudah tidak ada baru akan

menyesal”

(Anonim)



## KATA PENGANTAR

Alhamdulillah, puji dan syukur penulis panjatkan kehadirat Allah SWT karena berkat Rahmat dan Hidayah-Nya penulis dapat menyelesaikan Tesis yang berjudul **“Implementasi Regresi Pada Prediksi Jumlah Pendaftar Mahasiswa Baru (Studi Kasus : IAIN Madura)”**. Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada pihak-pihak yang telah memberi izin, membantu serta membimbing dalam penyusunan laporan Tesis:

1. Bapak Prof. Dr. M. Suyanto, M.M., selaku Rektor Universitas Amikom Yogyakarta yang telah mendukung adanya Program Studi Pendidikan Jarak Jauh (PJJ) Magister Teknik Informatika.
2. Ibu Prof. Dr. Kusriani, M.Kom., selaku Direktur Program Pasca Sarjana Universitas Amikom Yogyakarta yang telah mengizinkan kami untuk menempuh studi lanjut di Program Studi Pendidikan Jarak Jauh (PJJ) Magister Teknik Informatika. Serta menjadi pembimbing utama yang di sela-sela kesibukan beliau tetap menyempatkan untuk membimbing, mengoreksi, mendukung, mengingatkan jadwal-jadwal penting dan memberi arahan untuk menuju proses kesempurnaan sehingga penulis dapat menyelesaikan penulisan tesis.
3. Bapak Kusnawi, S.Kom., M.Kom., selaku pembimbing pendamping yang di sela-sela kesibukan beliau tetap memberi arahan, pendapat serta memberi semangat berupa dukungan untuk dapatnya penulis menyelesaikan penulisan tesis.
4. Tim penguji ujian dari SPT, SHPT hingga UT yang telah memberikan arahan dan bahkan wawasan tambahan untuk proses penyempurnaan penulisan tesis.
5. Seluruh dosen pengajar Program Studi Pendidikan Jarak Jauh (PJJ) Magister Teknik Informatika Universitas Amikom Yogyakarta yang memberikan arahan dan tidak segan untuk sharing ilmu sehingga penulis mendapat tambahan wawasan dalam melaksanakan tugas.



6. Segenap pengelola Program Studi Pendidikan Jarak Jauh (PJJ) Magister Teknik Informatika Universitas Amikom Yogyakarta atas pelayanan dan bantuan kepada penulis.
7. Bapak Rektor IAIN Madura, saya haturkan terimakasih telah memberikan ijin untuk studi lanjut di Universitas Amikom Yogyakarta.
8. Kepala Unit TIPD IAIN Madura, saya haturkan terimakasih telah memberikan ijin untuk studi lanjut dan menggunakan data pendaftar mahasiswa baru untuk dijadikan dataset pada penulisan tesis.
9. Bapak saya yang tercinta, terindu dan terkasih (Alm) Mohammad Rasjid Sukram. Ibu saya Masrurah serta Adik saya Utari Sanaba yang tercinta dan tersayang yang selalu memberikan doa dan dukungan.
10. Seluruh teman-teman Mahasiswa Program Studi Pendidikan Jarak Jauh (PJJ) Magister Teknik Informatika Universitas Amikom Yogyakarta Angkatan 2021 yang telah banyak membantu dan kekeluargaannya.

Yogyakarta, 01 November 2023

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PENGESAHAN.....	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN PERNYATAAN KEASLIAN TESIS.....	v
HALAMAN PERSEMBAHAN.....	vi
HALAMAN MOTTO.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xv
DAFTAR GAMBAR.....	xvii
INTISARI.....	xix
<i>ABSTRACT</i> .....	xx
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang Masalah.....	1
1.2. Rumusan Masalah.....	5
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	6
1.5. Manfaat Penelitian.....	6
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>8</b>
2.1. Tinjauan Pustaka.....	8
2.2. Keaslian Penelitian.....	13

2.3. Landasan Teori.....	27
2.3.1. Penerimaan Mahasiswa Baru .....	27
2.3.2. Machine Learning.....	27
2.3.3. Deep Learning .....	28
2.3.4. Forecasting (Peramalan / Prediksi).....	29
2.3.5. Times Series Data.....	30
2.3.6. Dekomposisi Data .....	34
2.3.7. ACF dan PACF.....	35
2.3.8. Data Stasioneritas .....	38
2.3.9. Diferensiasi Data .....	41
2.3.10. Persamaan Garis .....	42
2.3.11. Least Square.....	43
2.3.12. Semi Average.....	44
2.3.13. Quadratic.....	45
2.3.14. Regresi Linear.....	46
2.3.15. Logika Fuzzy .....	49
2.3.16. Fuzzy Mamdani .....	51
2.3.17. Long Short-Term Memory (LSTM) .....	51
2.3.18. Min-Max Normalisasi.....	54
2.3.19. Denormalisasi .....	54

2.3.20. Pengujian Performa.....	55
<b>BAB III Metode PENELITIAN.....</b>	<b>57</b>
3.1. Jenis, Sifat, dan Pendekatan Penelitian.....	57
3.2. Metode Pengumpulan Data.....	57
3.3. Metode Analisis Data.....	58
3.4. Alur Penelitian.....	60
<b>BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....</b>	<b>64</b>
4.1. Pengumpulan Data.....	64
4.2. Preprocessing Data.....	65
4.2.1. Pembersihan Data.....	65
4.2.2. Decompose Data.....	68
4.2.3. ACF dan PCF Data.....	70
4.2.4. Uji Stasioneritas.....	73
4.2.5. Diferensiasi Data.....	74
4.3. Least Square.....	77
4.3.1. Pembentukan Model Prediksi.....	77
4.3.2. Hasil Prediksi.....	81
4.3.3. Uji Performance.....	82
4.3.4. Visualisasi Data Asli dan Data Prediksi.....	83
4.4. Semi Average.....	84
4.4.1. Pembentukan Model Prediksi.....	84

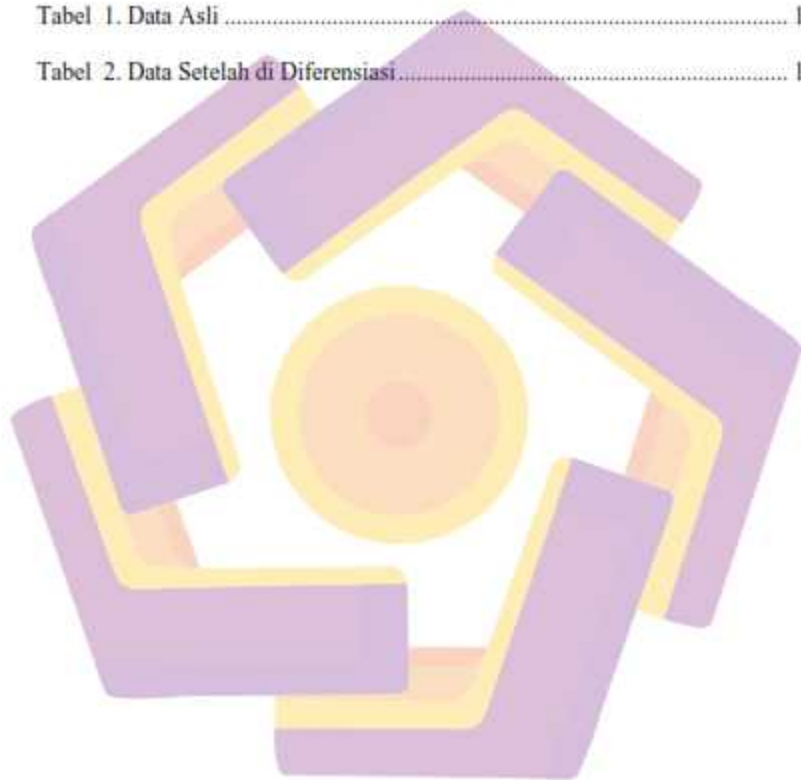
4.4.2. Hasil Prediksi.....	86
4.4.3. Uji Performance.....	87
4.4.4. Visualisasi Data Asli dan Data Prediksi.....	89
4.5. Quadratic.....	89
4.5.1. Pembentukan Model Prediksi.....	89
4.5.2. Hasil Prediksi.....	94
4.5.3. Uji Performance.....	95
4.5.4. Visualisasi Data Asli dan Data Prediksi.....	96
4.6. Long Short-Term Memory (LSTM).....	97
4.6.1. Pembentukan Model Prediksi.....	97
4.6.2. Hasil Prediksi.....	103
4.6.3. Uji Performance.....	104
4.6.4. Visualisasi Data Asli dan Data prediksi.....	105
4.7. Regresi Linear.....	106
4.7.1. Pembentukan Model Prediksi.....	106
4.7.2. Hasil Prediksi.....	108
4.7.3. Uji Performance.....	109
4.7.4. Visualisasi Data Asli dan Data Prediksi.....	111
4.8. Fuzzy Mamdani .....	111
4.8.1. Pembentukan Model Prediksi.....	111

4.8.2. Hasil Prediksi.....	115
4.8.3. Uji Performance.....	117
4.8.4. Visualisasi Data Asli dan Data Prediksi.....	118
4.9. Perbandingan Hasil Performance.....	119
4.9.1 Menggunakan Data Pendaftar.....	119
4.9.2 Menggunakan Data Pendaftar, Lulus dan Registrasi.....	121
BAB V PENUTUP.....	123
5.1. Kesimpulan.....	123
5.2. Saran.....	124
DAFTAR PUSTAKA.....	125
LAMPIRAN.....	129

## DAFTAR TABEL

Tabel 2.1. Matriks literatur review dan posisi penelitian.....	13
Tabel 4. 1. Total Data PMB Perjalur.....	64
Tabel 4. 2. Total Data PMB Sebelum Preprocessing.....	66
Tabel 4. 3. Total Data PMB yang Dihapus.....	66
Tabel 4. 4. Data PMB Setelah Preprocessing.....	67
Tabel 4. 5. Data PMB Pertahun Pelaksanaan.....	67
Tabel 4. 6. Data PMB Siap Digunakan.....	68
Tabel 4. 7. Hasil Identifikasi Least Square.....	79
Tabel 4. 8. Data Aktual dan Prediksi (Least Square).....	82
Tabel 4. 9. Selisih Antara Data Aktual dan Prediksi Least Square.....	82
Tabel 4. 10. Data Aktual dan Prediksi (Semi Average).....	87
Tabel 4. 11. Selisih Antara Data Aktual dan Prediksi Semi Average.....	88
Tabel 4. 12. Hasil Identifikasi Quadratic.....	91
Tabel 4. 13. Data Aktual dan Prediksi (Quadratic).....	94
Tabel 4. 14. Selisih Antara Data Aktual dan Prediksi Quadratic.....	95
Tabel 4. 15. Data Aktual dan Prediksi (LSTM).....	104
Tabel 4. 16. Selisih Antara Data Aktual dan Prediksi (LSTM).....	104
Tabel 4. 17. Data Aktual dan Prediksi (Regresi Linear).....	109
Tabel 4. 18. Selisih Antara Data Aktual dan Prediksi Regresi Linear.....	109
Tabel 4. 19. Data Aktual dan Prediksi (Fuzzy Mamdani).....	116
Tabel 4. 20. Selisih Antara Data Aktual dan Prediksi Fuzzy Mamdani.....	117

Tabel 4. 21. Perbandingan Hasil Performance (Least Square, Semi Average, Quadratic dan LSTM).....	119
Tabel 4. 22. Perbandingan Hasil Performance (Regresi Linear dan Fuzzy Mamdani) .....	121
Tabel 1. Data Asli .....	129
Tabel 2. Data Setelah di Diferensiasi.....	132

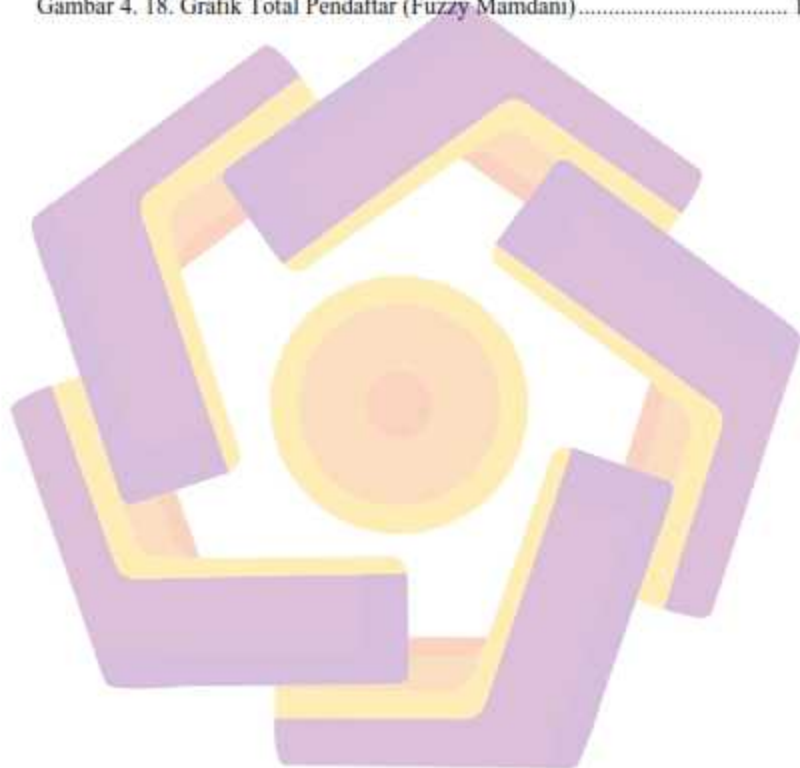




## DAFTAR GAMBAR

Gambar 2. 1. Upaward Trend.....	31
Gambar 2. 2. Downward Trend.....	31
Gambar 2. 3. Level Trend.....	32
Gambar 2. 4. Seasonality.....	32
Gambar 2. 5. Cyclical.....	33
Gambar 2. 6. Random Variations.....	33
Gambar 2. 7. Proses Penalaran Min-Max.....	50
Gambar 3.1. Alur Penelitian.....	61
Gambar 4. 1. Hasil Decompose Data Asli.....	69
Gambar 4. 2. Hasil ACF Data Asli.....	72
Gambar 4. 3. Hasil PACF Data Asli.....	72
Gambar 4. 4. Hasil Diferensiasi Data.....	75
Gambar 4. 5. Hasil Diferensiasi Data Tanpa Nilai NaN.....	76
Gambar 4. 6. Penentuan Unit Waktu (Least Square).....	77
Gambar 4. 7. Grafik Total Pendaftar (Least Square).....	84
Gambar 4. 8. Penentuan Unit Waktu (Semi Average).....	84
Gambar 4. 9. Grafik Total Pendaftar (Semi Average).....	89
Gambar 4. 10. Grafik Total Pendaftar (Quadratic).....	97
Gambar 4. 11. Hasil Min-Max Normalisasi.....	98
Gambar 4. 12. Grafik Loss Epoch LSTM.....	101
Gambar 4. 13. Hasil Prediksi LSTM (Min-Max Normalisasi).....	103

Gambar 4. 14. Grafik Total Pendaftar (LSTM) .....	106
Gambar 4. 15. Pembentukan Model Prediksi Regresi Linear .....	107
Gambar 4. 16. Grafik Total Pendaftar (Regresi Linear) .....	111
Gambar 4. 17. Install Library Fuzzy .....	112
Gambar 4. 18. Grafik Total Pendaftar (Fuzzy Mamdani) .....	119



## INTISARI

Pendaftaran Mahasiswa Baru digelar disetiap tahunnya dengan beberapa jalur yang dibuka. Namun, di 3 (tiga) tahun terakhir jumlah pendaftar selalu mengalami penurunan padahal di tahun-tahun sebelumnya mengalami penambahan pendaftar. Sehingga, perlu adanya suatu prediksi untuk meramalkan jumlah pendaftar. Selain itu, hasil dari prediksi tersebut dapat menjadi bahan pertimbangan penentuan kuota/pagu jumlah penerimaan mahasiswa baru di tahun akademik selanjutnya.

Penelitian ini menerapkan algoritma *Least Square*, *Semi Average*, *Quadratic*, *Long Short Term Memory (LSTM)*, *Regresi Linear*, dan *Fuzzy Mamdani* untuk memprediksi jumlah pendaftar mahasiswa baru berdasarkan data jumlah pendaftar dari tahun akademik 2014/2015 sampai 2022/2023 dengan dikelompokkan perbulan berdasarkan bulan pembuatan akun pendaftaran. Pengujian performance untuk algoritma tersebut di uji dengan dua (2) pengujian yakni *RSquare* dan *RMSE*. Selain itu, ada dua (2) jenis data yang digunakan untuk yang pertama menggunakan data pendaftar sedangkan untuk yang kedua menggunakan data pendaftar, lulus dan registrasi.

Berdasarkan hasil evaluasi pada penggunaan jenis data pertama yakni data pendaftar saja, memperoleh hasil bahwa algoritma *LSTM* yang paling efektif dalam memprediksi dengan nilai *Rsquare* 0,85 dan nilai *RMSE* 131,59. Sedangkan pada skema kedua yang menggunakan data pendaftar, lulus dan registrasi diperoleh hasil cukup memuaskan pada kedua algoritma tersebut dengan *Regresi Linear* (*Rsquare* 0,88, *RMSE* 119,00) dan *Fuzzy Mamdani* (*Rsquare* 0,86, *RMSE* 125,74).

Kata kunci: Fuzzy Mamdani, LSTM, Semi Average, Regresi Linear, Quadratic.

## **ABSTRACT**

*New Student Registration is held every year with several pathways open. However, in the last 3 (three) years the number of registrants has always decreased even though in previous years there was an increase in registrants. So, there needs to be a prediction to predict the number of registrants. Apart from that, the results of these predictions can be used as material for consideration in determining the quota/ceiling for the number of new student admissions in the next academic year.*

*This research applies the Least Square, Semi Average, Quadratic, Long Short Term Memory (LSTM), Linear Regression, and Fuzzy Mamdani algorithms to predict the number of new student registrants based on data on the number of registrants from the 2014/2015 to 2022/2023 academic year grouped by month. The month the registration account was created. Performance testing for this algorithm was tested using two (2) tests, namely RSquare and RMSE. Apart from that, there are two (2) types of data used. The first uses registrant data, while the second uses registrant, pass and registration data.*

*Based on the evaluation results using the first type of data, namely registrant data only, the results were that the LSTM algorithm was the most effective in predicting with an Rsquare value of 0.85 and an RMSE value of 131.59. Meanwhile, in the second scheme which uses registrant, pass and registration data, the results obtained are quite satisfactory for the two algorithms with Linear Regression (Rsquare 0.88, RMSE 119.00) and Fuzzy Mamdani (Rsquare 0.86, RMSE 125.74).*

*Keyword: Fuzzy Mamdani, LSTM, Semi Average, Regresi Linear Quadratic.*

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Pendaftaran Mahasiswa Baru digelar setiap tahunnya dengan melalui beberapa jalur yang diminati. Pada pendaftaran mahasiswa baru di tingkat UIN/IAIN/STAIN terdapat 2 jalur yang digelar secara nasional yakni SPAN-PTKIN dan UM-PTKIN. Sedangkan tingkat lokal di IAIN Madura menambahkan dua jalur yaitu Mandiri Reguler dan baru-baru ini menambahkan jalur Mandiri Prestasi. Sehingga di IAIN Madura menggelar empat jalur masuk yakni SPAN-PTKIN, UM-PTKIN, Mandiri Prestasi dan Mandiri Reguler setiap tahunnya.

Namun, beberapa tahun terakhir pendaftaran mahasiswa baru mengalami perubahan atau lebih tepatnya mengalami penurunan pendaftar. Sehingga, di beberapa tahun tersebut tidak dapat memenuhi pagu yang sudah ditetapkan. Maka dari itu dengan menggunakan data pendaftar mahasiswa baru dari tahun-tahun sebelumnya akan dilakukan prediksi pendaftar di tahun selanjutnya. Beberapa algoritma akan diterapkan dengan menggunakan data tersebut dan dilakukan perbandingan antar algoritma tersebut untuk mendapatkan algoritma mana yang terbaik. Sebagai landasan penentuan algoritma yang akan digunakan, akan dipaparkan beberapa penelitian terkait algoritma tersebut walau beberapa penelitian yang akan dipaparkan ini tidak berkaitan dengan data pendaftaran mahasiswa. Namun, dilihat dari seberapa baik algoritma tersebut digunakan pada data di penelitian dan kemiripan variabel data yang akan digunakan.

Seperti yang pernah Icha Yulian, Dini Sri Anggraeni dan Qurrotul Aini dalam penelitiannya yang bertujuan untuk menghindari penumpukan barang pada inventory agar tidak terjadi kelebihan barang dengan menggunakan algoritma Regresi Linear. Data yang digunakan terdiri dari Bulan dan jumlah penjualan pada tahun penjualan tersebut sebanyak 19 data bulanan. Hasil pengukuran performa menggunakan perhitungan MAD/MAE sebesar 62.50%, MSE sebesar 6391.34% dan MAPE sebesar 0.47%. Sedangkan hasil prediksi yang disertai dengan perhitungan indeks musim didapat nilai MAD/MAE sebesar 17.28%, MSE sebesar 5988.57% dan MAPE adalah 0.45%. Sehingga dapat disimpulkan bahwa pengaruh nilai indeks musim dapat mengurangi nilai error dalam peramalan sebesar 0.02% (Yulian et al., 2020).

Selanjutnya yakni Ines Saraswati Machfiroh dkk, yang membahas tentang trend penjualan sepeda motor untuk periode mendatang dengan membandingkan hasil dari dua (2) algoritma yakni Semi Average Method dan Least Square Method. Data yang digunakan merupakan data penjualan sepeda motor dari tahun 2015-2019 (tahun penjualan dan total unit motor yang terjual) dengan total data 5 data tahunan. Penelitiannya memperoleh hasil performa sebesar 31.89% pada Least Square dan 43,96% pada Semi Average yang dimana pengujian performa tersebut menggunakan evaluasi MAPE (Machfiroh et al., 2022).

Sedangkan Dalam bidang Institusi lebih tepatnya tentang peramalan jumlah calon mahasiswa yang akan mendaftar juga tidak luput dari penelitian. Ahmad Rifa'i melakukan penelitian tersebut untuk meramalkan jumlah pendaftar pada tahun akademik 2023/2024 sampai dengan tahun akademik 2026/2027. Data yang

digunakan yakni tahun akademik dan jumlah pendaftar pada tahun akademik tersebut sebanyak 7 data tahunan. Algoritma yang digunakan adalah algoritma Quadratic/parabolic. Hasil peramalannya menyatakan bahwa Jumlah calon mahasiswa baru yang akan mendaftar pada periode tahun akademik 2023/2024 adalah 651 orang Mahasiswa, 2024/2025 adalah 818 orang Mahasiswa, 2025/2026 adalah 1001 orang Mahasiswa, 2026/2027 adalah 1200 orang Mahasiswa. (Rifa'i, 2019)

Pada penelitian Laras Purwati Ayuningtias, Mohammad Irfan dan Jumadi menyatakan bahwa nilai performa dengan menggunakan algoritma AFFER pada Algoritma Fuzzy Mamdani mempunyai tingkat eror sebesar 19,76%, Fuzzy Tsukamoto sebesar 39,03% dan Fuzzy Sugeno sebesar 86,41%. Data yang digunakan yakni data jumlah mahasiswa baru sehingga ini menjadi pertimbangan untuk digunakan pada penelitian ini. Selain itu, penggunaan dari komponen lain selain jumlah pendaftar yakni data mahasiswa baru lulus seleksi dan mahasiswa baru yang melakukan registrasi ulang. (Ayuningtias et al., 2017)

Selain itu, Dwi Harini dan Lilia Sinta Wahyuniar dengan menggunakan algoritma Regresi Linear sedangkan data yang digunakan merupakan data jumlah murid baru. Penelitian tersebut mendapatkan hasil prediksi ada sebanyak 185 murid baru pada tahun ajaran 2022/2023 dan masuk ke kategori baik dengan nilai MSE 8,7% dan MAPE 15,12%.(Harini & Wahyuniar, 2021)

Sedangkan Westlie Wiratama, dkk dalam pembahasan penelitiannya mengenai prediksi turis mancanegara memprediksi jumlah wisatawan tertinggi

terjadi pada tahun 2024 bulan Agustus akan ada sebanyak 580 ribu yang berasal dari benua ASIA dengan nilai MAPE 13,84. (Wiratama et al., 2023)

Dari pemaparan beberapa hasil penelitian tersebut, pada penelitian ini akan menggunakan beberapa algoritma untuk dibandingkan hasil kinerjanya. Terdapat dua konsep perbandingan yang akan dibandingkan, pertama yang menggunakan data pendaftar saja dengan membandingkan hasil evaluasi prediksi dari Least Square, Semi Average, Quadratic, dan Long Short Term Memory (LSTM). Kedua yakni menggunakan data pendaftar, lulus dan registrasi dengan membandingkan hasil evaluasi prediksi dari Regresi Linear, Fuzzy Mamdani. Pemilihan algoritma tersebut didasari pada penelitian-penelitian sebelumnya menggunakan data tahunan dan jenis data yang digunakan relatif tren menaik dan stasioner, hasil evaluasi dari masing-masing algoritma masuk ke kategori bagus. Sedangkan pada penelitian ini menggunakan data yang pola trennya menaik lalu menurun serta masuk ke data tidak stasioner sehingga memerlukan beberapa tahapan lanjutan.

Prediksi jumlah pendaftar mahasiswa baru ini sangat diperlukan untuk mengantisipasi kekurangan mahasiswa yang diterima. Karena pada dasarnya, jumlah pendaftar ini juga yang bisa menggambarkan seberapa terkenal kampus kita di lingkungan siswa-siswa menengah atas serta menjadi landasan pertama untuk menjadi pertimbangan dalam menentukan seberapa banyak yang akan diluluskan untuk dapat terpenuhinya pagu penerimaan mahasiswa baru apabila melakukan registrasi ulang nantinya.

Dalam penggunaan keenam algoritma tersebut tentunya diperlukan pengujian performa pada algoritma-algoritma tersebut untuk mengetahui algoritma



mana yang baik untuk digunakan dengan jenis data yang digunakan. Ujian performa akan menggunakan perhitungan RMSE dan Rsquare. Dikarenakan Rsquare mampu menjelaskan seberapa besar model tersebut efektif dalam menangkap pola dan tren dalam data jumlah pendaftar mahasiswa baru, dimana nilai evaluasi model tersebut mendekati nilai 1 maka model tersebut dikatakan bagus dan direkomendasikan untuk digunakan. Sedangkan besar kecilnya RMSE menunjukkan bahwa prediksi model memiliki tingkat akurasi yang baik tidaknya, semakin besar nilai RMSE maka model tersebut melakukan kesalahan prediksi sebesar nilai tersebut.

### **1.2. Rumusan Masalah**

Berdasarkan apa yang sudah dipaparkan dalam latar belakang penelitian, maka rumusan masalah pada penelitian ini diantaranya:

- a. Bagaimana kinerja algoritma Least Square, Semi Average, Quadratic, Long Short Term Memory (LSTM), Regresi Linear, dan Fuzzy Mamdani pada prediksi jumlah pendaftar mahasiswa baru dengan perhitungan hasil dari RMSE dan RSquare ?
- b. Faktor apa saja yang mempengaruhi kinerja algoritma Least-Square, Semi Average, Quadratic, Long Short Term Memory (LSTM), Regresi Linear, dan Fuzzy Mamdani ?

### **1.3. Batasan Masalah**

Berikut batasan masalah dalam penelitian ini :

- a. Data yang akan digunakan merupakan data dari penerimaan mahasiswa baru antara lain data pendaftar, data lulus seleksi dan data yang melakukan registrasi ulang di lingkungan IAIN Madura.

- b. Data yang akan digunakan merupakan data dari tiga (3) jalur masuk penerimaan mahasiswa baru. Dimulai dari tahun akademik 2014/2015 sampai 2022/2023 (SPAN-PTKIN, UM-PTKIN dan Mandiri Reguler). Data tersebut dikelompokkan berdasarkan bulan pembuatan akun pendaftarannya.
- c. Skema pengujiannya, data PMB yang digunakan sebanyak 108 baris data dengan rincian 96 data train dan 12 untuk data test.
- d. Algoritma Least Square, Semi Average, Quadratic, Long Short Term Memory (LSTM), Regresi Linear, dan Fuzzy Mamdani yang akan digunakan untuk memprediksi jumlah pendaftar mahasiswa baru.
- e. Parameter yang digunakan untuk pengujian performa algoritma adalah Root Mean Square Error (RMSE) dan RSquare.

#### **1.4. Tujuan Penelitian**

Tujuan penelitian ini adalah :

- a. Untuk mengetahui kinerja algoritma Least Square, Semi Average, Quadratic, Long Short Term Memory (LSTM), Regresi Linear, dan Fuzzy Mamdani pada prediksi jumlah pendaftar mahasiswa baru.
- b. Untuk mengetahui algoritma mana yang cocok pada jenis data di penelitian ini dilihat dari penilaian performanya.

#### **1.5. Manfaat Penelitian**

Manfaat dari adanya penelitian ini :

- a. Memberikan pengetahuan tambahan tentang keilmuan pada bidang data mining terutama dalam hal prediksi.

- b. Hasil penelitian ini diharapkan dapat menjadi bahan masukan untuk penelitian yang berkaitan dengan komparasi algoritma data mining.
- c. Memberikan pengetahuan tambahan tentang keilmuan pada bidang machine learning dan deep learning.
- d. Mengetahui faktor yang mempengaruhi performa dari masing-masing algoritma.
- e. Mempermudah prediksi awal tentang jumlah pendaftar mahasiswa baru di tahun akademik mendatang.
- f. Membantu dalam penentuan Pagu agar terpenuhi Pagu Mahasiswa Baru.
- g. Civitas akademika dapat mengambil langkah awal ketika prediksi jumlah pendaftar mahasiswa belum melebihi target Pagu Mahasiswa Baru.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Tinjauan Pustaka

Teknik forecasting sangat diperlukan diberbagai bidang usaha maupun Institusi untuk memperoleh prediksi di tahun/periode mendatang agar yang direncanakan/diproduksi tidak jauh beda.

Seperti yang dilakukan Icha Yulian, Dini Sri Anggraeni dan Qurrotul Aini dalam penelitiannya yang bertujuan untuk menghindari penumpukan barang pada inventory agar tidak terjadi kelebihan barang juga menggunakan algoritma Regresi Linear. Data yang digunakan terdiri dari Bulan di tahun penjualan dan jumlah penjualan pada waktu tersebut. Pengukuran performa menggunakan perhitungan MAD/MAE sebesar 62.50%, MSE sebesar 6391.34% dan MAPE sebesar 0.47%. Sedangkan hasil prediksi yang disertai dengan perhitungan indeks musim didapat nilai MAD/MAE sebesar 17.28%, MSE sebesar 5988.57% dan MAPE adalah 0.45%. Sehingga, pengaruh nilai indeks musim dapat mengurangi nilai error dalam peramalan sebesar 0.02%. (Yulian et al., 2020)

Amiruddin dan Rezqiwati Ishak juga melakukan penelitian dengan algoritma yang sama yang dimana algoritma tersebut digunakan untuk memprediksi lebih awal terkait jumlah mahasiswa registrasi karena 4 tahun terakhir jumlah mahasiswa yang tidak registrasi cenderung meningkat dan mahasiswa registrasi cenderung menurun. Berdasarkan hasil penelitian dari 2 prodi yang dipilih yakni prodi Teknik Informatika didapatkan hasil tingkat error 4.24% atau tingkat akurasi

95.76%, dan untuk prodi Ilmu Hukum didapatkan tingkat error 7.69% atau tingkat akurasi 92.31%. (Amiruddin & Ishak, 2018)

Algoritma semi average juga dapat digunakan untuk forecasting seperti yang diteliti oleh Ines Saraswati Machfiroh dkk, yang membahas tentang trend penjualan sepeda motor untuk periode mendatang dengan membandingkan hasil dari dua (2) algoritma yakni Semi Average Method dan Least Square Method. Data yang digunakan merupakan data penjualan sepeda motor dari tahun 2015-2019 (tahun penjualan dan total unit motor yang terjual). Penelitiannya memperoleh hasil performa sebesar 31.89% pada Least Square dan 43,96% pada Semi Average yang dimana pengujian performa tersebut menggunakan algoritma MAPE (Machfiroh et al., 2022).

Peramalan juga berkaitan dengan kebijakan pemerintah seperti yang diteliti oleh Irfan Zikri dkk, melakukan analisis trend menggunakan tiga (3) macam algoritma yakni Least Square, Quadratic dan eksponensial dalam penelitian tren impor kedelai. Melalui Pangan dan Pertanian Organization (FAO), Badan Pusat Statistik (BPS), the Pusat Data dan Sistem Informasi (PUSDATIN) diperoleh data impor kedelai dari tahun 2003 sampai 2017. Temuan dari penelitian yang dilakukan, tren produksi kedelai cenderung menurun dengan rata-rata 6,12% pertahun sedangkan permintaan dan impor cenderung meningkat dengan rata-rata 3,30% dan 3,03% pertahun (Zikri et al., 2020).

Bukan hanya itu, peramalan tanaman pangan yang toleran terhadap perubahan iklim juga dapat membantu bidang pertanian. Hal ini diteliti oleh Sitti Raodah Garuda dan Yuliantoro Baliadi dengan menggunakan data daerah Provinsi

Papua dengan membandingkan beberapa algoritma yakni Exponential, linear, quadratic dan moving average pada penelitian Sitti Raodah Garuda dan Yuliantoro Baliadi di peramalan toleransi komoditas pangan. Data yang digunakan merupakan data produksi dalam periode 1970-2015 dengan empat (4) jenis tanaman yakni Padi, Jagung, Kedelai dan Ubi Jalar. Hasil dari peramalan keempat algoritma tersebut dibandingkan dan dicari mana yang terbaik ketika pengujian performanya (MSD/MSE, MAD/MAE dan MAPE). Model tren moving average ordo 2 sesuai untuk memprediksi produksi padi dan kedelai. Sedangkan Model tren Quadratic lebih sesuai untuk memprediksi produksi jagung dan ubi jalar (Garuda & Baliadi, 2018).

Peramalan dalam bidang farmasi juga tidak luput dari penelitian, seperti yang dilakukan oleh Indah Rahmawati dan Rony Wijanarko. Bertujuan untuk menghindari tragedi kehabisan persediaan obat dan kesalahan pelaporan karena di Apotek Demak Farma Jaya masih manual sehingga sering terjadi kesalahan. Dari penelitiannya pihak Apotek Demak Farma Jaya sudah dapat melakukan pembuatan laporan untuk peramalan penjualan Obat dengan efektif dan efisien. Adanya program untuk peramalan penjualan Obat ini, diharapkan nantinya pihak Apotek Demak Farma Jaya sudah dapat merencanakan anggaran yang akan dibuat. Pihak pemilik apotek dapat mengetahui secara langsung laporan terhadap perkiraan penjualan obat secara langsung, karena pada program yang dibuat pihak pemilik apotek dapat mengakses program secara langsung untuk melihat laporan perkiraan penjualan obat yang ada. (Rahmawati & Wijanarko, 2019)

Medyantiwi Rahmawita dan Ilham Fazri juga melakukan penelitian dalam bidang farmasi. Penelitiannya bertujuan untuk meramalkan penjualan obat-obatan di Rumah Sakit Bhayangkara dengan menggunakan algoritma least square berdasarkan data penjualan sebelumnya dan meringankan petugas dalam pengolaan data persediaan obat. Dari penelitiannya tersebut, algoritma least square Mempunyai error (tingkat kesalahan) yang diukur dengan MAPE adalah 3%. (Rahmawita & Fazri, 2018)

Dalam bidang Pendidikan lebih tepatnya tentang peramalan jumlah calon mahasiswa yang akan mendaftar juga tidak luput dari penelitian. Ahmad Rifa'i melakukan penelitian tersebut untuk meramalkan tahun akademik 2023/2024 Sampai dengan tahun akademik 2026/2027. Algoritma yang digunakan adalah algoritma Quadratic/parabolic. Hasil peramalannya menyatakan bahwa Jumlah calon mahasiswa baru yang akan mendaftar pada periode tahun akademik 2023/2024 adalah 651 orang Mahasiswa, 2024/2025 adalah 818 orang Mahasiswa, 2025/2026 adalah 1001 orang Mahasiswa, 2026/2027 adalah 1200 orang Mahasiswa. (Rifa'i, 2019)

Bidang pendidikan lainnya juga diteliti oleh Laras Purwati Ayuningtias, Mohammad Irfan dan Jumadi dengan menggunakan data jumlah mahasiswa baru menyatakan bahwa nilai performa dengan menggunakan algoritma AFFER pada Algoritma Fuzzy Mamdani mempunyai tingkat eror sebesar 19,76%, Fuzzy Tsukamoto sebesar 39,03% dan Fuzzy Sugeno sebesar 86,41%. (Ayuningtias et al., 2017)

Selanjutnya diteliti oleh Dwi Harini dan Lilia Sinta Wahyuniar dengan menggunakan algoritma Regresi Linear sedangkan data yang digunakan merupakan data jumlah murid baru. Penelitian tersebut mendapatkan hasil prediksi ada sebanyak 185 murid baru pada tahun ajaran 2022/2023 dan masuk ke kategori baik dengan nilai MSE 8,7% dan MAPE 15,12%. (Harini & Wahyuniar, 2021)

Dalam bidang deep learning kini juga dikembangkan algoritma Long Short Term Memory (LSTM) yang dapat digunakan pada bidang prediksi seperti yang dilakukan oleh Westlie Wiratama, dkk dalam memprediksi turis mancanegara ke Indonesia. Dengan menerapkan algoritma LSTM pada data wisatawan yang berasal dari benua ASIA menghasilkan MAPE sebesar 13,84 sedangkan pada data wisatawan yang berasal dari negara ASEAN menghasilkan MAPE sebesar 15,96. (Wiratama et al., 2023)

Nurfatima Selle, dkk dalam penelitian komparasi antara algoritma RNN dan LSTM. LSTM mampu menghasilkan kinerja yang lebih baik pada penggunaan deret waktu yang lebih panjang. Berdasarkan penerapan metode LSTM dan pengujian pada fitur data siang dan malam, didapatkan kondisi terbaik pada penggunaan untuk fitur data siang dengan panjang sequence 20, hidden size 8, 3 LSTM layer, dan 70% data training menghasilkan rata-rata RMSE 46,72, sedangkan untuk fitur data malam didapatkan panjang sequence 30, hidden size 8, 1 LSTM layer, dan 80% data training menghasilkan rata-rata RMSE 51,05. (Selle et al., 2022)



## 2.2. Keaslian Penelitian

Tabel 2.1. Matriks literatur review dan posisi penelitian

Implementasi Regresi Pada Prediksi Jumlah Pendaftar Mahasiswa Baru (Studi Kasus : IAIN Madura)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Suran atau Kelemahan	Perbandingan
1	Implementasi Least Square dalam untuk Prediksi Penjualan Sepeda Motor (Studi Kasus : PT. Graha Auto Pratama)  (Putra Manurung, 2015)	Bangun Unedo Putra Manurung, Jurnal Riset Komputer (JURIKOM), 2015	Memprediksi jumlah penjualan sepeda motor pada periode yang akan datang	Nilai MAPE dari tiap-tiap jenis kendaraan yang diprediksi berada di bawah angka 10%	-	<p><b>Jurnal</b> Data : penjualan sepeda Algoritma : Least Square Parameter : MAD, MSE, MAPE</p> <p><b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare</p>

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
2	Implementasi prediksi penjualan obat menggunakan algoritma least square pada apotek demak farma jaya  (Rahmawati & Wijanarko, 2019)	Indah Rahmawati dan Rony Wijanarko, Jurnal informatika dan rekayasa perangkat lunak, 2018	Untuk menghindari tragedi kehabisan persediaan obat dan kesalahan pelaporan	Pihak Apotek Demak Farma Jaya sudah melakukan pembuatan laporan untuk peramalan penjualan Obat dengan efektif dan efisien. Pihak pemilik apotek dapat mengetahui secara langsung laporan terhadap perkiraan penjualan obat secara langsung.	Penambahan fitur perhitungan pada perkiraan penjualan yang bersifat lebih dinamis, dan otomatis sesuai dengan jangka waktu yang diinginkan.	<b>Jurnal</b> Data : penjualan obat Algoritma : Least Square Parameter : - <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
3	Aplikasi peramalan penjualan obat menggunakan algoritma least square di rumah sakit bhayangkara  (Rahmawita Fazri, 2018)	Medyaniwi Rahmawita dan Ilham Fazri, Jurnal Ilmiah Rekayasa dan Manajemen Sistem Informasi, 2018	Meramalkan penjualan obat-obatan di Rumah	Mempunyai error (tingkat kesalahan)	Untuk penelitian selanjutnya, dapat digunakan algoritma peramalan yang lain untuk membandingkan	<b>Jurnal</b> Data : penjualan obat Algoritma : Least Square Parameter : - <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
4	Trend semi average and least square in forecasting yamaha motorcycle sales  (Machfiroh et al., 2022)	Ines Saraswati Machfiroh, Widiya Astuti Alam Sur dan Robby Tri Pangestu. Jurnal Ilmu Matematika dan Terapan (BAREKENG), 2022	Untuk mengetahui trend penjualan sepeda motor Yamaha sehingga diperoleh algoritma prediksi penjualan sepeda motor terbaik pada CV Surya Prima Pelaihari.	Algoritma Semi Average didasarkan pada nilai MAPE sebesar 43,96%. Algoritma Least Square memiliki nilai MAPE sebesar 31,89%.	Penilaian performa menggunakan satu (1) algoritma pengujian yakni MAPE	<b>Jurnal</b> Data : penjualan sepeda Algoritma : Semi Average dan Least Square Parameter : MAPE <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
5	Analisa Perbandingan Logic Fuzzy Algoritma Tsukamoto, Sugeno dan Mamdani  (Ayuningtias et al., 2017)	Laras Purwati Ayuningtias, Mohammad Irfan dan Jumadi. Jurnal Teknik Informatika, 2017	Untuk memprediksi jumlah pendaftar mahasiswa baru	Fuzzy mamdani mempunyai tingkat eror sebesar 19,76%, Tsukamoto sebesar 39,03% dan Sugeno sebesar 86,41%	Membandingkan algoritma lain dengan algoritma Fuzzy Mamdani	<b>Jurnal</b> Data : jumlah mahasiswa baru Algoritma : Tsukamoto, Sugeno dan Mamdani Parameter : AFFER <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
6	Prediksi Penjualan Menggunakan Algoritma Fuzzy Mamdani Pada PT. XYZ (Roza et al., 2022)	Yuni Roza, Yonky Fernando, Ihsan Verdian, Eka Lia Febrianti dan Ilwan Syafrinal - JURIKOM (Jurnal Riset Komputer), 2022	menentukan sebuah keputusan pada prediksi penjualan periode berikutnya ditentukan oleh sisa persediaan dari satu periode sebelumnya dan juga prediksi jumlah	perhitungan dan perbandingan dari jumlah permintaan dan persediaan yang telah terjadi pada bulan Desember 2019 adalah 1803 kemasan	-	<b>Jurnal</b> Data : jumlah penjualan Algoritma : Fuzzy Mamdani Parameter : - <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
7	Aplikasi algoritma tren waktu satu ragam dalam peramalan toleransi komoditas pangan terhadap perubahan iklim di papua (Garuda & Baliadi, 2018)	Sitti Ruodah dan Yuliantoro Baliadi, Jurnal Informatika Pertanian, 2018	Untuk meramalkan tanaman pangan yang toleran terhadap perubahan iklim di Provinsi Papua dengan algoritma trend satu ragam	<b>Model tren moving average ordo 2</b> sesuai untuk menduga ekspektasi produksi padi dan kedelai pada periode 1970-2015. <b>Model tren quadratic</b> lebih sesuai digunakan untuk menduga nilai ekspektasi produksi jagung dan ubi jalar yang didasarkan atas nilai MAPE, MAD, dan MSD	Penilaian performa menggunakan 3 algoritma pengujian yakni MAD, MSD dan MSD	<b>Jurnal</b> Data : produksi tanaman Algoritma : Linear, Quadratic, Exponential dan Moving Average Parameter : MAD, MSD dan MAPE <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
8	<p>Prediksi Jumlah Mahasiswa Registrasi Per Semester Menggunakan Linear Regresi pada Universitas Ihsan Gorontalo</p> <p>(Amiruddin Ishak, 2018)</p>	<p>Amiruddin dan Reziqiwati Ishak, ILKOM Jurnal Ilmiah, 2018</p>	<p>Salah satu faktor dalam mutu pelayanan dan peningkatan nilai akreditasi yakni meningkatnya jumlah mahasiswa yang melakukan registrasi pembayaran setiap semester. Berdasarkan laporan di PDPT Dikti tahun pelaporan 2017/2018 memiliki jumlah mahasiswa ± 9.000 orang, dari analisa 4 tahun terakhir jumlah mahasiswa yang tidak registrasi cenderung meningkat dan mahasiswa registrasi cenderung menurun, jika berlanjut, maka akan berdampak pada pencapaian harapan tersebut di atas.</p>	<p>Berdasarkan hasil penelitian dari 2 prodi yang dipilih yakni prodi Teknik Informatika didapatkan hasil tingkat error 4,24% atau tingkat akurasi 95,76%, dan untuk prodi Ilmu Hukum didapatkan tingkat error 7,69% atau tingkat akurasi 92,31%, dengan demikian aplikasi yang sudah dibangun layak untuk digunakan.</p>	<p>penelitian ini dapat diteliti kembali dengan menggunakan algoritma prediksi yang lain dan jumlah dataset yang lebih banyak lagi, agar hasilnya bisa dibandingkan mana yang lebih baik</p>	<p><b>Jurnal</b>            Data : Jumlah Mahasiswa Registrasi            Algoritma : Regresi Linear            Parameter : MAPE</p> <p><b>Penelitian</b>            Data : pendaftar mahasiswa baru            Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM)            Parameter : RMSE dan RSquare</p>

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
9	The statistical parabolic projection method untuk forecasting dalam penerimaan mahasiswa baru islam universitas indragiri di masa mendatang (Rifa'i, 2019)	Ahmad Rifa'i, Jurnal Manajemen dan Bisnis, 2019	Untuk membantu meramalkan jumlah calon mahasiswa baru yang akan mendaftar pada tahun akademik 2023/2024 Sampai dengan tahun akademik 2026/2027	Jumlah calon mahasiswa baru yang akan mendaftar pada periode tahun akademik 2023/2024 adalah 651 orang Mahasiswa, 2024/2025 adalah 818 orang Mahasiswa, 2025/2026 adalah 1001 orang Mahasiswa, 2026/2027 adalah 1200 orang Mahasiswa	Saran untuk Peneliti yang akan datang dan masih dalam tema yang sama dengan penelitian	<b>Jurnal</b> Data : pendaftar mahasiswa baru Algoritma : kombinasi Quadratic dengan Least Square Parameter : - <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
10	Estimasi Jumlah Murid Baru Menggunakan Algoritma Forecasting (Harini & Wahyuniar, 2021)	Dwi Harini, Lili Sinta Wahyuniar, Journal of Instructional Mathematics, 2021	Untuk memprediksi jumlah mahasiswa baru	Diprediksi ada sebanyak 185 murid baru pada tahun ajaran 2022/2023 dan masuk ke kategori baik dengan nilai MSE 8,7 dan MAPE 15,12%	-	<b>Jurnal</b> Data : jumlah murid baru Algoritma : Regresi Linear Parameter : MSE, MAPE <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
11	Peramalan Jumlah Siswa-Siswi Melalui Pendekatan Algoritma Regresi Linear  (Trianggana, 2020)	Dimas Aulia Trianggana, Jurnal Media Infotama, 2020	Aplikasi dapat membantu sekolah menjaga kualitas pendidikan dan menarik minat siswa untuk mendaftar berdasarkan jurusan yang tersedia.	Aplikasi mampu menampilkan hasil peramalan berdasarkan data trend yang diinputkan	Dapat mempergunakan aplikasi ini agar dapat mendapatkan informasi hasil peramalan jumlah pendaftaran siswa baru pada tahun pelajaran berikutnya	<b>Jurnal</b> Data : Jumlah pendaftar siswa baru perjurusan Algoritma : Regresi Linear Parameter : - <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
12	Forecasting the Sum of New College Students with Linear Regression Approach  (Utami et al., 2022)	Yulia Utami, Desi Vinsensia, Aura Nissa dan Sulastri, Jurnal Teknik Informatika C.I.T Medicom, 2022	Mengembangkan model regresi linier untuk memprediksi jumlah mahasiswa baru pada tahun berikutnya	Berdasarkan hasil diperoleh jumlah mahasiswa jurusan Teknik Informatika sebanyak 198 orang dengan nilai MAPE (Mean Absolute Percentage Error) sebesar 16,5%, dan untuk mahasiswa baru jurusan Manajemen Informatika sebanyak 8 orang dengan nilai MAPE sebesar 16,1%.	dipertimbangkan untuk menambahkan variabel independen ke kriteria peramalan atau menggabungkannya dengan algoritma lain untuk lebih mendekati hasil nyata.	<b>Jurnal</b> Data : Jumlah mahasiswa jurusan teknik informatika dan manajemen informatika selama 5 tahun terakhir Algoritma : Regresi Linear Parameter : MAPE <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
13	Perbandingan Algoritma Regresi Linier dan Regresi Random Forest Dalam Memprediksi Kasus Positif Covid-19  (Fachid & Triayudi, 2022)	Syakirah Fachid dan Agung Triayudi, Jurnal Media Informatika Budidarma, 2022	COVID-19 dengan cepat menyebar ke seluruh dunia, termasuk Indonesia. COVID-19 menyebar dengan cepat, menyebabkan kecacauan di masyarakat. Situasi ini memengaruhi banyak aspek kehidupan bangsa. Studi ini dilakukan untuk mengetahui jumlah orang yang terkena, meninggal, dan sembuh dari COVID-19. Selain itu, mereka ingin belajar bagaimana menggunakan regresi linier dan regresi hutan random.	Hasil penelitian penulis menghasilkan nilai RMSE sebesar 3031.127 MAPE 47.66 dan tingkat akurasi sebesar 94% pada algoritma Regresi Linier sedangkan untuk algoritma Random Forest menghasilkan nilai RMSE sebesar 1886.555, MAPE 14.85 dan tingkat akurasi sebesar 97.7%. Maka dapat disimpulkan bahwa pada penelitian ini regresi random forest lebih baik digunakan dibandingkan regresi linier.	-	<b>Jurnal</b> Data : Tanggal, Kasus Harian, Kasus Aktif, Kasus Aktif Dalam %, Sembuh (Harian), Sembuh, Meninggal(Harian) , Meninggal Algoritma : Regresi Linear dan Random Forest Parameter : RMSE dan MAPE <b>Penelitian</b> Data : pendafur mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare



Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
14	Perbandingan Algoritma Linear Regression dan Exponential Smoothing Dalam Peramalan Penerimaan Mahasiswa Baru  (Apriliza et al., 2022)	Fayza Apriliza, Darmansah, Annisa Oktavyani dan Dzakiyyah Al Kaazhim, JURIKOM (Jurnal Riset Komputer), 2022	pihak institusi membutuhkan data acuan seperti data prediksi jumlah mahasiswa yang akan diterima dalam lima tahun ke depan	Hasil pengujian algoritma linear regression memiliki nilai MAE 145,28; MSE 15238,46; dan MAPE 0,1216054793, sedangkan algoritma exponential smoothing dengan $\alpha = 0,1$ memiliki nilai MAE 327,2938; MSE 137036,2639; dan MAPE 0,2875524468	-	<b>Jurnal</b> Data : jumlah mahasiswa baru yang diterima Algoritma : Regresi Linear dan Exponential Smoothing Parameter : MAE, MSE, MAPE <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
15	Prediksi Jumlah Penduduk Provinsi Kalimantan Selatan Menggunakan Algoritma Semi Average  (Ruslan & Banjarbaru, 2016)	M. Ruslan, IJSE – Indonesian Journal on Software Engineering, 2016	memprediksi perolehan jumlah penduduk periode yang akan datang berdasarkan data penduduk tahun sebelumnya	Dari hasil uji coba 3 tahun terakhir menunjukkan validitas Algoritma Semi Average adalah 98,34% sehingga dinyatakan valid	-	<b>Jurnal</b> Data : Jumlah Penduduk Algoritma : Semi Average Parameter : Akurasi = 100% - MAPE <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
16	Analisa Forecasting Pendaftaran Mahasiswa Baru Universitas Muria Kudus Pusat Studi: Sains Dan Teknologi  (Evanita et al., 2022)	Evanita, Tutik Khotimah dan Muhamad Bakhar, Smart Comp, 2022	Pendaftaran siswa baru melonjak tajam bahkan pada tahun 2017, Universitas harus melakukan persiapan tambahan yang belum dijadwalkan untuk menangani penambahan mahasiswa baru yang kadang-kadang di luar ekspektasi. Ini termasuk meningkatkan jumlah guru, menambah kelas ke program studi, dan menyusun jadwal untuk sistem registrasi. Namun, masuknya siswa baru setiap tahun tidak selalu membuat universitas nyaman dan aman.	Nilai RMSE nya sebesar 49,6 5 dan nilai SE nya sebesar 2467,865	-	<p><b>Jurnal</b> Data : jumlah mahasiswa, dana promosi dan biaya masuk mahasiswa baru Algoritma : Regresi Linear Parameter : SE, RMSE</p> <p><b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare</p>

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
17	Aplikasi Forecasting Penjualan Menggunakan Algoritma Semi Average pada Toko Rumah Kita Makassar  (Patandean et al., 2019)	Silce Patandean, Askar dan Mashud, Inspiration : Jurnal Teknologi Informasi dan Komunikasi, 2019	Merancang dan mengimplementasikan Aplikasi Forecasting penjualan menggunakan algoritma semi average pada Toko Rumah Kita Makassar	Keakuratan peramalan sebesar 98,5%	Adanya pengembangan aplikasi yang dapat terintegrasi dengan sistem lain yang dapat mendukung kegiatan perusahaan dan dapat dikembangkan dengan menggunakan algoritma lain	<b>Jurnal</b> Data : Data penjualan kompor gas Algoritma : Semi Average Parameter : Akurasi <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
18	Analisis Perancangan Aplikasi Peramalan Persediaan Barang Dagang Sederhana dengan Algoritma Semi Average Method  (Saryanti & Sidhiantari, 2020)	I Gusti Ayu Desi Saryanti, I Gusti Ayu Putri Indah Sidhiantari, Jurnal Teknik Elektro dan Komputer, 2020	Beberapa pedagang yang tidak menentukan pada tahap pembelian karena kebanyakan aplikasi POS tidak memiliki fitur yang memungkinkan untuk memprediksi pembelian barang.	menampilkan kemungkinan error rate yang dicapai adalah kurang lebih 5,26%	Perlu juga untuk mempertimbangkan penggunaan dari library yang sudah tersedia untuk membantu dalam proses pembuatannya.	<b>Jurnal</b> Data : Data penjualan Algoritma : Semi Average Parameter : MAPE <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
19	Forecasting Analysis of MSME Sales of Nata De Coco Products in PT Shenovia Using Least Square, Moving Average, and Semi Average Methods  (Sudiatini & Ara, 2022)	Dian Sudiantini dan Sheren Novita Ara Journal Operational Research and Economics, 2022	Untuk mengetahui pengaruh ramalan penjualan pada PT Shenovia	Algoritma least square meramalkan sebanyak 16.126,69 bungkus dengan nilai MAPE 67%. Algoritma rata-rata bergerak meramalkan 15.758,33 bungkus dengan nilai MAPE 2,91%. Algoritma semi rata-rata meramalkan 17.025,08 bungkus dengan nilai MAPE 2,27%.	Manajemen perlu mempertimbangkan penggunaan algoritma rata-rata bergerak atau semi rata-rata dalam membuat peramalan penjualan karena algoritma ini paling efisien dalam proses perhitungan, dan algoritma ini lebih stabil.	<b>Jurnal</b> Data : Data penjualan produk Nata De Coco Algoritma : Semi Average, Moving Average, Least Square Parameter : MAPE <b>Penelitian</b> Data : pendftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
20	Decision Support System for Predicting Increased Data on Objects of Motor Vehicle Name Transfer (BBNKB 1) Using Regresi Linear Method (Case Study : Wheels 2 and wheels 4 in Samarinda)  (Pakpahan et al., 2019)	H S Pakpahan, O Anandiya, U Hairah dan Masma Wati, The 3rd International Conference On Science, 2019	Memperkirakan akan jumlah kendaraan yang akan masuk ke Samarinda	Hasil dari prediksi peningkatan data objek kendaraan menunjukkan bahwa nilai MAPE roda 2 rata-rata adalah 0,55% pada tahun 2017, 1,92% pada tahun 2018, dan 1,31% pada tahun 2019. Nilai MAPE roda 4 rata-rata adalah 0,57% pada tahun 2017, 0,77% pada tahun 2018, dan 0,47% pada tahun 2019. Kedua kendaraan memenuhi kriteria dengan nilai MAPE di bawah 10%.	Tidak melakukan perbandingan dengan algoritma lain	<b>Jurnal</b> Data : kendaraan masuk Algoritma : Tend Moment Parameter : MAPE <b>Penelitian</b> Data : pendftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
21	Penerapan algoritma Regresi Linear dalam forecasting penjualan produk cv rabbani asyisa  (Yulian et al., 2020)	Icha Yulian, Dini Sri Anggraeni dan Qurrotul Aini, Jurnal Teknologi dan Sistem Informasi (JURTEKSI), 2020	Untuk menghindari penumpukan barang pada inventory agar tidak terjadi kelebihan barang	Pengukuran performa menggunakan nilai MAD/MAE sebesar 62.50%, MSE sebesar 6391.34%, dan MAPE sebesar 0.47%. Sementara hasil prediksi yang menggunakan nilai indeks musim menghasilkan nilai MAD/MAE sebesar 17.28%, MSE sebesar 5988.57%, dan MAPE sebesar 0.45%, sehingga nilai indeks musim dapat mengurangi nilai error peramalan sebesar 0.02%.	Melakukan prediksi dengan menggunakan algoritma lain.	<b>Jurnal</b> Data : penjualan produk Algoritma : Regresi Linear Parameter : MAD/MAE, MSE dan MAPE <b>Penelitian</b> Data : pendftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
22	Perbandingan Prediksi Penggunaan Listrik dengan Menggunakan Metode Long Short Term Memory (LSTM) dan Recurrent Neural Network (RNN)  (Selle et al., 2022)	Nurfatima Selle, Novanto Yudistira dan Candra Dewi  Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) 2022	Energi listrik yang tidak dapat disimpan dalam waktu yang lama dan harus dapat selalu tersalurkan. Sehingga penyedia harus dapat mampu menyediakan energi listrik dengan tepat.	kondisi terbaik untuk penggunaan untuk fitur data siang adalah panjang sequence 20, ukuran tersembunyi 8, 3 lapisan LSTM, dan 70% instruksi data. Untuk fitur data malam, kondisi terbaik adalah panjang sequence 30, ukuran tersembunyi 8, 1 lapisan LSTM, dan 80% instruksi data menghasilkan rata-rata RMSE 51,05.	-	<b>Jurnal</b> Data : penggunaan listrik Algoritma : RNN dan LSTM Parameter : RMSE <b>Penelitian</b> Data : pendftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

Tabel 2.1. (Lanjutan)

No	Judul	Peneliti, Media Publikasi, dan Tahun	Tujuan Penelitian	Kesimpulan	Saran atau Kelemahan	Perbandingan
23	Prediksi Turis Mancanegara ke Indonesia Menggunakan Metode EDA Time Series dan LSTM  (Wiratama et al., 2023)	Westlie Wiratama, Lutfi Aulia Alifah, Alta Gurusinga dan Evta Indra  Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK) 2023	Memprediksi pola pergerakan jumlah wisatawan di masa depan, sehingga dapat digunakan sebagai pedoman dalam menentukan kebijakan yang akan diambil	Pada hasil EDA, benua asia menerima kunjungan wisatawan terbanyak, dengan Malaysia sebagai negara dengan jumlah tertinggi, diikuti oleh China dan Timor Leste. Sedangkan LSTM, benua asia menerima kunjungan wisatawan tertinggi pada tahun 2024 (580 ribu) dan ASEAN pada tahun 2025 (580 ribu), masing-masing dengan nilai MAPE 13.84 dan ASEAN dengan nilai 15.	Menyediakan data per hari memberikan tingkat detail yang lebih tinggi, memungkinkan kita untuk menganalisis tren dan pola secara lebih mendalam serta.	<b>Jurnal</b> Data : kunjungan turis mancanegara ke Indonesia Algoritma : EDA dan LSTM Parameter : MAPE <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare
24	Prediksi Harga Beras Berdasarkan Kualitas Beras dengan Metode Long Short Term Memory  (Nafi'iyah & Wulandari, 2022)	Nur Nafi'iyah, Putri Anggelia Wulandari Jurnal Inovtek Polbeng - Seri Informatika 2022	Membuat model arsitektur LSTM untuk memprediksi harga beras berdasarkan kualitas, yaitu premium, medium, dan luar kulit.	Arsitektur yang dibuat adalah layer input, 3 layer hidden LSTM, dan layer output, yaitu 4-50-50-1. Hasil evaluasi MAE dari prediksi harga beras premium, medium, dan luar kulit secara berurutan adalah 83,49, 89,6, 96,99.	-	<b>Jurnal</b> Data : produksi beras Algoritma : LSTM Parameter : MAE <b>Penelitian</b> Data : pendaftar mahasiswa baru Algoritma : Least Square, Semi Average, Quadratic, Regresi Linear, Fuzzy Mamdani dan Longs Short-Trem Memory (LSTM) Parameter : RMSE dan RSquare

### **2.3. Landasan Teori**

#### **2.3.1. Penerimaan Mahasiswa Baru**

##### **1) Jalur SPAN-PTKIN**

Model seleksi yang dilaksanakan berdasarkan prestasi calon mahasiswa dan digelar secara bersamaan / nasional. (Direktorat Jenderal Pendidikan Islam Kementerian Agama RI, 2019)

##### **2) Jalur UM-PTKIN**

Model seleksi yang dilaksanakan berdasarkan hasil ujian tulis calon mahasiswa serta ujian ini digelar secara bersamaan / nasional. (Direktorat Jenderal Pendidikan Islam Kementerian Agama RI, 2019)

##### **3) Jalur Mandiri Reguler**

Model seleksi nasional namun bersifat lokal karena digelar oleh pihak IAIN Madura sendiri, berdasarkan ujian tulis calon mahasiswa.

#### **2.3.2. Machine Learning**

Machine Learning / Pembelajaran Mesin merupakan studi ilmiah tentang algoritma yang membangun model matematika ataupun statistik komputasi berdasarkan data sampel / data training untuk membuat suatu prediksi ataupun keputusan tanpa diprogram secara eksplisit dalam melakukan tugasnya. (Pratama, 2020)

Dalam Machine Learning terdapat 3 jenis yakni :

1) Supervised Learning

Teknik yang mengidentifikasi hubungan / mengenali pola antara input data dan label output. Contoh dari teknik ini salah satunya ada prediksi / peramalan.

2) Unsupervised Learning

Teknik yang mengidentifikasi hubungan / mengenali pola yang sebelumnya belum dapat ditemukan pada data yang tidak berlabel / tidak memiliki output yang sudah ada sebelumnya dengan pengawasan manusia yang minimal.

3) Semi-Unsupervised Learning

Teknik yang menggabungkan sejumlah kecil data berlabel dan data tidak berlabel selama dilakukan training data untuk menemukan pola diantara keduanya.

### 2.3.3. Deep Learning

Semenjak tahun 1950-an, salah satu cabang dari AI yang disebut dengan Machine Learning (ML) telah berkembang cukup pesat dengan implementasi di beberapa bidang. NN adalah salah satu implementasi dari ML, sedangkan Deep Learning (DL) merupakan salah satu implementasi dari NN. DL, yang mulai populer digunakan sejak tahun 2006, menggunakan mekanisme deep architecture of learning atau pendekatan hierarchical learning.

Learning atau pembelajaran dalam hal ini adalah sebuah prosedur yang berisi proses estimasi parameter-parameter suatu model sehingga model yang



dikembangkan (algoritma) dapat menyelesaikan suatu tugas atau permasalahan tertentu. DL menggunakan beberapa lapisan (layers) di antara lapisan masukan (input layer) dan lapisan keluaran (output layer). Arsitektur tersebut dapat digunakan untuk melakukan pemrosesan nonlinier dengan beberapa tahap yang hasilnya dapat digunakan untuk feature learning dan klasifikasi pola (pattern classification). (Diponegoro et al., 2021)

#### **2.3.4. Forecasting (Peramalan / Prediksi)**

*Forecasting* / peramalan / prediksi merupakan proses untuk menduga suatu kejadian mendatang berdasarkan historis dan pengalaman dari data yang digunakan dengan menggunakan perhitungan matematis untuk memperkecil resiko kesalahan. *Forecasting* / peramalan / prediksi merupakan salah satu fungsi dari dari penggunaan data mining menurut Daniel T. Larose. (Kurniadi, 2018)

Peramalan pada dasarnya merupakan perkiraan suatu peristiwa di masa mendatang. Dimana situasi peramalan sangat beragam dalam horison waktu peramalan, faktor yang menentukan hasil sebenarnya, tipe pola data dan berbagai aspek lainnya. Sebelum melakukan peramalan harus diketahui terlebih dahulu apa sebenarnya persoalan dalam pengambilan keputusan itu. Setiap pengambilan keputusan yang menyangkut keadaan di masa yang akan datang, maka pasti ada peramalan yang melandasi pengambilan keputusan. Tujuan peramalan adalah untuk meredam ketidakpastian, sehingga diperoleh suatu perkiraan yang mendekati keadaan yang sebenarnya. Jika hasil peramalan mendekati akurat, maka hal ini

sangat berpengaruh besar untuk proses pengambilan keputusan pada perusahaan ataupun instansi. (Hatta & Fauziah Fitri, 2020)

Ada beberapa macam tipe peramalan yang dapat digunakan, antara lain (Amiruddin & Ishak, 2018):

#### 1) Times Series Model

Metode time series adalah metode peramalan secara kuantitatif dengan menggunakan waktu sebagai dasar peramalan.

#### 2) Causal Model

Metode peramalan yang menggunakan hubungan sebab-akibat sebagai asumsi, yaitu bahwa apa yang terjadi di masa lalu akan terulang pada saat ini.

#### 3) Judgemental Model

Bila time series dan causal model bertumpu pada kuantitatif, pada judgemental mencakup untuk memasukkan faktor-faktor kuantitatif/subjektif ke dalam metode peramalan. Secara khusus berguna bilamana faktor-faktor subjektif yang diharapkan menjadi sangat penting bilamana data kuantitatif yang akurat sudah diperoleh.

#### 2.3.5. Times Series Data

Waktu berperan sebagai index dari Time Series Data selain itu, index waktu memiliki interval yang konsisten misalnya harian, bulanan, tahunan dan seterusnya. Terdapat empat (4) komponen yang terdapat pada Time Series Data, antara lain :

##### 1) Trend

Trend merupakan tendensi atau kecenderungan pergerakan data yakni bisa cenderung naik, turun atau tetap.

a) Upward Trend (cenderung naik)

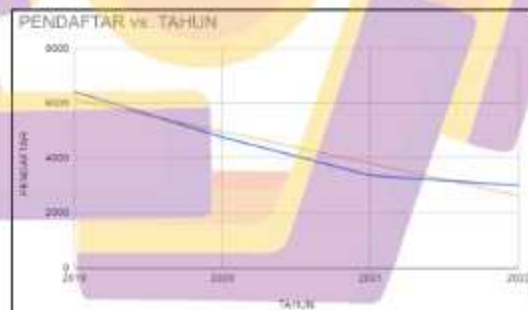
Data yang mengalami kenaikan berdasarkan garis trend (orange)  
dicontohkan pada Gambar 2.1. :



Gambar 2. 1. Upaward Trend

b) Downward Trend (cenderung turun)

Data yang mengalami penurunan berdasarkan garis trend (orange)  
dicontohkan pada Gambar 2.2 :



Gambar 2. 2. Downward Trend

c) Level Trend (cenderung tetap)

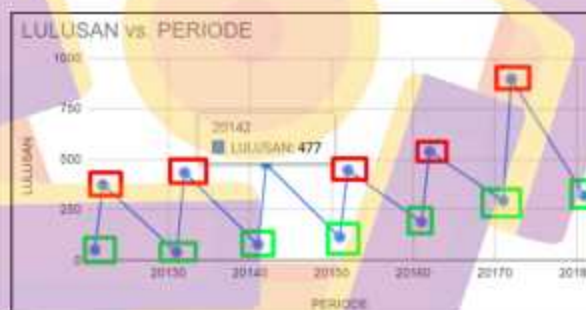
Data yang tidak mengalami penurunan maupun kenaikan  
(cenderung tetap) berdasarkan garis trend (orange) dicontohkan pada  
Gambar 2.3 :



Gambar 2. 3. Level Trend

## 2) Seasonality / Variasi Musiman

Pola yang ditemukan pada data, dimana data tersebut mengalami kenaikan/penurunan secara berulang dan sifatnya periodik dicontohkan pada Gambar 2.4 :

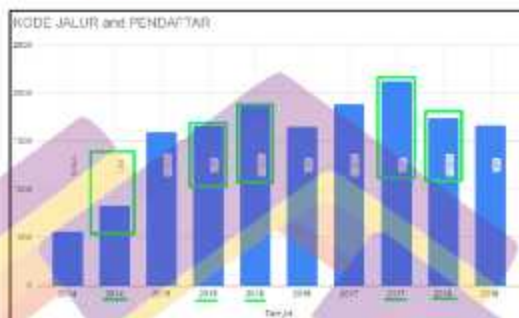


Gambar 2. 4. Seasonality

Pada Gambar 4 tersebut, kotak merah merupakan data lulusan yang mengalami kenaikan dan selalu terjadi di tahun periode genap. Sedangkan kotak hijau merupakan penurunan jumlah lulusan mahasiswa yang selalu terjadi di tahun periode ganjil.

### 3) Cyclical / Variasi Siklis

Di dalam data terdapat pola yang berulang namun tidak bersifat periodik (regular) dicontohkan pada Gambar 2.5 :



Gambar 2. 5. Cyclical

Terlihat pada Gambar 5 bahwa tiap tahun akademik berjalan, kenaikan pendaftar selalu ada namun tidak di satu jalur masuk.

### 4) Random Variations / Variasi Random

Random Variations ini merupakan gerakan data yang tidak memiliki pola atau tidak beraturan (irregular) dicontohkan pada Gambar 2.6 :



Gambar 2. 6. Random Variations

### 2.3.6. Dekomposisi Data

Dekomposisi Data merupakan proses untuk memecah suatu rangkaian waktu suatu data menjadi lebih sederhana yang bertujuan untuk memecah data deret waktu menjadi empat pola yaitu trend, musiman, siklis, dan perubahan-perubahan yang bersifat random, kemudian diidentifikasi masing-masing komponen tersebut secara terpisah. Proses decompose umumnya membagi data time series menjadi tiga komponen utama: tren (trend), musiman (seasonal), dan sisa (residual). Berikut adalah cara membaca hasil grafik decompose data:

- a. Original time series plot : Plot awal dari rangkaian waktu sebelum proses decompose.
- b. Trend component : Jika grafik ini naik atau turun secara konsisten, ini menunjukkan tren umum dalam data. Komponen tren mencerminkan perubahan umum dalam data sepanjang waktu.
- c. Seasonal component : Jika ada pola musiman yang berulang setiap periode tertentu (misalnya, setiap tahun atau setiap bulan), grafik ini akan menunjukkan fluktuasi yang terjadi pada saat yang sama setiap periode.
- d. Residual component : Grafik sisa (residual) menunjukkan bagian dari data yang tidak dapat dijelaskan oleh tren atau fluktuasi musiman. Jika grafik sisa acak dan tidak menunjukkan pola tertentu, ini adalah indikasi bahwa tren dan fluktuasi musiman telah dihilangkan secara efektif.
- e. Perhatikan pola tren-musiman : Perhatikan apakah ada pola tertentu dalam tren atau fluktuasi musiman yang dapat memberikan wawasan tentang sifat data.

- f. Korelasi antar komponen : Evaluasi apakah ada korelasi antara komponen tren dan musiman atau antara musiman dan sisa. Korelasi yang signifikan dapat mengindikasikan adanya interaksi antara komponen-komponen tersebut.
- g. Cek stationarity : Jika sisa tampak stasioner, ini menunjukkan bahwa proses decompose telah berhasil menghasilkan data yang lebih stasioner.
- h. Adjustments dan model lebih lanjut.

### 2.3.7. ACF dan PACF

ACF atau Autocorrelation Function yakni untuk mengukur seberapa kuat hubungan antara sebuah observasi pada waktu tertentu dengan observasi yang terjadi pada waktu sebelumnya. Ini memberikan gambaran umum tentang seberapa besar pengaruh suatu nilai pada waktu tertentu terhadap nilai pada waktu-waktu sebelumnya.

Melalui analisis ACF:

- a. Autocorrelation yang menurun secara berangsur: Jika nilai ACF menurun secara berangsur, ini mungkin menunjukkan adanya tren dalam data. Diferensiasi orde pertama  $Y'_t = Y_t - Y_{t-1}$  seringkali efektif untuk menghilangkan tren tersebut.
- b. Autocorrelation yang berulang pada lag-lag tertentu: Jika ada pola musiman dalam ACF, ini mungkin menunjukkan adanya efek musiman. Seasonal differencing atau transformasi musiman mungkin diperlukan untuk menghilangkan efek ini.

- c. Autocorrelation yang signifikan pada lag yang lebih tinggi: Autocorrelation yang signifikan pada lag yang lebih tinggi dapat menunjukkan adanya pola autoregressive dalam data. Transformasi ARIMA dengan orde autoregressive ( $p$ ) mungkin diperlukan.

Auto correlation merupakan korelasi antar data di dalam time series tersebut yang dipisahkan dalam lag. Artinya, jika lag =2, maka nilai  $t$  akan dikorelasikan dengan  $t+2$  (artinya data pertama akan dikorelasikan dengan data ketiga), jika menggunakan lag=4, maka kemudian  $t$  akan dikorelasikan dengan  $t+4$ , begitu seterusnya. Hal ini biasanya ditulis dengan rumus korelasi ( $X_t, X_{k+t}$ ) dengan  $k$  adalah jumlah lag dan  $t$  adalah nilainya.

PACF atau Partial Autocorrelation Function yakni mengukur hubungan antara observasi pada waktu tertentu dengan observasi yang terjadi pada waktu-waktu sebelumnya dengan menghilangkan efek dari observasi di antara keduanya. Dengan kata lain, PACF mengukur korelasi langsung antara dua waktu tertentu setelah menghapus pengaruh waktu-waktu di antaranya.

Melalui analisis PACF:

- a. Autocorrelation parsial yang signifikan pada lag ke- $p$ : Jika terdapat autocorrelation parsial yang signifikan pada lag ke- $p$  pada PACF, ini mungkin menunjukkan adanya orde autoregressive (AR) pada  $p$ . Transformasi ARIMA dengan orde autoregressive ( $p$ ) dapat diterapkan.
- b. Autocorrelation parsial yang nol setelah lag ke- $p$ : Jika terdapat autocorrelation parsial yang tiba-tiba menjadi nol setelah lag ke- $p$ , ini



mungkin menunjukkan adanya orde moving average (MA) pada  $q$ . Transformasi ARIMA dengan orde moving average ( $q$ ) dapat diterapkan.

- c. Autocorrelation parsial yang berulang pada lag-lag tertentu: Autocorrelation parsial yang berulang pada lag-lag tertentu dapat menunjukkan adanya pola musiman. Transformasi musiman atau seasonal differencing mungkin diperlukan.

Sama halnya dengan Auto Correlation, PACF juga menghitung korelasi antar data di time series. Perbedaannya adalah jika ACF mencari korelasi antara  $X_t$  dan  $X_{t+k}$  ( $k = \text{lag}$ ), maka PACF juga menghitung korelasi antara  $X_t$  dan  $X_{t+k}$  namun sebelumnya menghilangkan data antara  $X_t$  dan  $X_{t+k}$  terlebih dahulu. Berarti data yang dihilangkan adalah data  $X_{t+1}$  sampai dengan  $X_{t+k-1}$ .

Kombinasi analisis ACF dan PACF:

- a. ACF dan PACF bersama-sama: Pertimbangkan hasil ACF dan PACF bersama-sama untuk mendapatkan gambaran yang lebih lengkap. Misalnya, jika ACF menunjukkan musiman, sedangkan PACF menunjukkan orde autoregressive, maka mungkin diperlukan kombinasi seasonal differencing dan transformasi ARIMA.
- b. Eksplorasi visual: Selain ACF dan PACF, juga perhatikan plot data yang sudah diferensiasi atau ditransformasi. Eksplorasi visual membantu memahami apakah transformasi tersebut berhasil menghasilkan data yang lebih stasioner.

### 2.3.8. Data Stasioneritas

Data stasioneritas mengacu pada sifat statistik yang tetap atau konstan dari suatu rangkaian waktu sepanjang waktu. Dalam konteks data time series, sifat ini penting karena banyak metode analisis statistik dan model memerlukan asumsi bahwa data tersebut bersifat stasioner atau mendekati stasioneritas.

Beberapa alasan utama untuk menggunakan data stasioner dalam analisis time series melibatkan upaya untuk membuat analisis statistik lebih dapat diandalkan dan menghasilkan model prediksi yang lebih baik. Berikut adalah beberapa alasan utama:

- a. Stabilitas statistik: Dengan menggunakan data stasioner, properti statistik seperti rata-rata dan varians tetap sepanjang waktu. Ini membuat lebih mudah untuk menarik kesimpulan dan membuat prediksi yang lebih stabil.
- b. Menghilangkan tren dan musiman: Stasioneritas membantu menghilangkan tren dan komponen musiman dari data, membuatnya lebih mudah untuk fokus pada pola-pola yang mungkin tersembunyi dalam data dan meningkatkan interpretasi hasil analisis.
- c. Mempermudah prediksi: Model prediksi, seperti model regresi, lebih efektif saat bekerja dengan data stasioner. Prediksi yang dibuat pada data yang tidak stasioner mungkin sulit diandalkan karena fluktuasi alamiah data dapat menyebabkan kesalahan prediksi.
- d. Asumsi model statistik: Banyak model statistik, seperti model autoregressive integrated moving average (ARIMA), memerlukan asumsi

stasioneritas dalam analisisnya. Melanggar asumsi ini dapat menyebabkan hasil analisis yang tidak akurat.

- e. Estimasi parameter yang stabil: Estimasi parameter dalam model statistik seringkali lebih stabil pada data yang stasioner. Parameter yang stabil memungkinkan model memberikan hasil yang lebih konsisten sepanjang waktu.
- f. Menghindari spurious correlation: Data non-stasioner dapat menghasilkan korelasi yang keliru antar variabel yang sebenarnya tidak memiliki hubungan sebab-akibat. Stasioneritas membantu mengurangi risiko spurious correlation.
- g. Analisis spektral yang lebih baik: Analisis spektral, seperti analisis spektral daya, lebih efektif pada data yang stasioner karena fokus pada distribusi frekuensi yang konstan sepanjang waktu.
- h. Peningkatan performa model machine learning: Algoritma machine learning, terutama yang memanfaatkan konsep temporal, seperti recurrent neural networks (RNN) atau long short-term memory (LSTM), cenderung memberikan hasil yang lebih baik pada data yang stasioner.

Dalam memastikan stasioneritas data, dilakukan uji statistik seperti Augmented Dickey-Fuller (ADF) atau uji Kwiatkowski-Phillips-Schmidt-Shin (KPSS) pada data time series. Jika data tidak stasioner, transformasi seperti diferensiasi dapat diterapkan untuk mencapai stasioneritas.

a. Uji Augmented Dickey-Fuller (ADF)

Uji ADF digunakan untuk menguji apakah suatu rangkaian waktu memiliki akar unit (unit root), yang menunjukkan ketidakstasioneritas. Akar unit dapat ditemukan pada suatu rangkaian waktu yang memiliki tren atau pola musiman yang menyebabkan fluktuasi berkepanjangan. Uji ADF membandingkan perbandingan antara dua hipotesis:

- 1) Hipotesis Nol ( $H_0$ ): Rangkaian waktu memiliki akar unit, yang berarti tidak stasioner.
- 2) Hipotesis Alternatif ( $H_1$ ): Rangkaian waktu tidak memiliki akar unit, yang berarti stasioner.

Langkah-langkah uji ADF melibatkan regresi suatu rangkaian waktu pada dirinya sendiri dan menguji keberadaan akar unit. Hasil uji ini menghasilkan nilai p-nilai yang digunakan untuk menentukan apakah dapat menolak hipotesis nol atau tidak.

b. Uji Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

Uji KPSS, sebaliknya, digunakan untuk menguji hipotesis terkait dengan sifat stasioneritas suatu rangkaian waktu. Uji ini membandingkan dua hipotesis:

- 1) Hipotesis Nol ( $H_0$ ): Rangkaian waktu stasioner di sekitar suatu trend.
- 2) Hipotesis Alternatif ( $H_1$ ): Rangkaian waktu memiliki akar unit, yang menunjukkan tidak stasioner.

Perbandingan antara ADF dan KPSS:

- a. Jika kedua uji ADF dan KPSS menunjukkan stasioneritas, maka rangkaian waktu tersebut dapat dianggap stasioner.
- b. Jika uji ADF menunjukkan ketidakstasioneran dan uji KPSS menunjukkan stasioneritas, mungkin perlu dilakukan diferensiasi atau transformasi data untuk mencapai stasioneritas.

Interpretasi hasil:

- a. **ADF Statistic:** Nilai ini digunakan untuk membandingkan dengan nilai-nilai ambang batas kritis. Jika nilai ADF Statistik lebih kecil dari nilai-nilai ambang batas kritis, maka dapat menolak hipotesis nol (hipotesis non-stasioneritas) dan menyimpulkan bahwa deret waktu tersebut stasioner.
- b. **p-value:** P-value yang rendah menunjukkan kemungkinan kecil bahwa deret waktu tersebut non-stasioner. Umumnya, jika  $p\text{-value} < 0.05$ , kita dapat menolak hipotesis nol.
- c. **Critical value:** Nilai-nilai ini digunakan untuk menentukan ambang batas kritis. Jika nilai ADF Statistik lebih rendah dari nilai kritis yang sesuai pada tingkat kepercayaan tertentu, maka kita dapat menolak hipotesis nol.

### 2.3.9. Diferensiasi Data

Diferensiasi data yakni suatu proses di mana nilai-nilai dalam suatu rangkaian waktu diubah menjadi selisih antara nilai saat ini dan nilai sebelumnya. Tujuan dari diferensiasi adalah untuk menghilangkan tren atau pola musiman dari suatu rangkaian waktu, sehingga dapat membuat data menjadi lebih stasioner.

Stasioneritas memudahkan dalam pemodelan dan memungkinkan interpretasi yang lebih baik tentang perilaku data. Diferensiasi dinyatakan seperti persamaan (1):

$$Y'_t = Y_t - Y_{t-1} \dots\dots\dots(1)$$

Dimana :

$Y'_t$  : Nilai diferensiasi pada waktu t

$Y_t$  : Nilai pada waktu t

$Y_{t-1}$  : Nilai pada waktu sebelumnya

Proses tersebut dapat diulang jika diperlukan untuk mencapai stasioneritas yang diinginkan.

### 2.3.10. Persamaan Garis

Trend linear merupakan tren yang mempunyai variabel X (periode waktu) berpangkat paling tinggi satu. Rumus persamaan garis dari tren linier/garis lurus dapat dilihat pada persamaan (2) berikut:

$$Y' = a + bX \dots\dots\dots(2)$$

Dimana :

$Y$  : Nilai ramalan periode ke t

$a$  : *Intersept*

$b$  : *Slope* dari garis kecenderungan, merupakan tingkat perubahan

$X$  : Indeks waktu ( $t = 1, 2, 3, \dots, n$ ). n adalah banyaknya periode waktu

Persamaan (2) ini digunakan pada least square, semi average dan Regresi Linear sederhana untuk mendapatkan nilai prediksi.

Namun pada algoritma *Quadratic* ini mempunyai persamaan yang berbeda dalam menghitung nilai hasil peramalannya dikarenakan line yang akan terbentuk nantinya seperti parabola, seperti pada persamaan (3) :

$$Y' = a + bX + cX^2 \dots\dots\dots(3)$$

Dimana :

$Y$ : Nilai asli	$X$ : Unit waktu
$a$ : Konstanta	$b$ : Variabelitas per X
$c$ : Variabelitas per X kuadrat	$Y'$ : Nilai peramalan
	$n$ : Banyak data

### 2.3.11. Least Square

Least Square / kuadrat terkecil merupakan algoritma yang penentuan persamaan trend datanya mencakup analisis time series dengan dua kasus yakni data genap dan ganjil (Putra Manurung, 2015).

Langkah perhitungannya :

- 1) Mencari nilai tengah dari data :
  - a) Jika data ganjil maka data tengahnya akan dilakukan perindeksan X sebagai 0, sehingga X yang akan terbentuk ...., -3, -2, -1, 0, 1, 2, 3, ...
  - b) Jika data genap maka perindeksan X nya menjadi ...., -5, -3, -1, 1, 3, 5, ...
- 2) Setelah itu lakukan perhitungan XY yakni perkalian dari data riil dan unit waktu
- 3) Selanjutnya lakukan perhitungan kuadrat dari perindeks X atau  $X^2$ .

Persamaan (4) dan (5) merupakan rumus dalam pencarian konstanta a dan variabelitas b dalam Algoritma *Least Square* :

$$a = \frac{\sum Y}{n} \dots\dots\dots(4)$$

$$b = \frac{\sum XY}{\sum X^2} \dots\dots\dots(5)$$

Dimana :

$Y$  : Nilai asli

$X$  : Unit waktu

$a$  : Konstanta

$b$  : Variabelitas per  $X$

$n$  : Banyak data

### 2.3.12. Semi Average

Semi Average / setengah rata-rata merupakan penentuan model trend dengan cara membagi sekelompok data menjadi dua bagian yang sama (Ruslan & Banjarbaru, 2016). Semi average ini bermaksud untuk menentukan model trend dengan cara membagi sekelompok data menjadi 2 (dua) bagian yang sama. Apabila jumlah datanya ganjil, maka data yang berada ditengah dapat dihilangkan atau dapat dihitung 2 kali.

Langkah perhitungannya :

- 1) Mengelompokkan data menjadi 2 bagian (K1 dan K2). Jika jumlah datanya ganjil maka nilai yang berada di tengah dapat dihilangkan. Jika jumlah datanya genap maka nilai yang berada di tengahnya akan dimasukkan ke masing-masing kelompok. Namun perhitungan  $X$  nya hanya akan berlaku di satu kelompok saja.
- 2) Menghitung rata-rata dari masing kelompok K1 maupun K2.

Persamaan (6) dan (7) merupakan rumus dalam pencarian konstanta  $a$  dan variabelitas  $b$  dalam Algoritma *Semi Average* :



$$a = \frac{\sum K2 \text{ atau } K1}{n \text{ dari } K2 \text{ atau } K1} \dots\dots\dots(6)$$

$$b = \frac{K2 - K1}{\text{tahun dasar } K2 - \text{tahun dasar } K1} \dots\dots\dots(7)$$

Dimana :

$K1$  : Kelompok data 1

$n$  : Banyak data

$K2$  : Kelompok data 2

### 2.3.13. Quadratic

Quadratic / Kuadratik / Parabolik merupakan trend yang nilai variabel tak bebasnya naik atau turun secara linier atau berbentuk parabola bila datanya dibuat scatter plot. (Zikri et al., 2020)

Langkah perhitungannya :

- 1) Mencari nilai tengah data.
  - a) Jika data ganjil maka data tengahnya akan dilakukan perindexkan X sebagai 0, sehingga X yang akan terbentuk ..., -3, -2, -1, 0, 1, 2, 3, ...
  - b) Jika data genap maka perindexkan X nya menjadi ..., -5, -3, -1, 1, 3, 5, ...
- 2) Setelah itu lakukan perhitungan XY yakni perkalian dari data riil (Y) dan perindex X.
- 3) Selanjutnya lakukan perhitungan kuadrat dari perindex X atau  $X^2$ .
- 4) Hasil dari langkah 3 dikalikan dengan data riil (Y), untuk menjadi nilai  $X^2Y$ .
- 5) Lakukan perhitungan perindex X dikuadratkan 4 untuk menjadi  $X^4$ .

Persamaan (8), (9) dan (10) merupakan rumus dalam pencarian konstanta a, variabelitas b dan c variabelitas per X kuadrat, dalam Algoritma *Quadratic* :

$$a = \frac{(\sum Y \sum X^4) - (\sum X^2 Y \sum X^2)}{(n \sum X^4) - (\sum X^2)^2} \dots\dots\dots(8)$$

$$b = \frac{\sum XY}{\sum X^2} \dots\dots\dots(9)$$

$$c = \frac{(n \sum X^2 Y) - (\sum X^2 \sum Y)}{(n \sum X^4) - (\sum X^2)^2} \dots\dots\dots(10)$$

Dimana :

$Y$  : Nilai asli

$X$  : Unit waktu

$a$  : Konstanta

$b$  : Variabelitas per  $X$

$n$  : Banyak data

$c$  : Variabelitas per  $X$  kuadrat

#### 2.3.14. Regresi Linear

Trend Linear / Regresi Linear merupakan teknik peramalan yang dalam perhitungannya mempunyai unsur statistik dan matematis tertentu dengan bertujuan untuk menentukan garis lurus yang dibentuk berdasarkan data historis perusahaan / organisasi dan lain-lain. (Candra et al., 2018)

Regresi Linear Sederhana adalah algoritma statistik yang berfungsi untuk menguji sejauh mana hubungan sebab akibat antar variabel faktor penyebab ( $X$ ) terhadap variabel akibatnya. Faktor penyebab pada umumnya dilambangkan dengan  $x$  atau disebut juga dengan prediktor, sedangkan variabel akibat dilambangkan dengan  $y$  atau disebut juga dengan respon.

Analisis regresi bersifat asimetri atau dua arah. Teknik regresi membuat prediksi nilai dengan nilai yang ada pada satu variabel (yang disebut variabel independen) pada variabel lain, yang disebut variabel dependen. Dalam hal ini tujuannya bukan bermaksud membuat prediksi yang sempurna. Dengan informasi

pada independen bermaksud membuat prediksi nilai variabel dependen dengan error yang sekecil-kecilnya. Proposisi yang digunakan pada analisis regresi yaitu dari variabel independen X dan variabel dependen Y dinyatakan dengan regresi Y pada X.

Dalam regresi, variabel yang diprediksi disebut kriterium dan variabel yang digunakan untuk memprediksi disebut prediktor. Persamaan yang menyatakan hubungan antara variabel kriterium dan variabel prediktor disebut persamaan regresi.

Langkah perhitungannya :

- 1) Perindexkan dimulai dari 0, 1, 2, 3,... berlaku untuk data genap maupun ganjil
- 2) Setelah itu lakukan perhitungan XY yakni perkalian dari data riil (Y) dan perindex X.
- 3) Selanjutnya lakukan perhitungan kuadrat dari perindex X atau  $X^2$ .

Persamaan (11) dan (12) merupakan rumus dalam pencarian konstanta a dan variabelitas b dalam Algoritma *Regresi Linear* :

$$\sum Y = a.n + b.\sum X \dots\dots\dots(11)$$

$$\sum XY = a.\sum X + b.\sum X^2 \dots\dots\dots(12)$$

Dari persamaan (11) dan (12) kita lakukan penyederhanaan dengan memanfaatkan sistem substitusi dan eliminasi. Proses substitusi dan eliminasi dari persamaan (11) dan (12) ini akan menghasilkan persamaan yang sering dicantumkan di penelitian-penelitian yang lain sehingga jika kita tidak memperhatikan secara seksama, kita bisa menganggapnya sebagai persamaan yang

berbeda padahal tidak. Berikut proses eliminasi dan substitusi dari persamaan (11) dan (12):

$$\begin{array}{r|l} \sum XY = a \cdot \sum X + b \cdot \sum X^2 & *n \\ \hline \sum Y = a \cdot n + b \cdot \sum X & * \sum X \end{array}$$

$$\begin{array}{r} n \cdot \sum XY = a \cdot n \cdot \sum X + b \cdot n \cdot \sum X^2 \\ \hline \sum X \cdot \sum Y = a \cdot n \cdot \sum X + b \cdot \sum X^2 \end{array}$$

$$\begin{array}{r} n \cdot \sum XY = b \cdot n \cdot \sum X^2 \\ \hline \sum X \cdot \sum Y = b \cdot \sum X^2 \end{array}$$

$$(b \cdot n \cdot \sum X^2) - (b \cdot \sum X^2) = (n \cdot \sum XY) - (\sum X \cdot \sum Y)$$

$$b((n \cdot \sum X^2) - (\sum X^2)) = (n \cdot \sum XY) - (\sum X \cdot \sum Y)$$

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2}$$

Selanjutnya penyederhanaan untuk mendapat nilai dari konstanta a, prosesnya seperti di bawah ini:

$$\sum Y = a \cdot n + b \cdot \sum X$$

$$\sum Y - b \cdot \sum X = a \cdot n$$

$$a \cdot n = \sum Y - b \cdot \sum X$$

$$a = \frac{\sum Y - b \sum X}{n}$$

Sehingga didapat persamaan seperti persamaan (13) dan (14) di bawah ini:

$$a = \frac{\sum Y - b \sum X}{n} \dots \dots \dots (13)$$

$$b = \frac{n(\sum XY) - (\sum X)(\sum Y)}{n(\sum X^2) - (\sum X)^2} \dots \dots \dots (14)$$

Dimana :

$Y$  : Variabel terikat/variabel yang diramalkan  
 $X$  : Variabel bebas/variabel waktu

$a$  : Konstanta/intersep  
 $b$  : konstanta regresi/slope

$n$  : Banyak data

### 2.3.15. Logika Fuzzy

*Fuzzy Logic* merupakan perluasan dari banyaknya nilai logika di dalam arti dari pembentukan menjadi sistem yang banyak nilai logika. Dalam logika fuzzy dikenal juga dengan sistem kendali logika fuzzy disebut sebagai sistem inferensi fuzzy atau *fuzzy inference engine*, yakni sistem yang dapat melakukan penalaran dengan prinsip serupa seperti manusia melakukan penalaran dengan nalurnya. Sistem kendali logika *fuzzy* terdiri dari beberapa tahapan yakni Input Data, Fuzzyfikasi, Penalaran yang berisi Aturan dasar, lalu Defuzzyfikasi dan terakhir yakni Output (Roza et al., 2022).

#### 1) Fuzzyfikasi

*Fuzzyfikasi* merupakan pemetaan nilai input yang merupakan nilai tegas ke dalam fungsi keanggotaan himpunan *Fuzzy*, untuk kemudian diolah di dalam mesin penalaran.

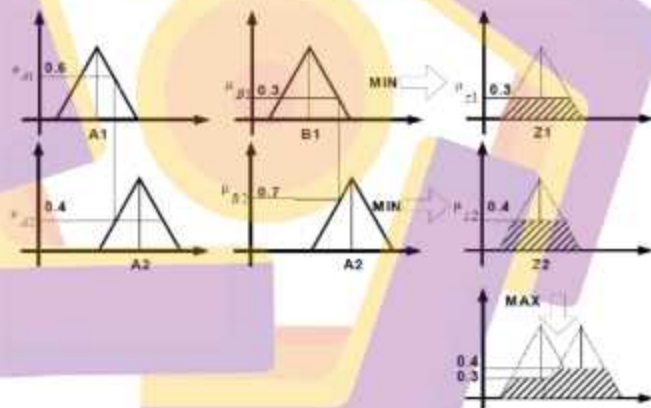
#### 2) Aturan Dasar

Aturan Dasar dalam kendali logika *Fuzzy* merupakan aturan implikasi dalam bentuk “jika ... maka ...”. Aturan dasar tersebut ditentukan dengan bantuan seorang pakar yang mengetahui karakteristik objek yang akan dikendalikan. Misalnya bentuk implikasi yang digunakan adalah sebagai berikut :

Jika  $X = A$  dan  $Y = B$  maka  $Z = C$

### 3) Penalaran

Pada proses penalaran ini sistem menalar ilai masukan untuk menentukan nilai keluaran sebagai bentuk pengambil keputusan. Sistem terdiri dari beberapa aturan, maka kesimpulan diperoleh dari kumpulan dan korelasi antar aturan. Terdapat 3 algoritma yang digunakan pada inferensi sistem *Fuzzy*, yaitu max, additive dan probabilistik OR. Pada algoritma max, solusi himpunan *Fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerah *Fuzzy* dan mengaplikasikannya ke output dengan menggunakan operator OR (union). Proses penalaran max-min dijelaskan dalam grafik berikut pada Gambar 2.7:



Gambar 2. 7. Proses Penalaran Min-Max

### 4) Defuzzyfikasi

*Defuzzyfikasi* merupakan kebalikan dari *fuzzyfikasi* yakni pemetaan dari himpunan *fuzzy* ke himpunan tegas. Inputan dari proses *defuzzyfikasi* adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*. Hasil dari *defuzzyfikasi* ini merupakan output dari sistem kendali logika *fuzzy*.

### 2.3.16. Fuzzy Mamdani

Algoritma *Fuzzy Mamdani* atau algoritma MIN-MAX, yang dimana menggunakan MIN pada fungsi implikasi dan MAX pada komposisi antar fungsi implikasi yang diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. *Fuzzy Inference System Mamdani* ini saat melakukan evaluasi aturan dalam mesin inferensi, menggunakan fungsi MIN dan komposisi antar rule menggunakan fungsi MAX untuk menghasilkan himpunan *fuzzy* baru (Ayuningtias et al., 2017). Dalam proses *defuzzyfikasi* pada algoritma ini menggunakan algoritma *Centroid* dengan menggunakan rumus pada persamaan (15);

$$Z = \frac{\int \mu(x) \cdot x \, dx}{\int \mu(x) \, dx} \dots \dots \dots (15)$$

### 2.3.17. Long Short-Term Memory (LSTM)

Salah satu metode *deep learning* yang pertama kali dikembangkan oleh Hochreiter dan Schmidhuber (1997) yang memiliki struktur kompleks yang di dalamnya memiliki lapisan tersembunyi. LSTM merupakan teknik turunan dari model *Recurrent Neural Network (RNN)*.

Dalam membuat model yang baik ada beberapa yang harus dipertimbangkan dalam pemrosesan LSTM yaitu, dengan menentukan jumlah *hidden layer* dan menentukan *hyperparameter* input yang sesuai. Arsitektur LSTM berisikan tiga lapisan yaitu, lapisan input, lapisan tersembunyi, dan lapisan output.

Tahapan-tahapan yang dilalui dalam proses model LSTM ditunjukkan pada persamaan di bawah :

c) Forget gate ( $f_t$ )

Forget gate memiliki fungsi untuk menentukan data yang akan dihilangkan dan menentukan waktu latensi yang optimal untuk input selanjutnya. Berikut adalah fungsi aktivasi sigmoid yang ditunjukkan pada persamaan (16):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \dots\dots\dots(16)$$

Dimana :

$f_t$ : forget gate	$\sigma$ : fungsi aktivasi sigmoid
$W_f$ : bobot forget gate	$h_{t-1}$ : nilai hidden state cell sebelum
$x_t$ : nilai input	$b_f$ : bias foget gate

d) Input gate ( $i_t$ )

Input gate memiliki fungsi untuk membawa titik masuk data dari luar dan mengolah data yang masuk. Berikut adalah dua fungsi aktivasi yang ditunjukkan pada persamaan (17) dan (18).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \dots\dots\dots(17)$$

$$C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \dots\dots\dots(18)$$

Dimana :

$i_t$ : input gate	$\sigma$ : fungsi aktivasi sigmoid
$W_i$ : bobot input gate	$h_{t-1}$ : nilai hidden state cell sebelum
$x_t$ : nilai input	$b_i$ : bias input gate
$C_t$ : candidate gate	$W_c$ : bobot candidate gate
$\tanh$ : fungsi aktivasi tanh	$b_c$ : bias candidate gate



e) Cell state ( $c_t$ )

Cell state memiliki fungsi untuk menggantikan nilai sebelumnya pada memory cell dengan nilai memory cell yang baru. Berikut adalah fungsi pada cell state yang ditunjukkan pada persamaan (19):

$$c_t = (i_t * C_t) + (f_t * C_{t-1}) \dots\dots\dots(19)$$

Dimana :

$c_t$ : cell gate	$i_t$ : input gate
$f_t$ : forget gate	$c_{t-1}$ : nilai cell state sebelum
$C_t$ : candidate gate	

f) Output gate ( $o_t$ )

Output gate untuk melakukan seluruh proses input dan output perhitungan atau output dalam sel LSTM. Berikut adalah fungsi pada output gate yang ditunjukkan pada persamaan (20) dan (21):

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \dots\dots\dots(20)$$

$$h_t = o_t * \tanh(c_t) \dots\dots\dots(21)$$

Dimana :

$o_t$ : output gate	$\sigma$ : fungsi aktivasi sigmoid
$W_o$ : bobot input gate	$h_{t-1}$ : nilai hidden state cell sebelum
$x_t$ : nilai input	$b_o$ : bias input gate
$h_t$ : hidden state	$c_t$ : cell gate

$\tanh$  : fungsi aktivasi tanh

### 2.3.18. Min-Max Normalisasi

Normalisasi data adalah proses transformasi nilai suatu data dalam rentang nilai tertentu. Salah satu metode yang dapat digunakan untuk melakukan normalisasi data adalah Min-Max Normalization. Min-Max Normalization adalah metode normalisasi sederhana yang melakukan transformasi linear terhadap data aktual sehingga dapat menghasilkan keseimbangan nilai antara data sebelum dan sesudah diproses (Selle et al., 2022). Persamaan (22) yang digunakan sebagai berikut:

$$d' = \frac{d - \min}{\max - \min} \dots \dots \dots (22)$$

Dimana :

$d'$  : hasil transformasi suatu nilai

$d$  : nilai data aktual yang akan dinormalisasi

$\min$  : nilai terkecil pada dataset fitur x

$\max$  : nilai terbesar pada dataset fitur x

### 2.3.19. Denormalisasi

Proses denormalisasi adalah proses membangkitkan kembali nilai yang telah dinormalisasi ke bentuk nilai aktual (Budiprasetyo et al., 2023). Proses denormalisasi dilakukan dengan menggunakan persamaan (23) sebagai berikut:

$$d = d'(\max - \min) + \min \dots \dots \dots (23)$$

Dimana :

$d$  : nilai asli setelah didenormalisasi

$d'$  : nilai hasil prediksi yang dinormalisasi

$min$  : nilai terkecil pada dataset fitur x

$max$  : nilai terbesar pada dataset fitur x

### 2.3.20. Pengujian Performa

#### 1) Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) digunakan sebagai perbedaan antara nilai yang diprediksi dengan nilai sebenarnya. Semakin tinggi nilai RMSE yang dihasilkan maka tingkat keakuratan yang didapatkan semakin rendah. Jika semakin rendah nilai RMSE yang dihasilkan maka tingkat keakuratan semakin tinggi (Roosaputri & Dewi, 2023). Adapun rumus untuk menghitung RMSE yang dapat dilihat pada persamaan (24):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^l (y_i - y'_i)^2} \dots \dots \dots (24)$$

Dimana :

$y_i$  : Nilai asli

$y'_i$  : Nilai hasil prediksi

$n$  : banyaknya data

$i$  : (1,2,..,l), l merupakan keseluruhan data

#### 2) RSquare

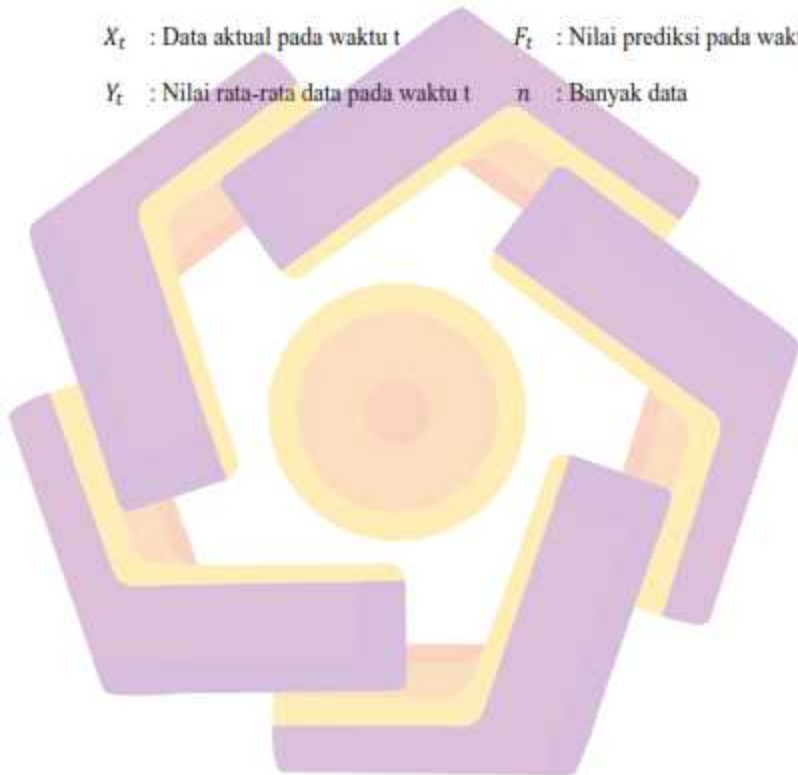
RSquare mengukur seberapa cocok garis prediksi dengan data yang ada. RSquare dapat diartikan sebagai persentase varian dalam variabel dependen yang dapat di jelaskan oleh model dan digunakan untuk mengukur seberapa jauh kemampuan model dalam menerangkan variasi variabel terikat. Dimana nilai

RSquare mendekati nilai 1 itu menandakan bahwa performa yang didapatkan sangat baik. RSquare dapat didefinisi-kan sebagai persamaan (25) berikut:

$$R^2 = 1 - \frac{\sum_{t=1}^n (X_t - F_t)^2}{\sum_{t=1}^n (Y_t - \bar{Y})^2} \dots\dots\dots(25)$$

Dimana :

- $X_t$  : Data aktual pada waktu t       $F_t$  : Nilai prediksi pada waktu t  
 $Y_t$  : Nilai rata-rata data pada waktu t       $n$  : Banyak data



## **BAB III**

### **METODE PENELITIAN**

#### **3.1. Jenis, Sifat, dan Pendekatan Penelitian**

Pada penelitian ini menggunakan jenis penelitian eksperimen yakni melakukan pengujian performa serta memprediksi jumlah pendaftar mahasiswa baru dengan menggunakan Algoritma Least Square, Semi Average, Quadratic, Long Short Term Memory (LSTM), Regresi Linear, dan Fuzzy Mamdani.

Penelitian ini dilakukan secara mandiri menggunakan sifat deskriptif dimana data yang diperoleh akan dibandingkan dengan data yang diuji. Penelitian ini menggunakan pendekatan kuantitatif.

Data yang digunakan pada penelitian ini merupakan data pendaftar mahasiswa baru dari tahun akademik 2014/2015 sampai 2022/2023. Dimana data tersebut akan diolah untuk dapat memprediksi jumlah pendaftar mahasiswa baru serta untuk mendapatkan algoritma mana yang terbaik.

#### **3.2. Metode Pengumpulan Data**

Pada pengumpulan data ini, peneliti memperoleh data mahasiswa baru dari akun jalur PMB Nasional yakni jalur SPAN-PTKIN dan UM-PTKIN. Serta jalur PMB lokal yang diadakan oleh kampus sendiri yakni Mandiri Reguler. Variabel yang akan digunakan adalah Bulan, jumlah calon mahasiswa baru yang mendaftar, jumlah calon mahasiswa yang dinyatakan lulus seleksi dan jumlah calon mahasiswa yang melakukan registrasi pada Bulan tersebut.

### 3.3. Metode Analisis Data

Analisa data pada penelitian ini menggunakan enam algoritma yang nantinya akan dilakukan perbandingan hasil pengujian performancenya pada prediksi jumlah pendaftar dengan menggunakan data test, untuk mengetahui algoritma peramalan mana yang terbaik untuk jenis data pada penelitian ini.

- a. Kinerja Algoritma Least Square, Semi Average, Quadratic, Long Short Term Memory (LSTM), Regresi Linear, dan Fuzzy Mamdani pada prediksi jumlah pendaftar mahasiswa baru:

- 1) Menginisialisasi nilai data riil (tahun, total pendaftar, total lulus seleksi, total registrasi).
- 2) Membagi data menjadi data training dan data testing.
- 3) Menghitung banyak data training untuk mendapat informasi bahwa data tersebut berjumlah genap atau ganjil.
- 4) Melakukan proses pembentukan model prediksi :

a) Least Square dan Quadratic

- Jika datanya ganjil maka data tengahnya akan dilakukan perindexkan  $X$  sebagai 0, sehingga  $X$  yang akan terbentuk ..., -3, -2, -1, 0, 1, 2, 3, ...

Jika datanya genap maka perindexkan  $X$  nya akan terbentuk ..., -5, -3, -1, 1, 3, 5, ...

- Lalu lakukan perhitungan  $X$ ,  $X^2$ ,  $X^4$ ,  $Y^2$ ,  $XY$ ,  $X^2Y$ ,  $XY^2$ ,  $\ln Y$ ,  $X^2 \ln Y$ .
- Substitusikan ke persamaan (4), (5), (8), (9) dan (10) untuk mendapatkan nilai  $a$ ,  $b$  maupun  $c$ .

b) Semi Average

Data dibagi menjadi 2 kelompok terlebih dahulu lalu masing-masing kelompok dicari nilai tengahnya untuk menjadi tahun dasar yang nilai  $X$  nya 0 (nol). Masing-masing kelompok dihitung rata-ratanya untuk menjadi nilai  $a$  seperti pada persamaan (6) dan mendapatkan nilai  $b$  berdasarkan berdasarkan persamaan (7).

c) Regresi Linear Sederhana

- Lalu lakukan perhitungan  $X^2$ ,  $XY$ .
- Substitusikan ke persamaan (13), (14) untuk mendapatkan nilai  $a$  dan  $b$ .

d) Fuzzy Mamdani

- Melakukan proses Fuzzyfikasi
- Penentuan Aturan Dasar
- Mensubstitusi ke penalaran berdasarkan aturan dasar
- Melakukan proses Defuzzyfikasi
- Substitusikan ke persamaan (15).

e) Long Short-term Memory (LSTM)

- Dilakukan proses min-max scaler pada data
- Menentukan banyak inputan data
- Menentukan banyak hidden layer yang digunakan
- Pembentukan model prediksi

- 5) Hitung prediksi total pendaftar tahun 2022 dengan menggunakan model prediksi yang sudah terbentuk.

- 6) Didapat nilai  $Y$  (riil) dan  $Y'$  (hasil peramalan dari masing-masing algoritma).
  - 7) Melakukan perhitungan performa masing-masing algoritma dengan mensubstitusikan ke persamaan (24) dan (25).
  - 8) Bandingkan nilai dari hasil performa (RMSE dan RSquare) untuk mendapatkan algoritma terbaik dalam kasus di penelitian ini berdasarkan jenis data yang digunakan.
- b. Faktor yang mempengaruhi kinerja Algoritma Least Square, Semi Average, Quadratic, Long Short Term Memory (LSTM), Regresi Linear, dan Fuzzy Mamdani

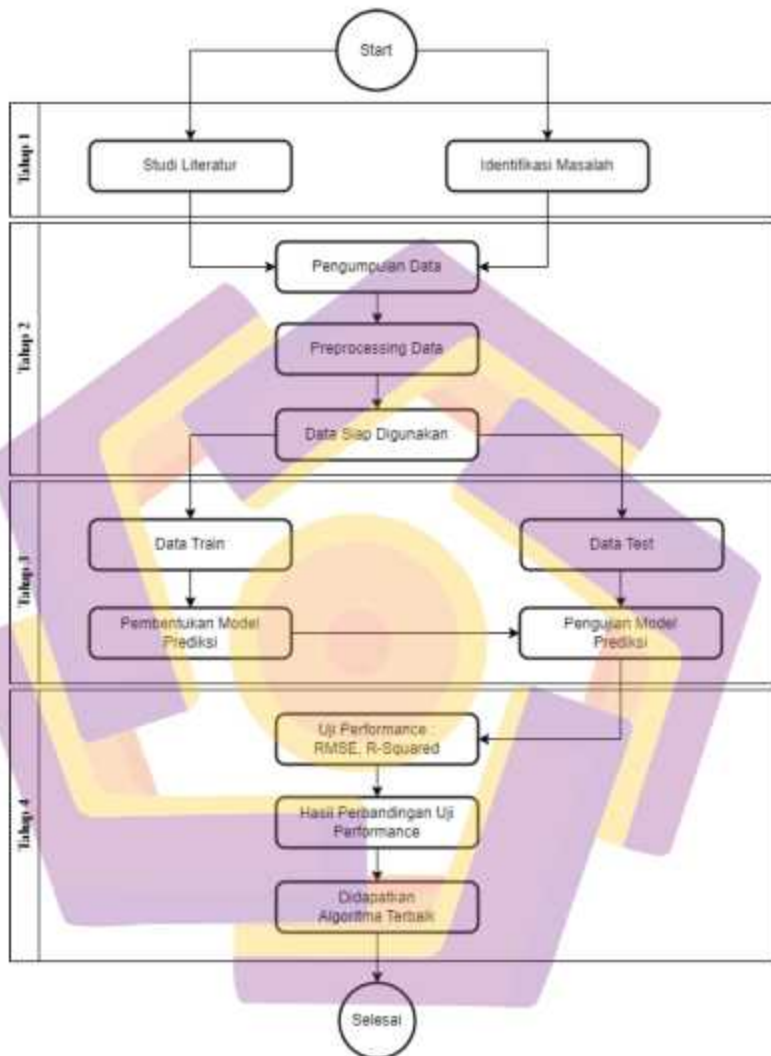
Faktor yang mempengaruhi kinerja Algoritma tersebut akan dilihat dari:

- 1) Penggunaan data yang berupa runtun waktu,
- 2) Data stasioner tidaknya,
- 3) Kecenderungan data aktual,
- 4) Kecenderungan data hasil peramalan.

### 3.4. Alur Penelitian

Berikut merupakan alur penelitian yang dilakukan pada penelitian ini yang digambarkan melalui flowchart pada Gambar 3.1.





Gambar 3.1. Alur Penelitian

Berikut penjelasan mengenai alur penelitian pada Gambar 3.1:

#### 1. Tahap 1

Mengidentifikasi masalah di tempat penelitian untuk diangkat menjadi tujuan penelitian dengan menggunakan data yang tersedia. Selain itu, juga diimbangi dengan mencari dan mempelajari studi literatur yang berkaitan dengan algoritma yang digunakan pada penelitian ini guna mengetahui sebaik apa algoritma tersebut di penelitian-penelitian sebelumnya ketika digunakan.

#### 2. Tahap 2

Pengumpulan data dari akun resmi yang dimiliki oleh IAIN Madura dari tahun 2014 sampai tahun 2022. Data yang digunakan merupakan data pendaftar pada pelaksanaan penerimaan mahasiswa baru. Dari data yang diambil tersebut dipreprocessing dengan di cek terlebih dahulu terutama pada bagian data pendaftar yang diselenggarakan lokal. Dikarenakan para pendaftar sering melakukan pendaftaran akun lebih dari satu kali hal ini terjadi karena sistem penerimaan mahasiswa baru milik IAIN Madura masih belum terkoneksi dengan data NISN pusat. Data yang double dihapus dan yang tidak memiliki notes juga dihapus. Selanjutnya data tersebut dikelompokkan/ditotalkan berdasarkan bulan pembuatan akun pendaftarannya pada tiap tahun pelaksanaan. Serta, dibentuk dalam 12 bulan tanpa memandang bulan tersebut terdapat pendaftaran tidaknya. Setelah itu dicek data tersebut stasioner tidaknya melalui uji stasioner, apabila tidak stasioner maka perlu dilakukan diferensiasi data. Dari hasil proses diferensiasi data tersebut dicek kembali data tersebut

sudah stasioner tidaknya. Apabila sudah masuk dalam kategori data stasioner maka data tersebut sudah siap digunakan untuk masuk ke tahap selanjutnya.

### 3. Tahap 3

Data yang sudah siap digunakan selanjutnya dibagi menjadi dua (2) bagian yakni data training dan data testing. Dengan skema data tahun terakhir digunakan sebagai data test, lalu sisanya digunakan sebagai data train. Data train ini yang akan digunakan dalam proses pembentukan model prediksi serta untuk melatih model prediksi tersebut.

Ketika model prediksi sudah selesai terbentuk, maka model tersebut akan di test keakuratannya menggunakan data test yang ada untuk menghasilkan nilai prediksi.

### 4. Tahap 4

Dari hasil prediksi yang didapat selanjutnya melakukan perbandingan dengan data test aslinya dengan menggunakan evaluasi performance menggunakan RSquare dan RMSE. Dari nilai RSquare mendekati nilai 1 dan RMSE yang mendekati nilai 0 maka model prediksi tersebut semakin akurat prediksinya.

## BAB IV

### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1. Pengumpulan Data

Prediksi total pendaftar mahasiswa baru pada penelitian ini menggunakan beberapa algoritma untuk dikomparasi yakni Least Square, Semi Average, Quadratic, Long Short Term Memory (LSTM), Regresi Linear, dan Fuzzy Mamdani. Namun sebelum masuk pada pemodelan prediksi berdasarkan algoritma tersebut, harus dilakukan preprocessing terlebih dahulu untuk dapat diolah.

Data pertama yang dikumpulkan melalui akun resmi PMB IAIN Madura yakni <https://pmb.iainmadura.ac.id/> untuk jalur Mandiri Reguler, <https://span-ptkin.ac.id/> untuk jalur SPAN-PTKIN dan <https://um.ptkin.ac.id/> untuk jalur UM-PTKIN. Data yang digunakan merupakan data para calon mahasiswa baru dari tahun 2014 sampai 2022 dengan menggunakan 3 jalur yakni SPAN-PTKIN, UM-PTKIN dan Mandiri Reguler. Berikut rincian total data pendaftar PMB per jalur ditampilkan dalam bentuk tabel oleh Tabel 4.1 di bawah ini:

Tabel 4. 1. Total Data PMB Perjalur

Tahun Akademik	SPAN-PTKIN	UM-PTKIN	Mandiri Reguler
2014/2015	563	827	468
2015/2016	1596	1668	1041
2016/2017	1866	1649	1039
2017/2018	1887	2125	1384
2018/2019	1746	1665	1962
2019/2020	3049	2086	1879
2020/2021	2386	1776	995
2021/2022	2085	1198	773
2022/2023	1743	1141	479

Tabel 4.1 merupakan tampilan rincian total data pendaftar pada akun PMB yang terekam sebanyak 41.076 record data, terhitung dari tahun 2014 sampai tahun 2022 dengan kondisi menggunakan tiga jalur (SPAN-PTKIN, UM-PTKIN dan Mandiri Reguler).

## **4.2. Preprocessing Data**

Pada tahap preprocessing data time series ini, ada beberapa tahapan yang perlu dilakukan untuk membuat data tersebut dapat digunakan. Tahapan tersebut antara lain pembersihan data, melakukan decompose data, menganalisis hasil ACF dan PACF. Selanjutnya mengecek data tersebut masuk jenis data stasioner tidaknya, apabila data tersebut tidak stasioner maka perlu dilakukan diferensiasi data untuk mengubahnya menjadi data stasioner.

### **4.2.1. Pembersihan Data**

Tahapan selanjutnya yakni preprocessing data yang dimana pada tahap ini dilakukan pembersihan data. Data yang akan dihapus merupakan data yang double. Hal ini dilakukan karena pada jalur lokal/Mandiri Reguler yang digelar para siswa diharuskan membuat akun dulu setelah itu melakukan pembayaran pendaftaran berdasarkan kode bayar yang didapat. Setelah melakukan pembayaran berdasarkan kode bayar, siswa tersebut dapat melakukan pengisian data yakni pemilihan prodi yang dituju, upload foto dan upload ijazah. Sedangkan mahasiswa yang mengalami masalah dalam pembayaran, mereka cenderung melakukan pembuatan akun kembali sehingga memiliki kode bayar yang berbeda dengan identitas yang sama. Maka dari itu, pembersihan data perlu dilakukan untuk menghapus data tersebut.

Agar lebih mudah membandingkan dari dan sampai proses preprocessing selesai sehingga dapat digunakan, maka kami menampilkannya dalam bentuk total peserta PMB pada tiap jalur masuk pertahunnya. Tabel 4.2 merupakan total data PMB yang belum dilakukan preprocessing:

Tabel 4. 2. Total Data PMB Sebelum Preprocessing

Tahun Akademik	SPAN-PTKIN	UM-PTKIN	Mandiri Reguler	Total
2014/2015	563	827	468	1858
2015/2016	1596	1668	1041	4305
2016/2017	1866	1649	1039	4554
2017/2018	1887	2125	1384	5396
2018/2019	1746	1665	1962	5373
2019/2020	3049	2086	1879	7014
2020/2021	2386	1776	995	5157
2021/2022	2085	1198	773	4056
2022/2023	1743	1141	479	3363

Data yang didapat dari penerimaan PMB sebelum preprocessing sebanyak 41.076 record data terhitung dari tahun 2014/2022. Dari data tersebut menghasilkan Tabel 4.3 yang merupakan total data yang dihapus dan hanya terdapat pada jalur Mandiri Reguler dikarenakan jalur SPAN-PTKIN dan UM-PTKIN sudah terkoneksi pada data Kemdikbud sehingga siswa tersebut hanya bisa melakukan pendaftaran sekali.

Tabel 4. 3. Total Data PMB yang Dihapus

Tahun Akademik	SPAN-PTKIN	UM-PTKIN	Mandiri Reguler
2014/2015	0	0	0
2015/2016	0	0	0
2016/2017	0	0	0
2017/2018	0	0	0
2018/2019	0	0	728
2019/2020	0	0	591
2020/2021	0	0	394
2021/2022	0	0	399
2022/2023	0	0	215

Sebanyak 2.327 record data yang dihapus, selanjutnya dilakukan penjumlahan pendaftar berdasarkan tahun pelaksanaan PMB-nya masing-masing seperti pada Tabel 4.4:

Tabel 4. 4. Data PMB Setelah Preprocessing

Tahun	SPAN-PTKIN	UM-PTKIN	Mandiri Reguler	Total
2014	563	827	468	1858
2015	1596	1668	1041	4305
2016	1866	1649	1039	4554
2017	1887	2125	1384	5396
2018	1746	1665	1234	4645
2019	3049	2086	1288	6423
2020	2386	1776	601	4763
2021	2085	1198	374	3657
2022	1743	1141	264	3148

Sebanyak 38.749 record data setelah mengalami preprocessing, sehingga data pendaftar, lulus dan registrasi yang digunakan sudah siap digunakan untuk diproses ke masing-masing algoritma. Tabel 4.5 merupakan data yang sudah siap digunakan untuk diproses ke tahap pembentukan model prediksi pada masing-masing algoritma:

Tabel 4. 5. Data PMB Per Tahun Pelaksanaan

TAHUN	PENDAFTAR	LULUS	REGISTER
2014	1858	1618	1418
2015	4305	2195	1795
2016	4554	2476	1906
2017	5396	2188	1989
2018	4645	2598	2183
2019	6423	2581	2187
2020	4763	2417	2007
2021	3657	2174	1739
2022	3148	2021	1624

Selanjutnya, dari data tersebut dikelompokkan lagi berdasarkan bulan pembuatan akun pendaftarannya, sehingga didapat didapatkan data seperti pada Tabel 4.6 untuk rentang waktu 12 bulan setahun:

Tabel 4. 6. Data PMB Siap Digunakan

Date	BULAN	TAHUN	PENDAFTAR	LULUS	REGISTER
2014-01-01	01	2014	0	0	0
2014-02-01	02	2014	313	278	193
2014-03-01	03	2014	250	218	158
2014-04-01	04	2014	718	684	655
2014-05-01	05	2014	292	248	235
2014-06-01	06	2014	184	141	130
2014-07-01	07	2014	101	49	47
...	...	...	...	...	...
2022-07-01	07	2022	115	115	82
2022-08-01	08	2022	0	0	0
2022-09-01	09	2022	0	0	0
2022-10-01	10	2022	0	0	0
2022-11-01	11	2022	0	0	0
2022-12-01	12	2022	0	0	0

Tabel 4.6 menampilkan 7 data teratas dan 6 data terbawah. Total data yang digunakan pada Tabel 4.6 sebanyak 108 row data dengan 4 kolom yakni Date, Bulan, Tahun, Pendaftar, Lulus dan Register.

#### 4.2.2. Decompose Data

Tabel 4.6 selanjutnya masuk ke tahapan decompose data, dimana pada proses ini data tersebut akan diuraikan berdasarkan komponen-komponen yang mengandung dalam data tersebut seperti pola tren, musiman dan residu. Pada proses penguraian ini memerlukan library:

```
from statsmodels.tsa.seasonal import seasonal_decompose
```

Selanjutnya, membaca dataset yang dimana harus diatur bahwa kolom Date dijadikan index dan menjadi frekuensi yang teratur. Berikut source codenya:



```
df = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/THESIS/BULANAN_THESIS/DATA_BULANAN/DATA_LSTM.csv', in
dex_col='Date', parse_dates=True)
df.index.freq='MS')

results = seasonal_decompose(df['PENDAFTAR'])
results.plot();
```

seasonal\_decompose, source code tersebut memanggil fungsi untuk menguraikan data Pendaftar ke dalam bentuk 3 pola yakni tren, musiman dan residu lalu ditampilkan dengan fungsi plot() hasilnya seperti pada Gambar 4.1:



Gambar 4. 1. Hasil Decompose Data Asli

Gambar 4.1 menunjukkan hasil komponen-komponen pola trend, seasonal dan residu yang terdapat pada data asli, data asli ditampilkan pada grafik paling atas. Dapat kita lihat pada grafik line trend, 2015-2019 cukup stabil mengalami peningkatan pendaftar. Namun di tahun 2020 sampai 2022 mengalami tren yang menurun. Sedangkan pada grafik line musiman yang terjadi hampir sama di setiap tahunnya yang dimana jumlah pendaftar tinggi pada bulan februari, april dan juni.

Grafik residu merupakan sisa dari penguraian data asli yang diuraikan menjadi data trend dan musiman.

#### 4.2.3. ACF dan PCF Data

Selanjutnya melakukan analisis dengan menggunakan ACF dan PACF pada data tersebut dengan menggunakan source code:

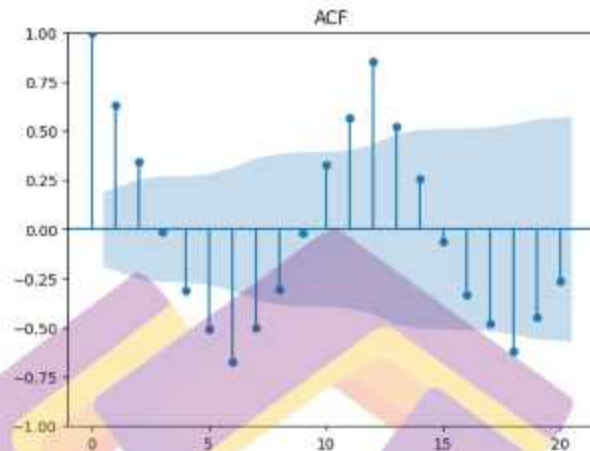
```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
plt.figure(figsize=(12, 8))
plot_acf(data, lags=20, title='ACF')
plt.show()

plt.figure(figsize=(12, 8))
plot_pacf(data, lags=20, title='PACF')
plt.show()
```

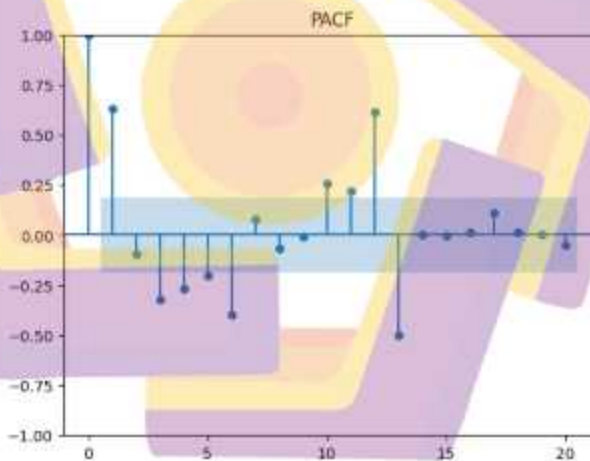
Pada baris `from statsmodels.graphics.tsaplots import plot_acf, plot_pacf` mengimpor fungsi untuk menggambar plot oto-korelasi dan oto-korelasi parsial. Plot ini berguna untuk memvisualisasikan struktur korelasi dalam deret waktu. `plt.figure(figsize=(12, 8))` membuat sebuah figur dengan ukuran 12x8 inci. `plot_acf(data, lags=20, title='ACF')` menggambar plot fungsi oto-korelasi (ACF) dari data. Seperti sebelumnya, `lags=20` berarti hanya menghitung oto-korelasi parsial hingga lag ke-20. `title='ACF'` memberikan judul untuk plot tersebut. `plt.show()` menampilkan plot ACF yang telah digambar.

Pada Gambar 4.2 dan Gambar 4.3 terdapat beberapa komponen penting, antara lain:

- a. Batas horizontal: Garis biru pucat di area bayangan menunjukkan batas kepercayaan 95%. Nilai oto-korelasi di luar area bayangan ini dianggap signifikan.
- b. Batas vertical (lags): Pada sumbu x (horizontal), garis vertikal menunjukkan nilai-nilai lag. Pada sumbu y (vertikal), nilai-nilai oto-korelasi terkait dengan lags tersebut.
- c. Interpretasi: Lag yang melewati batas bayangan dengan signifikan dapat menjadi indikasi pola oto-korelasi yang penting. Untuk ACF jika terdapat pola siklus di sekitar nilai-nilai lag tertentu, ini dapat menjadi petunjuk untuk pemilihan model. Sedangkan untuk PACF membantu mengidentifikasi "efek langsung" dari suatu lag tanpa melibatkan lag-lag di antara keduanya. Sebagai contoh, jika ada puncak di lag 3, itu berarti korelasi parsial antara observasi saat ini dan observasi yang terjadi pada lag 3 signifikan.



Gambar 4. 2. Hasil ACF Data Asli



Gambar 4. 3. Hasil PACF Data Asli

Pada Gambar 4.2 dan Gambar 4.3 dapat diperhatikan pada lag 1 yang signifikan di masing-masing grafik ACF maupun PACF. Namun pada lag 2 hanya pada grafik ACF yang masih signifikan walaupun nilainya lebih rendah daripada lag 1. Lalu terjadi lagi pada lag 11, 12, dan 13 pada grafik ACF sedangkan pada

grafik PACF signifikan pada lag 10, 11, 12. Sehingga dapat disimpulkan terjadi pola musiman di data tersebut.

#### 4.2.4. Uji Stasioneritas

Selanjutnya pengecekan stasioner tidaknya dengan menggunakan metode ADF (Augmented Dickey-Fuller ) berikut source codenya:

```
from statsmodels.tsa.stattools import adfuller

# Uji stasioneritas (ADF Test)
def adf_test(timeseries):
    result = adfuller(timeseries, autolag='AIC')
    print("ADF Statistic:", result[0])
    print("p-value:", result[1])
    print("Critical Values:")
    for key, value in result[4].items():
        print(f"{key}: {value}")

print("Uji Stasioneritas:")
adf_test(data)
```

`adf_test` adalah fungsi yang mengambil deret waktu sebagai argumen dan menghitung uji ADF untuk menilai stasioneritasnya. `adfuller(timeseries, autolag='AIC')` adalah fungsi utama yang melakukan uji ADF. Parameter `autolag='AIC'` mengatur penggunaan kriteria informasi Akaike (AIC) untuk menentukan jumlah lag yang digunakan dalam regresi. Mencetak hasil uji stasioneritas untuk suatu deret waktu (`data`). Hasil yang ditampilkan mencakup nilai ADF Statistik, nilai p-value, dan nilai-nilai ambang batas kritis. Dari source code tersebut menghasilkan nilai pengujian berikut:

```
Uji Stasioneritas:
ADF Statistic: -2.5087398050999705
p-value: 0.11336229186262187
Critical Values:
1%: -3.5011373281819504
5%: -2.8924800524857854
10%: -2.5832749307479226
```

Berdasarkan hasil tersebut Nilai ADF Statistik (-2.5087398050999705) lebih besar daripada nilai kritis pada tingkat signifikansi 1%, 5%, dan 10%. Oleh karena itu, tidak dapat menolak hipotesis nol bahwa deret waktu tersebut non-stasioner. Nilai p-value (0.11336229186262187) lebih besar dari tingkat signifikansi yang umumnya digunakan (0,05). Ini juga mendukung kesimpulan bahwa kita tidak dapat menolak hipotesis nol. Sehingga disimpulkan hasil uji ADF tersebut, tidak dapat menolak hipotesis nol bahwa deret waktu tersebut non-stasioner. P-value yang relatif tinggi dan nilai ADF Statistik yang lebih besar daripada nilai-nilai kritis pada tingkat signifikansi yang umumnya digunakan menunjukkan bahwa deret waktu mungkin tidak stasioner. Oleh karena itu, kemungkinan besar, deret waktu tersebut belum stasioner dan perlu diberlakukan transformasi atau diferensiasi untuk mencapai stasioneritas.

#### 4.2.5. Diferensiasi Data

Berdasarkan hasil uji ADF data tersebut tidak stasioner maka perlu dilakukan Langkah diferensiasi dengan menggunakan source code berikut:

```
data['differenced_data'] = data.diff()
data.head(13)
```

`data.diff()` fungsi ini melakukan diferensiasi pada deret waktu `data`.

Diferensiasi pada suatu deret waktu menghasilkan deret waktu baru yang terdiri dari selisih antara setiap nilai dan nilai sebelumnya. `data['differenced_data']` menetapkan hasil diferensiasi ke dalam kolom baru dengan nama `differenced_data` di dalam objek DataFrame `data`. `data.head(13)` untuk menampilkan data sebanyak 13 baris teratas dari `data`.

PENDAFTAR		differenced_data
Date		
2014-01-01	0	NaN
2014-02-01	313	313.0
2014-03-01	250	-63.0
2014-04-01	718	468.0
2014-05-01	292	-426.0
2014-06-01	184	-108.0
2014-07-01	101	-83.0
2014-08-01	0	-101.0
2014-09-01	0	0.0
2014-10-01	0	0.0
2014-11-01	0	0.0
2014-12-01	0	0.0
2015-01-01	0	0.0

Gambar 4. 4. Hasil Diferensiasi Data

Gambar 4.4 merupakan hasil dari proses diferensiasi, dapat diperhatikan pada baris pertama nilai di kolom `differenced_data` terdapat nilai NaN sehingga kita perlu melakukan penghapusan baris yang bernilai NaN dengan perintah source code berikut:

```
data_asli_diff = data.dropna()
```

`dropna()` perintah untuk menghapus baris yang mengandung nilai NaN lalu menyimpannya ke dalam dataframe baru yang bernama `data_asli_diff`. Dari data yang sudah di diferensiasi tersebut selanjutnya di cek kembali data tersebut sudah stasioner atau tidak, apabila tidak stasioner maka perlu dilakukan diferensiasi kedua. Berikut Gambar 4.5 yang merupakan 15 baris pertama dari hasil data diferensiasi setelah baris yang bernilai NaN dihapus:

PENDAFTAR differenced_data		
Date		
2014-02-01	313	313.0
2014-03-01	250	-63.0
2014-04-01	718	468.0
2014-05-01	292	-426.0
2014-06-01	184	-108.0
2014-07-01	101	-83.0
2014-08-01	0	-101.0
2014-09-01	0	0.0
2014-10-01	0	0.0
2014-11-01	0	0.0
2014-12-01	0	0.0
2015-01-01	0	0.0
2015-02-01	858	858.0

Gambar 4. 5. Hasil Diferensiasi Data Tanpa Nilai NaN

Berikut hasil pengecekan uji stioner tidaknya diperoleh hasil:

```
Uji Stasioneritas:
ADF Statistic: -4.25392340786855
p-value: 0.0005332989854118362
Critical Values:
1%: -3.5011373281819504
5%: -2.8924800524857854
10%: -2.5832749307479226
```

Nilai ADF Statistik adalah  $-4.25392340786855$ . Nilai ini lebih kecil dari nilai kritis pada tingkat signifikansi 1%, 5%, dan 10%. Oleh karena itu, kita dapat menolak hipotesis nol bahwa deret waktu tersebut stasioner. Nilai p-value adalah  $0.0005332989854118362$ . Karena nilai p-value lebih kecil dari tingkat signifikansi yang umumnya digunakan (0.05), kita dapat menolak hipotesis nol. Maka dapat disimpulkan berdasarkan hasil uji ADF, kita dapat menyimpulkan bahwa deret waktu tersebut stasioner. Ini didukung oleh nilai ADF Statistik yang lebih kecil

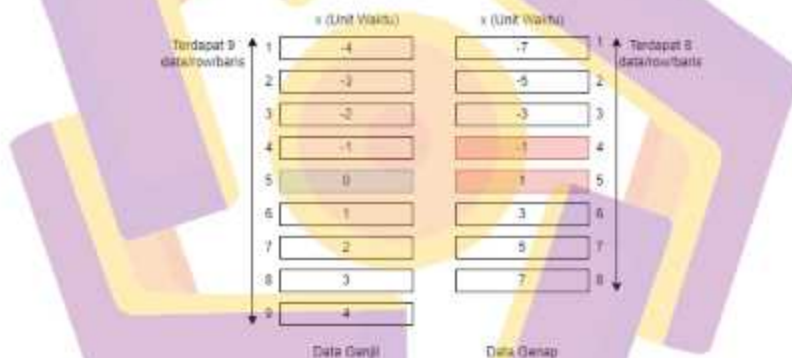


daripada nilai-nilai kritis pada tingkat signifikansi yang umumnya digunakan, dan nilai p-value yang rendah. Oleh karena itu, dapat diasumsikan bahwa deret waktu tersebut sudah stasioner.

### 4.3. Least Square

#### 4.3.1. Pembentukan Model Prediksi

Sebelum pembentukan model harus melalui proses identifikasi kebutuhan untuk substitusikan ke persamaan (4) dan (5). Menentukan  $x$  (unit waktu) di tiap tahunnya terlebih dahulu, dengan ketentuan seperti pada Gambar 4.6:



Gambar 4. 6. Penentuan Unit Waktu (Least Square)

Penjelasan mengenai Gambar 4.6 dalam penentuan X (Unit Waktu) :

#### 1. Kategori Ganjil / Data Ganjil

Misal terdapat data sebanyak 9 data / baris / row, dibagi menjadi 2 bagian sehingga di dapat 4,5. Maka data dari baris 1 – 4 merupakan baris yang nilai X (Unit Waktu) kurang dari 0. Baris ke 5 merupakan nilai X (Unit Waktu) yang bernilai 0 karena berada ditengah. Selanjutnya, baris ke 6 – 9 mendapat

nilai X (Unit Waktu) lebih dari 0. Jika diurutkan dari data ke 1 – ke 9 maka nilai X (Unit Waktu) nya yakni -4,-3,-2,-1,0,1,2,3,4.

## 2. Kategori Genap / Data Genap

Misal terdapat data sebanyak 8 data / baris / row, dibagi menjadi 2 bagian sehingga di dapat 4. Maka nilai X (Unit Waktu) yang nilainya 0 berada di antara data baris ke 4 dan 5. Sehingga, baris ke 4 – 1 akan memiliki nilai X (Unit Waktu) kurang dari 0. Sedangkan baris ke 5 – 8 akan memiliki nilai X (Unit Waktu) lebih dari 0. Jika diurutkan dari data ke 1 – ke 8 maka nilai X (Unit Waktu) nya yakni -7,-5,-3,-1,1,3,5,7.

Berikut source code yang digunakan untuk melakukan penentuan nilai x (unit waktu) tersebut:

```
if (n % 2 == 0): #jika datanya genap
    pertama = (n*2)-1
    kedua = n-pertama
    ketiga = n-1
    x_biasa = np.linspace(kedua, ketiga, n)
    kondisi = 'GENAP'
else:
    nilai_b = n-1
    ketiga = nilai_b/2
    kedua = ketiga *(-1)
    x_biasa = np.linspace(kedua, ketiga, n)
    kondisi = 'GANJIL'
train['X'] = x_biasa
```

`np.linspace(kedua, ketiga, n)` untuk menampilkan deret angka dengan kondisi deret sebanyak n, dimulai dari kedua dan diakhiri oleh nilai ketiga. Setelah ditentukan nilai x / unit waktunya selanjutnya melakukan perhitungan berdasarkan kebutuhan untuk mendapat nilai a dan b dengan menggunakan source code di bawah ini:

```

train['XY'] = (train['X'])*(train['PENDAFTAR'])
train['Y'] = (train['PENDAFTAR'])
train['X_KUADRAT'] = (train['X'])**2
jumlah_xy = sum(train['XY'])
jumlah_x_kuadrat = sum(train['X_KUADRAT'])
jumlah_y = sum(train['Y'])

```

Agar lebih mudah, hasil dari source code kami pindah ke dalam bentuk

Tabel 4.7:

Tabel 4. 7. Hasil Identifikasi Least Square

Date	Pendaftar	Y	X	XY	X <sup>2</sup>
2014-02-01	313	313	-47	-14711	2209
2014-03-01	-63	-63	-46	2898	2116
...	...	...	...	...	...
2017-11-01	0	0	-2	0	4
2017-12-01	0	0	-1	0	1
2018-01-01	0	0	0	0	0
2018-02-01	973	973	1	973	1
...	...	...	...	...	...
2021-10-01	0	0	45	0	2025
2021-11-01	0	0	46	0	2116
2021-12-01	0	0	47	0	2209

Selanjutnya dilakukan perhitungan untuk mendapat nilai a konstanta dan b variabelitas per x menggunakan source code di bawah ini:

```

b = jumlah_xy / jumlah_x_kuadrat
a = jumlah_y / n

```

Hasil yang didapat yakni a = 0,0 dan b = -0,494 dengan hitungan manual seperti di bawah ini:

$$a = \frac{\sum Y}{n} = \frac{0,0}{95} = 0,0$$

$$b = \frac{\sum XY}{\sum X^2} = \frac{-35297,0}{71440,0} = -0,494$$

Dari hasil yang diperoleh selanjutnya membuat model prediksi, model prediksi yang didapatkan dengan menggunakan source code:

```
n = len(train)
banyak_train = n-1
x_terakhir = train.loc[banyak_train]
banyak_test = len(test)
if (kondisi=='GENAP'):
    awal = x_terakhir['X']+2
    akhir = (banyak_test*2)+x_terakhir['X']
    test['X'] = np.linspace(awal, akhir, banyak_test)
else :
    awal = x_terakhir['X']+1
    akhir = banyak_test+x_terakhir['X']
    test['X'] = np.linspace(awal, akhir, banyak_test)
Y_prediksinya = (b*(test['X'])) + a
test['PREDIKSI'] = Y_prediksinya
```

`train.loc[banyak_train]` memfilter data train pada index terakhir. `if (kondisi=='GENAP')`: untuk membaca pesan kondisi pada proses penentuan unit waktu di data train 'GENAP' atau `else : Ganjil`. `awal = x_terakhir['X'] +1` penentuan deret pertama untuk data test dengan menambahkan 1 jika `kondisi=='GENAP'` atau 2 jika `kondisi=='GANJIL'`. Penentuan deret terakhir untuk data test dengan mengalikan 2 (`banyak_test*2`) lalu di tambakan `x_terakhir['X']` jika `kondisi=='GENAP'` dan cukup menambahkan `banyak_test` dan `x_terakhir['X']` jika `kondisi=='GANJIL'`. `np.linspace(awal, akhir, banyak_test)` dengan menggunakan source code tersebut, akan membantu dalam pembentukan X/unit waktu dengan rentang yang sudah ditentukan. Lalu menyimpan hasilnya ke kolom baru `test['X']` di data `test`.

Berikut contoh perhitungan nilai prediksi dari *Least Square* untuk Date 2022-01-01 yang dimana  $X$  atau unit waktu untuk Date tersebut adalah 48 berdasarkan urutan  $X$  dari data terakhir di data train. Selanjutnya disubstitusikan ke persamaan (1) seperti di bawah ini :

$$\begin{aligned} Y' &= a + bX \\ &= 0,0 + (-0,494 \cdot (48)) \\ &= 0,0 + (-23,7157) \\ &= -23,7157 \end{aligned}$$

-23,7157 merupakan hasil nilai prediksi jumlah pendaftar untuk Date 2022-01-01.

#### 4.3.2. Hasil Prediksi

Selanjutnya merupakan source code untuk menghitung nilai prediksi dengan menggunakan data test:

```
Y_prediksinya = (b*(test['X'])) + a
test['PREDIKSI'] = Y_prediksinya
```

Dari `Y_prediksinya` yang disimpan pada kolom baru `test['PREDIKSI']` dengan nama kolom `'PREDIKSI'`, karena hasil prediksi tersebut merupakan hasil pengolahan diferensiasi data maka pada hasil akhir prediksi perlu kita dediferensiasi dengan data aslinya dengan menggunakan source code:

```
test['PREDIKSI_FIX'] =
test['ASLI'].shift(1) + test['PREDIKSI']
```

Hasil prediksi pendaftar disimpan di kolom baru `test['PREDIKSI_FIX']` pada DataFrame `test` dengan nama kolom `'PREDIKSI_FIX'`.

Hasil pada source code tersebut kami pindahkan dalam bentuk tabel agar mudah pembacaanya, hasil prediksi seperti pada Tabel 4.8:

Tabel 4. 8. Data Aktual dan Prediksi (Least Square)

Tahun Akademik	Data Aktual	Data Prediksi
2022-01-01	0	0,000
2022-02-01	961	-24,210
2022-03-01	782	936,296
2022-04-01	605	756,802
2022-05-01	536	579,308
2022-06-01	149	509,814
2022-07-01	115	122,320
2022-08-01	0	87,826
2022-09-01	0	-27,668
2022-10-01	0	-28,163
2022-11-01	0	-28,657
2022-12-01	0	-29,151

Data pada Tabel 4.8 disandingkan dengan data aktual untuk menjadi pembandingan antara data aktual dan data prediksi yang diperoleh.

#### 4.3.3. Uji Performance

Selanjutnya pengujian performance dari hasil Tabel 4.8 menghasilkan Tabel 4.9 untuk melihat hasil selisih antara data aktual dan data prediksi:

Tabel 4. 9. Selisih Antara Data Aktual dan Prediksi Least Square

Tahun Akademik	Data Aktual	Data Prediksi	Selisih
2022-01-01	0	0,000	0,000
2022-02-01	961	-24,210	985,210
2022-03-01	782	936,296	154,296
2022-04-01	605	756,802	151,802
2022-05-01	536	579,308	43,308
2022-06-01	149	509,814	360,814
2022-07-01	115	122,320	7,320
2022-08-01	0	87,826	87,826
2022-09-01	0	-27,668	27,668
2022-10-01	0	-28,163	28,163
2022-11-01	0	-28,657	28,657
2022-12-01	0	-29,151	29,151

Pada python sudah disediakan library untuk melakukan pengujian performance RMSE dan RSquare dengan memanfaatkan library dari `from sklearn.metrics`. Sedangkan untuk mendapatkan nilai RMSE maka diperlukan library dari matematika yakni `from math` dengan memanfaatkan fungsi `sqrt`, maka source code lengkapnya seperti di bawah ini:

```
from sklearn.metrics import mean_squared_error, RSquare_score
from math import sqrt
print("RSQUARE: %.2f" % RSquare_score(test['PENDAFTAR'],
test['PREDIKSI_FIX']))
print("RMSE: %.2f" %
sqrt(mean_squared_error(test['PENDAFTAR'],
test['PREDIKSI_FIX'])))
```

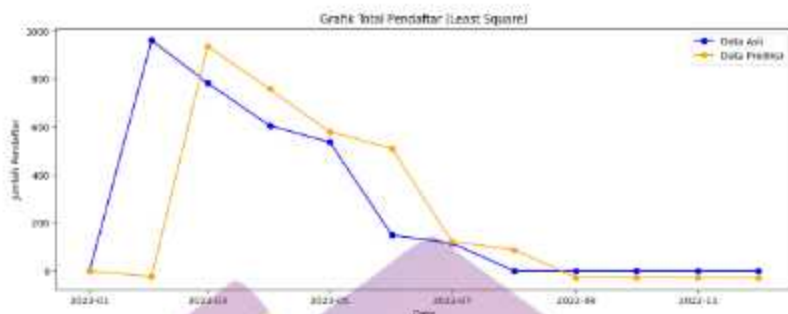
Hasil dari source code pengujian performance di atas yakni di bawah ini:

```
RSQUARE: 0.17
RMSE: 310.99
```

Berdasarkan hasil dari uji performance, RSquare sebesar 0.17 menunjukkan bahwa model hanya mampu menjelaskan sekitar 17% dari variasi dalam data jumlah pendaftar mahasiswa baru. Artinya, sebagian besar variasi tidak dapat dijelaskan oleh model, dan terdapat faktor-faktor lain yang mempengaruhi jumlah pendaftar. RMSE sebesar 310.99 menunjukkan rata-rata kesalahan prediksi model adalah sekitar 310.99 unit. Ini memberikan gambaran tentang seberapa akurat model dalam memprediksi jumlah pendaftar.

#### 4.3.4. Visualisasi Data Asli dan Data Prediksi

Data prediksi yang diperoleh selanjutnya di visualisasikan dalam bentuk grafik garis sehingga dapat terlihat perbedaan dan jarak garis dari data asli dan data prediksi seperti pada Gambar 4.7:

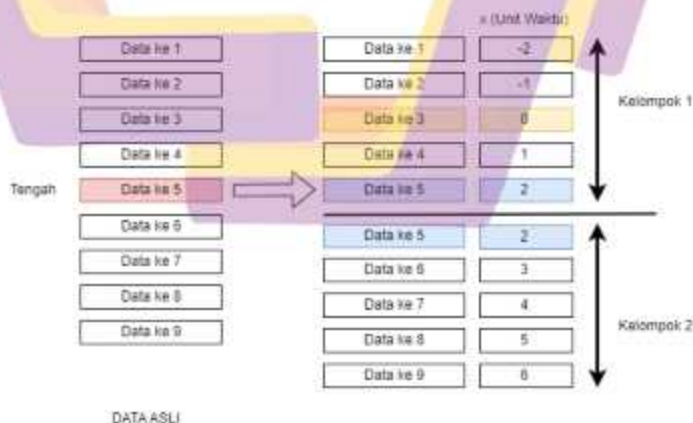


Gambar 4. 7. Grafik Total Pendaftar (Least Square)

#### 4.4. Semi Average

##### 4.4.1. Pembentukan Model Prediksi

Sebelum pembentukan model harus melalui proses identifikasi kebutuhan berdasarkan persamaan (6) dan (7), salah satunya yakni penentuan nilai  $x$  (unit waktu). Karena dalam penentuannya, mempunyai aturan sendiri seperti yang diilustrasikan oleh Gambar 4.8:



Gambar 4. 8. Penentuan Unit Waktu (Semi Average)



Penjelasan mengenai Gambar 4.8 dalam penentuan X (Unit Waktu) :

1. Data yang digunakan dicek ada berapa banyak, masuk ke data ganjil atau genap.
2. Pada Gambar 4.8 dicontohkan untuk data ganjil yang sebanyak 9 row. Kemudian dicari data tengahnya yakni data ke 5. Kemudian data ke 5 ini di kloning sebanyak 1 kali dan diletakkan dibawahnya sehingga data tersebut tetap berurutan.
3. Selanjutnya data yang ada terdapat 10 data dengan kondisi data ke 5 ada 2. Lalu 10 data tersebut dikelompokkan menjadi 2 bagian sama rata/sama banyak.
4. Dalam penentuan nilai x (unit waktu), mengecek kelompok 1 data tengahnya terdapat pada data ke berapa. Dalam kasus, kelompok 1 data tengahnya terdapat pada data ke 3. Maka unit waktu untuk data ke 3 bernilai 0, untuk data yang berada di atas data ke 3 seperti data ke-2 dan ke 1 bernilai negatif secara berurutan. Sedangkan data ke 4 sampai terakhir bernilai positif secara berurutan. Namun pada data ke 5 hanya berlaku 1 unit waktu saja.

Berikut source code yang digunakan untuk melakukan penentuan x/unit waktu tersebut:

```
# Ubah kolom tanggal di DataFrame A ke format datetime
train['TH_BU_TG'] = pd.to_datetime(train['TH_BU_TG'])

# Menghitung nilai rata-rata (konstanta) dari data historis
nilai_konstanta = train['PENDAFTAR'].mean()

# Menghitung simpangan baku (variabilitas) dari data historis
simpangan_baku = train['PENDAFTAR'].std()

# Tampilkan nilai konstanta dan variabilitas
print(f'Nilai Konstanta (Rata-rata)/a: {nilai_konstanta}')
print(f'Variabilitas (Simpangan Baku)/b: {simpangan_baku}')
```

`pd.to_datetime(train['TH_BU_TG'])` untuk menjadikan data pada kolom tersebut menjadi `datetime`. `train['PENDAFTAR'].mean()` menggunakan fungsi `mean()` dari Pandas, ini menghitung nilai rata-rata dari kolom 'PENDAFTAR' di DataFrame `train`. `simpangan_baku = train['PENDAFTAR'].std()` menggunakan fungsi `std()` dari Pandas, ini menghitung simpangan baku dari kolom 'PENDAFTAR' di DataFrame `train`. Nilai konstanta `a` yakni `0.0` dan nilai variabelitas `b` yakni `381.318`.

#### 4.4.2. Hasil Prediksi

Selanjutnya merupakan source code untuk menghitung nilai prediksi dengan menggunakan data test:

```
# Ubah kolom tanggal prediksi ke format datetime
test['TH_BU_TG'] = pd.to_datetime(test['Date'])
# Hitung metode semi-average dari data historis di tahun yang sesuai
tahun_prediksi = test['Date'].iloc[0].year
data_sebelumnya = train[train['Date'].dt.year ==
tahun_prediksi - 1]
semi_average = data_sebelumnya['PENDAFTAR'].mean()
# Prediksi jumlah pendaftar pada tanggal yang diinginkan di
tahun yang diinginkan
test['PREDIKSI'] = semi_average
```

`iloc[0].year` mengambil nilai `year` data pada kolom `test['TH_BU_TG']` di baris index ke 0 `iloc[0]`. `test['PREDIKSI'] = semi_average` Menambahkan kolom 'PREDIKSI' ke DataFrame `test` dan mengisinya dengan nilai rata-rata yang telah dihitung sebelumnya. Ini dapat dianggap sebagai prediksi sederhana berbasis rata-rata dari data historis. Dari source code tersebut pada kolom baru `test['PREDIKSI']` dengan nama kolom 'PREDIKSI' menyimpan hasil prediksinya, karena hasil prediksi tersebut

merupakan hasil pengolahan diferensiasi data maka pada hasil akhir prediksi perlu kita dediferensiasi dengan data aslinya dengan menggunakan source code:

```
test['PREDIKSI_FIX'] =
test['ASLI'].shift(1) + test['PREDIKSI']
```

Hasil prediksi pendaftar disimpan di kolom baru `test['PREDIKSI_FIX']` pada DataFrame `test` dengan nama kolom `'PREDIKSI_FIX'`.

Hasil pada source code tersebut kami pindahkan dalam bentuk tabel agar mudah pembacaanya, hasil prediksi seperti pada Tabel 4.10:

Tabel 4. 10. Data Aktual dan Prediksi (Semi Average)

Tahun Akademik	Data Aktual	Data Prediksi
2022-01-01	0	0
2022-02-01	961	0
2022-03-01	782	961
2022-04-01	605	782
2022-05-01	536	605
2022-06-01	149	536
2022-07-01	115	149
2022-08-01	0	115
2022-09-01	0	0
2022-10-01	0	0
2022-11-01	0	0
2022-12-01	0	0

Data pada Tabel 4.10 disandingkan dengan data aktual untuk menjadi pembandingan antara data aktual dan data prediksi yang diperoleh.

#### 4.4.3. Uji Performance

Selanjutnya pengujian performance dari hasil Tabel 4.10 menghasilkan Tabel 4.11 untuk melihat hasil selisih antara data aktual dan data prediksi:

Tabel 4. 11. Selisih Antara Data Aktual dan Prediksi Semi Average

Tahun Akademik	Data Aktual	Data Prediksi	Selisih
2022-01-01	0	0	0
2022-02-01	961	0	961
2022-03-01	782	961	179
2022-04-01	605	782	177
2022-05-01	536	605	69
2022-06-01	149	536	387
2022-07-01	115	149	34
2022-08-01	0	115	115
2022-09-01	0	0	0
2022-10-01	0	0	0
2022-11-01	0	0	0
2022-12-01	0	0	0

Pada python sudah disediakan library untuk melakukan pengujian performance RMSE dan RSquare dengan memanfaatkan library dari `from sklearn.metrics`. Sedangkan untuk mendapatkan nilai RMSE maka diperlukan library dari matematika yakni `from math` dengan memanfaatkan fungsi `sqrt`, maka source code lengkapnya seperti di bawah ini:

```
from sklearn.metrics import mean_squared_error, RSquare_score
from math import sqrt
print("RSQUARE: %.2f" % RSquare_score(test['PENDAFTAR'],
test['PREDIKSI_FIX']))
print("RMSE: %.2f" %
sqrt(mean_squared_error(test['PENDAFTAR'],
test['PREDIKSI_FIX'])))
```

Hasil dari source code pengujian performance di atas yakni di bawah ini:

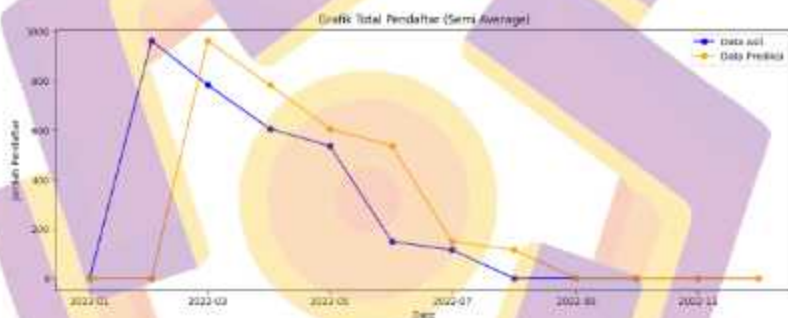
```
RSQUARE: 0.17
RMSE: 310,35
```

Berdasarkan hasil dari uji performance, Nilai RSquare sebesar 0.17 menunjukkan bahwa model menggunakan metode semi-average hanya mampu menjelaskan sekitar 17% dari variasi yang ada dalam data jumlah pendaftar

mahasiswa baru. Ini mengindikasikan bahwa model ini tidak secara efektif mewakili variasi yang signifikan dalam data. RMSE sebesar 310.35 menunjukkan rata-rata kesalahan prediksi model sekitar 310.35 unit. RMSE yang tinggi dapat mengindikasikan bahwa prediksi model memiliki kesalahan yang signifikan.

#### 4.4.4. Visualisasi Data Asli dan Data Prediksi

Data prediksi yang diperoleh selanjutnya di visualisasikan dalam bentuk grafik garis sehingga dapat terlihat perbedaan dan jarak garis data asli dan data prediksi seperti pada Gambar 4.9:



Gambar 4. 9. Grafik Total Pendaftar (Semi Average)

## 4.5. Quadratic

### 4.5.1. Pembentukan Model Prediksi

Sebelum pembentukan model harus melalui proses identifikasi kebutuhan berdasarkan persamaan (8), (9) dan (10). Penentuan x (unit waktu harus dilakukan, untuk ketentuan dalam penentuan x (unit waktu) aturannya sama dengan ketentuan penentuan x (unit waktu) milik least square seperti yang diilustrasikan oleh Gambar

4.6. Berikut source code yang digunakan untuk melakukan penentuan nilai x (unit waktu) tersebut:

```
n = len(dataset) #untuk cek panjang/banyaknya baris pada data
if (n % 2 == 0): #jika datanya genap
    pertama = (n*2)-1
    kedua = n-pertama
    ketiga = n-1
    x_biasa = np.linspace(kedua, ketiga, n) #-5,-3,-1,1,3,3
else:
    pertama = n-1
    ketiga = pertama /2
    kedua = ketiga*(-1)
    x_biasa = np.linspace(kedua, ketiga, n) #-2,-2,-1,0,1,2,3
train['X'] = x_biasa
```

`np.linspace(kedua, ketiga, n)` untuk menampilkan deret angka dengan kondisi deret sebanyak n, dimulai dari kedua dan diakhiri oleh nilai ketiga. Setelah ditentukan nilai x / unit waktunya selanjutnya menginputkannya ke kolom baru `train['X']` di DataFrame `train`. Selanjutnya melakukan perhitungan dengan menggunakan source code di bawah ini:

```
train['XY'] = (train['X'] )*(train['PENDAFTAR'] )
train['Y'] = (train['PENDAFTAR'] )
train['X_KUADRAT'] = (train['X'] )**2
train['X_KUADRAT_Y'] = ((train['X'] )**2)*(train['PENDAFTAR'] )
train['X_KUADRAT_EMPAT'] = (train['X'] )**4

jumlah_xy = sum(train['XY'])
jumlah_y = sum(train['Y'])
jumlah_x_kuadrat = sum(train['X_KUADRAT'])
jumlah_x_kuadrat_y = sum(train['X_KUADRAT_Y'])
jumlah_x_kuadrat_empat = sum(train['X_KUADRAT_EMPAT'])
```

`train['X']` merupakan kolom yang terdapat pada DataFrame `test` yang berisi unit waktu berdasarkan historis unit waktu dari DataFrame `train`. Dengan adanya kolom tersebut dapat lebih mudah dalam perhitungan untuk memperoleh

nilai konstanta a, variabelitas b dan variabelitas kuadrat c. `(train['X'] )**2` menandakan bahwa data pada kolom tersebut dikuadratkan. `sum(train['XY'])` merupakan fungsi untuk menjumlahkan semua DataFrame yang terdapat pada kolom 'XY'. Agar lebih mudah, hasil dari source code kami pindah ke dalam bentuk Tabel 4.12:

Tabel 4. 12. Hasil Identifikasi Quadratic

Date	Pendaftar	Y	X	XY	X <sup>2</sup>	X <sup>2</sup> Y	X <sup>4</sup>
2014-02-01	313	313	-47	-14711	2209	691417	4879681
2014-03-01	250	250	-46	2898	2116	-133308	4477456
...	...	...	...	...	...	...	...
2017-11-01	0	0	-2	0	4	0	16
2017-12-01	0	0	-1	0	1	0	1
2018-01-01	0	0	0	0	0	0	0
2018-02-01	973	973	1	973	1	973	1
...	...	...	...	...	...	...	...
2021-10-01	0	0	45	0	2025	0	4100625
2021-11-01	0	0	46	0	2116	0	4477456
2021-12-01	0	0	47	0	2209	0	4879681

Selanjutnya dilakukan perhitungan untuk mendapat nilai a konstanta, b variabelitas per x dan c variabelitas per x kuadrat menggunakan source code di bawah ini:

```

a1 = jumlah_y * jumlah_x_kuadrat_empat
a2 = jumlah_x_kuadrat_y * jumlah_x_kuadrat
a3 = n * jumlah_x_kuadrat_empat
a4 = (jumlah_x_kuadrat)**2
a = (a1 - a2)/(a3 - a4)

b = jumlah_xy / jumlah_x_kuadrat

c1 = n * jumlah_x_kuadrat_y
c2 = jumlah_x_kuadrat * jumlah_y
c3 = n * jumlah_x_kuadrat_empat
c4 = (jumlah_x_kuadrat)**2
c = (c1-c2)/(c3-c4)

```

Hasil yang didapat yakni  $a = 1.2675$ ,  $b = -0.4940$  dan  $c = -0.0016$ .

Berikut untuk lebih jelasnya dicontohkan untuk hitungan manualnya:

$$\begin{aligned}
 a &= \frac{(\sum y \cdot \sum x^*) - (\sum x^2 y \cdot \sum x^2)}{(n \cdot \sum x^*) - (\sum x^2)^2} \\
 &= \frac{0.0 - (-5173470480.0)}{9185255120.0 - 5103673600.0} \\
 &= 1.2675 \\
 b &= \frac{\sum xy}{\sum x^2} \\
 &= \frac{-35297.0}{71440.0} = -0.4940 \\
 c &= \frac{(n \sum x^2 y) - (\sum x^2 \sum y)}{(n \sum x^*) - (\sum x^2)^2} \\
 &= \frac{-6879615.0 - 0.0}{9185255120.0 - 5103673600.0} \\
 &= -0.0016
 \end{aligned}$$

Setelah didapatkan nilai *konstanta* a, *variabelitas* b dan *variabelitas* kuadrat c, selanjutnya melakukan perhitungan nilai prediksi jumlah pendaftar mahasiswa baru pada tiap Bulan dengan menggunakan source code:

```

banyak_train = n-1
x_terakhir = train.loc[banyak_train]
banyak_test = len(test)

if (kondisi=='GENAP'):
    awal = x_terakhir['X']+2
    akhir = (banyak_test*2)+x_terakhir['X']
    test['X'] = np.linspace(awal, akhir, banyak_test)
else :
    awal = x_terakhir['X']+1
    akhir = banyak_test+x_terakhir['X']
    test['X'] = np.linspace(awal, akhir, banyak_test)

Y_prediksinya = a + (b*x_fix) + (c*x_kuadrat)

```



`train.loc[banyak_train]` memfilter data train pada index terakhir. `if (kondisi=='GENAP')`: untuk membaca pesan kondisi pada proses penentuan unit waktu di data train 'GENAP' atau `else` : Ganjil. `awal = x_terakhir['X'] +1` penentuan deret pertama untuk data test dengan menambahkan 1 jika `kondisi=='GENAP'` atau 2 jika `kondisi=='GANJIL'`. Penentuan deret terakhir untuk data test dengan mengalikan 2 (`banyak_test*2`) lalu di tambakan `x_terakhir['X']` jika `kondisi=='GENAP'` dan cukup menambahkan `banyak_test` dan `x_terakhir['X']` jika `kondisi=='GANJIL'`. `np.linspace(awal, akhir, banyak_test)` dengan menggunakan source code tersebut, akan membantu dalam pembentukan  $X$ /unit waktu dengan rentang yang sudah ditentukan. Lalu menyimpan hasilnya ke kolom baru `test['X']` di data `test`.

Berikut contoh perhitungan nilai prediksi dari *Quadratic* untuk Date 2022-01-01 yang dimana  $X$  atau unit waktu untuk Date tersebut adalah 48 sedangkan  $X^2$ nya 2304 atau  $(48)^2$  berdasarkan hasil  $X$  dari Daterakhir di data train. Selanjutnya disubstitusikan ke persamaan (2) seperti di bawah ini :

$$\begin{aligned}
 Y' &= a + bX + cX^2 \\
 &= 1.2675 + (-0.4940 * (48)) + (-0.0016 * (48)^2) \\
 &= 472,40 + (-23,712) + (-3,6864) \\
 &= -23,3317
 \end{aligned}$$

-23,3317 merupakan hasil perhitungan prediksinya pada Date 2022-01-01.

Perhitungan nilai prediksi untuk Date selanjutnya menggunakan cara yang sama.

#### 4.5.2. Hasil Prediksi

Selanjutnya berikut source code untuk menghitung nilai prediksi dengan menggunakan data test dan memanfaatkan kolom `test['X']` yang dibuat di DataFrame test sebelumnya:

```
test['X_KUADRAT'] = test['X']**2
Y_prediksinya = a + (b*(test['X'])) + (c*(test['X']**2))
test['PREDIKSI'] = Y_prediksinya
```

Dari `Y_prediksinya` yang disimpan pada kolom baru `test['PREDIKSI']` dengan nama kolom 'PREDIKSI', karena hasil prediksi tersebut merupakan hasil pengolahan diferensiasi data maka pada hasil akhir prediksi perlu kita dediferensiasi dengan data aslinya dengan menggunakan source code:

```
test['PREDIKSI_FIX'] =
test['ASLI'].shift(1) + test['PREDIKSI']
```

Hasil prediksi pendaftar disimpan di kolom baru `test['PREDIKSI_FIX']` pada DataFrame test dengan nama kolom 'PREDIKSI\_FIX'.

Hasil pada source code tersebut kami pindahkan dalam bentuk tabel agar mudah pembacaanya, hasil prediksi seperti pada Tabel 4.13:

Tabel 4. 13. Data Aktual dan Prediksi (Quadratic)

Tahun Akademik	Data Aktual	Data Prediksi
2022-01-01	0	0,00
2022-02-01	961	-26,99
2022-03-01	782	933,35
2022-04-01	605	753,69
2022-05-01	536	576,02
2022-06-01	149	506,35
2022-07-01	115	118,67
2022-08-01	0	83,99
2022-09-01	0	-31,69

Tabel 4.13. (Lanjutan)

Tahun Akademik	Data Aktual	Data Prediksi
2022-10-01	0	-32,37
2022-11-01	0	-33,06
2022-12-01	0	-33,75

Data pada Tabel 4.13 disandingkan dengan data aktual untuk menjadi pembandingan antara data aktual dan data prediksi yang diperoleh.

#### 4.5.3. Uji Performance

Selanjutnya pengujian performance dari hasil Tabel 4.13 menghasilkan Tabel 4.14 untuk melihat hasil selisih antara data aktual dan data prediksi:

Tabel 4. 14. Selisih Antara Data Aktual dan Prediksi Quadratic

Tahun Akademik	Data Aktual	Data Prediksi	Selisih
2022-01-01	0	0,00	0,00
2022-02-01	961	-26,99	987,99
2022-03-01	782	933,35	151,35
2022-04-01	605	753,69	148,69
2022-05-01	536	576,02	40,02
2022-06-01	149	506,35	357,35
2022-07-01	115	118,67	3,67
2022-08-01	0	83,99	83,99
2022-09-01	0	-31,69	31,69
2022-10-01	0	-32,37	32,37
2022-11-01	0	-33,06	33,06
2022-12-01	0	-33,75	33,75

Pada python sudah disediakan library untuk melakukan pengujian performance RMSE dan RSquare dengan memanfaatkan library dari `from sklearn.metrics`. Sedangkan untuk mendapatkan nilai RMSE maka diperlukan library dari matematika yakni `from math` dengan memanfaatkan fungsi `sqrt`, maka source code lengkapnya seperti di bawah ini:

```

from sklearn.metrics import mean_squared_error, RSquare_score
from math import sqrt
print("RSQUARE: %.2f" % RSquare_score(test['PENDAFTAR'],
test['PREDIKSI_FIX']))
print("RMSE: %.2f" %
sqrt(mean_squared_error(test['PENDAFTAR'],
test['PREDIKSI_FIX'])))

```

Hasil dari source code pengujian performance di atas yakni di bawah ini:

```

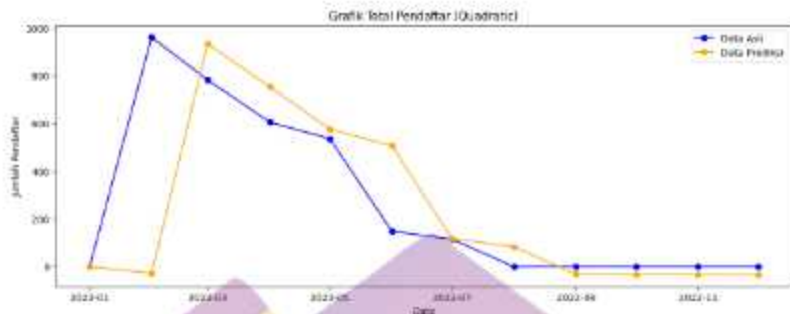
RSQUARE: 0.17
RMSE: 311.15

```

Berdasarkan hasil dari uji performance, Nilai RSquare sebesar 0.17 menunjukkan bahwa model quadratic hanya mampu menjelaskan sekitar 17% dari variasi yang ada dalam data jumlah pendaftar mahasiswa baru. Nilai ini menunjukkan bahwa model ini tidak secara efektif mewakili variasi yang signifikan dalam data. RMSE sebesar 311.15 menunjukkan rata-rata kesalahan prediksi model sekitar 311.15 unit. Model quadratic ini belum mampu memberikan peningkatan yang signifikan dalam menjelaskan variasi dalam data jumlah pendaftar mahasiswa baru.

#### 4.5.4. Visualisasi Data Asli dan Data Prediksi

Data prediksi yang diperoleh selanjutnya di visualisasikan dalam bentuk grafik garis sehingga dapat terlihat perbedaan dan jarak garis data asli dan data prediksi seperti pada Gambar 4.10:



Gambar 4. 10. Grafik Total Pendaftar (Quadratic)

## 4.6. Long Short-Term Memory (LSTM)

### 4.6.1. Pembentukan Model Prediksi

Membaca dataset dan menjadikan kolom Date sebagai index yang dimana Date tersebut menjadi frekuensi, source codenya:

```
df = pd.read_csv('//content/drive/MyDrive/Colab  
Notebooks/THESIS/BULANAN_THESIS/DATA_BULANAN/DATA_LSTM.csv', in  
dex_col='Date', parse_dates=True)  
df.index.freq= ('MS')
```

Selanjutnya melakukan normalisasi pada data agar data tersebut memiliki rentang yang sama dengan menggunakan fungsi dari library yang ada yakni seperti pada source code:

```
from sklearn.preprocessing import MinMaxScaler  
scaler = MinMaxScaler()
```

Dilanjutkan dengan memanggil data train dan test untuk dirubah dengan menggunakan skala tersebut, source codenya:

```
scaler.fit(train)  
scaled_train = scaler.transform(train)  
scaled_test = scaler.transform(test)
```

Berikut nilai hasil dari proses penskalaan data pendaftar dengan menampilkan 10 data teratas dari data train seperti pada Gambar 4.11:

```
scaled_train[:10]
array([[0.42025862],
       [0.25818966],
       [0.48706897],
       [0.10172414],
       [0.2307931 ],
       [0.20956897],
       [0.24181034],
       [0.28534483],
       [0.28534483],
       [0.28534483]])
```

Gambar 4. 11. Hasil Min-Max Normalisasi

Memanggil library `TimeseriesGenerator` untuk secara otomatis mengubah data waktu menjadi sampel yang siap untuk melatih model prediksi, source code librarynya:

```
from keras.preprocessing.sequence import TimeseriesGenerator
```

Selanjutnya, memberikan nilai inputan sebanyak 12 bulan untuk dapat memprediksi nilai bulan ke 13 (januari di tahun selanjutnya). `batch_size` Disini diberi angka 1, sehingga dapat diartikan bahwa di setiap 1 iterasinya akan ada 12 inputan untuk memprediksi data ke 13. Di iterasi kedua, data 2 – data 13 akan digunakan untuk menjadi nilai inputan dalam memprediksi data ke 14 begitu seterusnya.

```
n_input = 12
n_features = 1
generator = TimeseriesGenerator(scaled_train, scaled_train,
length=n_input, batch_size=1)
```

Batch adalah kelompok yang digunakan untuk menggabungkan satu atau lebih sampel data. Sedangkan Batch size adalah jumlah total sampel data training

yang ada di setiap batch. Langkah berikutnya, memanggil library untuk model dan layer seperti source code:

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
```

Model `Sequential` cocok untuk lapisan layer biasa di mana setiap lapisan memiliki tepat sebuah input dan sebuah output. `Dense` digunakan untuk membuat single layer pada Deep Learning Sequential kita. Lalu, dilakukan pendefinisian model dengan source code:

```
model = Sequential()
model.add(LSTM(100, activation='relu', input_shape=(n_input,
n_features)))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse')
```

`Sequential()` untuk memastikan penambahan layer satu demi satu. Lalu ditambahkan dengan layer LSTM dengan 100 neuron dengan kondisi aktif dengan jenis aktivasi reLU. `model.add(Dense(1))` merupakan lapisan untuk output yang jumlah outputnya sebanyak 1 neuron. Di compile dengan menggunakan optimasi dari adam dan menampilkan nilai `loss` / kesalahan / error dalam bentuk nilai MSE. Selanjutnya melakukan proses training model sebanyak 70 epochs dengan `batch_size 1` dan `batch 83`.

```
model.fit(generator, epochs=70)
```

`epochs` merupakan hyperparameter yang menentukan berapa kali algoritma deep learning bekerja melewati seluruh dataset baik secara forward maupun backward. Satu `epochs` tercapai ketika semua `batch` telah berhasil dilewatkan

melalui jaringan saraf satu kali. Dalam kasus penelitian ini, 1 epochs tercapai ketika 84 batch sampel data training selesai diproses.

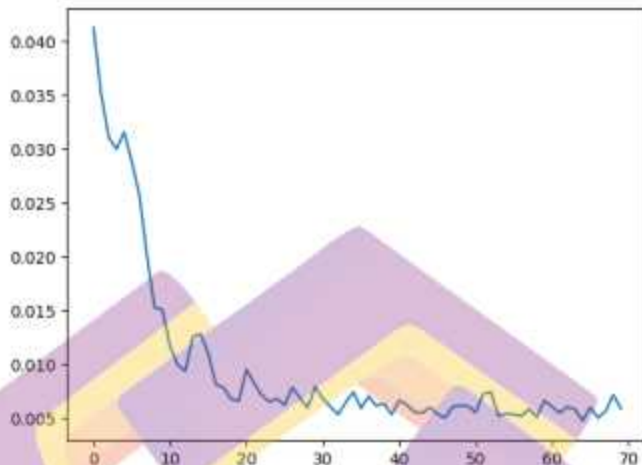
```
Epoch 1/70
83/83 [=====] - 2s 8ms/step - loss: 0.0413
Epoch 2/70
83/83 [=====] - 1s 15ms/step - loss: 0.0351
Epoch 3/70
83/83 [=====] - 1s 15ms/step - loss: 0.0310
Epoch 4/70
83/83 [=====] - 1s 8ms/step - loss: 0.0300
Epoch 5/70
83/83 [=====] - 1s 8ms/step - loss: 0.0316
...
Epoch 66/70
83/83 [=====] - 1s 8ms/step - loss: 0.0060
Epoch 67/70
83/83 [=====] - 1s 15ms/step - loss: 0.0051
Epoch 68/70
83/83 [=====] - 1s 14ms/step - loss: 0.0057
Epoch 69/70
83/83 [=====] - 1s 8ms/step - loss: 0.0072
Epoch 70/70
83/83 [=====] - 1s 12ms/step - loss: 0.0059
```

Untuk menampilkan grafik loss perepochnya kita menggunakan source code:

```
loss_per_epoch = model.history.history['loss']
plt.plot(range(len(loss_per_epoch)),loss_per_epoch)
```

Dari source code tersebut akan menampilkan grafik line histori epoch dari 0-70 seperti pada Gambar 4.12:





Gambar 4. 12. Grafik Loss Epoch LSTM

Dapat diperhatikan, loss yang dihasilkan mulai berkurang dari yang awal nilai mse-nya 0.0413 sampai menjadi 0.0059 di epoch ke 70. Langkah berikutnya melakukan prediksi untuk 12 bulan selanjutnya dengan menggunakan 12 bulan terakhir dari data train. Berikut source codenya:

```
test_predictions = []

first_eval_batch = scaled_train[-n_input:]
current_batch = first_eval_batch.reshape((1, n_input,
n_features))

for i in range(len(test)):
    current_pred = model.predict(current_batch)[0]
    test_predictions.append(current_pred)
    current_batch =
np.append(current_batch[:,1:,:], [[current_pred]], axis=1)
```

`test_predictions=[]` untuk menyetel data frame hasil prediksi dalam keadaan kosong sebagai awal. `scaled_train[-n_input:]` mengambil 12 nilai terakhir (januari – desember 2021) dari data train untuk dijadikan inputan awal yang

nantinya akan memprediksi bulan ke 13 (januari 2022). Yang dimana data testnya dimulai dari januari – desember 2022. Selanjutnya `first_eval_batch.reshape((1, n_input, n_features))` untuk pengaturan batch seperti pada pembuatan model prediksi. Dilanjut dengan menghitung jumlah data test `len(test)`, sehingga proses prediksinya akan dilakukan sebanyak jumlah data test yang ada untuk dapat dibandingkan nantinya. Tahap berikutnya, `model.predict(current_batch)` memulai prediksi dimana hasil prediksi akan tersimpan sebagai `current_pred`. Data yang tersimpan pada `current_pred` akan ditambahkan ke dataframe `test_predictions`. Selanjutnya, melakukan penghapusan data pertama pada inputan awal menggunakan `current_batch[:, 1:, :]`, Dimana tadi inputan awal menggunakan data 2021-01-01 sampai 2021-12-01, dari hasil prediksi memperoleh data untuk 2022-01-01. Sehingga pada proses `current_batch[:, 1:, :]` data 2021-01-01 akan dihapus dan `append(current_batch[:, 1:, :], [[current_pred]], axis=1)` data inputan selanjutnya menggunakan data dari 2021-02-01 sampai 2022-01-01. Proses tersebut berlanjut sampai memenuhi sebanyak hasil dari `len(test)`.

Dari proses tersebut didapatkan hasil prediksi dalam bentuk skala, seperti pada Gambar 4.13:

```
test_predictions:
[array([0.25014964], dtype=float32),
 array([0.7547104], dtype=float32),
 array([0.18058658], dtype=float32),
 array([0.17912674], dtype=float32),
 array([0.17666969], dtype=float32),
 array([0.17536035], dtype=float32),
 array([0.17867722], dtype=float32),
 array([0.18621174], dtype=float32),
 array([0.21489178], dtype=float32),
 array([0.22384611], dtype=float32),
 array([0.2325195], dtype=float32),
 array([0.24073456], dtype=float32)]
```

Gambar 4. 13. Hasil Prediksi LSTM (Min-Max Normalisasi)

#### 4.6.2. Hasil Prediksi

Dari Gambar hasil prediksi tersebut masih berupa skala, sehingga perlu dilakukan denormalisasi / lakukan inverse dengan menggunakan source code:

```
true_predictions = scaler.inverse_transform(test_predictions)
data_test['Predictions'] = true_predictions
```

Dari `true_predictions` yang disimpan pada kolom baru `data_test['Predictions']` dengan nama kolom 'Predictions', karena hasil prediksi tersebut merupakan hasil pengolahan diferensiasi data maka pada hasil akhir prediksi perlu kita dediferensiasi dengan data aslinya dengan menggunakan source code:

```
test['PREDIKSI_FIX'] =
test['ASLI'].shift(1) + test['Predictions']
```

Hasil prediksi pendaftar disimpan di kolom baru `test['PREDIKSI_FIX']` pada DataFrame `test` dengan nama kolom 'PREDIKSI\_FIX'. Hasil pada source code tersebut kami pindahkan dalam bentuk tabel agar mudah pembacaanya, hasil prediksi seperti pada Tabel 15:

Tabel 4. 15. Data Aktual dan Prediksi (LSTM)

Tahun Akademik	Data Aktual	Data Prediksi
2022-01-01	0	0,00
2022-02-01	961	1088,93
2022-03-01	782	717,96
2022-04-01	605	535,57
2022-05-01	536	352,87
2022-06-01	149	280,84
2022-07-01	115	-98,47
2022-08-01	0	-114,99
2022-09-01	0	-163,45
2022-10-01	0	-142,68
2022-11-01	0	-122,55
2022-12-01	0	-103,50

#### 4.6.3. Uji Performance

Hasil prediksi kita coba cek selisihnya dengan data aktualnya sehingga menghasilkan data seperti pada Tabel 4.16:

Tabel 4. 16. Selisih Antara Data Aktual dan Prediksi (LSTM)

Tahun Akademik	Data Aktual	Data Prediksi	Selisih
2022-01-01	0	0,00	0,00
2022-02-01	961	1088,93	127,93
2022-03-01	782	717,96	64,04
2022-04-01	605	535,57	69,43
2022-05-01	536	352,87	183,13
2022-06-01	149	280,84	131,84
2022-07-01	115	-98,47	213,47
2022-08-01	0	-114,99	114,99
2022-09-01	0	-163,45	163,45
2022-10-01	0	-142,68	142,68
2022-11-01	0	-122,55	122,55
2022-12-01	0	-103,50	103,50

Pada python sudah disediakan library untuk melakukan pengujian performance RMSE dan RSquare dengan memanfaatkan library dari `from sklearn.metrics`. Sedangkan untuk mendapatkan nilai RMSE maka diperlukan

library dari matematika yakni `from math` dengan memanfaatkan fungsi `sqrt`, maka source code lengkapnya seperti di bawah ini:

```
from sklearn.metrics import mean_squared_error, RSquare_score
from math import sqrt
print("RSQUARE: %.2f" % RSquare_score(test['PENDAFTAR'],
test['PREDIKSI_FIX']))
print("RMSE: %.2f" %
sqrt(mean_squared_error(test['PENDAFTAR'],
test['PREDIKSI_FIX'])))
```

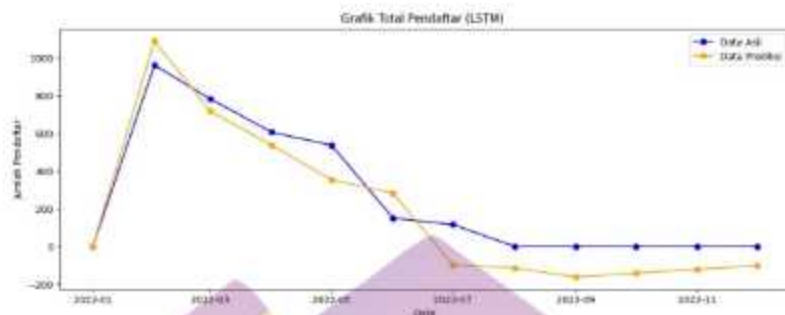
Hasil dari source code pengujian performance di atas yakni di bawah ini:

```
RSQUARE: 0.85
RMSE: 131.59
```

Berdasarkan hasil dari uji performance, Nilai RSquare sebesar 0.85 menunjukkan bahwa model LSTM mampu menjelaskan sekitar 85% dari variasi yang ada dalam data jumlah pendaftar mahasiswa baru. Ini adalah nilai RSquare yang tinggi, menandakan bahwa model ini cukup efektif dalam mewakili variasi dalam data. Model ini mungkin dapat mengidentifikasi pola dan tren yang kompleks dalam data. RMSE sebesar 131.59 menunjukkan rata-rata kesalahan prediksi model sekitar 131.59 unit (mungkin jumlah mahasiswa). Ini adalah nilai RMSE yang relatif rendah, menunjukkan bahwa model ini memiliki tingkat akurasi yang baik dalam memprediksi data.

#### 4.6.4. Visualisasi Data Asli dan Data prediksi

Untuk membedakan hasil data prediksi dan data asli dapat ditampilkan dalam bentuk grafik line dengan memanfaatkan fungsi sehingga menampilkan grafik line seperti pada Gambar 4.14:



Gambar 4. 14. Grafik Total Pendaftar (LSTM)

Dapat diperhatikan pada line orange tersebut yakni data hasil prediksi mendekati line biru yakni data asli. Bukan hanya itu, bentuk dari line orange hampir menyerupai bentuk dari line biru yang dimana pada bulan februari sampai bulan juni berada di rentang hasil 1000 – 200 dan di bulan-bulan tersebut memang merupakan pelaksanaan pendaftaran mahasiswa baru.

## 4.7. Regresi Linear

### 4.7.1. Pembentukan Model Prediksi

Pada algoritma ini tersedia library yang tinggal digunakan sehingga class yang terdapat dalam modul tersebut digunakan untuk melakukan proses prediksinya. Library yang digunakan seperti di bawah ini:

```
from sklearn.linear_model import LinearRegression
from sklearn import datasets, linear_model
```

`LinearRegression` digunakan untuk pembentukan model Regresi Linear.

Selanjutnya melakukan inisialisasi data terlebih dahulu dengan source code:

```
pendukung = train[['LULUS', 'REGISTRASI']]
cari = train['PENDAFTAR']
```

`train[['LULUS', 'REGISTRASI']]` untuk mengambil kolom dan dijadikan DataFrame `pendukung`. `cari = train['PENDAFTAR']` untuk mengambil kolom dan dijadikan DataFrame `cari`. Setelah inialisasi selesai, maka siap untuk dilakukan proses pembentukan model prediksi dengan menggunakan source code:

```
regressor = LinearRegression()
regressor.fit(pendukung, cari)
```

`regressor = LinearRegression()` membuat objek `regressor` dari kelas `LinearRegression` dan objek `regressor` ini yang akan digunakan sebagai model regresi linear yang akan dilatih dan digunakan untuk proses prediksi. `fit(..., ...)` digunakan untuk melatih model menggunakan data `train`. `pendukung` merupakan data independen yang digunakan untuk memprediksi target atau dependen. `cari` merupakan variabel target atau dependen yang akan diprediksi, bagian ini merupakan jumlah pendaftar. Hasil proses pembentukan model prediksi dari source code tersebut seperti pada Gambar 4.15:

```
#melakukan fitting simple linear regression
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(pendukung, cari) #
```

```
- LinearRegression
LinearRegression()
```

Gambar 4.15. Pembentukan Model Prediksi Regresi Linear

Untuk menampilkan nilai dari intercept dan koefisien dari model yang tersebut menggunakan source code:

```
print("Intercept: ", regressor.intercept_)
print("Coefficients: ", regressor.coef_)
```

menghasilkan nilai Intercept = 0.0 dan Coefficients = [4.18869059 - 2.9373329].

#### 4.7.2. Hasil Prediksi

Model prediksi yang sudah terbentuk dan tersimpan dalam objek regressor, selanjutnya digunakan untuk memprediksi pendaftar dengan menggunakan data test yang tersimpan di DataFrame pendukung\_test, berikut source codenya:

```
pendukung_test = test[['LULUS', 'REGISTRASI']]
Y_prediksinya = regressor.predict(pendukung_test)
test['PREDIKSI'] = Y_prediksinya
```

Hasil prediksi pendaftar disimpan di kolom baru test['PREDIKSI'] pada DataFrame test dengan nama kolom 'PREDIKSI'. Dari Y\_prediksinya yang disimpan pada kolom baru test['PREDIKSI'] dengan nama kolom 'PREDIKSI', karena hasil prediksi tersebut merupakan hasil pengolahan diferensiasi data maka pada hasil akhir prediksi perlu kita dediferensiasi dengan data aslinya dengan menggunakan source code:

```
test['PREDIKSI_FIX'] =
test['ASLI'].shift(1) + test['PREDIKSI']
```

Hasil prediksi pendaftar disimpan di kolom baru test['PREDIKSI\_FIX'] pada DataFrame test dengan nama kolom 'PREDIKSI\_FIX'. Hasil pada source code tersebut kami pindahkan dalam bentuk tabel agar mudah pembacaanya, hasil prediksi seperti pada Tabel 4.17:



Tabel 4. 17. Data Aktual dan Prediksi (Regresi Linear)

Tahun Akademik	Data Aktual	Data Prediksi
2022-01-01	0	0,00
2022-02-01	961	792,49
2022-03-01	782	901,04
2022-04-01	605	873,15
2022-05-01	536	618,21
2022-06-01	149	-32,19
2022-07-01	115	121,14
2022-08-01	0	-125,84
2022-09-01	0	0,00
2022-10-01	0	0,00
2022-11-01	0	0,00
2022-12-01	0	0,00

Data pada Tabel 4.17 disandingkan dengan data aktual untuk menjadi pembandingan antara data aktual dan data prediksi yang diperoleh.

#### 4.7.3. Uji Performance

Selanjutnya pengujian performance dari hasil Tabel 4.17 menghasilkan Tabel 4.18 untuk melihat hasil selisih antara data aktual dan data prediksi:

Tabel 4. 18. Selisih Antara Data Aktual dan Prediksi Regresi Linear

Tahun Akademik	Data Aktual	Data Prediksi	Selisih
2022-01-01	0	0,00	0,00
2022-02-01	961	792,49	168,51
2022-03-01	782	901,04	119,04
2022-04-01	605	873,15	268,15
2022-05-01	536	618,21	82,21
2022-06-01	149	-32,19	181,19
2022-07-01	115	121,14	6,14
2022-08-01	0	-125,84	125,84
2022-09-01	0	0,00	0,00
2022-10-01	0	0,00	0,00
2022-11-01	0	0,00	0,00
2022-12-01	0	0,00	0,00

Pada python sudah disediakan library untuk melakukan pengujian performance RMSE dan RSquare dengan memanfaatkan library dari `from sklearn.metrics`. Sedangkan untuk mendapatkan nilai RMSE maka diperlukan library dari matematika yakni `from math` dengan memanfaatkan fungsi `sqrt`, maka source code lengkapnya seperti di bawah ini:

```
from sklearn.metrics import mean_squared_error, RSquare_score
from math import sqrt
print("RSQUARE: %.2f" % RSquare_score(test['PENDAFTAR'],
test['PREDIKSI_FIX']))
print("RMSE: %.2f" %
sqrt(mean_squared_error(test['PENDAFTAR'],
test['PREDIKSI_FIX'])))
```

Hasil dari source code pengujian performance di atas yakni di bawah ini:

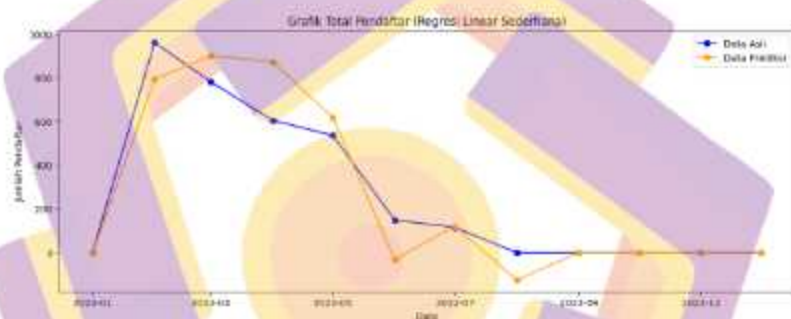
```
RSQUARE: 0.88
RMSE: 119,00
```

Berdasarkan hasil dari uji performance, data jumlah pendaftar, lulus, dan registrasi adalah variabel yang digunakan dalam model. Ketiga variabel ini dapat dianggap sebagai fitur-fitur atau faktor yang memengaruhi jumlah pendaftar mahasiswa baru. Nilai RSquare sebesar 0.88 menunjukkan bahwa model Regresi Linear dapat menjelaskan sekitar 88% variasi dalam jumlah pendaftar mahasiswa baru dengan mempertimbangkan variabel jumlah lulus dan jumlah registrasi. Ini menandakan bahwa hubungan linier antara variabel tersebut cukup kuat dan dapat diandalkan untuk membuat prediksi. RMSE sebesar 119.00 menunjukkan rata-rata kesalahan prediksi model sekitar 119.00 unit. Semakin rendah nilai RMSE, semakin baik model dalam memprediksi data. Dengan nilai RMSE yang relatif rendah, model ini memberikan prediksi yang akurat. Variabel jumlah lulus dan jumlah registrasi mungkin memiliki dampak signifikan terhadap jumlah pendaftar

mahasiswa baru. Keterlibatan variabel-variabel ini dalam model membantu meningkatkan kemampuan model untuk menjelaskan variasi dalam jumlah pendaftar.

#### 4.7.4. Visualisasi Data Asli dan Data Prediksi

Data prediksi yang diperoleh selanjutnya di visualisasikan dalam bentuk grafik garis sehingga dapat terlihat perbedaan dan jarak garis data asli dan data prediksi seperti pada Gambar 4.16:



Gambar 4. 16. Grafik Total Pendaftar (Regresi Linear)

## 4.8. Fuzzy Mamdani

### 4.8.1. Pembentukan Model Prediksi

Hal pertama yang perlu dilakukan yakni menginstall library fuzzy karena di google colaboratory disediakan library fuzzy yang dapat dimanfaatkan pengguna. Berikut library yang perlu diinstall source codenya:

```
!pip install pandas numpy matplotlib scikit-fuzzy
```



data, seperti grafik dan `plot.matplotlib.patches` untuk membuat objek-objek geometri (seperti persegi panjang atau lingkaran) yang dapat digunakan dalam visualisasi. Dilanjut dengan melakukan pengecekan nilai minimal dan maksimal dari data yang akan diinisialisasi dengan source code:

```
kolom_tertentu = ['PENDAFTAR', 'LULUS', 'REGISTRASI']
nilai_max_tertentu = data[kolom_tertentu].max()
nilai_min_tertentu = data[kolom_tertentu].min()

print("\nNilai Maksimal untuk Kolom Tertentu:")
print(nilai_max_tertentu)

print("\nNilai Minimal untuk Kolom Tertentu:")
print(nilai_min_tertentu)

kolom_tertentu = ['PENDAFTAR', 'LULUS', 'REGISTRASI']
```

membuat list `kolom_tertentu` yang berisi nama kolom yang ingin dievaluasi. Menggunakan `max()` dan `min()` pada DataFrame `data` untuk mendapatkan nilai maksimal dan minimal dari kolom-kolom tertentu. Menampilkan hasil nilai maksimal dan minimal untuk kolom tertentu menggunakan `print()`. Hasil dari source code tersebut:

```
Nilai Maksimal untuk Kolom Tertentu:
PENDAFTAR    1658.0
LULUS         572.0
REGISTRASI    510.0

Nilai Minimal untuk Kolom Tertentu:
PENDAFTAR    -662.0
LULUS        -440.0
REGISTRASI   -420.0
```

Data tersebut menjadi dasar dalam penentuan ambang batas. Selanjutnya, lakukan inisialisasi variabel input dan output bertujuan untuk menggambarkan hubungan antara berbagai faktor masukan (seperti tingkat kelulusan dan jumlah registrasi) dan hasil keluaran (jumlah pendaftar). Berikut source codenya:

```
# Inisialisasi variabel input
lulus = ctrl.Antecedent(np.arange(-400, 600, 1), 'LULUS')
register = ctrl.Antecedent(np.arange(-400, 600, 1),
'REGISTRASI')
# Inisialisasi variabel output
pendaftar = ctrl.Consequent(np.arange(-600, 1700, 1),
'PENDAFTAR')
```

`ctrl.Antecedent` merupakan tanda sebagai variabel input, sedangkan `ctrl.Consequent` merupakan variabel output. `(-400, 600, 1), 'LULUS')` memiliki rentang -400 hingga 600, dengan langkah 1, yang berarti setiap nilai antara -400 sampai 600 akan menjadi nilai yang mungkin dalam variabel bersangkutan. Setelah itu, penentuan fungsi keanggotaan untuk variabel input output dengan source code:

```
# Fungsi keanggotaan untuk variabel input dan output
lulus['low'] = fuzz.trimf(lulus.universe, [-400, -400, 300])
lulus['high'] = fuzz.trimf(lulus.universe, [300, 600, 600])

register['low'] = fuzz.trimf(register.universe, [-400, -400,
300])
register['high'] = fuzz.trimf(register.universe, [300, 600,
600])

pendaftar['low'] = fuzz.trimf(pendaftar.universe, [-600, -600,
500])
pendaftar['high'] = fuzz.trimf(pendaftar.universe, [500, 1700,
1700])
```

`fuzz.trimf(lulus.universe, [-400, -400, 300])` merupakan cara untuk mendefinisikan fungsi keanggotaan segitiga (`trimf`) untuk kategori 'low'. Fungsi ini menggambarkan bagaimana nilai-nilai pada variabel `lulus` terkait dengan tingkat 'low'. Parameter yang diberikan adalah `[-400, -400, 300]` yang menggambarkan bahwa kategori 'low' memiliki nilai minimum -400, nilai puncak pada -400 (dalam hal ini, titik puncak tidak memiliki arti nyata), dan nilai

maksimum 300. Dilanjutkan dengan penentuan rule aturan fuzzy dengan source code:

```
# Membuat aturan fuzzy
rule1 = ctrl.Rule(lulus['low'] & register['high'],
pendaftar['low'])
rule2 = ctrl.Rule(lulus['low'] & register['low'],
pendaftar['low'])
rule3 = ctrl.Rule(lulus['high'] & register['high'],
pendaftar['high'])
rule4 = ctrl.Rule(lulus['high'] & register['low'],
pendaftar['high'])
```

`rule1 = ctrl.Rule (lulus['low'] & register['high'] , pendaftar['low'] )` bisa diartikan sebagai : jika lulus['low'] masuk dalam rentang nilai 'low' dan register['high'] masuk dalam rentang 'high' maka prediksi pendaftar masuk dalam rentang nilai pendaftar['low']. Kemudian pembuatan sistem fuzzy itu sendiri menggunakan source code:

```
# Membuat sistem fuzzy
pendaftar_ctrl = ctrl.ControlSystem([rule1, rule2, rule3,
rule4])
pendaftar_sim = ctrl.ControlSystemSimulation(pendaftar_ctrl)
```

`ctrl.ControlSystem([rule1, rule2, rule3, rule4])` untuk membuat kontrol sistem logika fuzzy. `ctrl.ControlSystemSimulation` membuat objek simulasi dari kontrol sistem logika kabur yang telah Anda buat sebelumnya.

#### 4.8.2. Hasil Prediksi

Sistem fuzzy yang sudah dibuat selanjutnya melakukan prediksi untuk tiap tahunnya dengan menggunakan source code:

```

prediksi_pendaftar = []
for index, row in test.iterrows():
    pendaftar_sim.input['LULUS'] = row['LULUS']
    pendaftar_sim.input['REGISTRASI'] = row['REGISTRASI']
    pendaftar_sim.compute()
    prediksi_pendaftar.append(pendaftar_sim.output['PENDAFTAR']
])

test['PREDIKSI'] = prediksi_pendaftar

```

Hasil prediksi pendaftar disimpan di kolom baru `test['PREDIKSI']` pada DataFrame `test` dengan nama kolom `'PREDIKSI'`. Dari `prediksi_pendaftar` yang disimpan pada kolom baru `test['PREDIKSI']` dengan nama kolom `'PREDIKSI'`, karena hasil prediksi tersebut merupakan hasil pengolahan diferensiasi data maka pada hasil akhir prediksi perlu kita dediferensiasi dengan data aslinya dengan menggunakan source code:

```

test['PREDIKSI_FIX'] =
test['ASLI'].shift(1) + test['PREDIKSI']

```

Hasil prediksi pendaftar disimpan di kolom baru `test['PREDIKSI_FIX']` pada DataFrame `test` dengan nama kolom `'PREDIKSI_FIX'`. Hasil pada source code tersebut kami pindahkan dalam bentuk tabel agar mudah pembacaanya, hasil prediksi seperti pada Tabel 4.19:

Tabel 4. 19. Data Aktual dan Prediksi (Fuzzy Mamdani)

Tahun Akademik	Data Aktual	Data Prediksi
2022-01-01	0	0,00
2022-02-01	961	1121,22
2022-03-01	782	785,32
2022-04-01	605	690,01
2022-05-01	536	439,59
2022-06-01	149	311,87
2022-07-01	115	-18,51
2022-08-01	0	-66,30
2022-09-01	0	-157,14



Tabel 4. 19. (Lanjutan)

Tahun Akademik	Data Aktual	Data Prediksi
2022-10-01	0	-157,14
2022-11-01	0	-157,14
2022-12-01	0	-157,14

#### 4.8.3. Uji Performance

Selanjutnya pengujian performance dari hasil Tabel 4.19 menghasilkan Tabel 4.20 untuk melihat hasil selisih antara data aktual dan data prediksi:

Tabel 4. 20. Selisih Antara Data Aktual dan Prediksi Fuzzy Mamdani

Tahun Akademik	Data Aktual	Data Prediksi	Selisih
2022-01-01	0	0,00	0,00
2022-02-01	961	1121,22	160,22
2022-03-01	782	785,32	3,32
2022-04-01	605	690,01	85,01
2022-05-01	536	439,59	96,41
2022-06-01	149	311,87	162,87
2022-07-01	115	-18,51	133,51
2022-08-01	0	-66,30	66,30
2022-09-01	0	-157,14	157,14
2022-10-01	0	-157,14	157,14
2022-11-01	0	-157,14	157,14
2022-12-01	0	-157,14	157,14

Pada python sudah disediakan library untuk melakukan pengujian performance RMSE dan RSquare dengan memanfaatkan library dari `from sklearn.metrics`. Sedangkan untuk mendapatkan nilai RMSE maka diperlukan library dari matematika yakni `from math` dengan memanfaatkan fungsi `sqrt`, maka source code lengkapnya seperti di bawah ini:

```

from sklearn.metrics import mean_squared_error, RSquare_score
from math import sqrt
print("RSQUARE: %.2f" % RSquare_score(test['PENDAFTAR'],
test['PREDIKSI_FIX']))
print("RMSE: %.2f" %
sqrt(mean_squared_error(test['PENDAFTAR'],
test['PREDIKSI_FIX'])))

```

Hasil dari source code pengujian performance di atas yakni di bawah ini:

```

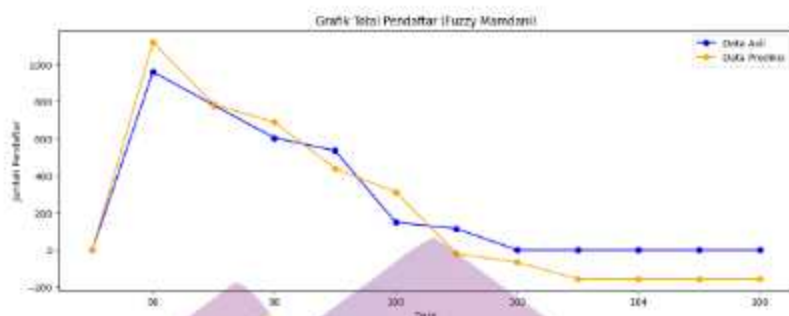
RSQUARE: 0.86
RMSE: 125,74

```

Berdasarkan hasil dari uji performance, data jumlah pendaftar, lulus, dan registrasi adalah variabel yang digunakan dalam model. Aturan fuzzy yang tepat sehingga menghasilkan nilai RSquare sebesar 0.86 menunjukkan bahwa model ini dapat menjelaskan sekitar 86% variasi dalam jumlah pendaftar mahasiswa baru dengan mempertimbangkan ketepatan aturan fuzzy dan keanggotaannya. RMSE sebesar 125.74 menunjukkan rata-rata kesalahan prediksi model sekitar 125.74 unit. Semakin rendah nilai RMSE, semakin baik model dalam memprediksi data. Dengan nilai RMSE yang relatif rendah, model ini memberikan prediksi yang akurat. Namun, pada model ini yang harus diperhatikan yakni ketepatan aturan fuzzy, fungsi keanggotaan dan hubungan antar variabel.

#### 4.8.4. Visualisasi Data Asli dan Data Prediksi

Data prediksi yang diperoleh selanjutnya di visualisasikan dalam bentuk grafik garis sehingga dapat terlihat perbedaan dan jarak garis data asli dan data prediksi seperti pada Gambar 4.18:



Gambar 4. 18. Grafik Total Pendaftar (Fuzzy Mamdani)

#### 4.9. Perbandingan Hasil Performance

##### 4.9.1 Menggunakan Data Pendaftar

Bagian ini merupakan perbandingan hasil uji performance dari algoritma yang menggunakan data pendaftar dalam proses memprediksi jumlah pendaftarnya, ditampilkan pada Tabel 4.21:

Tabel 4. 21. Perbandingan Hasil Performance (Least Square, Semi Average, Quadratic dan LSTM)

Algoritma	RSQUARE	RMSE
Least Square	0,17	310,99
Semi Average	0,17	310,35
Quadratic	0,17	311,15
Long Short-Term Memory (LSTM)	0,85	131,59

Least square (RSquare 0,17 dan RMSE 310,99) Rendahnya nilai RSquare dan tingginya nilai RMSE menunjukkan bahwa model least square tidak efektif dalam menjelaskan variasi dalam data jumlah pendaftar. Model ini tidak mampu menangkap pola atau tren yang kompleks dalam data. Semi average (RSquare 0,17

dan RMSE 310,35) hasil yang mirip dengan least square menunjukkan bahwa metode semi-average juga tidak memberikan peningkatan yang signifikan dalam kinerja dibandingkan dengan model yang lebih sederhana. Keduanya memiliki nilai RSquare dan RMSE yang rendah. Quadratic (RSquare 0,17 dan RMSE 311,15) model tersebut memberikan hasil yang serupa dengan least square dan semi average. RSquare yang rendah menunjukkan bahwa model ini tidak mampu menjelaskan variasi yang signifikan dalam data jumlah pendaftar, dan RMSE yang tinggi menandakan adanya kesalahan prediksi yang besar. LSTM (RSquare 0,85 dan RMSE 131,59) model tersebut memberikan hasil yang jauh lebih baik dibandingkan dengan tiga model sebelumnya. RSquare yang tinggi (0,85) menandakan bahwa model ini efektif dalam menjelaskan variasi dalam data. RMSE yang relatif rendah (131,59) menunjukkan bahwa prediksi model memiliki tingkat akurasi yang baik.

Model LSTM secara konsisten memberikan hasil evaluasi yang jauh lebih baik dibandingkan dengan tiga model lainnya. RSquare yang tinggi menandakan bahwa model mampu menggambarkan variasi dalam data dengan baik, sedangkan RMSE yang rendah menunjukkan tingkat akurasi yang baik. Model least square, semi average, dan quadratic tampaknya tidak mampu menangani kompleksitas pola atau tren dalam data jumlah pendaftar. Semua tiga model ini memberikan hasil evaluasi yang serupa dan menunjukkan keterbatasan dalam melakukan prediksi yang akurat. Oleh karena itu, untuk prediksi data jumlah pendaftar mahasiswa baru, model LSTM dapat dianggap sebagai pilihan yang lebih baik dan lebih canggih dibandingkan dengan model-model linier yang lebih sederhana. LSTM dapat

menangani pola waktu dan dependensi jangka panjang yang mungkin ada dalam data tersebut.

#### 4.9.2 Menggunakan Data Pendaftar, Lulus dan Registrasi

Bagian ini merupakan perbandingan hasil uji performance dari algoritma yang menggunakan data pendaftar, data lulus dan data registrasi dalam proses memprediksi jumlah pendaftarnya, ditampilkan pada Tabel 4.22:

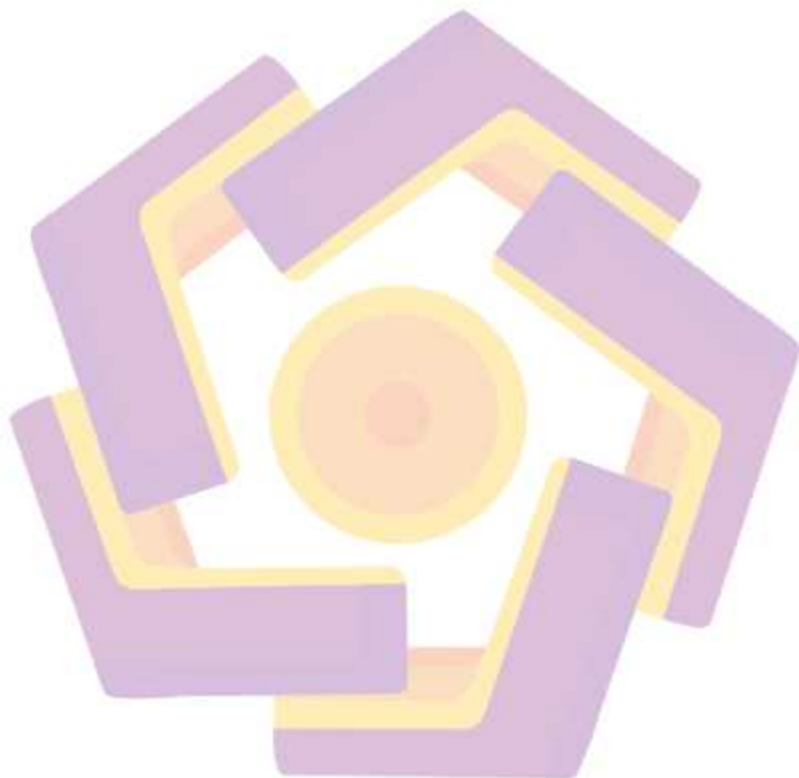
Tabel 4. 22. Perbandingan Hasil Performance (Regresi Linear dan Fuzzy Mamdani)

Algoritma	RSQUARE	RMSE
Regresi Linear	0,88	119,00
Fuzzy Mamdani	0,86	125,74

Regresi Linear (RSquare 0.88 dan RMSE 119,00) model ini memperoleh nilai RSquare yang tinggi (0.88) menandakan bahwa model regresi linear efektif dalam menjelaskan variasi dalam data jumlah pendaftar. Tingginya akurasi prediksi ditunjukkan oleh nilai RMSE yang rendah (119.00). Model ini secara umum memberikan hasil yang sangat baik. Fuzzy mamdani (RSquare 0.86 dan RMSE 125,74) meskipun model ini memiliki nilai RSquare yang tinggi (0.86) menunjukkan kemampuan model Fuzzy Mamdani dalam menjelaskan variasi dalam data, nilai RMSE yang lebih tinggi (125.74) mengindikasikan tingkat kesalahan prediksi yang sedikit lebih tinggi dibandingkan dengan regresi linear.

Baik regresi linear maupun model Fuzzy Mamdani memberikan hasil yang baik, dengan nilai RSquare yang mendekati 1, menunjukkan kemampuan keduanya dalam menjelaskan variasi dalam data jumlah pendaftar mahasiswa baru. Meskipun regresi linear memiliki nilai RMSE yang lebih rendah, perbedaan ini mungkin tidak

signifikan dalam konteks aplikasi praktis. Model Fuzzy Mamdani mungkin memiliki kompleksitas aturan dan keanggotaan yang memberikan konteks interpretatif yang lebih baik dalam beberapa kasus.



## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Penelitian ini melakukan prediksi jumlah pendaftar pada penerimaan mahasiswa baru dengan menggunakan algoritma Least Square, Semi Average, Quadratic, Long Short-Term Memory (LSTM), Regresi Linear, dan Fuzzy Mamdani. Penelitian yang sudah dilakukan menghasilkan:

1. Least Square, Semi Average, dan Quadratic, ketiga model tersebut menunjukkan hasil evaluasi yang serupa dengan nilai RSquare yang rendah (0.17) dan nilai RMSE yang tinggi (310.99, 310.35, 311.15). Hasil ini menandakan bahwa model-model tersebut tidak efektif dalam menjelaskan variasi dalam data time series jumlah pendaftar bulanan. Dimana data tersebut merupakan yang awalnya tidak stasioner.
2. Model LSTM memberikan hasil evaluasi yang jauh lebih baik dengan nilai RSquare yang tinggi (0.85) dan nilai RMSE yang rendah (131.59). Nilai RSquare yang tinggi menandakan kemampuan model dalam menjelaskan variasi dalam data time series dengan baik. RMSE yang rendah menunjukkan tingkat akurasi yang baik dalam prediksi jumlah pendaftar bulanan.
3. Regresi Linear memberikan hasil yang sangat baik dengan nilai RSquare sebesar 0.88, menunjukkan kemampuan yang tinggi dalam menjelaskan variasi dalam data jumlah pendaftar. RMSE yang rendah (119.00)

menandakan tingkat akurasi yang baik dalam prediksi jumlah pendaftar. Model ini memiliki interpretabilitas yang tinggi dan cocok untuk kasus-kasus di mana hubungan linier antara variabel dapat dengan jelas diidentifikasi.

4. Fuzzy mamdani juga memberikan hasil yang baik dengan nilai RSquare sebesar 0.86, menunjukkan kemampuan untuk menjelaskan variasi dalam data jumlah pendaftar. Meskipun RMSE (125.74) sedikit lebih tinggi dibandingkan dengan regresi linear, perbedaan ini mungkin tidak signifikan dalam konteks praktis. Fuzzy Mamdani memberikan interpretabilitas yang baik dan dapat memberikan wawasan tambahan melalui aturan linguistik.

## 5.2. Saran

Berdasarkan kesimpulan yang sudah di paparkan, ada beberapa saran untuk penelitian selanjutnya yang dapat diberikan antara lain:

1. Pola data dan sifat algoritma harus saling mendukung untuk mendapatkan prediksi yang tepat.
2. Skema pembentukan model prediksi dengan menggunakan data train yang diuji dengan data test perlu dicoba dengan berbagai skema untuk memperoleh kondisi model prediksi yang lebih baik.
3. Jumlah data yang akan digunakan menjadi faktor pertimbangan untuk menghasilkan prediksi yang lebih baik dan data historis yang runtun waktu.
4. Pemanfaatan unsur seasonal dari data untuk meningkatkan ketepatan prediksi.



## DAFTAR PUSTAKA

### PUSTAKA BUKU

Direktorat Jenderal Pendidikan Islam Kementerian Agama RI. (2019). *Panduan Perencanaan & Penganggaran Program Pendidikan Islam 2020 Pada Perguruan Tinggi Keagamaan Islam Negeri (PTKIN)* (Direktorat Jenderal Pendidikan Islam Kementerian Agama RI (ed.); 2nd ed.). Direktorat Jenderal Pendidikan Islam Kementerian Agama RI.

### PUSTAKA MAJALAH, JURNAL ILMIAH ATAU PROSIDING

Amiruddin, & Ishak, R. (2018). Prediksi Jumlah Mahasiswa Registrasi Per Semester Menggunakan Linier Regresi Pada Universitas Ichsan Gorontalo. *ILKOM Jurnal Ilmiah*, 10(2), 136–143. <https://doi.org/10.33096/ilkom.v10i2.274.136-143>

Apriliza, F., Darmansah, Oktavyani, A., & Kaazhim, D. Al. (2022). Perbandingan Metode Linear Regression dan Exponential Smoothing Dalam Peramalan Penerimaan Mahasiswa Baru. *JURIKOM (Jurnal Riset Komputer)*, 9(3), 726–732. <https://doi.org/10.30865/jurikom.v9i3.4300>

Ayuningtias, L. P., Irfan, M., & Jumadi, J. (2017). Analisa Perbandingan Logic Fuzzy Metode Tsukamoto, Sugeno, Dan Mamdani. *Jurnal Teknik Informatika*, 10(1), 9–16. <https://doi.org/10.15408/jti.v10i1.6810>

Budiprasetyo, G., Hani'ah, M., & Aflah, D. Z. (2023). Prediksi Harga Saham Syariah Menggunakan Algoritma Long Short-Term Memory (LSTM). *Jurnal Nasional Teknologi Dan Sistem Informasi*, 8(3), 164–172. <https://doi.org/10.25077/teknosi.v8i3.2022.164-172>

Candra, B. P., Kusriani, & Fatta, H. A. (2018). Implementation of Trend Moment Method for Stock Prediction as Supporting Production. *Journal of Physics: Conference Series*, 1140(1). <https://doi.org/10.1088/1742-6596/1140/1/012031>

Diponegoro, M. H., Kusumawardani, S. S., & Hidayah, I. (2021). Tinjauan Pustaka Sistematis: Implementasi Metode Deep Learning pada Prediksi Kinerja Murid. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, 10(2), 131–138. <https://doi.org/10.22146/jnteti.v10i2.1417>

Direktorat Jenderal Pendidikan Islam Kementerian Agama RI. (2019). *Panduan Perencanaan & Penganggaran Program Pendidikan Islam 2020 Pada Perguruan Tinggi Keagamaan Islam Negeri (PTKIN)* (Direktorat Jenderal Pendidikan Islam Kementerian Agama RI (ed.); 2nd ed.). Direktorat Jenderal

## Pendidikan Islam Kementerian Agama RI.

- Evanita, E., Khotimah, T., & Bakhar, M. (2022). Analisa Forecasting Pendaftaran Mahasiswa Baru Universitas Muria Kudus Pusat Studi: Sains dan Teknologi. *Smart Comp: Jurnalnya Orang Pintar Komputer*, 11(2), 162–174. <https://doi.org/10.30591/smartcomp.v11i2.3449>
- Fachid, S., & Triayudi, A. (2022). Perbandingan Algoritma Regresi Linier dan Regresi Random Forest Dalam Memprediksi Kasus Positif Covid-19. *Jurnal Media Informatika Budidarma*, 6(1), 68. <https://doi.org/10.30865/mib.v6i1.3492>
- Garuda, S. R., & Baliadi, Y. (2018). Aplikasi Metode Tren Waktu Satu Ragam Dalam Peramalan Toleransi Komoditas Pangan Terhadap Perubahan Iklim Di Papua. *Jurnal Informatika Pertanian*, 27(1), 35–46.
- Harini, D., & Wahyuniar, L. S. (2021). Estimasi Jumlah Murid Baru Menggunakan Metode Forecasting. *Journal of Instructional Mathematics*, 2(2), 64–70. <https://doi.org/10.37640/jim.v2i2.1025>
- Hatta, M., & Fauziah Fitri, A. (2020). Sistem Prediksi Persediaan Stok Darah Dengan Metode Least Square Pada Unit Transfusi Darah Studi Kasus PMI Kota Cirebon. *Jurnal Ilmiah Ilmu Komputer*, 6(1), 41–45. <https://doi.org/10.35329/jiik.v6i1.130>
- Kurniadi, W. (2018). Pendukung Keputusan Dalam Peramalan Penjualan Ayam Broiler Dengan Metode Trend Moment Dan Simple Moving Average Pada CV. Merdeka Adi Perkasa. *Jurnal Media Informatika Budidarma*, 2(3), 76–90. <https://doi.org/10.30865/mib.v2i3.652>
- Machfiroh, I. S., Alam Sur, W. A., & Pangestu, R. T. (2022). Trend Semi Average and Least Square in Forecasting Yamaha Motorcycle Sales. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, 16(1), 343–354. <https://doi.org/10.30598/barekengvol16iss1pp341-352>
- Nafi'iyah, N., & Wulandari, P. A. (2022). Prediksi Harga Beras Berdasarkan Kualitas Beras dengan Metode Long Short Term Memory. *Jurnal Inovtek Polbeng*, 7(2), 278–288.
- Pakpahan, H. S., Anandiya, O., Hairah, U., & Wati, M. (2019). Decision support system for predicting increased data on objects of motor vehicle name transfer (BBNKB I) using trend moment method (Case study: Wheels 2 and wheels 4 in Samarinda). *Journal of Physics: Conference Series*, 1341(4). <https://doi.org/10.1088/1742-6596/1341/4/042014>
- Patandean, S., Askar, A., & Mashud, M. (2019). Aplikasi Forecasting Penjualan Menggunakan Metode Semi Average Pada Toko Rumah Kita Makassar.

*Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, 9(1), 25–32.  
<https://doi.org/10.35585/inspir.v9i1.2493>

Pratama, R. R. (2020). Analisis Model Machine Learning Terhadap Pengenalan Aktifitas Manusia. *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 19(2), 302–311.  
<https://doi.org/10.30812/matrik.v19i2.688>

Putra Manurung, B. U. (2015). Implementasi Least Square Dalam Untuk Prediksi Penjualan Sepeda Motor ( Studi Kasus : Pt . Graha Auto Pratama ). *Jurnal Riset Komputer (JURIKOM)*, ISSN 2407-389X, 2(6), 21–24.

Rahmawati, I., & Wijanarko, R. (2019). Implementasi Prediksi Penjualan Obat Menggunakan Metode Least Square Pada Apotek Demak Farma Jaya. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 1(1), 19–23.  
<https://doi.org/10.36499/jinrpl.v1i1.2762>

Rahmawita, M., & Fazri, I. (2018). Aplikasi Peramalan Penjualan Obat Menggunakan Metode Least Square di Rumah Sakit Bhayangkara. *Jurnal Ilmiah Rekayasa Dan Manajemen Sistem Informasi*, 4(2), 201.  
<https://doi.org/10.24014/rmsi.v4i2.5685>

Rifa'i, A. (2019). The Statistical Parabolic Projection Method Untuk Forecasting. *Jurnal Manajemen Dan Bisnis*, 8(2), 354–365.  
<https://doi.org/10.34006/jmbi.v8i2.137>

Roosaputri, D. R., & Dewi, C. (2023). Perbandingan Algoritma ARIMA , Prophet , dan LSTM dalam Prediksi Penjualan Tiket Wisata Taman Hiburan ( Studi Kasus : Saloka Theme Park ). *Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, 4(3), 507–517. <https://doi.org/10.30645/kesatria.v4i3.199>

Roza, Y., Pernando, Y., Verdian, I., Febrianti, E. L., & Syafrinal, I. (2022). Prediksi Penjualan Menggunakan Metode Fuzzy Mamdani Pada PT. XYZ. *JURIKOM (Jurnal Riset Komputer)*, 9(6), 1989–1995.  
<https://doi.org/10.30865/jurikom.v9i6.5333>

Ruslan, M., & Banjarbaru, S. (2016). Prediksi Jumlah Penduduk Provinsi Kalimantan Selatan Menggunakan Metode Semi Average. *IJSE – Indonesian Journal on Software Engineering*, 2(1), 2461–2690.

Saryanti, I. G. A. D., & Sidhiantari, I. G. A. P. I. (2020). Analisis Perancangan Aplikasi Peramalan Barang Dadang Sederhana dengan Metode Semi Average Method. *Jurnal Teknik Elektro Dan Komputer*, 9(J. Tek. Elektro dan Komput.), 205–212. <https://doi.org/10.35793/jtek.v9i3.31415>

Selle, N., Yudistira, N., & Dewi, C. (2022). Perbandingan Prediksi Penggunaan Listrik dengan Menggunakan Metode Long Short Term Memory (LSTM) dan

- Recurrent Neural Network (RNN). *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 9(1), 155–162. <https://doi.org/10.25126/jtiik.2022915585>
- Sudiatini, D., & Ara, S. N. (2022). Forecasting Analysis of MSME Sales of Nata De Coco Products in PT Shenovia Using Least Square, Moving Average, and Semi Average Methods. *Journal Operational Research, Manajemen and Economics*, 2(3), 15–21. <https://doi.org/10.32424/1.jame.2022.24.3.5981>
- Trianggana, D. A. (2020). Peramalan Jumlah Siswa-Siswi Melalui Pendekatan Metode Regresi Linear. *Jurnal Media Infotama*, 16(2), 115–120. <https://doi.org/10.37676/jmi.v16i2.1149>
- Utami, Y., Vinsensia, D., Nissa, A., & Sulastri, S. (2022). Forecasting the Sum of New College Students with Linear Regression Approach. *Jurnal Teknik Informatika C.I.T Medicom*, 14(1), 10–15. <https://doi.org/10.35335/cit.vol14.2022.231.pp10-15>
- Wiratama, W., Alifah, L. A., Gurusinga, A., & Indra, E. (2023). Prediksi Turis Mancanegara ke Indonesia Menggunakan Metode EDA Time Series dan LSTM. *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, 8(2), 524–537. <https://doi.org/10.30645/jurasik.v8i2.637>
- Yulian, I., Anggraeni, D. S., & Aini, Q. (2020). Penerapan Metode Trend Moment Dalam Forecasting Penjualan Produk CV. Rabbani Asyisa. *JURTEKSI (Jurnal Teknologi Dan Sistem Informasi)*, 6(2), 193–200.
- Zikri, I., Safrida, S., Susanti, E., & Putri, R. A. (2020). Analysis of trend and determinant factors of imported soybean in the period of 2003-2022. *Advances in Food Science, Sustainable Agriculture and Agroindustrial Engineering*, 3(1), 17–24. <https://doi.org/10.21776/ub.afssaae.2020.003.01.3>

## LAMPIRAN

### I. Data Asli

Tabel 1. Data Asli

NO	TH_BU_TG	BULAN	TAHUN	PENDAFTAR	LULUS	REGISTRASI
1	01/01/2014	1	2014	0	0	0
2	01/02/2014	2	2014	313	278	193
3	01/03/2014	3	2014	250	218	158
4	01/04/2014	4	2014	718	684	655
5	01/05/2014	5	2014	292	248	235
6	01/06/2014	6	2014	184	141	130
7	01/07/2014	7	2014	101	49	47
8	01/08/2014	8	2014	0	0	0
9	01/09/2014	9	2014	0	0	0
10	01/10/2014	10	2014	0	0	0
11	01/11/2014	11	2014	0	0	0
12	01/12/2014	12	2014	0	0	0
13	01/01/2015	1	2015	0	0	0
14	01/02/2015	2	2015	858	414	252
15	01/03/2015	3	2015	529	195	104
16	01/04/2015	4	2015	1127	668	614
17	01/05/2015	5	2015	902	513	444
18	01/06/2015	6	2015	522	322	304
19	01/07/2015	7	2015	130	83	77
20	01/08/2015	8	2015	0	0	0
21	01/09/2015	9	2015	0	0	0
22	01/10/2015	10	2015	0	0	0
23	01/11/2015	11	2015	0	0	0
24	01/12/2015	12	2015	0	0	0
25	01/01/2016	1	2016	0	0	0
26	01/02/2016	2	2016	958	572	396
27	01/03/2016	3	2016	880	472	321
28	01/04/2016	4	2016	1018	494	410
29	01/05/2016	5	2016	623	291	229
30	01/06/2016	6	2016	457	275	234
31	01/07/2016	7	2016	582	372	316
32	01/08/2016	8	2016	0	0	0
33	01/09/2016	9	2016	0	0	0
34	01/10/2016	10	2016	0	0	0
35	01/11/2016	11	2016	0	0	0
36	01/12/2016	12	2016	0	0	0

Tabel 1. (Lanjutan)

NO	TH_BU_TG	BULAN	TAHUN	PENDAFTAR	LULUS	REGISTRASI
37	01/01/2017	1	2017	0	0	0
38	01/02/2017	2	2017	998	399	358
39	01/03/2017	3	2017	889	261	232
40	01/04/2017	4	2017	1156	328	299
41	01/05/2017	5	2017	969	255	220
42	01/06/2017	6	2017	722	506	463
43	01/07/2017	7	2017	662	440	417
44	01/08/2017	8	2017	0	0	0
45	01/09/2017	9	2017	0	0	0
46	01/10/2017	10	2017	0	0	0
47	01/11/2017	11	2017	0	0	0
48	01/12/2017	12	2017	0	0	0
49	01/01/2018	1	2018	0	0	0
50	01/02/2018	2	2018	973	547	392
51	01/03/2018	3	2018	773	435	310
52	01/04/2018	4	2018	889	435	396
53	01/05/2018	5	2018	776	399	355
54	01/06/2018	6	2018	669	419	390
55	01/07/2018	7	2018	565	363	340
56	01/08/2018	8	2018	0	0	0
57	01/09/2018	9	2018	0	0	0
58	01/10/2018	10	2018	0	0	0
59	01/11/2018	11	2018	0	0	0
60	01/12/2018	12	2018	0	0	0
61	01/01/2019	1	2019	0	0	0
62	01/02/2019	2	2019	1658	382	275
63	01/03/2019	3	2019	1391	316	210
64	01/04/2019	4	2019	1179	530	470
65	01/05/2019	5	2019	907	400	366
66	01/06/2019	6	2019	722	524	469
67	01/07/2019	7	2019	566	429	397
68	01/08/2019	8	2019	0	0	0
69	01/09/2019	9	2019	0	0	0
70	01/10/2019	10	2019	0	0	0
71	01/11/2019	11	2019	0	0	0
72	01/12/2019	12	2019	0	0	0
73	01/01/2020	1	2020	0	0	0
74	01/02/2020	2	2020	1339	477	352
75	01/03/2020	3	2020	1047	387	269
76	01/04/2020	4	2020	979	534	460

Tabel 1. (Lanjutan)

NO	TH_BU_TG	BULAN	TAHUN	PENDAFTAR	LULUS	REGISTRASI
77	01/05/2020	5	2020	766	447	398
78	01/06/2020	6	2020	355	335	315
79	01/07/2020	7	2020	246	237	213
80	01/08/2020	8	2020	0	0	0
81	01/09/2020	9	2020	0	0	0
82	01/10/2020	10	2020	0	0	0
83	01/11/2020	11	2020	0	0	0
84	01/12/2020	12	2020	0	0	0
85	01/01/2021	1	2021	0	0	0
86	01/02/2021	2	2021	1176	458	318
87	01/03/2021	3	2021	909	458	336
88	01/04/2021	4	2021	709	532	442
89	01/05/2021	5	2021	489	379	322
90	01/06/2021	6	2021	261	247	231
91	01/07/2021	7	2021	113	100	90
92	01/08/2021	8	2021	0	0	0
93	01/09/2021	9	2021	0	0	0
94	01/10/2021	10	2021	0	0	0
95	01/11/2021	11	2021	0	0	0
96	01/12/2021	12	2021	0	0	0
97	01/01/2022	1	2022	0	0	0
98	01/02/2022	2	2022	961	415	322
99	01/03/2022	3	2022	782	353	254
100	01/04/2022	4	2022	605	508	444
101	01/05/2022	5	2022	536	481	401
102	01/06/2022	6	2022	149	149	121
103	01/07/2022	7	2022	115	115	82
104	01/08/2022	8	2022	0	0	0
105	01/09/2022	9	2022	0	0	0
106	01/10/2022	10	2022	0	0	0
107	01/11/2022	11	2022	0	0	0
108	01/12/2022	12	2022	0	0	0

## 2. Data Setelah di Diferensiasi

Tabel 2. Data Setelah di Diferensiasi

NO	TH_BU_TG	BULAN	TAHUN	diff_pendaftar	diff_lulus	diff_registrasi
1	01/02/2014	2	2014	313	278	193
2	01/03/2014	3	2014	-63	-60	-35
3	01/04/2014	4	2014	468	466	497
4	01/05/2014	5	2014	-426	-436	-420
5	01/06/2014	6	2014	-108	-107	-105
6	01/07/2014	7	2014	-83	-92	-83
7	01/08/2014	8	2014	-101	-49	-47
8	01/09/2014	9	2014	0	0	0
9	01/10/2014	10	2014	0	0	0
10	01/11/2014	11	2014	0	0	0
11	01/12/2014	12	2014	0	0	0
12	01/01/2015	1	2015	0	0	0
13	01/02/2015	2	2015	858	414	252
14	01/03/2015	3	2015	-329	-219	-148
15	01/04/2015	4	2015	598	473	510
16	01/05/2015	5	2015	-225	-155	-170
17	01/06/2015	6	2015	-380	-191	-140
18	01/07/2015	7	2015	-392	-239	-227
19	01/08/2015	8	2015	-130	-83	-77
20	01/09/2015	9	2015	0	0	0
21	01/10/2015	10	2015	0	0	0
22	01/11/2015	11	2015	0	0	0
23	01/12/2015	12	2015	0	0	0
24	01/01/2016	1	2016	0	0	0
25	01/02/2016	2	2016	958	572	396
26	01/03/2016	3	2016	-78	-100	-75
27	01/04/2016	4	2016	138	22	89
28	01/05/2016	5	2016	-395	-203	-181
29	01/06/2016	6	2016	-166	-16	5
30	01/07/2016	7	2016	125	97	82
31	01/08/2016	8	2016	-582	-372	-316
32	01/09/2016	9	2016	0	0	0
33	01/10/2016	10	2016	0	0	0
34	01/11/2016	11	2016	0	0	0
35	01/12/2016	12	2016	0	0	0
36	01/01/2017	1	2017	0	0	0
37	01/02/2017	2	2017	998	399	358
38	01/03/2017	3	2017	-109	-138	-126



Tabel 2. (Lanjutan)

NO	TH_BU_TG	BULAN	TAHUN	diff_pendaftar	diff_lulus	diff_registrasi
39	01/04/2017	4	2017	267	67	67
40	01/05/2017	5	2017	-187	-73	-79
41	01/06/2017	6	2017	-247	251	243
42	01/07/2017	7	2017	-60	-66	-46
43	01/08/2017	8	2017	-662	-440	-417
44	01/09/2017	9	2017	0	0	0
45	01/10/2017	10	2017	0	0	0
46	01/11/2017	11	2017	0	0	0
47	01/12/2017	12	2017	0	0	0
48	01/01/2018	1	2018	0	0	0
49	01/02/2018	2	2018	973	547	392
50	01/03/2018	3	2018	-200	-112	-82
51	01/04/2018	4	2018	116	0	86
52	01/05/2018	5	2018	-113	-36	-41
53	01/06/2018	6	2018	-107	20	35
54	01/07/2018	7	2018	-104	-56	-50
55	01/08/2018	8	2018	-565	-363	-340
56	01/09/2018	9	2018	0	0	0
57	01/10/2018	10	2018	0	0	0
58	01/11/2018	11	2018	0	0	0
59	01/12/2018	12	2018	0	0	0
60	01/01/2019	1	2019	0	0	0
61	01/02/2019	2	2019	1658	382	275
62	01/03/2019	3	2019	-267	-66	-65
63	01/04/2019	4	2019	-212	214	260
64	01/05/2019	5	2019	-272	-130	-104
65	01/06/2019	6	2019	-185	124	103
66	01/07/2019	7	2019	-156	-95	-72
67	01/08/2019	8	2019	-566	-429	-397
68	01/09/2019	9	2019	0	0	0
69	01/10/2019	10	2019	0	0	0
70	01/11/2019	11	2019	0	0	0
71	01/12/2019	12	2019	0	0	0
72	01/01/2020	1	2020	0	0	0
73	01/02/2020	2	2020	1339	477	352
74	01/03/2020	3	2020	-292	-90	-83
75	01/04/2020	4	2020	-68	147	191
76	01/05/2020	5	2020	-213	-87	-62
77	01/06/2020	6	2020	-411	-112	-83
78	01/07/2020	7	2020	-109	-98	-102

Tabel 2. (Lanjutan)

NO	TH_BU_TG	BULAN	TAHUN	diff_pendaftar	diff_lulus	diff_registrasi
79	01/08/2020	8	2020	-246	-237	-213
80	01/09/2020	9	2020	0	0	0
81	01/10/2020	10	2020	0	0	0
82	01/11/2020	11	2020	0	0	0
83	01/12/2020	12	2020	0	0	0
84	01/01/2021	1	2021	0	0	0
85	01/02/2021	2	2021	1176	458	318
86	01/03/2021	3	2021	-267	0	18
87	01/04/2021	4	2021	-200	74	106
88	01/05/2021	5	2021	-220	-153	-120
89	01/06/2021	6	2021	-228	-132	-91
90	01/07/2021	7	2021	-148	-147	-141
91	01/08/2021	8	2021	-113	-100	-90
92	01/09/2021	9	2021	0	0	0
93	01/10/2021	10	2021	0	0	0
94	01/11/2021	11	2021	0	0	0
95	01/12/2021	12	2021	0	0	0
96	01/01/2022	1	2022	0	0	0
97	01/02/2022	2	2022	961	415	322
98	01/03/2022	3	2022	-179	-62	-68
99	01/04/2022	4	2022	-177	155	190
100	01/05/2022	5	2022	-69	-27	-43
101	01/06/2022	6	2022	-387	-332	-280
102	01/07/2022	7	2022	-34	-34	-39
103	01/08/2022	8	2022	-115	-115	-82
104	01/09/2022	9	2022	0	0	0
105	01/10/2022	10	2022	0	0	0
106	01/11/2022	11	2022	0	0	0
107	01/12/2022	12	2022	0	0	0