



**PERAMALAN JUMLAH WISATAWAN MENGGUNAKAN
METODE TIME SERIES ARIMA DAN FEED-FORWARD
NEURAL NETWORK
(STUDI KASUS: TEMPAT WISATA PASIR PUTIH,
SITUBONDO)**

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Teknologi Informasi Universitas Jember dan mencapai gelar Sarjana Komputer

SKRIPSI

Oleh

**Muhammad Zulqarnain Hidayat
182410102074**

**KEMETERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN
TEKNOLOGI
UNIVERSITAS JEMBER
FAKULTAS ILMU KOMPUTER
TEKNOLOGI INFORMASI
JEMBER
2023**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam pengerjaan skripsi;
2. Ayahanda Mochammad Taufik Hidayat;
3. Ibunda Suci Setyo;
4. Saudari kandung Atiqah Hani Rahmatillah;
5. Keluarga besar;
6. Teman – teman dengan dukungan do'a nya;
7. Guru-guru baik dari Pendidikan formal dan non-formal;
8. Almamater Program Studi Teknologi Informasi Fakultas Ilmu Komputer Universitas Jember.

MOTTO

“it is what it is”

Lawrence

“be old, be bold”

Muhammad Zulqarnain Hidayat

PERNYATAAN ORISINALITAS

Saya yang bertanda tangan di bawah ini :

Nama : Muhammad Zulqarnain Hidayat

NIM : 182410102074

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul: *Peramalan Jumlah Wisatawan Menggunakan Metode Time Series ARIMA dan Feed-Forward Neural Network (Studi Kasus: Tempat Wisata Pasir Putih, Situbondo)* adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan skripsi ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 17 Juli 2023

Yang menyatakan,

(Meterai Rp 10.000,00)

Muhammad Zulqarnain Hidayat

NIM 182410102074

HALAMAN PERSETUJUAN

Skripsi berjudul *Peramalan Jumlah Wisatawan Menggunakan Metode Time Series ARIMA dan Feed-Forward Neural Network (Studi Kasus: Tempat Wisata Pasir Putih, Situbondo)* telah diuji dan disahkan oleh Fakultas Ilmu Komputer Universitas Jember pada:

Hari : Jum'at

Tanggal : 7 Juli 2023

Tempat : Fakultas Ilmu Komputer Universitas Jember

Pembimbing

Tanda Tangan

1. Pembimbing Utama

Nama : Prof. Dr. Saiful Bukhori, ST., M.Kom

(.....)

NIP : 196811131994121001

2. Pembimbing Pendamping

Nama : Priza Pandunata, S.Kom., M.Sc

(.....)

NIP : 198301312015041002

Penguji

1. Penguji 1

Nama : Nelly Oktavia A, S.Si., MT

(.....)

NIP : 198410242009122008

2. Penguji 2

Nama : Gama Wisnu Fajarianto, S.Kom., M.Kom

(.....)

NRP : 760015717

ABSTRACT

Tourism comes from Sanskrit which means to travel repeatedly. Tourism is the largest industry in Indonesia, since the covid-19 outbreak on March 3 2020, residents around Pasir Putih Beach Situbondo experienced economic problems due to a drastic decrease in the number of tourist visitors. Tourism office and the local management needs to do a fairly large and extensive promotion of the activities that will be held to boost the economy of the surrounding community, the implementation of activities can be based on the highest forecasting value in order to prepare infrastructure to compensate for the surge in the number of tourist visitors. Forecasting is done by implementing ARIMA time-series method and feed-forward Neural Network with forecasting accuracy score using Mean Absolute Percentage Error, this study uses monthly tourist visitor data of Pasir Putih Beach Situbondo from January 2017 to December 2021. The results showed that the data used has a seasonal data type with a stationary data pattern, so there is no need to do the process of differencing but need to add seasonal parameters. The best ARIMA model selected is seasonal ARIMA model and the best FFNN model is 1 input layer LSTM model with 72 nodes, 1 hidden layer LSTM with 64 nodes, 1 hidden layer Dense with 18 nodes, and 1 output layer with 1 node with an error value of 4.245956% or has an accuracy of 95.7% The hybrid forecasting value for the number of tourist visitors in 2022 was 16216 visitors in January, 12066 visitors in February, 10357 visitors in March, 7814 visitors in April, 8592 visitors in May, 10110 visitors in June, 13451 visitors in July, 13460 visitors in August, 13466 visitors in September, 13468 visitors in October, 13467 visitors in November, 13465 visitors in December.

Keywords: Tourist, Forecasting, ARIMA, Feed-Forward Neural Network, ARIMA-FFNN

RINGKASAN

Pariwisata berasal dari bahasa sansekerta yang berarti bepergian berulang kali. Pariwisata adalah industri terbesar di Indonesia, sejak adanya wabah Covid-19 pada 3 Maret 2020 lalu warga sekitar Pantai Pasir Putih Situbondo mengalami permasalahan ekonomi akibat penurunan jumlah pengunjung wisata yang sangat drastis. Dinas pariwisata dan pihak pengelola setempat perlu melakukan promosi yang cukup besar dan luas tentang kegiatan-kegiatan yang akan diadakan untuk meningkatkan kembali perekonomian masyarakat sekitar, pelaksanaan kegiatan dapat berdasarkan nilai peramalan tertinggi agar dapat menyiapkan sarana prasaran untuk mengimbangi pelonjakan jumlah pengunjung wisata. Peramalan dilakukan dengan mengimplementasikan metode *time-series* ARIMA dan *feed-forward Neural Network* dengan skor akurasi peramalan menggunakan *Mean Absolute Percentage Error* (MAPE), penelitian ini menggunakan data pengunjung wisata bulanan Pantai Pasir Putih Situbondo dari bulan Januari 2017 hingga bulan Desember 2021.

Hasil penelitian menunjukkan bahwa data yang digunakan memiliki tipe data musiman dengan pola data stasioner, sehingga tidak perlu dilakukan proses *differencing* (i) namun perlu menambahkan parameter musiman (M). Model ARIMA terbaik yang dipilih adalah model ARIMA musiman (1,0,0)(1,0,0,12) dan model FFNN terbaik adalah model 1 layer input LSTM sebanyak 72 node, 1 hidden layer LSTM sebanyak 64 node, 1 hidden layer Dense sebanyak 18 node, dan 1 output layer sebanyak 1 node dengan nilai error 4.245956 % atau memiliki akurasi sebesar 95.7%.

Nilai peramalan *hybrid* untuk jumlah pengunjung wisata pada tahun 2022 sebanyak 16.216 pengunjung bulan Januari, 12.066 pengunjung bulan Februari, 10.357 pengunjung bulan Maret, 7.814 pengunjung bulan April, 8.592 pengunjung bulan Mei, 10.110 pengunjung bulan Juni, 13.451 pengunjung bulan Juli, 13.460 pengunjung bulan Agustus, 13.466 pengunjung bulan September, 13.468 pengunjung bulan Oktober, 13.467 pengunjung bulan November, 13.465 pengunjung bulan Desember.

PRAKATA

Puji syukur atas kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga skripsi dengan judul “Peramalan Jumlah Pengunjung Wisatawan Menggunakan Metode *Time Series ARIMA* dan *Feed-Forward Neural Network* (Studi Kasus: Tempat Wisata Pasir Putih, Situbondo)” dapat terselesaikan dengan lancar. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan Pendidikan Strata Satu (S1) pada Program Studi Teknologi Informasi Fakultas Ilmu Komputer Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak Oleh karena itu penulis menyampaikan terimakasih kepada:

1. Prof. Dr. Antonius Cahya Prihandoko, M.App.Sc, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Jember;
2. Prof. Dr. Saiful Bukhori, ST., M.Kom selaku Dosen Pembimbing Utama dan Bapak Priza Pandunata, S. Kom., M.Sc selaku Dosen Pembimbing Pendamping yang telah meluangkan waktu, dan memberikan arahan, ilmu, nasihat serta saran dalam penulisan skripsi ini serta mendampingi saya selaku mahasiswa;
3. Ibu Nelly Oktavia A, S.Si.,MT selaku dosen penguji utama dan Bapak Gama Wisnu Fajarianto, S.Kom., M.Kom selaku dosen penguji anggota yang telah berkenan menguji skripsi ini dan memberikan masukan serta saran demi sempurnanya skripsi ini;
4. Seluruh Bapak dan Ibu Dosen serta Staff Karyawan Fakultas Ilmu Komputer Universitas Jember;
5. Ayahanda Mochammad Taufik Hidayat dan Ibunda Suci Setyo yang telah memberikan dukungan dan doanya selama masa perkuliahan hingga skripsi ini terselesaikan;
6. Saudari Atiqah Hani Rahmatillah dan Suami yang selalu mendukung dan memberi motivasi;
7. Keluarga besar yang telah mendukung dan mendoakan;

8. Keluarga besar Badan Perwakilan Mahasiswa Fakultas Ilmu Komputer Universitas Jember, 2020/2021 yang telah mengajarkan banyak pengalaman baru;
9. Mas Frandy, Mas Ridlo, Mas Satrio, Mbak Jahrotus, Mbak Erlin, Mbak Mutiya serta kakak-kakak Badan Perwakilan Mahasiswa Fakultas Ilmu Komputer Universitas Jember yang lain yang telah mengajarkan banyak pengalaman berorganisasi;
10. Mas Marco, Mbak Jahrotus, Mbak Fadah, yang telah mengajarkan banyak ilmu dalam penulisan dan perhitungan penelitian skripsi;
11. Keluarga Besar Anak Tuhan yang selalu memberi dukungan dimasa sulit
12. Keluarga Besar Kosan Band yang selalu menemani dan memberikan semangat semasa kuliah;
13. Teman – teman Tretan TI 18 yang selalu memberikan semangat sejak mahasiswa baru hingga akhir;
14. Seluruh Mahasiswa Fakultas Ilmu Komputer Universitas Jember yang telah mendukung selama menempuh pendidikan S1; dan
15. Seluruh pihak yang tidak dapat disebutkan satu-persatu.

DAFTAR ISI

HALAMAN JUDUL	i
PERSEMBAHAN.....	ii
MOTTO	iii
PERNYATAAN ORISINALITAS.....	iv
HALAMAN PERSETUJUAN	v
RINGKASAN	vii
PRAKATA	viii
DAFTAR ISI.....	x
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xiv
DAFTAR NOTASI.....	xvi
DAFTAR ISTILAH DAN SINGKATAN	xvii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Penelitian	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB 2. TINJAUAN TEORI.....	5
2.1 Kajian Literatur	5
2.2 Pariwisata	6
2.3 Covid-19.....	7
2.4 Peramalan (<i>Forecasting</i>).....	7
2.5 ARIMA.....	8
2.6 NEURAL NETWORK.....	10
2.7 MODEL HYBRID ARIMA-FFNN	11
2.8 MAPE.....	11
BAB 3. METODOLOGI PENELITIAN	12
3.1 Lokasi dan Waktu Penelitian.....	12
3.2 Populasi dan Sampel/Subyek Penelitian	12
3.3 Prosedur Penelitian.....	12
3.3.1. <i>Pengolahan data</i>	12
3.3.2. <i>Visualisasi Data</i>	13

3.3.3. Analisis Data	13
3.3.4. Pembuatan model ARIMA	13
3.3.5. Pengujian model ARIMA	13
3.3.6. Pembagian dataset	13
3.3.7. Pembuatan dan pengujian model FFNN.....	14
3.3.8. Pengujian Hybrid	14
3.3.9. Peramalan ARIMA	14
3.3.10. Peramalan FFNN	14
3.3.11. Peramalan Hybrid.....	14
BAB 4. HASIL DAN PEMBAHASAN	15
4.1 Pengolahan Data.....	15
4.2 Visualisasi Data.....	15
4.3 Analisis Data	17
4.4 Pembuatan Model ARIMA	18
4.5 Pengujian Model ARIMA	20
4.6 Pembagian Dataset	21
4.7 Pembuatan dan Pengujian Model FFNN.....	21
4.8 Pengujian <i>Hybrid</i>	23
4.9 Peramalan ARIMA.....	25
4.10 Peramalan FFNN	26
4.11 Peramalan <i>Hybrid</i>	26
BAB 5. KESIMPULAN, KETERBATASAN, DAN SARAN.....	33
5.1 Kesimpulan.....	33
5.2 Saran.....	34
DAFTAR PUSTAKA	35

DAFTAR TABEL

Tabel 1 Nilai Peramalan ARIMA musiman tahun 2020	23
Tabel 2 Nilai Peramalan ARIMA musiman 6 bulan	27

DAFTAR GAMBAR

Gambar 4.1 Grafik Jumlah Pengunjung Wisata.....	15
Gambar 4.2 Code Program Pemberian Index	16
Gambar 4.3 Visualisasi Dataset Dengan Nilai 0.....	16
Gambar 4.4 Uji ADF.....	17
Gambar 4.5 Dekomposisi Data	17
Gambar 4.6 Plot ACF dan PACF.....	18
Gambar 4.7 Model ARIMA (1,0,1)	18
Gambar 4.8 Model ARIMA (1,0,0)	19
Gambar 4.9 Plot ACF dan PACF musiman	19
Gambar 4.10 Plot Residual dan Plot Density.....	20
Gambar 4.11 Uji Nilai Error	20
Gambar 4.12 Code Program untuk Menormalisasi Dataset.....	21
Gambar 4.13 Pengujian Model pada Dimensi	21
Gambar 4.14 Code Program Peramalan FFNN.....	22
Gambar 4.15 Code Program Inverse Normalisasi.....	22
Gambar 4.16 Menghitung Nilai Error menggunakan RMSE	22
Gambar 4.17 Peramalan ARIMA musiman hybrid.....	23
Gambar 4.18 Pembaruan Dataset.....	24
Gambar 4.19 Peramalan ARIMA tahun 2022.....	25
Gambar 4.20 Peramalan FFNN tahun 2022.....	26
Gambar 4.21 Dataset Peramalan Hybrid model FFNN	27
Gambar 4.22 Grafik Peramalan Hybrid	28
Gambar 4.23 Perbandingan Grafik Peramalan Hybrid Dengan Dataset Nilai Rata-Rata	30
Gambar 4.24 Peramalan ARIMA Musiman Tahun 2022 Menggunakan Dataset Nilai Rata-Rata.....	31
Gambar 4.25 Grafik Peramalan Hybrid Menggunakan Dataset Nilai Rata-Rata .	32

DAFTAR LAMPIRAN

Lampiran 1 Tabel Jumlah Pengunjung Objek Wisata Kabupaten Situbondo	39
Lampiran 2 Tabel Jumlah Pengunjung Wisata Pantai Pasir Putih Situbondo tahun 2017-2021	41
Lampiran 3 library dan import dataset	42
Lampiran 4 visualisasi dataset	42
Lampiran 5 Peramalan Data Testing	43
Lampiran 6 Pembagian Dataset untuk Pembuatan Model FFNN	43
Lampiran 7 Pembutan Dimensi FFNN	43
Lampiran 8 Pembuatan Model FFNN	44
Lampiran 9 Visualisasi Peramalan Data Testing FFNN	44
Lampiran 10 Dataset Pengujian Hybrid model FFNN	45
Lampiran 11 Normalisasi Data	45
Lampiran 12 Pembuatan Dimensi	45
Lampiran 13 Pembuatan Dimensi	45
Lampiran 14 Pembuatan Model FFNN	46
Lampiran 15 Pengujian Model pada Dimensi	46
Lampiran 16 Peramalan Tahun 2021	46
Lampiran 17 Inverse Normalisasi Nilai Peramalan 2021	47
Lampiran 18 Pengambilan Index dari Dataset	47
Lampiran 19 Pemberian Index pada Nilai Peramalan	47
Lampiran 20 Visualisasi Nilai Peramalan	48
Lampiran 21 Membulatkan Nilai Peramalan	48
Lampiran 22 Menggabungkan Nilai Peramalan pada Dataset	49
Lampiran 23 Visualisasi Peramalan Hybrid	49
Lampiran 24 Peramalan ARIMA musiman 6 bulan	50
Lampiran 25 Nilai peramalan ARIMA musiman 6 bulan	50
Lampiran 26 Peramalan Hybrid model FFNN	50
Lampiran 27 Inverse Normalisasi	51
Lampiran 28 Pemberian Index Tanggal	51
Lampiran 29 Visualisasi Nilai Peramalan	51

Lampiran 30 Pembulatan nilai Peramalan	52
Lampiran 31 Penggabungan Nilai Peramalan pada Dataset.....	52
Lampiran 32 Pengisian Nilai Kosong Menggunakan Nilai Rata-Rata	53
Lampiran 33 Grafik Jumlah Pengunjung Wisata Menggunakan Nilai Rata-Rata	53
Lampiran 34 Tes ADF	53
Lampiran 35 Dekomposisi Dataset Rata-Rata	54
Lampiran 36 Model ARIMA (8,0,0).....	54
Lampiran 37 Model ARIMA (17,0,0).....	55
Lampiran 38 Plot ACF dan PACF musiman	55
Lampiran 39 Model ARIMA musiman.....	56
Lampiran 40 Plot Residual dan Plot Density	56
Lampiran 41 Grafik Prediksi Data Testing.....	57
Lampiran 42 Uji Nilai Error	57
Lampiran 43 Grafik Peramalan Data Testing ARIMA Musiman	58
Lampiran 44 Nilai Peramalan Data Testing ARIMA Musiman	58
Lampiran 45 Pembaruan Dataset Peramalan Data Testing	59
Lampiran 46 Grafik Dataset Baru	59
Lampiran 47 Pembuatan Model FFNN	59
Lampiran 48 Pembuatan Dimensi FFNN	60
Lampiran 49 Nilai Peramalan ARIMA musiman tahun 2022 Menggunakan Dataset Rata-Rata	60
Lampiran 50 Pembaruan Dataset Peramalan Hybrid	61
Lampiran 51 Pemberian Index Tanggal Pada Nilai Peramalan Hybrid Model FFNN	61
Lampiran 52 Nilai Peramalan Hybrid Model FFNN	62
Lampiran 53 Link Lampiran Online	62

DAFTAR NOTASI

X_t	: Nilai variabel X pada waktu ke-t
Φ_p	: Parameter <i>autoregressive</i> ke-p
α_t	: Nilai <i>error</i> pada waktu ke-t
θ_q	: Nilai koefisien orde q
$\omega_t / \nabla^d X_t$: Pembeda ordo ke-d
\hat{L}_t	: Model ARIMA
\hat{a}_t	: Model NEURAL NETWORK

DAFTAR ISTILAH DAN SINGKATAN

Singkatan/Istikal	Arti dan keterangan
BPS	Badan Pusat Statistik
ARIMA	<i>Autoregressive Moving Average</i>
FFNN	<i>Feed-Forward Neural Network</i>
<i>Hybrid</i>	Gabungan
LSTM	<i>Long Short Term Memory</i>

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Pariwisata adalah bepergian secara berulang-ulang, pariwisata merupakan industri terbesar saat ini. Berdasarkan Undang-Undang Pariwisata no 10 tahun 2009, industri pariwisata adalah kumpulan beberapa usaha barang dan atau jasa untuk memenuhi kebutuhan wisatawan. Kristalina Georgieva sebagai Direktur Pelaksana IMF menyatakan bahwa adanya wabah Covid- 19 sejak 3 Maret 2020 menyebabkan penurunan jumlah pengunjung wisata yang berdampak pada perlambatan sector perekonomian di Indonesia. Badan Pusat Statistik (BPS) mendata terjadi penyusutan kunjungan wisata di Indonesia sebesar 7.62% dari 1.37 juta pengunjung wisata di bulan Desember tahun 2019. Warga yang bertempat tinggal di sekitar Pantai Pasir Putih Situbondo mengalami permasalahan ekonomi akibat dari penurunan jumlah pengunjung wisata yang sangat drastis.

Dinas Pariwisata dan pihak pengelola setempat perlu melakukan promosi yang cukup besar dan luas tentang *event-event* yang akan diadakan. Promosi tersebut akan menunjukkan perubahan besar dari aspek perhotelan, kuliner, transportasi, dan paket perjalanan wisata di berbagai daerah, akibatnya terjadi ketimpangan antara infrastruktur dengan peningkatan jumlah wisatawan. Berdasarkan terjadinya ketimpangan tersebut, maka diperlukan peramalan jumlah pengunjung wisata pada masa mendatang untuk menyiapkan kebutuhan pembangunan sarana dan prasarana. agar mampu meningkatkan jumlah kunjungan wisata dengan tetap memberikan rasa aman dari penyebaran virus Covid-19 dan dapat meningkatkan kembali perekonomian masyarakat sekitar. Penelitian ini menggunakan data jumlah pengunjung wisata bulanan dari tahun 2017 hingga tahun 2021, penelitian ini mengimplementasikan metode *time series arima* dan *feed-forward neural network* serta dilengkapi dengan skor akurasi peramalan *Mean Absolute Percentage Error* (MAPE).

Metode ARIMA adalah metode yang digunakan untuk data deret waktu satu variable (*univariate*) yaitu data yang mengalami fluktuasi bersamaan dengan bertambahnya periode waktu. Metode ARIMA (p,d,q) merupakan perpaduan metode *Autoregressive* dan *Moving Average*, p dinotasikan sebagai AR, d sebagai hasil dari *differencing*, dan q sebagai MA. ARIMA sering digunakan untuk peramalan model acak deret waktu, model ARIMA lebih akurat untuk metode prediksi jangka pendek. Model ARIMA lebih memudahkan untuk pemahaman struktur dan karakteristik data deret waktu. Metode *Feed-Forward Neural Network* adalah pemrosesan data dari *input layer* menuju *hidden layer* sampai *output layer* tanpa melakukan proses *cycle / loop*. *Neural Network* memiliki kelebihan yaitu dapat menemukan hubungan antara input dan output dari data yang rumit, *Neural Network* sering digunakan untuk peramalan bisnis, pengolahan suara, penglihatan digital, sistem kontrol, dan rating karena nilai output yang dihasilkan mendekati nilai sebenarnya. *Neural Network* memiliki elemen-elemen pemrosesan informasi yang disebut neuron, unit, sel, atau node, setiap node terhubung dengan node lain dengan penghubung yang memiliki bobot. Penggabungan dua metode ini dilakukan dengan cara memproses data pengunjung wisata menggunakan metode ARIMA terlebih dahulu yang kemudian *output* dari metode ARIMA dijadikan *inputan* untuk proses perhitungan dengan metode *Feed-Forward Neural Network*.

1.2 Rumusan Masalah

Rumusan masalah dari penjelasan latar belakang diatas dapat diuraikan sebagai berikut:

1. Bagaimana proses peramalan jumlah wisatawan menggunakan metode *time series arima* dan *feed forward neural network* untuk peramalan di tahun 2022?
2. Bagaimana hasil peramalan jumlah wisatawan menggunakan metode *time series arima* dan *feed forward neural network* untuk peramalan di tahun 2022?

3. Bagaimana akurasi hasil peramalan jumlah wisatawan menggunakan metode *time series arima* dan *feed forward neural network* untuk peramalan di tahun 2022?

1.3 Batasan Penelitian

Batasan masalah dalam penelitian ini yaitu:

1. Menggunakan data pengunjung tempat wisata Pasir Putih Situbondo
2. Menggunakan data pengunjung dari tahun 2017 – 2021
3. Metode yang digunakan adalah metode *time series arima* dan *feed-forward neural network*

1.4 Tujuan Penelitian

Tujuan penelitian ini yaitu:

1. Menerapkan metode *time series arima* dan *feed forward neural network* untuk peramalan jumlah wisatawan di tahun 2022
2. Mengetahui hasil peramalan pengunjung wisata menggunakan metode *time series arima* dan *feed forward neural network* di tahun 2022
3. Mengetahui akurasi hasil peramalan pengunjung wisata menggunakan metode *time series arima* dan *feed forward neural network* di tahun 2022

1.5 Manfaat Penelitian

Manfaat penelitian ini dapat diruaikan sebagai berikut:

1. **Bagi Akademis.** Diharapkan dapat menyajikan informasi yang berkaitan dengan judul penelitian terhadap pembaca dan terhadap Program Studi Teknologi Informasi Universitas Jember
2. **Bagi Peneliti.** Dapat mengimplementasikan ilmu pengetahuan yang telah dipelajari untuk membantu tempat wisata Pasir Putih, Situbondo. Selain itu menambah referensi penulis tentang peramalan secara datamining

3. **Bagi Objek Penelitian.** Mendapatkan nilai peramalan jumlah kunjungan wisata di tahun mendatang agar dapat memilih waktu atau bulan dengan jumlah pengunjung tertinggi untuk mengadakan *event-event* yang dapat mengangkat kembali perekonomian masyarakat sekitar.

BAB 2. TINJAUAN TEORI

2.1 Kajian Literatur

Penelitian yang dilakukan oleh Li Lujin, Wang Yaru, Li, Xiaoyue pada tahun 2020 dengan judul “Tourists Forecast Lanzhou Based on the Baolan High-Speed Railway by the Arima Model” penelitian ini melakukan peramalan kunjungan wisata di Kota Lanzhou dengan data penumpang kereta dengan tujuan Kota Lanzhou dari tahun 2009 hingga tahun 2019 menggunakan metode ARIMA. Peneliti berpendapat bahwa metode ARIMA cocok dengan nilai acak data deret waktu dan menunjukkan hasil lebih akurat untuk peramalan jangka pendek (*short-term prediction*). Penelitian ini menunjukkan hasil peramalan sebesar 81.045.529 jumlah pengunjung dengan nilai error 1.03%.

Penelitian yang dilakukan oleh Laila, Sri Wahyuningsih, dan Fidia Deny tahun 2019 dengan judul “Peramalan Harga Minyak Mentah Menggunakan Model *Autoregressive Integrated Moving Average Neural Network* (ARIMA-NN) ” penelitian ini meramalkan harga minyak mentah pada periode mendatang, penelitian ini menggunakan data harga minyak mentah bulan July 2010 hingga bulan Desember 2017. Model ARIMA-NN terbaik yang digunakan adalah ARIMA (2,1,0) dengan komponen 2 node input dan 2 node neuron *hidden layer* untuk *Neural Network* yang digunakan. Penelitian ini menghasilkan peramalan harga minyak mentah di Indonesia untuk bulan Januari hingga Desember 2018 berada di kisaran 60 USD / barrel

Penelitian yang dilakukan oleh Shabri Ani, Samsudin Ruhaidah, dan Yussof Yusliza pada tahun 2020 dengan judul “Combining Deep Neural Network and Four Series for Tourist Arrival Forecasting” melakukan peramalan jumlah pengunjung wisata pada Kepulauan Langkawi, Malaysia. Penelitian ini menggunakan metode Deep Neural Network dengan menggunakan data pengunjung wisata dari bulan Januari tahun 2002 hingga Desember 2016. Untuk menguji tingkat akurasi dari metode yang digunakan, peneliti juga melakukan peramalan dengan menggunakan metode ARIMA, ANN, dan hybrid ARIMA ANN. Penelitian ini menunjukkan hasil

bahwa Four series-Deep Neural Network (FDNN) menghasilkan nilai peramalan yang sangat mirip dengan nilai data asli.

Penelitian yang dilakukan oleh Hermawati dan Latif pada tahun 2021 dengan judul “Forecasting tourist visits using data decomposition technique and learning optimization of artificial neural network” melakukan peramalan terhadap salah satu tempat wisata di daerah Sumenep, Madura, Indonesia dengan menggunakan metode Empirical Mode Decomposition (EEMD) yang di hybrid dengan metode Feed Forward Neural Network (FFNN) untuk menghasilkan nilai peramalan yang akurat. Penelitian ini menggunakan data bulanan dari tahun 2015 hingga tahun 2019, data diproses menggunakan metode dekomposisi untuk mengurutkan data dari data dengan frekuensi tertinggi ke terendah dengan standar deviasi sebesar 0.2, kemudian nilai residual data dinormalisasi untuk dilakukan perhitungan dengan metode FFNN. Penelitian ini mendapatkan hasil bahwa model 4-30-1 adalah model FFNN terbaik untuk peramalan dengan nilai RMSE sebesar 0.14592 dan nilai MSE sebesar 0.02129.

2.2 Pariwisata

Pariwisata berasal dari Bahasa sansekerta yaitu pari dan wisata. Pari bermakna berulang dan wisata artinya bepergian, sehingga pariwisata berarti bepergian berulang kali. Berpergian dapat dikatakan sebagai perjalanan wisata jika memenuhi 3 syarat, yaitu 1. Tidak dilakukan dalam waktu yang lama. 2. Tidak dilakukan karena paksaan orang lain. 3. Tidak dijadikan sebagai profesi. Para ahli sejarah kepariwisataan berpendapat bahwa elemen penting pariwisata yang perlu dikelola yaitu: jenis dan tujuan kegiatan kunjungan, lokasi kegiatan yang dituju, aksesibilitas lokasi tujuan, rencana lama tinggal di lokasi wisata, sarana prasarana lokasi wisata.

Kata wisatawan yang berasal dari kata “wisata” dan mendapat akhiran “wan” memiliki makna orang yang melakukan perjalanan wisata. Setiap wisatawan memiliki budaya dan ketertarikan berwisata yang berbeda-beda, untuk memenuhi kebutuhan setiap wisatawan maka diperlukan pendekatan yang lebih spesifik. Citra

positif pada tempat wisata ditimbulkan dari rasa puas terhadap lokasi yang mudah diakses, pemenuhan sarana prasarana, dan pelayanan yang berkualitas.

2.3 Covid-19

Kasus Covid-19 pertama di Indonesia tercatat pada tanggal 2 Maret 2020. *International Committee on Taxonomy of Viruses* menamai penyakit ini dengan SARS-CoV-2 karena dari hasil studi filogenetik menunjukkan bahwa Covid-19 diidentifikasi sebagai virus dengan subgenus yang sama dengan *sarbecovirus*. Dari hasil penelitian diperoleh hipotesis bahwa SARS-CoV-2 berasal dari kelelawar, karena sekuens SARS-CoV-2 serupa dengan *coronavirus* pada kelelawar yang kemudian bermutasi dan menginfeksi manusia. Covid-19 menular pada manusia melalui droplet saat batuk atau bersin. Covid-19 tidak hanya menginfeksi saluran pernafasan tetapi juga menginfeksi saluran pencernaan, terbukti dari 23% pasien masih terdeteksi virus Covid-19 pada fesesnya.

Covid-19 bermutasi pada saluran pernafasan dengan cara menempel dan menembus sel reseptor, kemudian memadukan protein yang diperlukan untuk membentuk sel baru yang akan timbul di permukaan saluran pernafasan. Jika imun tidak kuat dalam melawan virus maka akan terjadi replikasi virus, namun jika imun menanggapi berlebihan maka akan berdampak kerusakan jaringan dalam tubuh, hal ini dikarenakan sifat sitopatik virus COVID-19.

2.4 Peramalan (*Forecasting*)

Peramalan diartikan sebagai kegiatan menganalisa kebutuhan di masa depan. Peramalan bertujuan untuk mengurangi tingkat kesalahan antara pengambil keputusan dan pembuat kebijakan dalam memikirkan tentang ketidakpastian dan akibat yang dapat muncul saat membuat rencana atau ketetapan yang berorientasi untuk masa depan. Buffa sebagai peneliti terdahulu menyatakan bahwa peramalan adalah teknik statistika untuk menentukan keadaan di masa depan berdasarkan pengolahan data dengan cara-cara tertentu yang didapatkan di masa lampau. Peramalan dibutuhkan karena adanya perbedaan waktu antara keadaan dan

kebijakan, menentukan waktu suatu peristiwa akan berlangsung sehingga sanggup menyiapkan langkah yang dibutuhkan.

2.5 ARIMA

Aswi dan Sukarna pada tahun 2006 menyatakan bahwa metode ARIMA digunakan untuk data deret waktu satu variable (univariate). Metode ARIMA merupakan penggabungan dari dua metode, yaitu *autoregressive* dan *moving average*. Metode ARIMA dinotasikan sebagai (p,d,q) dengan p mewakili *autoregressive* (AR), d mewakili proses *differencing*, dan q mewakili *moving average*. Dalam penggunaan metode arima diperlukan pemilihan model arima dengan residu terbaik, yaitu yang bersifat white noise atau bersifat homoskedastisitas dan berdistribusi normal. Untuk mendapatkan model terbaik ada beberapa tahap yang perlu dilakukan yaitu identifikasi model, estimasi parameter, uji diagnostic, evaluasi model, dan peramalan.

Model AR (*autoregressive*)

$$\begin{aligned}x_t &= a_t + \phi a_{t-1} + \phi^2 a_{t-2} + \dots \\&= a_t + \phi(a_{t-1} + \phi a_{t-2} + \dots) \\&= \phi x_{t-1} + a_t \\x_t - \phi x_{t-1} &= a_t \\(1 - \phi B)x_t &= a_t\end{aligned}$$

$$\begin{aligned}x_t &= (1 - \phi B)^{-1} a_t = (1 + \phi B + \phi^2 B^2 + \dots) a_t \\&= a_t + \phi a_{t-1} + \phi^2 a_{t-2} + \dots\end{aligned}$$

$$x_t - \phi_1 x_{t-1} - \phi_2 x_{t-2} - \dots - \phi_q x_{t-q} = a_t$$

Model MA (*moving average*)

$$x_t = a_t - \theta a_{t-1}$$

$$x_t = (1 - \theta B)a_t$$

$$(1 - \theta B)^{-1}x_t = a_t$$

$$(1 + \theta B + \theta^2 B^2 + \dots)x_t = a_t$$

$$x_t = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

$$x_t = (1 - \theta_1 B - \dots - \theta_q B^q)a_t = \theta(B)a_t$$

Model ARIMA

ARIMA adalah penggabungan dari AR dan MA yang di tambahkan *differencial(I)*

ARMA

$$x_t - \phi x_{t-1} = a_t - \theta a_{t-1}$$

$$(1 - \phi B)x_t = (1 - \theta B)a_t$$

$$x_t - \phi_1 x_{t-1} - \dots - \phi_p x_{t-p} = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$

$$(1 - \phi_1 B - \dots - \phi_p B^p)x_t = (1 - \theta_1 B - \dots - \theta_q B^q)a_t$$

$$\phi(B)x_t = \theta(B)a_t$$

Differencial (pembeda ordo)

$$\nabla x_t = \theta_0 + a_t$$

Sehingga model ARIMA

$$\phi(B)\nabla^d x_t = \theta_0 + \theta(B)a_t$$

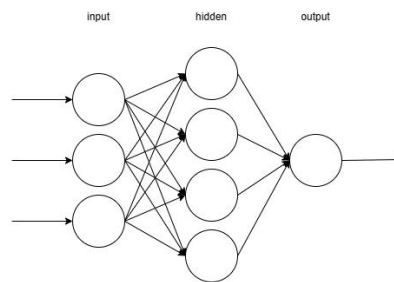
$$\phi(B)\nabla^d x_t = \theta(B)a_t$$

$$\phi(B)w_t = \theta(B)a_t$$

$$w_t = \nabla^d x_t$$

2.6 NEURAL NETWORK

Neural Network atau Jaringan Saraf Tiruan adalah pemrosesan informasi dengan algoritma yang bekerja seperti saraf otak manusia. Neural Network memiliki elemen-elemen pemrosesan informasi yang disebut neuron, unit, sel, atau node. Neural Network memiliki 3 layer dalam pemrosesan informasi, yaitu input layer, hidden layer, dan output layer. Input layer adalah node-node yang membawa data masuk untuk di proses pada layer selanjutnya, hidden layer adalah node-node dengan penambahan bobot untuk pemrosesan informasi, output layer adalah node-node untuk informasi yang telah di proses.



Proses feed forward neural network

1. Setiap node input menyebarkan data pada node hidden layer
2. Setiap node hidden layer menjumlahkan nilai inputan

$$z_{in_j} = \sum_{i=1}^n x_i v_{ij}$$

Nilai penjumlahan di proses dalam fungsi aktivasi sigmoid

Fungsi aktivasi sigmoid

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

$$z_j = f(z_{in_j})$$

Kemudian mengirim nilai hasil aktivasi sigmoid ke semua node output

3. Setiap node output layer menjumlahkan nilai masukan dari hidden layer

$$y_in_k = \sum_{j=1}^n z_j w_{jk}$$

Nilai penjumlahan di proses dalam fungsi aktivasi sigmoid

Fungsi aktivasi sigmoid

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

$$y_k = f(y_in_k)$$

2.7 MODEL HYBRID ARIMA-FFNN

Zhen dan Zhong pada tahun 2015 menyatakan bahwa model ARIMA-NN adalah kombinasi atau *hybrid* dari dua metode atau lebih dalam satu fungsi. Pada penelitian ini digunakan kombinasi metode antara ARIMA dan NEURAL NETWORK. Hal ini digunakan untuk saling melengkapi kekurangan dari setiap metode dan adanya deret waktu yang linear serta non linear. Secara umum kombinasi ARIMA-NN dituliskan sebagai berikut

$$\widehat{N}_t = \widehat{L}_t + \widehat{a}_t$$

2.8 MAPE

Setiap metode peramalan memiliki tingkat akurasi masing-masing. Semakin kecil nilai *error*nya maka semakin akurat hasil peramalan. *Mean Absolute Percentage Error* (MAPE) adalah metode evaluasi yang digunakan untuk mengukur akurasi peramalan. MAPE mengukur rata-rata dari nilai *error* absolute sebagai persentase dari nilai rata-rata *error rate* absolute periode data actual.

$$MAPE = \left(\frac{1}{N}\right) \sum_{t=1}^n \frac{|X_t - F_t|}{X_t} \times 100\%$$

BAB 3. METODOLOGI PENELITIAN

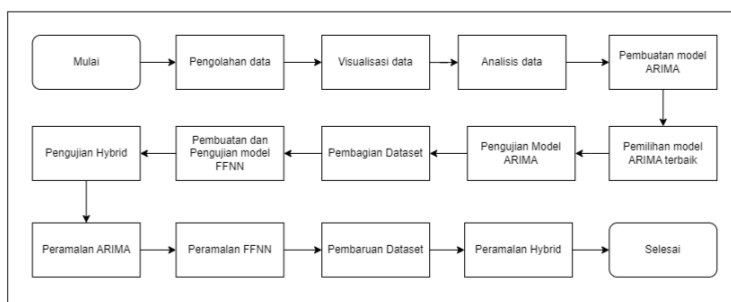
3.1 Lokasi dan Waktu Penelitian

Lokasi penelitian dilaksanakan di Pantai Pasir Putih, Kecamatan Bungatan, Kabupaten Situbondo, Jawa Timur, Indonesia. Waktu penelitian dimulai pada bulan Mei 2022 hingga bulan Desember 2022.

3.2 Populasi dan Sampel/Subyek Penelitian

Pantai Pasir Putih Situbondo berlokasi di Kecamatan Bungata, Kabupaten Situbondo, Provinsi Jawa Timur. Data pengunjung wisata didapatkan dari permintaan data kepada Kantor Dinas Pariwisata, Pemuda, dan Olahraga Kabupaten Situbondo, data yang diterima dalam bentuk file *excel* yang berisikan objek wisata, alamat, jumlah wisatawan nusantara dan wisatawan mancanegara perbulan, dan total wisatawan nusantara dan wisatawan mancanegara dalam satu tahun.

3.3 Prosedur Penelitian



3.3.1. Pengolahan data

Tahap ini adalah tahap untuk menjelaskan data yang akan digunakan sebagai dataset. data yang digunakan dalam penelitian ini adalah data pengunjung wisata Pantai Pasir Putih Situbondo dari tahun 2017 hingga tahun 2021. Tahap ini juga tahap mengolah data kotor menjadi data bersih dengan mengambil jumlah pengunjung wisata Pantai Pasir Putih, menjumlah wisatawan nusantara dan

wisatawan mancanegara, dan menjadikan kolom bulan sebagai kolom index. Data yang telah bersih dari atribut yang tidak digunakan akan menjadi dataset dalam analisis yang dilakukan

3.3.2. Visualisasi Data

Tahap ini adalah tahap pengolahan data pengunjung wisata menjadi dataset yang akan di olah pada *tools Jupyter Notebook* dalam bahasa pemrograman *python* versi 3 dengan berbagai macam *library* yang digunakan untuk memvisualisasikan data.

3.3.3. Analisis Data

Tahap ini adalah tahap peneliti menganalisa dataset yang digunakan untuk menentukan proses perhitungan selanjutnya. Hasil dari Analisa dataset menunjukkan bahwa dataset memiliki tipe data musiman, dengan hasil uji ADF menunjukkan bahwa dataset adalah data stasioner.

3.3.4. Pembuatan model ARIMA

Tahap ini adalah tahap pembuatan model ARIMA dari dataset yang telah di analisis. Karena dataset adalah data stasioner maka tidak perlu melakukan differencing (i), kemudian karena dataset memiliki tipe data musiman, maka ditambahkan parameter (m) yaitu jumlah observasi dalam satu tahun.

3.3.5. Pengujian model ARIMA

Tahap ini adalah tahap pembagian dataset menjadi data training dan data testing sebanyak 36 data training dan 24 data testing, kemudian model ARIMA yang telah dipilih di ujikan terhadap data training dan data testing.

3.3.6. Pembagian dataset

Tahap ini adalah tahap pembagian dataset baru menjadi data training dan data testing sebanyak 36 data training dan 24 data testing, kemudian nilai data training

dan data testing di normalisasikan dengan interval 0-1 agar lebih mudah dilakukan proses perhitungan, selanjutnya pembuatan dimensi dari data training dan data testing yang digunakan untuk proses pengujian model.

3.3.7. Pembuatan dan pengujian model FFNN

Tahap ini adalah tahap pembuatan model FFNN dengan menginisialisasi jumlah node input, jumlah hidden layer, jumlah node hidden layer, dan jumlah output layer. Kemudian model yang dibuat diujikan terhadap dimensi data training dan data testing untuk mendapatkan model FFNN dengan nilai *val_loss* terkecil.

3.3.8. Pengujian Hybrid

Tahap ini adalah tahap penggabungan model ARIMA dengan model FFNN. Dataset dibagi menjadi 36 data training dan 24 data testing, 12 data testing pertama dilakukan proses perhitungan dengan menggunakan model ARIMA dan 12 data testing kedua dilakukan dengan proses perhitungan model FFNN.

3.3.9. Peramalan ARIMA

Tahap ini adalah tahap peramalan dengan model ARIMA untuk memperkirakan jumlah pengunjung wisata 1 tahun selanjutnya.

3.3.10. Peramalan FFNN

Tahap ini adalah tahap peramalan dengan model FFNN untuk memperkirakan jumlah pengunjung wisata 1 tahun selanjutnya.

3.3.11. Peramalan Hybrid

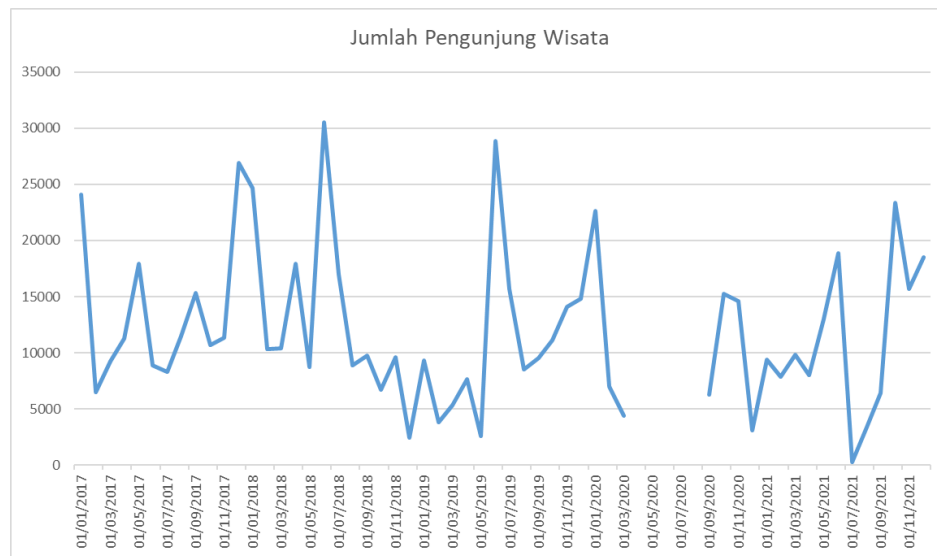
Tahap ini adalah tahap peramalan jumlah pengunjung wisata untuk bulan Januari 2022 hingga bulan Juni 2022 menggunakan model ARIMA musiman, kemudian hasil dari peramalan ARIMA musiman disimpan pada dataset awal untuk dijadikan dataset pada proses peramalan jumlah pengunjung wisata bulan Juli 2022 hingga bulan Desember 2022 menggunakan metode FFNN

BAB 4. HASIL DAN PEMBAHASAN

4.1 Pengolahan Data

Total data yang didapatkan sebanyak 360 data. Data yang terkumpul dilakukan *pre-processing* data, sehingga atribut yang tidak digunakan untuk penelitian, (contoh: alamat, objek wisata selain pasir putih, kolom wisnus dan wisman) dihapus atau digabungkan. Hasil dari *pre-processing* data menjadi 60 data.

Pada lampiran 1 menjelaskan tentang data pengunjung pada beberapa objek wisata tahun 2017 di Kabupaten Situbondo. Pengunjung didominasi oleh wisatawan lokal dan beberapa wisatawan asing, Pantai Pasir Putih Situbondo adalah objek wisata dengan jumlah pengunjung tertinggi. Dari lampiran 2 kemudian divisualisasikan menjadi grafik pada gambar 4.1. pada grafik tersebut menunjukkan bahwa pengunjung pada tahun 2020 hingga tahun 2021 mengalami penurunan dibanding tahun-tahun sebelumnya.



Gambar 4.1 Grafik Jumlah Pengunjung Wisata

4.2 Visualisasi Data

Pada tahap pengolahan data, atribut pada lampiran 1 akan di konversi agar dapat diolah pada *tools* dan *library* data mining. Atribut bulan dikonversi menjadi tipe *date-time* sebagai index dataset dengan nama *date* dan atribut wisnus dan wisman dikonversi menjadi tipe *int64* sebagai value dari index dengan nama *value*.

Lampiran 3 adalah proses *import file* excel menjadi dataset dengan nama variable “df”

```
In [5]: df['date'] = pd.to_datetime(df['date'], infer_datetime_format=True)
df = df.set_index(['date'])
#mengatur index dataset pada kolom date

In [6]: df.head()
Out[6]:
```

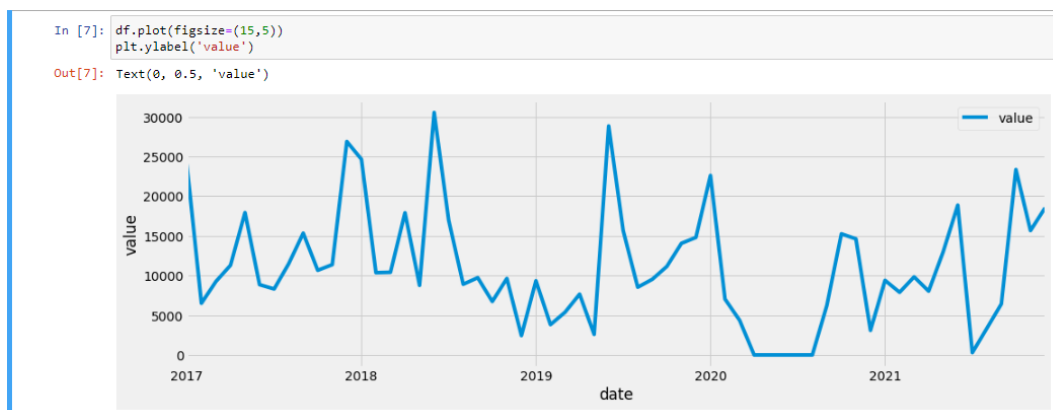
date	value
2017-01-31	24064
2017-02-28	6510
2017-03-31	9264
2017-04-30	11304
2017-05-31	17940

Gambar 4.2 Code Program Pemberian Index

Gambar 4.2 adalah code program pemberian index pada kolom date. Selanjutnya dataset divisualisasikan sehingga terbentuk grafik jumlah pengunjung wisata Pantai Pasir Putih Situbondo seperti pada lampiran 4.

Pada dataset terdapat 5 nilai yang kosong karena adanya penutupan wisata, yaitu pada bulan april, mei, juni, juli, agustus 2020. Untuk mengatasi hal tersebut dapat dilakukan dengan pengisian nilai kosong menggunakan nilai rata-rata pengunjung atau menghapus nilai kosong. Penghapusan nilai kosong berpengaruh pada jumlah dataset yang mengakibatkan index tidak berurutan, sehingga data tidak dapat di proses, untuk mengatasinya data diberi nilai 0.

Pengujian pertama ARIMA dilakukan dengan pemberian nilai 0 pada file excel yang kemudian diimport kembali pada jupyter notebook. Sehingga grafik dataset akan terlihat seperti gambar 4.3



Gambar 4.3 Visualisasi Dataset Dengan Nilai 0

4.3 Analisis Data

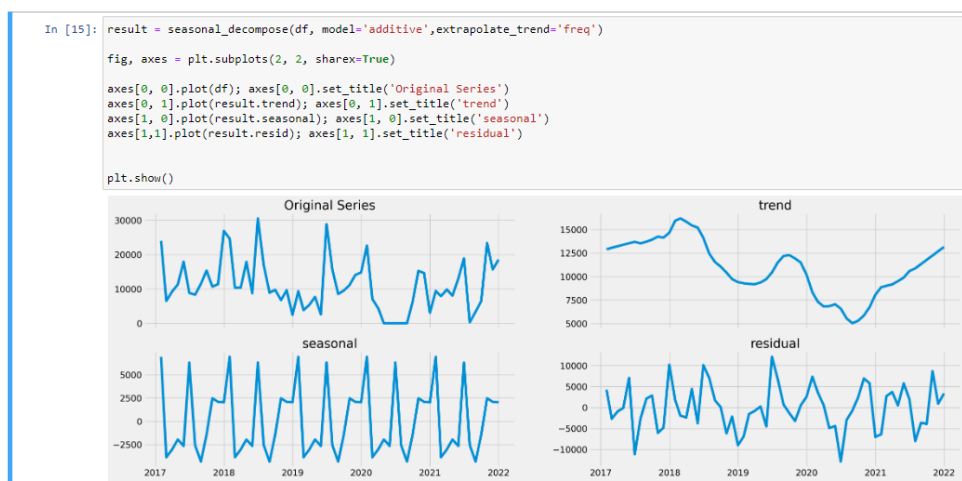
Terdapat 4 pola tipe data *time series* yaitu horizontal, trend, musiman, dan siklus. Sedangkan terdapat 2 model data *time series* yaitu stasioner dan non-stasioner.

```
In [8]: from statsmodels.tsa.stattools import adfuller
res = adfuller(df.value)
print('Augmented Dickey-Fuller Statistic: %f' % res[0])
print('p-value: %f' % res[1])
print('critical values at different levels:')
for k, v in res[4].items():
    print('\t%s: %.3f' % (k, v))

#menghitung adfuller
#p-val dengan nilai diatas 0.05 maka datanya tidak stasioner
#jika data tidak stasioner maka perlu dilakukan differencing

Augmented Dickey-Fuller Statistic: -5.624256
p-value: 0.000001
critical values at different levels:
1%: -3.546
5%: -2.912
10%: -2.594
```

Gambar 4.4 Uji ADF

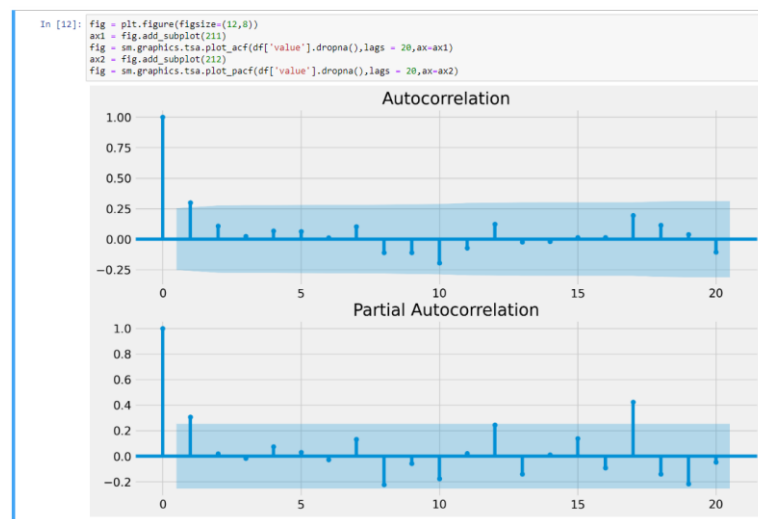


Gambar 4.5 Dekomposisi Data

Hasil dari analisis dataset menunjukkan bahwa dataset memiliki model data stasioner dengan nilai p-value pada gambar 4.4 sebesar 0.000001 dan tipe data musiman karena terdapat pola pengulangan periodik pada gambar 4.5, sehingga tahap selanjutnya adalah melakukan perhitungan untuk mendapatkan nilai AR dan MA tanpa menghitung *differencing* (i) dan menambahkan nilai parameter (M) saat membuat model ARIMA.

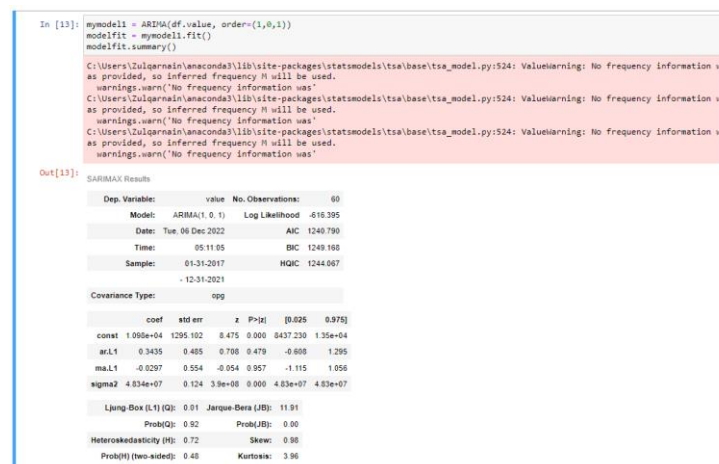
4.4 Pembuatan Model ARIMA

Nilai AR didapatkan dari plot PACF dan nilai MA didapatkan dari plot ACF, lag pertama yang keluar dari area biru grafik akan dijadikan nilai AR dan MA. Model ARIMA terbaik yang dipilih adalah model dengan nilai coef AR dan atau MA lebih besar dari 0, nilai $p > z$ AR dan atau MA mendekati 0 dan memiliki selisih yang relatif kecil, serta nilai AIC terkecil.



Gambar 4.6 Plot ACF dan PACF

Dari plot ACF dan PACF pada gambar 4.6 didapatkan nilai ARIMA (p,d,q) adalah (1,0,1). Kemudian model tersebut diproses dengan perhitungan ARIMA untuk melihat nilai-nilai parameter yang telah ditetapkan.



Gambar 4.7 Model ARIMA (1,0,1)

```
In [14]: mymodel2 = ARIMA(df.value, order=(1,0,0))
modelfit = mymodel2.Fit()
modelfit.summary()

C:\Users\zulqarnain\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information w
as provided, so inferred frequency H will be used.
warnings.warn("No frequency information was")
C:\Users\zulqarnain\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information w
as provided, so inferred frequency H will be used.
warnings.warn("No frequency information was")
C:\Users\zulqarnain\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information w
as provided, so inferred frequency H will be used.
warnings.warn("No frequency information was")

Out[14]: SARIMAX Results
-----
Dep. Variable:          value      No. Observations:      60
Model:                ARIMA(1, 0, 0)      Log Likelihood: -616.416
Date:                Tue, 06 Dec 2022      AIC: 1238.832
Time:                05:11:05             BIC: 1245.115
Sample:              01-31-2017           HQIC: 1241.290
                   - 12-31-2021

Covariance Type:      opg

coef      std err      z      P>|z|      [0.025      0.975]
-----
const    1.080e+04    1254.425    8.749    0.000    8516.955    1.34e+04
ar.L1    0.3171    0.129    2.451    0.014    0.064    0.571
sigma2    4.719e+07    0.123    3.85e+08    0.000    4.72e+07    4.72e+07

Ljung-Box (L1) (Q):    8.92      Jarque-Bera (JB):    11.97
Prob(Q):              0.90      Prob(JB):          0.00
Heteroskedasticity (H): 0.73      Skew:              0.98
Prob(H) (two-sided):  0.45      Kurtosis:          3.98
```

Gambar 4.8 Model ARIMA (1,0,0)

Pada gambar 4.7 nilai coef MA dibawah 0, nilai $p > z$ MA memiliki selisih yang besar terhadap nilai $p > z$ AR, sehingga diujikan model lain dengan menghapus nilai MA yaitu model (1,0,0). Dari uji model kedua pada gambar 4.8 didapatkan nilai coef, $p > z$, dan AIC yang lebih rendah, sehingga tahap selanjutnya adalah penambahan parameter musiman (M) pada model.

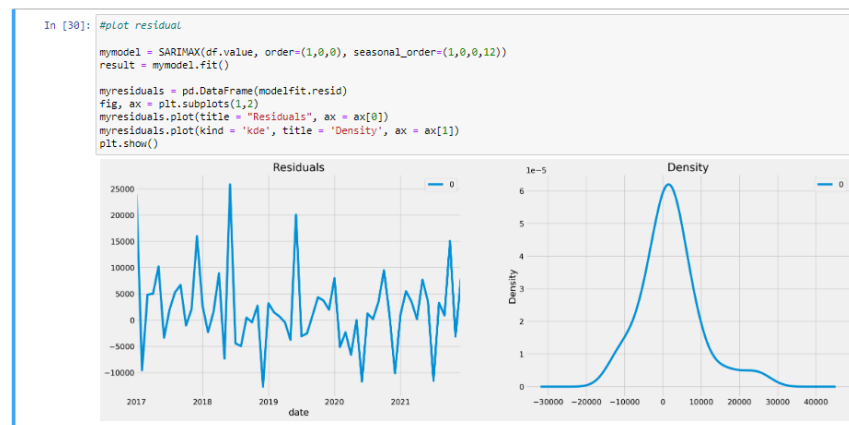


Gambar 4.9 Plot ACF dan PACF musiman

Nilai musiman diambil dari lag pada grafik yang memiliki pola *dying down* yaitu plot lag 12 pada plot ACF seperti pada gambar 4.9 dan karena plot PACF memiliki pola *cut-off* maka model ARIMA musiman memiliki tipe AR murni. Sehingga model ARIMA musiman yang terbentuk adalah (1,0,0)(1,0,0,12).

4.5 Pengujian Model ARIMA

Model ARIMA musiman yang dipilih selanjutnya diproses untuk plot residual dan density. Model ARIMA dikatakan baik ketika tidak ada pola atau penumpukan data pada plot residual, sedangkan plot density berfungsi untuk menunjukkan persebaran data, dan model ARIMA dikatakan baik ketika grafik berbentuk lonceng di sekitar nilai 0 pada sumbu X seperti pada gambar 4.10 berikut.



Gambar 4.10 Plot Residual dan Plot Density

Tahap selanjutnya adalah memprediksi nilai data testing, dataset dibagi menjadi 36 data training dan 24 data testing dengan grafik biru adalah data training, grafik merah adalah data testing, dan grafik kuning adalah grafik peramalan data testing seperti pada lampiran 5.

Berikutnya adalah perhitungan nilai error dengan menggunakan metode *Mean Percentage Absolute Error* (MAPE)

```
In [35]: test_act = test
test_fc = prediction

mape = np.mean(np.abs(test_fc - test_act)/np.abs(test_act))
mae = np.mean(np.abs(test_fc - test_act))
mpe = np.mean(test_fc - test_act)
rmse = np.mean((test_fc - test_act)**2)**.5

import pprint
pprint.pprint({'mape':mape,'mae':mae, 'mpe':mpe,
              'rmse':rmse})

{'mae': value 6319.463125
 'mpe': value 4.245946
 'rmse': value -3696.222977
 'mape': value 7661.823456
 'dtype': Float64}
```

Gambar 4.11 Uji Nilai Error

Dari gambar 4.11 didapatkan nilai error sebesar 4.245945% atau memiliki akurasi sebesar 94.7%.

4.6 Pembagian Dataset

Setelah dataset dibagi menjadi data training dan data testing seperti pada lampiran 6, selanjutnya data training dan data testing di normalisasi seperti gambar berikut

```
In [15]: nndatatrain = np.array(trainnn2).reshape(-1,1)
nndatatest = np.array(testnn2).reshape(-1,1)
scalernn = MinMaxScaler(feature_range=(0,1))
realdatann = scalernn.fit_transform(realdata)
scaled_nntrain = scalernn.fit_transform(nndatatrain)
scaled_nntest = scalernn.fit_transform(nndatatest)

seq_sizenn = 6
```

Gambar 4.12 Code Program untuk Menormalisasi Dataset

Data training dan data testing di proses pada tahap pembuatan dimensi, dimensi yang di buat pada lampiran 7 adalah variable yang digunakan dalam model, selain itu pembuatan dimensi juga digunakan sebagai pengujian model FFNN yang dibuat.

4.7 Pembuatan dan Pengujian Model FFNN

Tahap ini adalah tahap inialisasi jumlah layer dan jumlah node input, hidden dan output yang akan digunakan. Lampiran 8 adalah code program pembuatan model FFNN. Setelah pembuatan model, selanjutnya adalah pengujian model pada dimensi, model dengan nilai *val_loss* terkecil dipilih menjadi model FFNN terbaik.

```
In [59]: result = modelnn.fit(nnttrain_x, nnttrain_y, validation_data=(nnttest_x,nnttest_y), verbose=2, epochs=10)

Epoch 1/10
1/1 - 2s - loss: 0.1842 - val_loss: 0.2359 - 2s/epoch - 2s/step
Epoch 2/10
1/1 - 0s - loss: 0.1711 - val_loss: 0.2220 - 29ms/epoch - 29ms/step
Epoch 3/10
1/1 - 0s - loss: 0.1592 - val_loss: 0.2091 - 29ms/epoch - 29ms/step
Epoch 4/10
1/1 - 0s - loss: 0.1482 - val_loss: 0.1964 - 30ms/epoch - 30ms/step
Epoch 5/10
1/1 - 0s - loss: 0.1374 - val_loss: 0.1835 - 27ms/epoch - 27ms/step
Epoch 6/10
1/1 - 0s - loss: 0.1267 - val_loss: 0.1704 - 29ms/epoch - 29ms/step
Epoch 7/10
1/1 - 0s - loss: 0.1160 - val_loss: 0.1572 - 27ms/epoch - 27ms/step
Epoch 8/10
1/1 - 0s - loss: 0.1056 - val_loss: 0.1441 - 29ms/epoch - 29ms/step
Epoch 9/10
1/1 - 0s - loss: 0.0955 - val_loss: 0.1313 - 29ms/epoch - 29ms/step
Epoch 10/10
1/1 - 0s - loss: 0.0860 - val_loss: 0.1191 - 29ms/epoch - 29ms/step
```

Gambar 4.13 Pengujian Model pada Dimensi

Pada tahap ini dibentuk 3 model yang pertama model 1 layer input LSTM sebanyak 72 node, 1 hidden layer LSTM sebanyak 64 node, 1 hidden layer Dense sebanyak 18 node, dan 1 output layer sebanyak 1 node, yang kedua model 1 layer input LSTM sebanyak 80 node, 1 hidden layer LSTM sebanyak 50 node, 1 hidden

layer Dense sebanyak 20 node, dan 1 output layer sebanyak 1 node, dan yang ketiga model 1 layer input LSTM sebanyak 40 node, 1 hidden layer LSTM sebanyak 32 node, 1 hidden layer Dense sebanyak 18 node, dan 1 output layer sebanyak 1 node.

Dari hasil pengujian model pada dimensi, model 1 layer input LSTM sebanyak 72 node, 1 hidden layer LSTM sebanyak 64 node, 1 hidden layer Dense sebanyak 18 node, dan 1 output layer sebanyak 1 node memiliki nilai *val_loss* terkecil seperti pada gambar 4.13.

Tahap selanjutnya adalah tahap meramalkan data training seperti pada gambar 4.14 setelah mendapat nilai peramalan data testing data dilakukan proses inverse normalisasi untuk mengembalikan nilai asli data dengan cara pada gambar 4.15

```
In [70]: trainPredict = model.predict(x_train_)
testPredict = model.predict(x_test_)

2/2 [=====] - 0s 3ms/step
1/1 [=====] - 0s 15ms/step
```

Gambar 4.14 Code Program Peramalan FFNN

```
In [71]: trainscaled = scaler.inverse_transform(trainPredict)
trainY_inverse = scaler.inverse_transform([y_train])
testscaled = scaler.inverse_transform(testPredict)
testY_inverse = scaler.inverse_transform([y_test])

print(trainscaled.shape, trainY_inverse.shape, testscaled.shape, testY_inverse.shape)

(47, 1) (1, 47) (5, 1) (1, 5)
```

Gambar 4.15 Code Program Inverse Normalisasi

```
In [62]: trainScorenn = math.sqrt(mean_squared_error(trainY_inversenn[0], trainscalednn[0]))
print('Train Score: %.2f RMSE' % (trainScorenn))

testScorenn = math.sqrt(mean_squared_error(testY_inversenn[0], testscalednn[0]))
print('Test Score: %.2f RMSE' % (testScorenn))

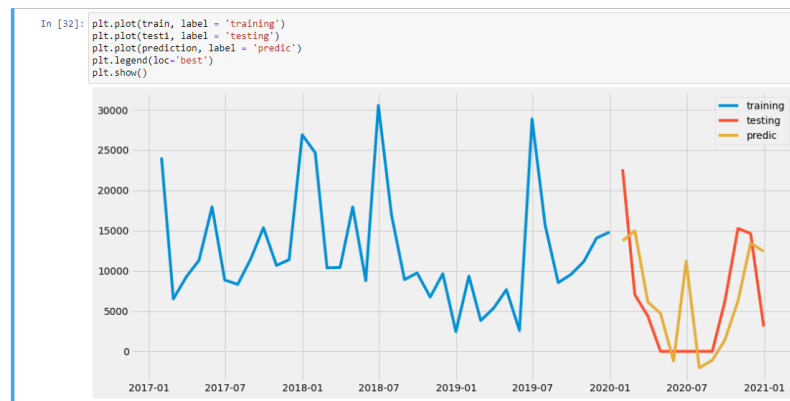
Train Score: 12295.65 RMSE
Test Score: 17274.88 RMSE
```

Gambar 4.16 Menghitung Nilai Error menggunakan RMSE

Setelah data dilakukan inverse normalisasi, dilakukan penghitungan nilai error menggunakan metode RMSE pada gambar 4.16, didapatkan nilai error untuk data training sebesar 12295.65 dan nilai error untuk data testing sebesar 17274.88. selanjutnya data divisualisasikan untuk melihat perbandingan grafiknya seperti pada lampiran 9.

4.8 Pengujian *Hybrid*

Tahap ini menggabungkan model ARIMA musiman dengan FFNN untuk meramalkan data testing. Tahap pertama adalah meramalkan nilai data testing untuk bulan Januari 2020 – bulan Desember 2020 menggunakan model ARIMA musiman.



Gambar 4.17 Peramalan ARIMA musiman hybrid

Nilai peramalan yang didapatkan pada gambar 4.17 dibulatkan dan di tulis ulang pada file excel lain seperti pada tabel 1 untuk di jadikan sebagai dataset pada gambar 4.18.

Tabel 1 Nilai Peramalan ARIMA musiman tahun 2020

31/01/2020	13726
29/02/2020	14960
31/03/2020	6122
30/04/2020	4705
31/05/2020	-1195
30/06/2020	11201
31/07/2020	-2042
31/08/2020	-1100
30/09/2020	1428
31/10/2020	6267

30/11/2020	13422
31/12/2020	12420

```
In [33]: rdarima = pd.read_excel('datasetbaru1.xlsx')

In [34]: rdarima['date'] = pd.to_datetime(rdarima['date'], infer_datetime_format=True)
rdarima = rdarima.set_index(['date'])
rdarima.tail()

Out[34]:
```

	value
date	
2020-08-31	-1100
2020-09-30	1428
2020-10-31	6267
2020-11-30	13422
2020-12-31	12420

Gambar 4.18 Pembaruan Dataset

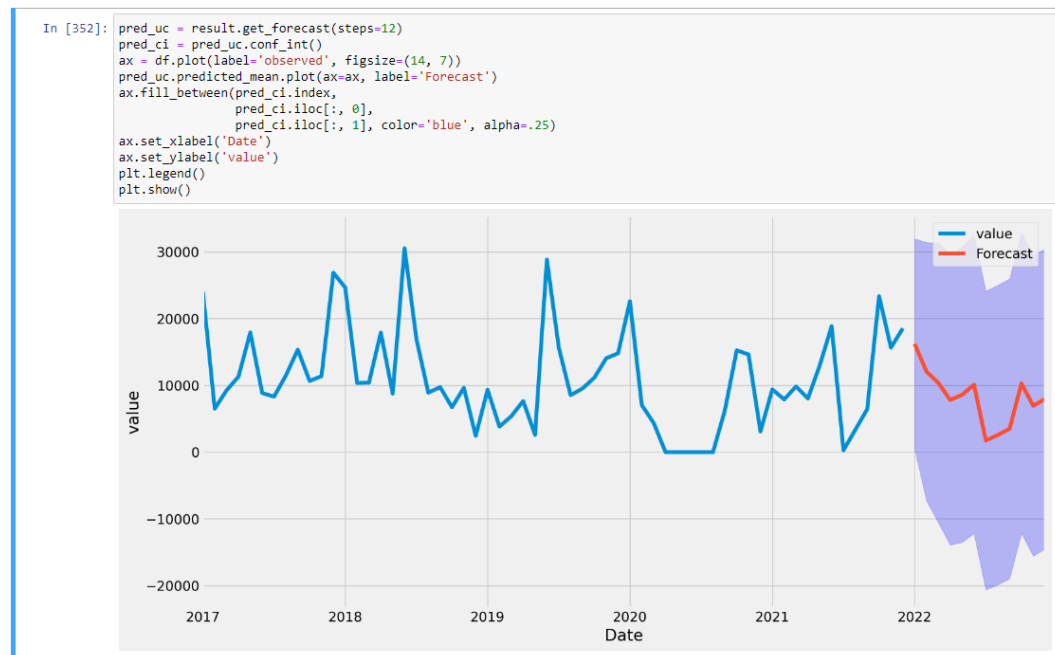
Sehingga dataset yang akan digunakan pada pengujian hybrid untuk model FFNN akan terlihat seperti gambar pada lampiran 10. Selanjutnya adalah tahap peramalan hybrid untuk metode FFNN yang terdiri dari tahap normalisasi data, pembuatan dimensi, pembuatan model, peramalan, dan inverse normalisasi seperti pada lampiran 11 hingga lampiran 17.

Setelah mendapatkan nilai peramalan FFNN dilakukan pemberian index tanggal mengikuti index dari dataset yang digunakan dengan code program seperti pada lampiran 18 hingga lampiran 23. Pada lampiran 17 terdapat 24 data nilai yang telah dilakukan inverse normalisasi, dari 24 data tersebut 12 data pertama adalah nilai peramalan 12 data terakhir dari data asli, dan 12 data berikutnya adalah nilai peramalan 12 bulan yang didapatkan dengan metode FFNN dan lampiran 18, 19 adalah code program untuk memberikan index pada nilai peramalan.

Setelah proses pemberian index tanggal, tahap selanjutnya adalah mengambil 12 nilai peramalan FFNN kemudian dibulatkan seperti pada lampiran 22 dan menggabungkan nilai tersebut pada dataset yang kemudian divisualisasikan seperti pada lampiran 23.

4.9 Peramalan ARIMA

Tahap ini adalah tahap peramalan ARIMA musiman untuk periode 12 bulan, yaitu bulan Januari tahun 2022 hingga bulan Desember tahun 2022.

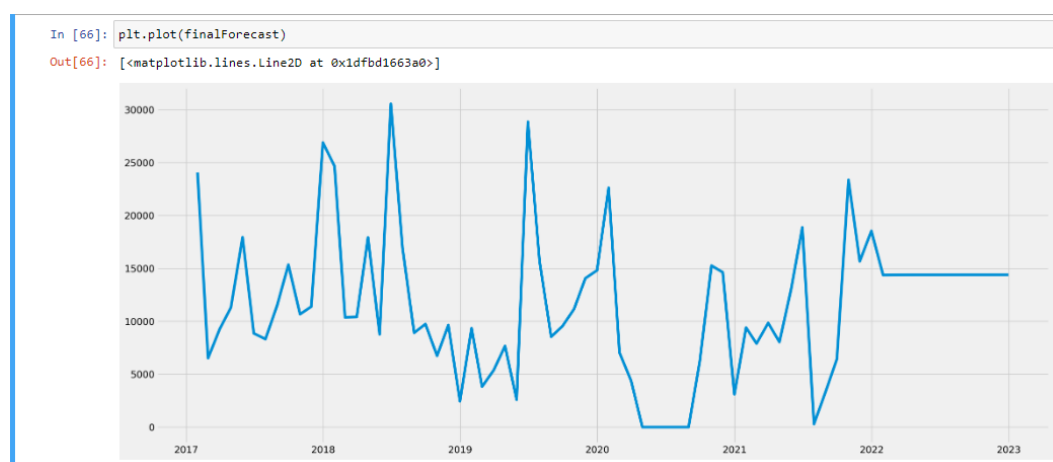


Gambar 4.19 Peramalan ARIMA tahun 2022

Gambar 4.19 adalah code program untuk meramalkan jumlah pengunjung pada tahun 2022, kode *steps* adalah parameter untuk menentukan berapa banyak nilai peramalan yang ingin dilakukan pada model ARIMA. Nilai peramalan ARIMA musiman jumlah pengunjung untuk tahun 2022 yang di dapatkan pada gambar 4.21 adalah sebanyak 16216 pengunjung pada bulan Januari, 12066 pengunjung pada bulan Februari, 10357 pengunjung pada bulan Maret, 7814 pengunjung pada bulan April, 8592 pengunjung pada bulan Mei, 10110 pengunjung pada bulan Juni, 1749 pengunjung pada bulan Juli, 2556 pengunjung pada bulan Agustus, 3499 pengunjung pada bulan September, 10281 pengunjung pada bulan Oktober, 6922 pengunjung pada bulan November, dan 7990 pengunjung pada bulan Desember.

4.10 Peramalan FFNN

Dengan mengubah nilai parameter *future* pada lampiran 16 dan parameter *month* pada lampiran 19 menjadi 24, serta mengubah dimensi *rescaled_prediction_new* pada lampiran 17 menjadi 24 dan mengubah nilai *slicing* kode *time_series_array* pada lampiran 19 menjadi 0 dengan tidak diberi nilai sebagai parameter index tanggal untuk menyesuaikan jumlah dari *rescaled_prediction_new*, maka grafik peramalan 12 bulan yaitu bulan Januari tahun 2022 hingga bulan Desember tahun 2022 akan membentuk grafik sebagai berikut.



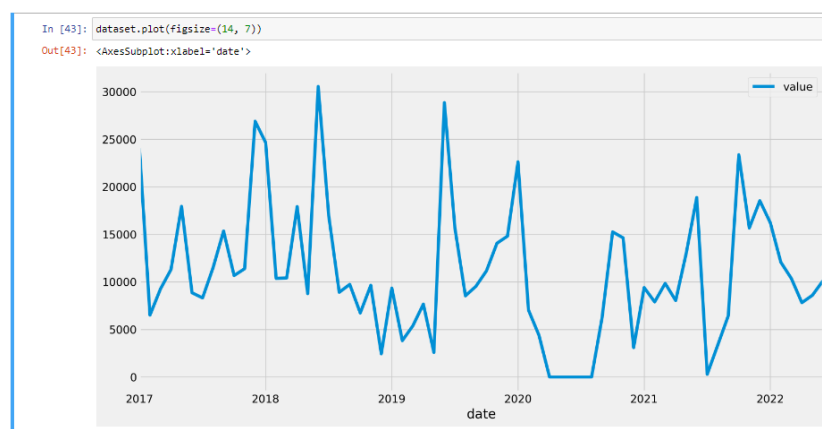
Gambar 4.20 Peramalan FFNN tahun 2022

4.11 Peramalan Hybrid

Tahap pertama adalah melakukan peramalan 6 bulan menggunakan model ARIMA musiman seperti pada lampiran 24, nilai peramalan ARIMA musiman yang didapatkan dapat dilihat pada lampiran 25. Nilai tersebut ditambahkan pada file excel dataset seperti pada tabel 2 untuk dijadikan dataset pada perhitungan peramalan menggunakan metode FFNN seperti pada gambar 4.21.

Tabel 2 Nilai Peramalan ARIMA musiman 6 bulan

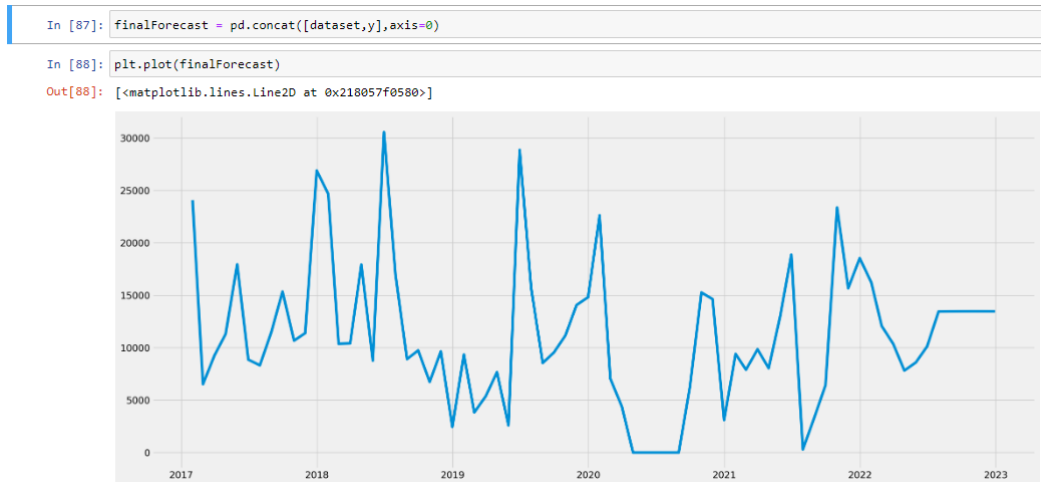
31/01/2022	16216
28/02/2022	12066
31/03/2022	10357
30/04/2022	7814
31/05/2022	8592
30/06/2022	10110



Gambar 4.21 Dataset Peramalan Hybrid model FFNN

Tahap kedua yaitu meramalkan jumlah pengunjung wisata untuk bulan Juli 2022 hingga Desember 2022 menggunakan metode FFNN, dengan proses pengerjaan yang sama seperti tahap pengujian hybrid yaitu normalisasi data, peramalan, inverse normalisasi, dan pemberian index tanggal dapat dilihat pada lampiran 26 hingga lampiran 31.

Pada tahap ini mengubah nilai parameter *future* menjadi 6 pada lampiran 26, mengubah nilai parameter *rescaled_prediction_new* menjadi 18 pada lampiran 27, mengubah nilai parameter *months* dan nilai parameter *time_series_array* menjadi 6 pada lampiran 28.



Gambar 4.22 Grafik Peramalan Hybrid

Sehingga grafik peramalan *hybrid* yang terbentuk akan seperti pada gambar 4.22 dengan nilai peramalan jumlah pengunjung dari gabungan metode ARIMA musiman dengan FFNN pada gambar 4.68 adalah sebanyak 16216 pengunjung bulan Januari, 12066 pengunjung bulan Februari, 10357 pengunjung bulan Maret, 7814 pengunjung bulan April, 8592 pengunjung bulan Mei, 10110 pengunjung bulan Juni, 13451 pengunjung bulan Juli, 13460 pengunjung bulan Agustus, 13466 pengunjung bulan September, 13468 pengunjung bulan Oktober, 13467 pengunjung bulan November, 13465 pengunjung bulan Desember.

Pengujian kedua ARIMA dilakukan dengan menggunakan nilai rata-rata untuk mengisi nilai kosong dan dengan langkah pengerjaan yang sama. Lampiran 32 adalah code program untuk mengisi nilai rata-rata pada nilai kosong, sehingga grafik jumlah pengunjung wisata yang terbentuk akan terlihat seperti pada lampiran 33.

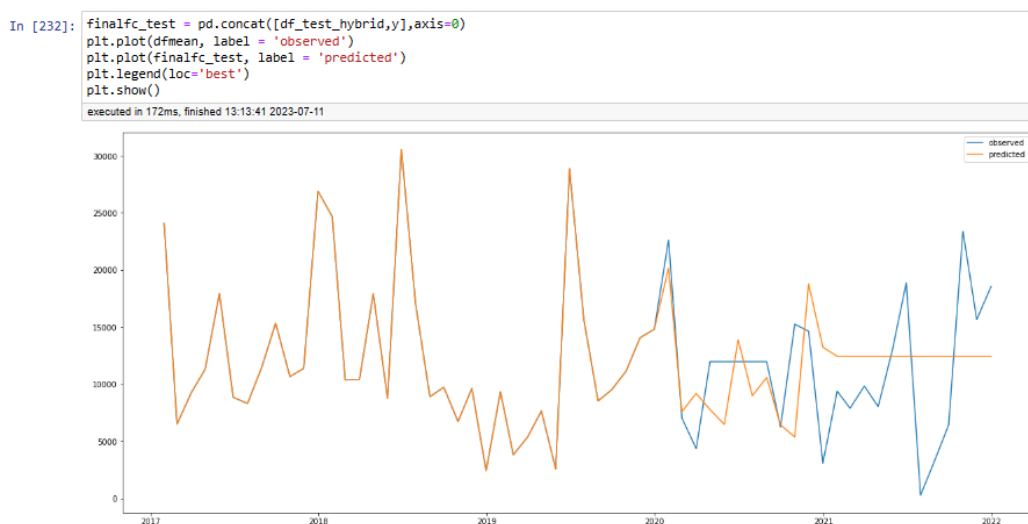
Tahap selanjutnya menguji dataset menggunakan test ADF seperti pada lampiran 34, hasil test ADF menunjukkan bahwa nilai p-value bernilai 0.000000 dengan nilai statistic sebesar -6.797056 dimana nilai tersebut lebih kecil dari dataset dengan menggunakan nilai 0, hasil test ADF ini menunjukkan bahwa dataset memiliki model data stasioner, dan pada lampiran 34 terdapat pola perulangan pada dataset, sehingga perlu dilakukan perhitungan musiman(M) dengan plot ACF dan plot PACF musiman.

Dari lampiran 35 terdapat 2 lag PACF yang keluar dari batas ketetapan, yaitu lag ke 8 dan lag 17, sehingga kemungkinan model yang terbentuk adalah ARIMA (8,0,0) dan ARIMA (17,0,0). Pada lampiran 36 dan lampiran 37 model (17,0,0) memiliki nilai coef AR, nilai $p > z$ dan AIC lebih rendah sehingga model (17,0,0) lebih baik dari pada model (8,0,0).

Pada lampiran 38 yaitu plot ACF dan plot PACF musiman, plot ACF berpola *dying down* dan plot PACF berpola *cut-off* dan didapatkan nilai musiman bernilai 2, sehingga model ARIMA musiman yang terbentuk adalah (17,0,0)(0,0,2) dengan nilai parameter coef AR, $p > z$, dan nilai AIC pada lampiran 39. Kemudian model ARIMA musiman di uji pada plot residual dan plot *density*. Pada lampiran 40 tidak terdapat penumpukan data pada plot residual dan terbentuknya grafik lonceng pada plot *density*.

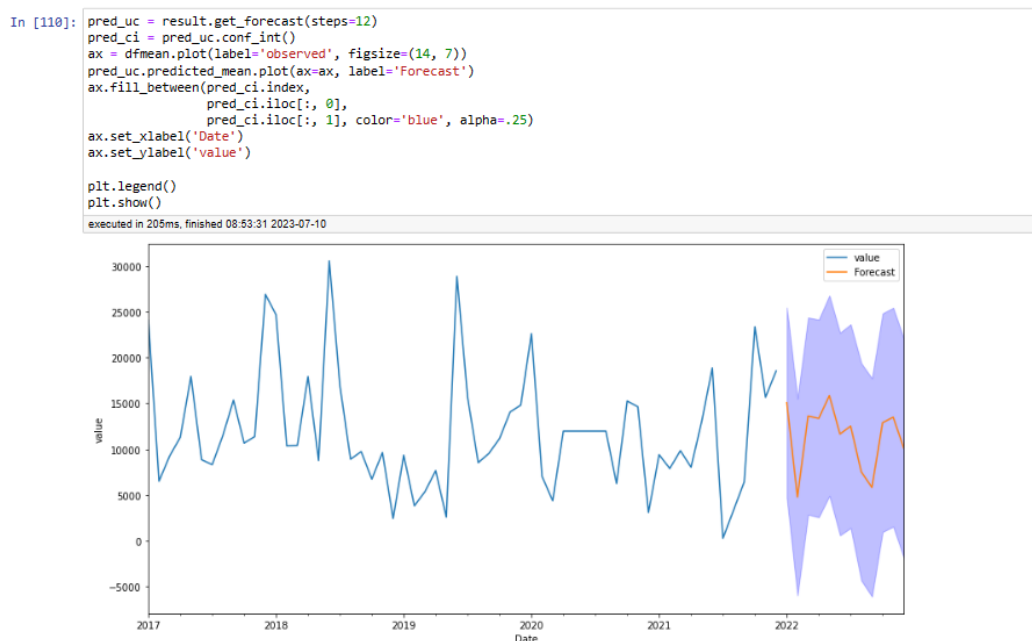
Tahap selanjutnya adalah peramalan nilai data testing dengan membagi dataset menjadi 36 data training dan 24 data testing, pada lampiran 41 grafik biru adalah grafik data training, grafik kuning adalah grafik data testing, dan grafik hijau adalah grafik prediksi data testing dengan nilai error sebesar 1.692819% menggunakan MAPE pada lampiran 42.

Selanjutnya adalah tahap peramalan hybrid pada data testing dengan pembagian dataset sebanyak 36 data training, 12 data testing ARIMA musiman, dan 12 data testing FFNN. Lampiran 43 dan lampiran 44 adalah grafik dan nilai untuk peramalan data testing menggunakan model ARIMA musiman. Selanjutnya nilai pada lampiran 44 ditulis pada file excel baru dan di import kembali untuk dijadikan peramalan hybrid data testing menggunakan metode FFNN seperti pada lampiran 45, maka dataset akan terlihat seperti lampiran 46. Kemudian dilakukan pembuatan model dan dimensi FFNN seperti pada lampiran 47 dan lampiran 48. Setelah dilakukan tahap-tahap peramalan model FFNN, nilai peramalan data testing yang didapatkan di gabungkan dengan dataset peramalan ARIMA musiman yang kemudian divisualisasikan seperti pada gambar



Gambar 4.23 Perbandingan Grafik Peramalan Hybrid Dengan Dataset Nilai Rata-Rata

Tahap selanjutnya adalah menghitung nilai peramalan jumlah pengunjung wisata pada tahun 2022 menggunakan metode ARIMA musiman dengan menggunakan dataset nilai rata-rata.

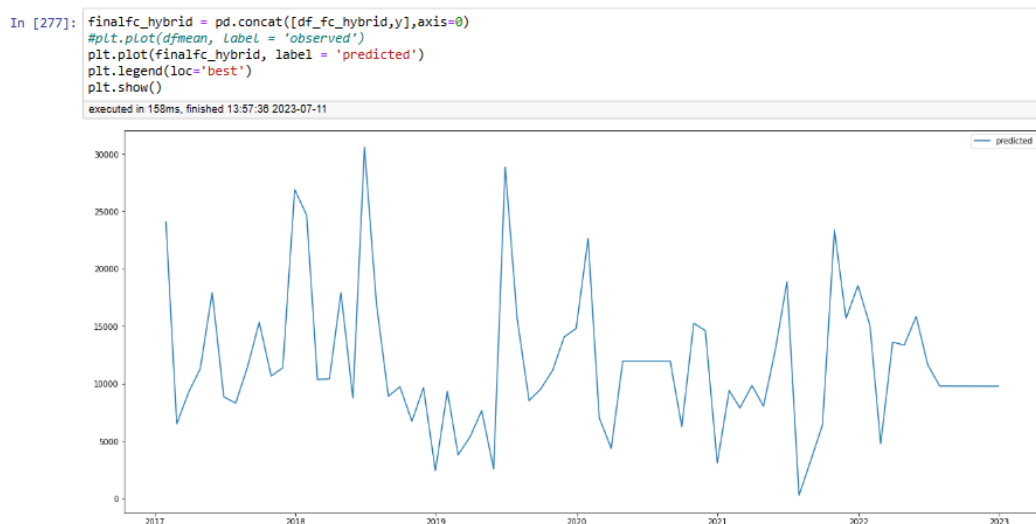


Gambar 4.24 Peramalan ARIMA Musiman Tahun 2022 Menggunakan Dataset Nilai Rata-Rata

Dari grafik peramalan ARIMA musiman pada gambar 4.24 dan lampiran 49 didapatkan nilai jumlah pengunjung wisata pada tahun 2022 sebanyak 15.082 pengunjung untuk bulan Januari, 4.784 pengunjung untuk bulan February, 13.617 pengunjung untuk bulan Maret, 13.356 pengunjung untuk bulan April, 15.845 pengunjung untuk bulan Mei, 11.657 pengunjung untuk bulan Juni, 12.519 pengunjung untuk bulan Juli, 7.534 pengunjung untuk bulan Agustus, 5.829 pengunjung untuk bulan September, 12.892 pengunjung untuk bulan Oktober, 13.494 pengunjung untuk bulan November, dan 10.106 pengunjung untuk bulan Desember.

Selanjutnya adalah menghitung nilai peramalan jumlah pengunjung wisata pada tahun 2022 menggunakan metode ARIMA musiman untuk bulan Januari 2022 – Juni 2022 dan metode FFNN untuk bulan Juli 2022 – Desember 2022. Setelah menghitung nilai peramalan 6 bulan menggunakan ARIMA musiman, selanjutnya nilai tersebut di simpan pada file excel baru untuk dijadikan dataset, sehingga grafik dataset baru akan terlihat seperti pada lampiran 50.

Dengan melakukan tahap-tahap peramalan FFNN, didapatkan grafik dan nilai peramalan FFNN untuk bulan Juli 2022 – Desember 2022 seperti pada lampiran 51 dan lampiran 52.



Gambar 4.25 Grafik Peramalan Hybrid Menggunakan Dataset Nilai Rata-Rata

Selanjutnya nilai peramalan FFNN tersebut digabungkan dengan dataset nilai peramalan ARIMA musiman sehingga grafik yang terbentuk akan terlihat pada gambar 4.25 dengan nilai jumlah peramalan hybrid pengunjung wisata pada tahun 2022 sebanyak 15.082 pengunjung untuk bulan Januari, 4.784 pengunjung untuk bulan February, 13.617 pengunjung untuk bulan Maret, 13.356 pengunjung untuk bulan April, 15.845 pengunjung untuk bulan Mei, 11.657 pengunjung untuk bulan Juni, 9.798 pengunjung untuk bulan Juli, 9.796 pengunjung untuk bulan Agustus, 9.793 pengunjung untuk bulan September, 9.790 pengunjung untuk bulan Oktober, 9.788 pengunjung untuk bulan November, dan 9.787 pengunjung untuk bulan Desember.

BAB 5. KESIMPULAN, KETERBATASAN, DAN SARAN

5.1 Kesimpulan

1. Berdasarkan penelitian yang menerapkan metode time series arima dan feed forward neural network, data pengunjung pariwisata adalah dataset dengan tipe data musiman dan dengan pola data stasioner, maka dari itu tidak perlu dilakukan perhitungan *differencing* (i), namun perlu menambahkan parameter (M) untuk pembuatan model ARIMA musiman.
2. Dataset yang digunakan memiliki kekosongan data selama 5 bulan, untuk mengatasi permasalahan tersebut dilakukan pengujian dengan mengisi nilai kosong dengan menghapus nilai kosong dan dengan mengisi nilai kosong dengan nilai rata-rata dataset, penghapusan nilai kosong kosong berpengaruh pada jumlah dataset yang mengakibatkan index tidak berurutan, sehingga data tidak dapat di proses, untuk mengatasinya data kosong diberi nilai 0.
3. Pengujian pertama dilakukan dengan menggunakan dataset dengan pemberian nilai 0 pada data kosong, hasil pengujian ARIMA musiman didapatkan model terbaik yaitu model (1,0,0)(1,0,0,12) dengan uji nilai error menggunakan metode MAPE sebesar 4.245956% , dan menggunakan model FFNN 1 layer input LSTM sebanyak 72 node, 1 hidden layer LSTM sebanyak 64 node, 1 hidden layer Dense sebanyak 18 node, dan 1 output layer sebanyak 1 node. Nilai peramalan hybrid yang didapatkan untuk tahun 2022 sebanyak 16.216 pengunjung bulan Januari, 12.066 pengunjung bulan Februari, 10.357 pengunjung bulan Maret, 7.814 pengunjung bulan April, 8.592 pengunjung bulan Mei, 10.110 pengunjung bulan Juni, 13.451 pengunjung bulan Juli, 13.460 pengunjung bulan Agustus, 13.466 pengunjung bulan September, 13.468 pengunjung bulan Oktober, 13.467 pengunjung bulan November, 13.465 pengunjung bulan Desember.
4. Pengujian kedua dilakukan dengan menggunakan dataset dengan pemberian nilai rata-rata dataset pada nilai kosong, hasil pengujian ARIMA musiman

didapatkan model terbaik yaitu model (17,0,0) (0,0,0,2) dengan uji nilai error menggunakan MAPE sebesar 1.692819%, dan menggunakan model FFNN 1 layer input LSTM sebanyak 72 node, 1 hidden layer LSTM sebanyak 64 node, 1 hidden layer Dense sebanyak 18 node, dan 1 output layer sebanyak 1 node. Nilai peramalan hybrid yang didapatkan untuk tahun 2022 sebanyak 15.082 pengunjung untuk bulan Januari, 4.784 pengunjung untuk bulan February, 13.617 pengunjung untuk bulan Maret, 13.356 pengunjung untuk bulan April, 15.845 pengunjung untuk bulan Mei, 11.657 pengunjung untuk bulan Juni, 9.798 pengunjung untuk bulan Juli, 9.796 pengunjung untuk bulan Agustus, 9.793 pengunjung untuk bulan September, 9.790 pengunjung untuk bulan Oktober, 9.788 pengunjung untuk bulan November, dan 9.787 pengunjung untuk bulan Desember.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan dan kesimpulan dari hasil penelitian ini, saran yang dapat diberikan untuk penelitian selanjutnya yaitu menggunakan dataset dengan data history lebih banyak dan membuat model FFNN yang lebih optimal, sehingga grafik dan nilai yang didapat dari proses peramalan lebih baik, dan juga dapat digunakan untuk peramalan jangka panjang (long-term prediction).

DAFTAR PUSTAKA

1. Isdarmanto. (2017). *Dasar-Dasar Kepariwisata dan Pengelolaan Destinasi Pariwisata*.
2. Susilo, A., Rumende, C. M., Pitoyo, C. W., Santoso, W. D., Yulianti, M., Herikurniawan, H., Sinto, R., Singh, G., Nainggolan, L., Nelwan, E. J., Chen, L. K., Widhani, A., Wijaya, E., Wicaksana, B., Maksum, M., Annisa, F., Jasirwan, C. O. M., & Yuniastuti, E. (2020). Coronavirus Disease 2019: Tinjauan Literatur Terkini. *Jurnal Penyakit Dalam Indonesia*, 7(1), 45. <https://doi.org/10.7454/jpdi.v7i1.415>
3. Syah, F., & Syafganti, I. (2021). *Kesiapan Pariwisata Depok di Era New Normal*. 4(1), 73–80. <http://ojs.stiami.ac.id>
4. Febrian, D., Idrus, S. I. Al, & Nainggolan, D. A. J. (2020). The Comparison of Double Moving Average and Double Exponential Smoothing Methods in Forecasting the Number of Foreign Tourists Coming to North Sumatera. *Journal of Physics: Conference Series*, 1462(1). <https://doi.org/10.1088/1742-k6596/1462/1/012046>
5. Ighravwe, D. E., & Anyaeche, C. O. (2019). A comparison of ARIMA and ANN techniques in predicting port productivity and berth effectiveness. *International Journal of Data and Network Science*, 3(1), 13–22. <https://doi.org/10.5267/j.ijdns.2018.11.003>
6. Safitri, B. A., Iriany, A., & Wardhani, N. W. S. (2021). Perbandingan Akurasi Peramalan Curah Hujan dengan menggunakan ARIMA, Hybrid ARIMA-NN, dan FFNN di Kabupaten Malang. *Seminar Nasional Official Statistics, 2021*(1), 245–253. <https://doi.org/10.34123/semnasoffstat.v2021i1.853>
7. Sari, V., & Maulidany, D. A. (2020). Prediksi Kecepatan Angin dalam Mendeteksi Gelombang Air Laut terhadap Skala Beaufort dengan Metode Hybrid Arima-Ann. *Jurnal Statistika*, 8(1), 8–17.
8. Qamara, L. N., Wahyuningsih, S., & Tisna, D. (2019). Peramalan Harga Minyak Mentah Menggunakan Model Autoregressive Integrated Moving Average Neural Network (ARIMA-NN). *Jurnal EKSPONENSIAL*, 10, 127–134.

9. Rianto, M., & Yunis, R. (2021). Analisis Runtun Waktu Untuk Memprediksi Jumlah Mahasiswa Baru Dengan Model Random Forest. *Paradigma - Jurnal Komputer Dan Informatika*, 23(1). <https://doi.org/10.31294/p.v23i1.9781>
10. Rola, J. B., Sta Romana, C. L. C., Feliscuzo, L. S., Lopez, I. F. M., & Rola, C. N. (2020). A comparative analysis of ARIMA and feed-forward neural network prognostic model for bull services. *Advances in Science, Technology and Engineering Systems*, 5(2), 411–418. <https://doi.org/10.25046/aj050253>
11. Baskoro, F., Alamsyah, F., & Suprianto, B. (2021). Peramalan Beban Listrik Harian Menggunakan Artificial Neural Network. *Jurnal Teknik Elektro*, 10(1), 203–209.
12. Rizk-Allah, R. M., & Hassanien, A. E. (2020). *COVID-19 forecasting based on an improved interior search algorithm and multi-layer feed forward neural network*. 1–24. https://doi.org/10.1007/978-3-030-91103-4_8
13. *Module II Perception networks – Learning rule – Training and testing algorithm , AdaptiveLinear Neuron , Back propagation Network – Architecture , Training algorithm* . (n.d).
14. Škrinjarić, T. (2019). Book review: Multivariate time series analysis and applications. In *Oeconomica Jadertina* (Vol. 9, Issue 2).
15. Li, L., Wang, Y., & Li, X. (2020). Tourists Forecast Based on the Baolan High-Speed Railway by the Arima Model. *Applied Mathematics and Nonlinear Sciences*, 5(1), 55–60. <https://doi.org/10.2478/amns.2020.1.00006>
16. Shabri, A., Samsudin, R., & Yusoff, Y. (2020). Combining Deep Neural Network and Fourier Series for Tourist Arrivals Forecasting. *IOP Conference Series: Materials Science and Engineering*, 864(1). <https://doi.org/10.1088/1757-899X/864/1/012094>
17. Herawati, S., & Latif, M. (2021). Forecasting tourist visits using data decomposition technique and learning optimization of artificial neural network. *Journal of Physics: Conference Series*. <https://doi.org/10.1088/1742-6596/1811/1/012092>

18. Ukhra, A. U. (2014). Pemodelan Dan Peramalan Data Deret Waktu Dengan Metode Seasonal Arima. *Jurnal Matematika UNAND*, 3(3), 59.
<https://doi.org/10.25077/jmu.3.3.59-67.2014>
19. Rizal, A. A., & Soraya, S. (2018). Multi Time Steps Prediction dengan Recurrent Neural Network Long Short Term Memory. *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, 18(1), 115–124.
<https://doi.org/10.30812/matrik.v18i1.344>
20. Rusdi. (2011). Uji Akar-Akar Unit dalam Model Runtun Waktu Autoregresif. *Statistika*, 11(2), 67–78.
21. Ritha, N., Bettiza, M., & Dufan, A. (2016). Prediksi Curah Hujan dengan Menggunakan Algoritma Levenberg-Marquardt dan Backpropagation. *Jurnal Sustainable*, 5(2), 11–16.
22. Rahmahwati, F. S., & Arifin, F. (2017). *DEVELOPING ARTIFICIAL NEURAL NETWORK*. 23(4), 402–411.
23. Sammen, S. S., Ghorbani, M. A., Malik, A., Tikhamarine, Y., AmirRahmani, M., Al-Ansari, N., & Chau, K. W. (2020). Enhanced artificial neural network with Harris hawks optimization for predicting scour depth downstream of ski-jump spillway. *Applied Sciences (Switzerland)*, 10(15).
<https://doi.org/10.3390/app10155160>
24. Prediction, S. W. S. (2019). *Optimization of Feedforward Neural Networks Using an Improved Flower Pollination Algorithm for Short-Term Wind Speed Prediction*. *Energies*, doi:10.3390/en12214126
25. Nanda, S. K., Tripathy, D. P., Nayak, S. K., & Mohapatra, S. (2013). Prediction of Rainfall in India using Artificial Neural Network (ANN) Models. *International Journal of Intelligent Systems and Applications*, 5(12), 1–22.
<https://doi.org/10.5815/ijisa.2013.12.01>
26. Toharudin, T., Caraka, R. E., Bachrudin, A., Bakar, S. A., & Ambarsari, D. (2019). Copula-based feedforward neural network genetic algorithm cargo forecasting. *International Journal of Advances in Soft Computing and Its Applications*, 11(1), 22–33.

27. Ayu, R., Gernowo, R., Fisika, D., Sains, F., Diponegoro, U., & E-, S. (2019). Metode Autoregressive Integrated Movingaverage (Arima) Dan Metode Adaptive Neuro Fuzzy Inference System (Anfis) Dalam Analisis Curah Hujan. *Berkala Fisika*, 22(1), 41–48.
28. Cheng, X., Feng, Z. K., & Niu, W. J. (2020). Forecasting Monthly Runoff Time Series by Single-Layer Feedforward Artificial Neural Network and Grey Wolf Optimizer. *IEEE Access*, 8, 157346–157355.
<https://doi.org/10.1109/ACCESS.2020.3019574>
29. Khan, S., Aslam, S., Mustafa, I., & Aslam, S. (2021). Short-Term Electricity Price Forecasting by Employing Ensemble Empirical Mode Decomposition and Extreme Learning Machine. *Forecasting*, 3(3), 460–477.
<https://doi.org/10.3390/forecast3030028>
30. Anggarini, D. T. (2021). Upaya Pemulihan Industri Pariwisata Dalam Situasi Pandemi Covid -19. *Jurnal Pariwisata*, 8(1), 22–31.
<https://doi.org/10.31294/par.v8i1.9809>
31. Budiyaniti, E. (2020). Dampak Virus Corona Terhadap Sektor Perdagangan Dan Pariwisata Indonesia. *Kajian Bidang Ekonomi Dan Kebijakan Publik*, XII(4), 19–24. http://berkas.dpr.go.id/puslit/files/info_singkat/Info_Singkat-XII-4-II-P3DI-Februari-2020-219.pdf
32. Rofifah, D. (2020). Pengantar Pariwisata. In *Paper Knowledge . Toward a Media History of Documents*.
33. Terence C. Mills. (2010). Applied time series analysis. In *Lecture Notes in Electrical Engineering: Vol. 59 LNEE*. https://doi.org/10.1007/978-90-481-8768-3_2
34. Saâdaoui, F., Saadaoui, H., & Rabbouch, H. (2020). Hybrid feedforward ANN with NLS-based regression curve fitting for US air traffic forecasting. *Neural Computing and Applications*, 32(14), 10073–10085.
<https://doi.org/10.1007/s00521-019-04539-5>

LAMPIRAN-LAMPIRAN

Lampiran 1 Tabel Jumlah Pengunjung Objek Wisata Kabupaten Situbondo

NO	OBYEK WISATA	ALAMAT	JUMLAH		JUMLAH		JUMLAH		JUMLAH		JUMLAH		JUMLAH		JUMLAH		JUMLAH		JUMLAH		JUMLAH		JUMLAH		Total (orang)			
			Januari		Pebruari		Maret		April		Mei		Juni		Juli		Agustus		Septemb er		Oktober		Nopemb er		Desembe r		Indonesia	Asing
			Indonesia	Asing	Indonesia	Asing	In do nes ia	A si ng	In do nes ia	A si ng	In do nes ia	A si ng	In do nes ia	A si ng	In do nes ia	A si ng	In do nes ia	A si ng	In do nes ia	A si ng	In do nes ia	A si ng	In do nes ia	A si ng	In do nes ia	A si ng		
1	TN Baluran	Banyuputih	12.549	34	4.683	56	4.820	69	5.631	74	6.102	13	874	56	16759	2199	5405	185	5520	74	4911	138	5464	39	12126	63	84.844	1.210
2	Pantai Pasir Putih	Bungatan	24.034	30	6.479	31	9.238	26	11.254	50	17.895	45	8.851	6	8.313	22	11.472	22	15.345	33	10.639	29	11.372	12	26289		161.181	256
3	Desa Kebangsaan Wonorejo	Wonorejo	100	15	120	8	252	4	456	33	650	22	789	5	524	33	456	11	210	11	215	11	193	22			3.965	45
4	Rafting Samir Situbondo Adventure	Asembagus	201	-	245	-	298	-	252	-	274	-	257	-	230	-	210	-	195	-	310	11	195	11			2.667	2

Lampiran 2 Tabel Jumlah Pengunjung Wisata Pantai Pasir Putih Situbondo tahun 2017-2021

Tanggal	Jumlah Pengunjung Wisata
31/01/2017	24064
28/02/2017	6510
31/03/2017	9264
30/04/2017	11304
31/05/2017	17940
30/06/2017	8857
31/07/2017	8315
31/08/2017	11494
30/09/2017	15348
31/10/2017	10668
30/11/2017	11384
31/12/2017	26892
31/01/2018	24658
28/02/2018	10368
31/03/2018	10417
30/04/2018	17918
31/05/2018	8767
30/06/2018	30549
31/07/2018	16968
31/08/2018	8908
30/09/2018	9738
31/10/2018	6735
30/11/2018	9638
31/12/2018	2440
31/01/2019	9348
28/02/2019	3822
31/03/2019	5382
30/04/2019	7659
31/05/2019	2586
30/06/2019	28854

Tanggal	Jumlah Pengunjung Wisata
31/07/2019	15673
31/08/2019	8535
30/09/2019	9536
31/10/2019	11157
30/11/2019	14065
31/12/2019	14810
31/01/2020	22614
29/02/2020	7035
31/03/2020	4373
30/04/2020	
31/05/2020	
30/06/2020	
31/07/2020	
31/08/2020	
30/09/2020	6252
31/10/2020	15259
30/11/2020	14629
31/12/2020	3092
31/01/2021	9401
28/02/2021	7894
31/03/2021	9841
30/04/2021	8039
31/05/2021	13007
30/06/2021	18875
31/07/2021	288
31/08/2021	3358
30/09/2021	6435
31/10/2021	23362
30/11/2021	15671
31/12/2021	18539

Lampiran 3 library dan import dataset

```
In [67]: import warnings
import itertools
import pandas as pd
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt
import math

from datetime import datetime
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from pmdarima.arima import auto_arima
from statsmodels.tsa.stattools import adfuller
from numpy import log
from keras.models import Sequential
from keras.layers import Dense
from tensorflow.keras.layers import LSTM, Dense, Dropout
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn import preprocessing

plt.style.use('fivethirtyeight')
```

```
In [2]: df = pd.read_excel('data.xlsx')
```

```
In [3]: df.info()
```

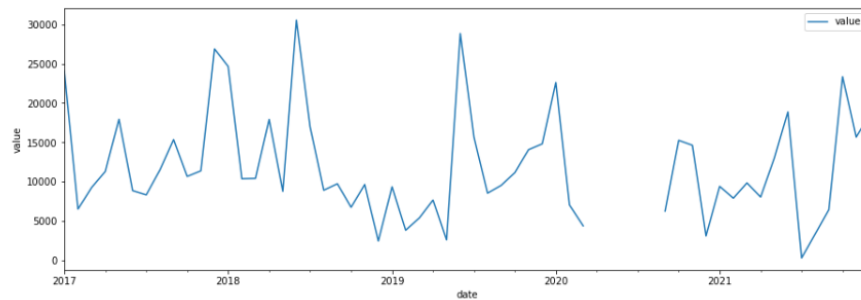
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 60 entries, 0 to 59
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0   date    60 non-null     datetime64[ns]
 1   value   60 non-null     int64
dtypes: datetime64[ns](1), int64(1)
memory usage: 1.1 KB
```

Lampiran 4 visualisasi dataset

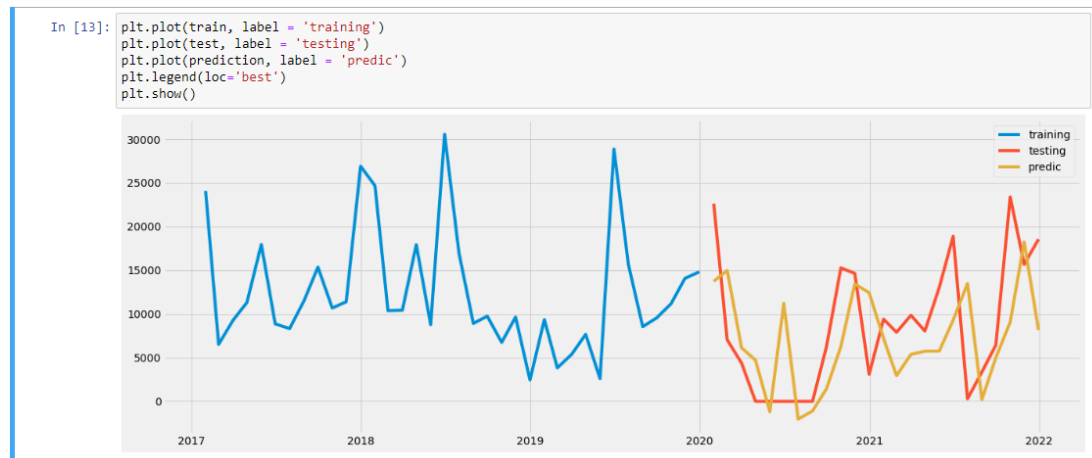
```
In [7]: dataset.plot(figsize=(15,5))
plt.ylabel('value')
```

```
executed in 700ms, finished 15:13:05 2023-07-09
```

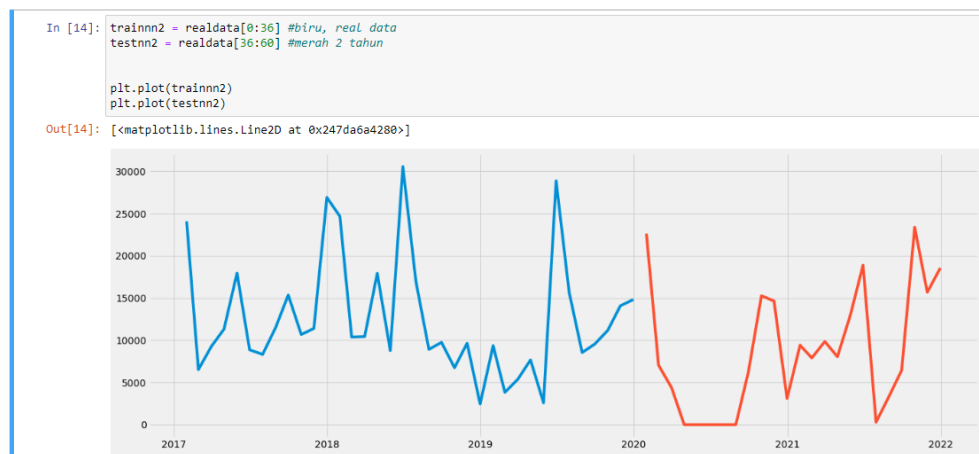
```
Out[7]: Text(0, 0.5, 'value')
```



Lampiran 5 Peramalan Data Testing



Lampiran 6 Pembagian Dataset untuk Pembuatan Model FFNN



Lampiran 7 Pembutan Dimensi FFNN

```
In [16]: nntrain_x = []
nntrain_y = []

for i in range(seq_sizenn, len(scaled_nnttrain)):
    nntrain_x.append(scaled_nnttrain[i-seq_sizenn:i])
    nntrain_y.append(scaled_nnttrain[i])
nntrain_x, nnttrain_y = np.array(nnttrain_x), np.array(nnttrain_y)
```

```
In [17]: nntest_x = []
nntest_y = []

for i in range(seq_sizenn, len(scaled_nnttest)):
    nntest_x.append(scaled_nnttest[i-seq_sizenn:i])
    nntest_y.append(scaled_nnttest[i])
nntest_x, nntest_y = np.array(nntest_x), np.array(nntest_y)
```

```
In [18]: print(nnttrain_x.shape, nnttrain_y.shape, nntest_x.shape, nntest_y.shape)

(30, 6, 1) (30, 1) (18, 6, 1) (18, 1)
```

Lampiran 8 Pembuatan Model FFNN

```
In [68]: model = Sequential()
model.add(LSTM(72, return_sequences = True, input_shape = (seq_size, 1), activation = 'relu'))
model.add(LSTM(64, activation = 'relu', return_sequences=False))
model.add(Dense(18))
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')
print(model.summary())
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 6, 72)	21312
lstm_1 (LSTM)	(None, 64)	35072
dense_2 (Dense)	(None, 18)	1170
dense_3 (Dense)	(None, 1)	19

=====
Total params: 57,573
Trainable params: 57,573
Non-trainable params: 0
=====
None

Lampiran 9 Visualisasi Peramalan Data Testing FFNN

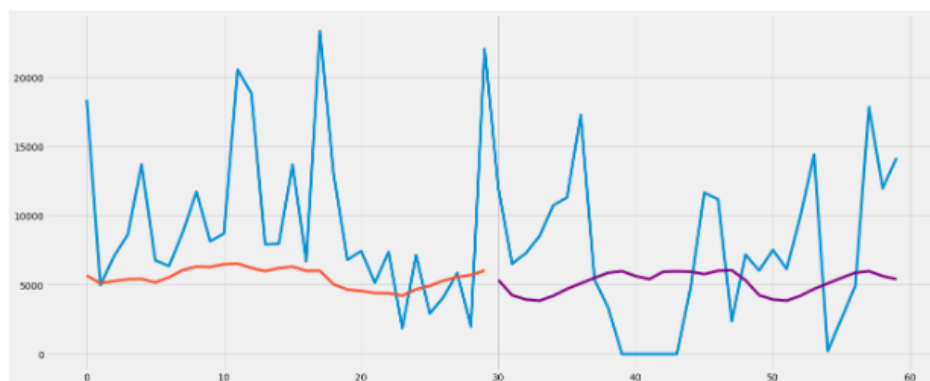
```
In [63]: ndf2n = np.resize(scaled_ntrain, (60,1))
trainPredictPlotnn = np.empty_like(ndf2n)
trainPredictPlotnn[:, :] = np.nan
print(trainPredictPlotnn.shape)
print(trainscalednn.shape, trainPredictnn.shape)
trainPredictPlotnn[:len(trainscalednn), :] = trainPredictnn

dfn_nn = np.resize(scaled_nntest, (60,1))
testPredictnn = np.resize(testPredictnn, (30,1))
#print(len(trainscaled))
testPredictPlotnn = np.empty_like(dfn_nn)
print(testPredictPlotnn.shape)
#print(len(testPredictPlotnn))
testPredictPlotnn[:, :] = np.nan
testPredictPlotnn[len(trainscalednn):len(dfn_nn), :] = testPredictnn

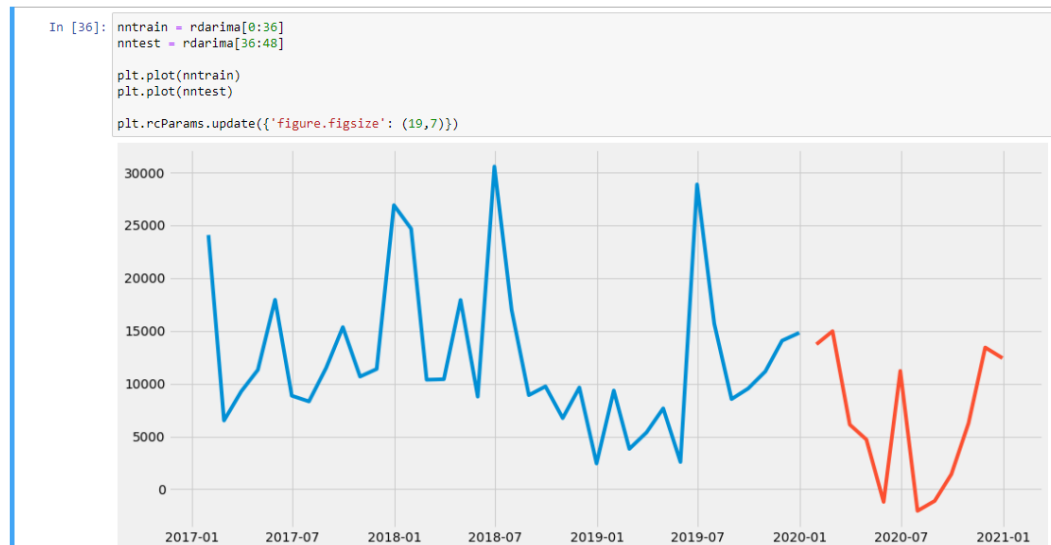
#testpredictplot dimensinya harus (x,1)

#plt.plot(trainnn2)
#plt.plot(testnn2)
plt.plot(scalernn.inverse_transform(realdatann))
#plt.plot(scaler.inverse_transform(scaled_nntest))
plt.plot(scalernn.inverse_transform(trainPredictPlotnn))
plt.plot(scalernn.inverse_transform(testPredictPlotnn), color='purple')
plt.rcParams.update({'figure.figsize': (21,9)})
plt.show()
```

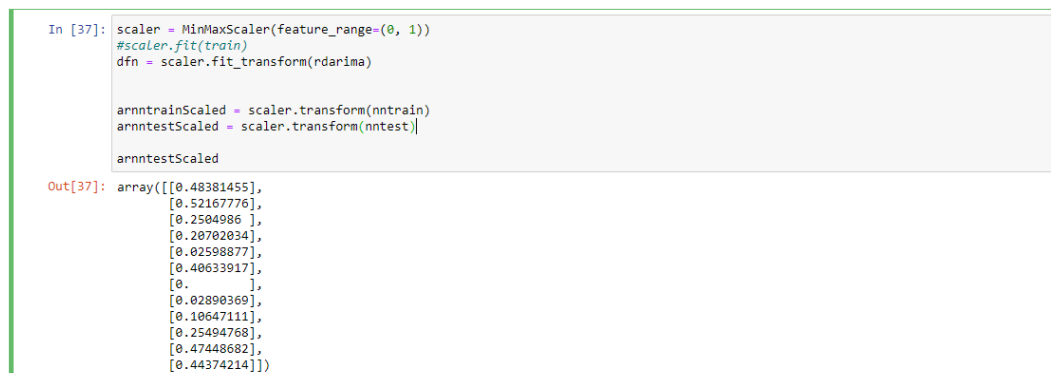
```
(60, 1)
(30, 1) (30, 1)
(60, 1)
```



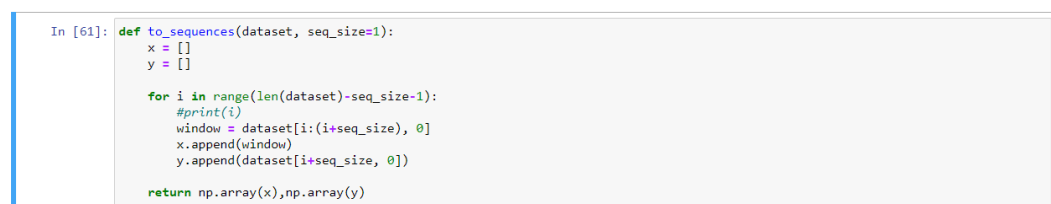
Lampiran 10 Dataset Pengujian Hybrid model FFNN



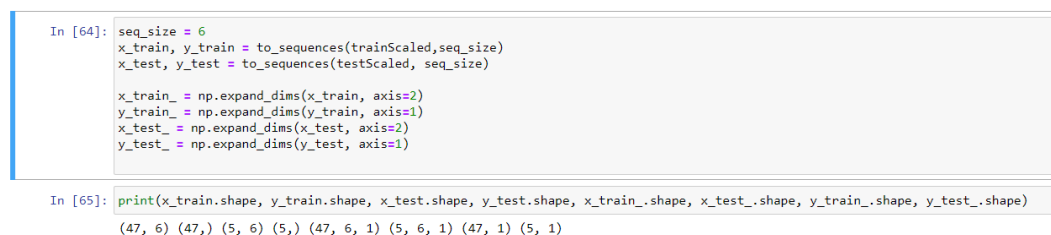
Lampiran 11 Normalisasi Data



Lampiran 12 Pembuatan Dimensi



Lampiran 13 Pembuatan Dimensi



Lampiran 14 Pembuatan Model FFNN

```
In [41]: model = Sequential()
model.add(LSTM(72, return_sequences = True, input_shape = (seq_size, 1), activation = 'relu'))
model.add(LSTM(64, activation = 'relu', return_sequences=False))
model.add(Dense(18))
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')
print(model.summary())

Model: "sequential_1"
-----
```

Layer (type)	Output Shape	Param #
lstm_2 (LSTM)	(None, 6, 72)	21312
lstm_3 (LSTM)	(None, 64)	35072
dense_2 (Dense)	(None, 18)	1170
dense_3 (Dense)	(None, 1)	19

```
-----
Total params: 57,573
Trainable params: 57,573
Non-trainable params: 0
-----
None
```

Lampiran 15 Pengujian Model pada Dimensi

```
In [42]: model.fit(x_train_, y_train_, validation_data=(x_test_,y_test_), verbose=2, epochs=10)

Epoch 1/10
1/1 - 2s - loss: 0.2395 - val_loss: 0.0542 - 2s/epoch - 2s/step
Epoch 2/10
1/1 - 0s - loss: 0.2178 - val_loss: 0.0492 - 33ms/epoch - 33ms/step
Epoch 3/10
1/1 - 0s - loss: 0.1975 - val_loss: 0.0449 - 28ms/epoch - 28ms/step
Epoch 4/10
1/1 - 0s - loss: 0.1782 - val_loss: 0.0414 - 26ms/epoch - 26ms/step
Epoch 5/10
1/1 - 0s - loss: 0.1599 - val_loss: 0.0386 - 25ms/epoch - 25ms/step
Epoch 6/10
1/1 - 0s - loss: 0.1422 - val_loss: 0.0365 - 27ms/epoch - 27ms/step
Epoch 7/10
1/1 - 0s - loss: 0.1253 - val_loss: 0.0355 - 27ms/epoch - 27ms/step
Epoch 8/10
1/1 - 0s - loss: 0.1091 - val_loss: 0.0357 - 25ms/epoch - 25ms/step
Epoch 9/10
1/1 - 0s - loss: 0.0940 - val_loss: 0.0374 - 30ms/epoch - 30ms/step
Epoch 10/10
1/1 - 0s - loss: 0.0803 - val_loss: 0.0409 - 27ms/epoch - 27ms/step

Out[42]: <keras.callbacks.History at 0x28735dce580>
```

Lampiran 16 Peramalan Tahun 2021

```
In [49]: future = 12 #months
print(prediction)
for i in range(len(nntest) + future):
    arnncurrent_pred = model.predict(current_batch_arnn)[0]
    print(arnncurrent_pred)
    prediction.append(arnncurrent_pred)
    current_batch_arnn = np.append(current_batch_arnn[:,1:,:], [[arnncurrent_pred]],axis=1)

print(current_batch_arnn.shape)
print(current_batch_arnn)

[]
1/1 [=====] - 0s 14ms/step
[0.29748937]
1/1 [=====] - 0s 15ms/step
[0.28951734]
1/1 [=====] - 0s 14ms/step
[0.2913922]
1/1 [=====] - 0s 15ms/step
[0.28918323]
1/1 [=====] - 0s 10ms/step
[0.2421036]
1/1 [=====] - 0s 15ms/step
[0.24205716]
(1, 6, 1)
[[[0.24248932]
[0.24236092]
[0.24225315]
[0.24216768]
[0.24210361]
[0.24205716]]]]
```

Lampiran 17 Inverse Normalisasi Nilai Peramalan 2021

```
In [66]: #arnnprediction = np.array(arnnprediction, dtype=int)
#arnnprediction = arnnprediction.reshape(1,-1)
#print(arnnprediction)
#print(prediction)
arnnrescaled_prediction = scaler.inverse_transform(prediction)
print(len(arnnrescaled_prediction))
print(arnnrescaled_prediction.shape)

rescaled_prediction_arnn = np.resize(arnnrescaled_prediction, (24,1))
print(rescaled_prediction_arnn)
print(rescaled_prediction_arnn.shape)

24
(24, 1)
[[7653.47621658]
 [7393.65972716]
 [7454.76342309]
 [7382.77062312]
 [7166.01130807]
 [6752.5993003 ]
 [6293.47073299]
 [6237.80818957]
 [6182.65168694]
 [6113.99914464]
 [6043.0024 ]
 [5980.32473269]
 [5938.75022806]
 [5920.92127551]
 [5903.97959349]
 [5888.622567 ]
 [5876.18765892]
 [5867.08038178]
 [5860.96952654]
 [5856.7847344 ]
 [5853.27255884]
 [5850.48690632]
 [5848.39863822]
 [5846.88488667]]
(24, 1)
```

Lampiran 18 Pengambilan Index dari Dataset

```
In [51]: time_series_array = nntest.index

print(len(time_series_array))
print(time_series_array.shape)
print(time_series_array)

12
(12,)
DatetimeIndex(['2020-01-31', '2020-02-29', '2020-03-31', '2020-04-30',
               '2020-05-31', '2020-06-30', '2020-07-31', '2020-08-31',
               '2020-09-30', '2020-10-31', '2020-11-30', '2020-12-31'],
              dtype='datetime64[ns]', name='date', freq=None)
```

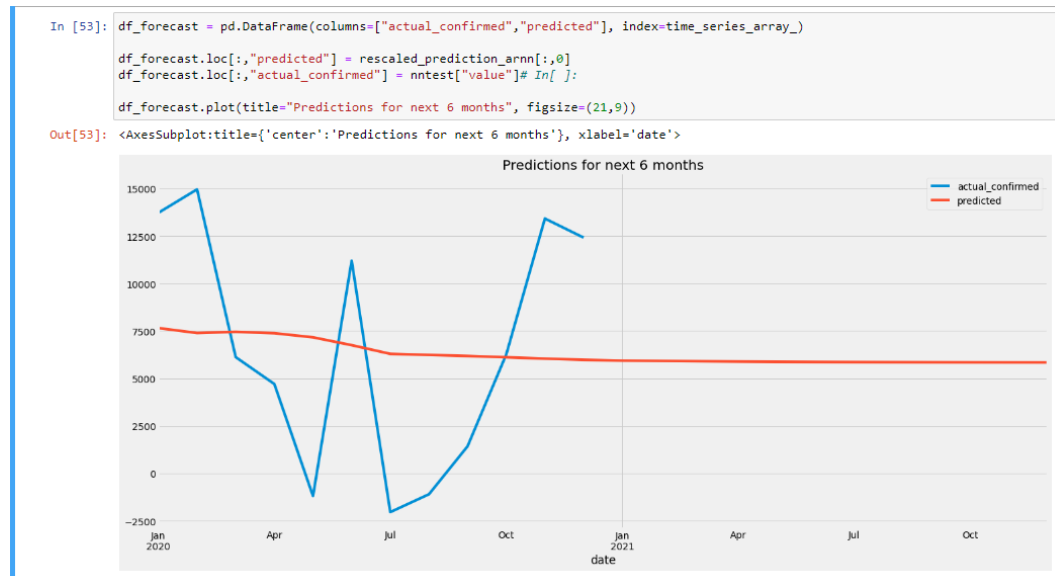
Lampiran 19 Pemberian Index pada Nilai Peramalan

```
In [52]: for k in range(0, future):
time_series_array_ = time_series_array.append(time_series_array[:] + pd.DateOffset(months=12))

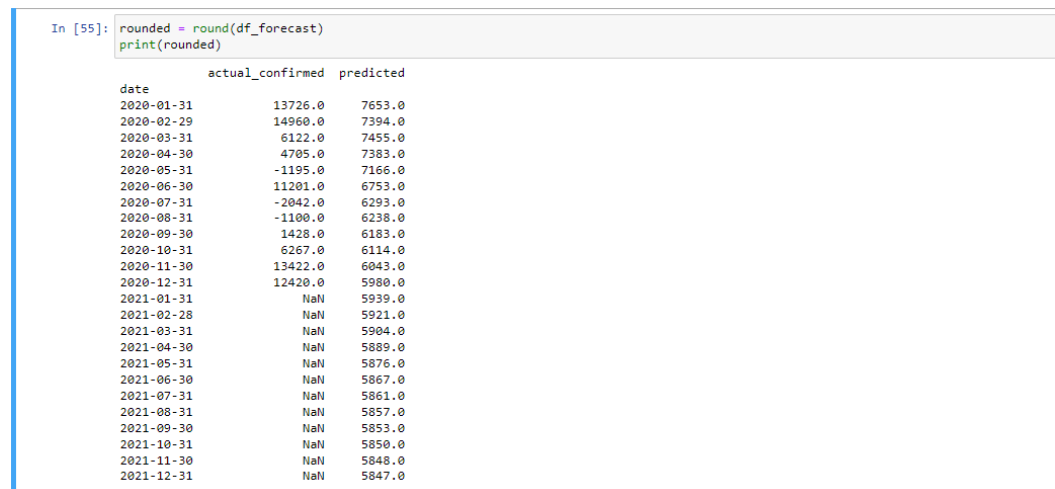
print(len(time_series_array_))
print(time_series_array_.shape)
print(time_series_array_)

24
(24,)
DatetimeIndex(['2020-01-31', '2020-02-29', '2020-03-31', '2020-04-30',
               '2020-05-31', '2020-06-30', '2020-07-31', '2020-08-31',
               '2020-09-30', '2020-10-31', '2020-11-30', '2020-12-31',
               '2021-01-31', '2021-02-28', '2021-03-31', '2021-04-30',
               '2021-05-31', '2021-06-30', '2021-07-31', '2021-08-31',
               '2021-09-30', '2021-10-31', '2021-11-30', '2021-12-31'],
              dtype='datetime64[ns]', name='date', freq=None)
```

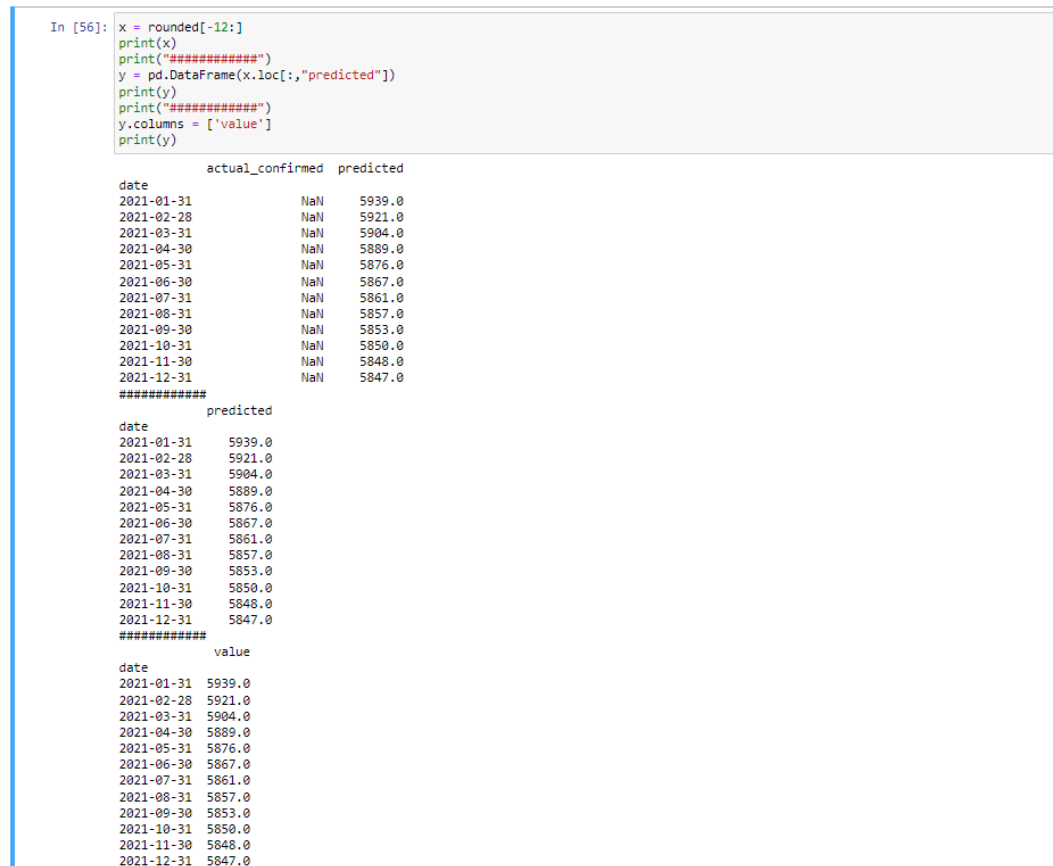

Lampiran 20 Visualisasi Nilai Peramalan



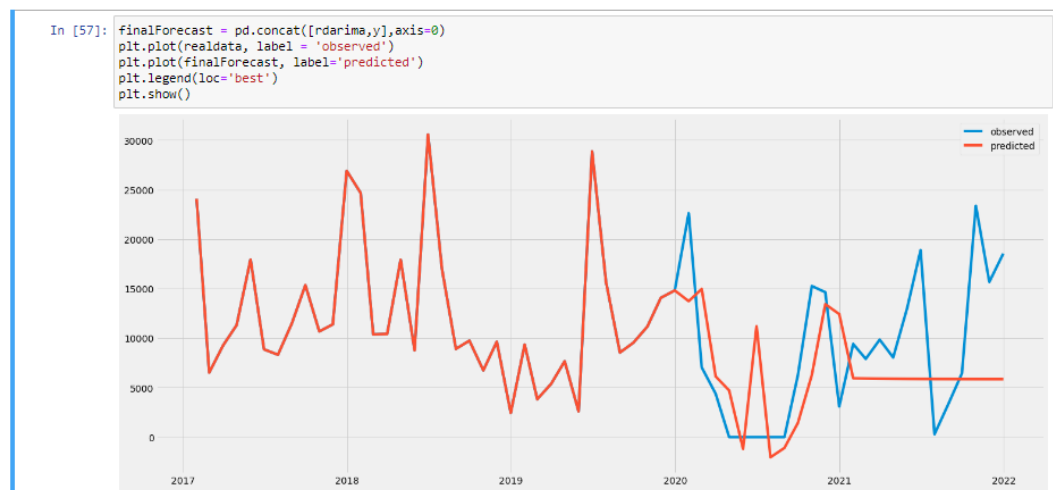
Lampiran 21 Membulatkan Nilai Peramalan



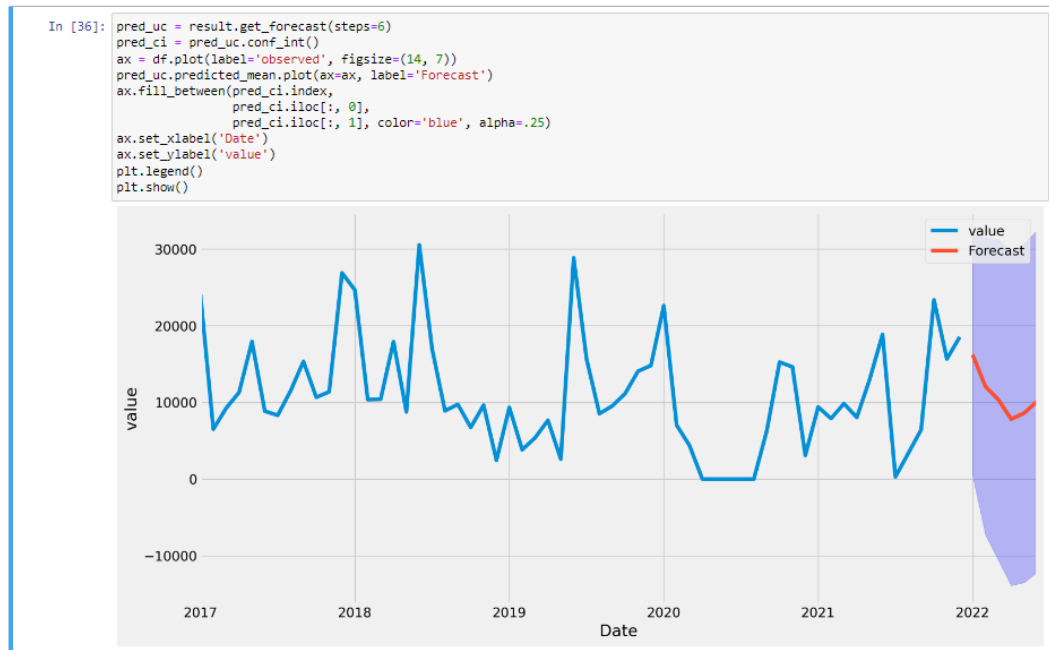
Lampiran 22 Menggabungkan Nilai Peramalan pada Dataset



Lampiran 23 Visualisasi Peramalan Hybrid



Lampiran 24 Peramalan ARIMA musiman 6 bulan



Lampiran 25 Nilai peramalan ARIMA musiman 6 bulan

```
In [37]: pred_uc.predicted_mean
Out[37]: 2022-01-31    16216.456910
         2022-02-28    12066.091194
         2022-03-31    10357.116075
         2022-04-30     7813.603406
         2022-05-31     8591.755976
         2022-06-30    10110.096510
         Freq: M, Name: predicted_mean, dtype: float64

In [38]: print(round(pred_uc.predicted_mean))
2022-01-31    16216.0
2022-02-28    12066.0
2022-03-31    10357.0
2022-04-30     7814.0
2022-05-31     8592.0
2022-06-30    10110.0
         Freq: M, Name: predicted_mean, dtype: float64
```

Lampiran 26 Peramalan Hybrid model FFNN

```
In [76]: future = 6 #months
for i in range(len(test) + future):
    current_pred = model.predict(current_batch_new)[0]
    print(current_pred)
    prediction.append(current_pred)
    current_batch_new = np.append(current_batch_new[:,1:,:], [[current_pred]], axis=1)

print(current_batch_new.shape)
print(current_batch_new)

1/1 [=====] - 0s 19ms/step
[0.40998858]
1/1 [=====] - 0s 16ms/step
[0.41969353]
1/1 [=====] - 0s 15ms/step
[0.42741963]
1/1 [=====] - 0s 15ms/step
[0.43796602]
1/1 [=====] - 0s 15ms/step
[0.44077384]
(1, 6, 1)
[[[0.44029891]
 [0.44061875]
 [0.44079369]
 [0.44086063]
 [0.44083822]
 [0.44077384]]]
```

Lampiran 27 Inverse Normalisasi

```
In [77]: rescaled_prediction = scaler.inverse_transform(prediction)
print(len(rescaled_prediction))
print(rescaled_prediction.shape)
|
rescaled_prediction_new = np.resize(rescaled_prediction, (18,1))
print(rescaled_prediction_new)
print(rescaled_prediction_new.shape)

18
(18, 1)
[[12524.74119562]
 [12821.21763593]
 [13057.24235854]
 [13379.42391184]
 [13668.10797197]
 [13811.25961331]
 [13334.32752585]
 [13399.44611362]
 [13446.3633619 ]
 [13480.4608292 ]
 [13492.54316095]
 [13482.31537744]
 [13450.69155157]
 [13460.46229863]
 [13465.80652946]
 [13467.85135782]
 [13467.1667136 ]
 [13465.20018232]]
(18, 1)
```

Lampiran 28 Pemberian Index Tanggal

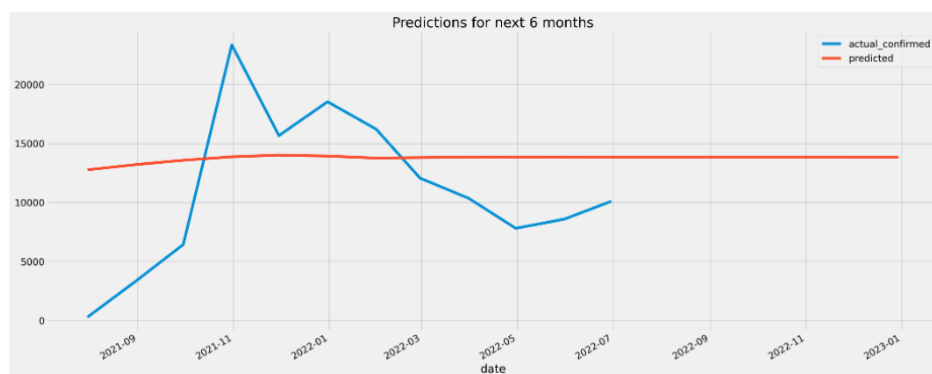
```
In [79]: for k in range(0, future):
time_series_array_ = time_series_array.append(time_series_array[6:] + pd.DateOffset(months=6))

print(len(time_series_array_))
print(time_series_array_.shape)
print(time_series_array_)

18
(18,)
DatetimeIndex(['2021-07-31', '2021-08-31', '2021-09-30', '2021-10-31',
               '2021-11-30', '2021-12-31', '2022-01-31', '2022-02-28',
               '2022-03-31', '2022-04-30', '2022-05-31', '2022-06-30',
               '2022-07-31', '2022-08-28', '2022-09-30', '2022-10-30',
               '2022-11-30', '2022-12-30'],
              dtype='datetime64[ns]', name='date', freq=None)
```

Lampiran 29 Visualisasi Nilai Peramalan

```
In [62]: df_forecast.plot(title="Predictions for next 6 months", figsize=(21,9))
Out[62]: <AxesSubplot:title={'center':'Predictions for next 6 months'}, xlabel='date'>
```



Lampiran 30 Pembulatan nilai Peramalan

```
In [85]: rounded = round(df_forecast)
print(rounded)
```

date	actual_confirmed	predicted
2021-07-31	288.0	12525.0
2021-08-31	3358.0	12821.0
2021-09-30	6435.0	13057.0
2021-10-31	23362.0	13379.0
2021-11-30	15671.0	13668.0
2021-12-31	18539.0	13811.0
2022-01-31	16216.0	13334.0
2022-02-28	12066.0	13399.0
2022-03-31	10357.0	13446.0
2022-04-30	7814.0	13480.0
2022-05-31	8592.0	13493.0
2022-06-30	10110.0	13482.0
2022-07-31	NaN	13451.0
2022-08-28	NaN	13460.0
2022-09-30	NaN	13466.0
2022-10-30	NaN	13468.0
2022-11-30	NaN	13467.0
2022-12-30	NaN	13465.0

Lampiran 31 Penggabungan Nilai Peramalan pada Dataset

```
In [86]: x = rounded[-6:]
print(x)
print("#####")
y = pd.DataFrame(x.loc[:, "predicted"])
print(y)
print("#####")
y.columns = ['value']
print(y)
```

date	actual_confirmed	predicted
2022-07-31	NaN	13451.0
2022-08-28	NaN	13460.0
2022-09-30	NaN	13466.0
2022-10-30	NaN	13468.0
2022-11-30	NaN	13467.0
2022-12-30	NaN	13465.0

#####

date	predicted
2022-07-31	13451.0
2022-08-28	13460.0
2022-09-30	13466.0
2022-10-30	13468.0
2022-11-30	13467.0
2022-12-30	13465.0

#####

date	value
2022-07-31	13451.0
2022-08-28	13460.0
2022-09-30	13466.0
2022-10-30	13468.0
2022-11-30	13467.0
2022-12-30	13465.0

Lampiran 32 Pengisian Nilai Kosong Menggunakan Nilai Rata-Rata

```
In [9]: dfmean['value'] = dataset['value'].replace(np.NaN, dataset['value'].mean())
print(dfmean['value'])
```

executed in 64ms, finished 15:13:05 2023-07-09

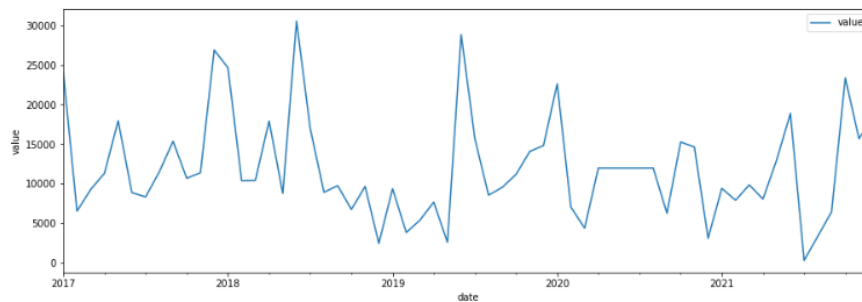
```
date
2017-01-31    24064.000000
2017-02-28    6510.000000
2017-03-31    9264.000000
2017-04-30   11384.000000
2017-05-31   17940.000000
2017-06-30    8857.000000
2017-07-31    8315.000000
2017-08-31   11494.000000
2017-09-30   15348.000000
2017-10-31   10668.000000
2017-11-30   11384.000000
2017-12-31   26892.000000
2018-01-31   24658.000000
2018-02-28   10368.000000
2018-03-31   10417.000000
2018-04-30   17918.000000
2018-05-31    8767.000000
2018-06-30   30549.000000
2018-07-31   16968.000000
2018-08-31   8908.000000
2018-09-30   9738.000000
2018-10-31   6735.000000
2018-11-30   9638.000000
2018-12-31   2440.000000
2019-01-31   9348.000000
2019-02-28   3022.000000
2019-03-31   5302.000000
2019-04-30   7059.000000
2019-05-31   2586.000000
2019-06-30   28854.000000
2019-07-31   15673.000000
2019-08-31   8535.000000
2019-09-30   9536.000000
2019-10-31   11157.000000
2019-11-30   14065.000000
2019-12-31   14810.000000
2020-01-31   22614.000000
2020-02-29   7035.000000
2020-03-31   4373.000000
2020-04-30   11973.363636
2020-05-31   11973.363636
2020-06-30   11973.363636
2020-07-31   11973.363636
2020-08-31   11973.363636
2020-09-30   6752.000000
```

Lampiran 33 Grafik Jumlah Pengunjung Wisata Menggunakan Nilai Rata-Rata

```
In [10]: dfmean.plot(figsize=(15,5))
plt.ylabel('value')
```

executed in 203ms, finished 15:13:05 2023-07-09

Out[10]: Text(0, 0.5, 'value')



Lampiran 34 Tes ADF

```
In [11]: from statsmodels.tsa.stattools import adfuller
res = adfuller(dfmean.value)
print('Augmneted Dickey_fuller Statistic: %f' % res[0])
print('p-value: %f' % res[1])
print('critical values at different levels:')
for k, v in res[4].items():
    print('\t%s: %.3f' % (k, v))
```

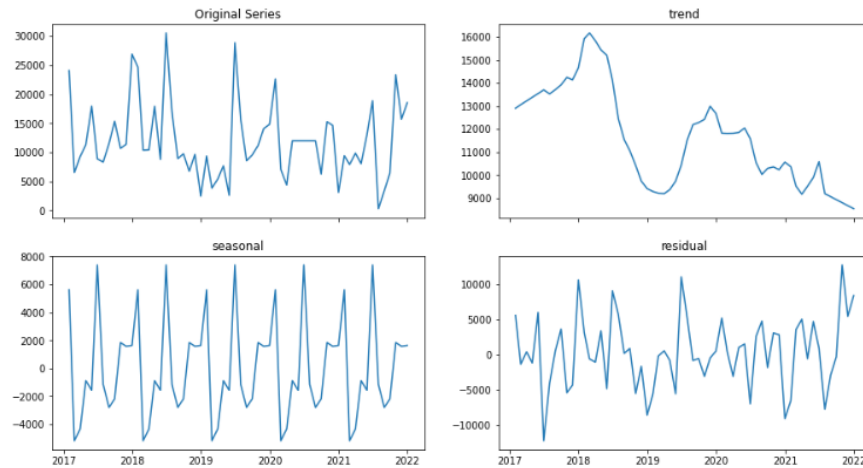
executed in 78ms, finished 15:13:05 2023-07-09

```
Augmneted Dickey_fuller Statistic: -6.797056
p-value: 0.000000
critical values at different levels:
1%: -3.546
5%: -2.912
10%: -2.594
```

Lampiran 35 Dekomposisi Dataset Rata-Rata

```
In [12]: result = seasonal_decompose(dfmean, model='additive', extrapolate_trend='freq')
fig, axes = plt.subplots(2, 2, sharex=True, figsize=(15,8))
axes[0, 0].plot(dfmean); axes[0, 0].set_title('Original Series')
axes[0, 1].plot(result.trend); axes[0, 1].set_title('trend')
axes[1, 0].plot(result.seasonal); axes[1, 0].set_title('seasonal')
axes[1, 1].plot(result.resid); axes[1, 1].set_title('residual')
plt.show()
```

executed in 485ms, finished 15:13:08 2023-07-09



Lampiran 36 Model ARIMA (8,0,0)

```
In [14]: mymodel1 = ARIMA(dfmean.value, order=(8,0,0))
modelFit = mymodel1.fit()
modelFit.summary()
```

executed in 159ms, finished 15:13:09 2023-07-09

C:\Users\Zulqarnain\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequency M will be used.
warnings.warn('No frequency information was provided, so inferred frequency M will be used.')
C:\Users\Zulqarnain\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequency M will be used.
warnings.warn('No frequency information was provided, so inferred frequency M will be used.')
C:\Users\Zulqarnain\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequency M will be used.
warnings.warn('No frequency information was provided, so inferred frequency M will be used.')

Out[14]:

SARIMAX Results

Dep. Variable:	value	No. Observations:	00
Model:	ARIMA(8, 0, 0)	Log Likelihood:	-507.924
Date:	Sun, 09 Jul 2023	AIC:	1235.849
Time:	15:13:08	BIC:	1256.762
Sample:	01-31-2017	HQIC:	1244.041
	-12-31-2021		
Covariance Type:	opg		

	coef	std err	z	P> z	[0.025	0.975]
const	1.197e+04	1041.748	11.494	0.000	9931.579	1.4e+04
ar.L1	0.2010	0.161	1.254	0.210	-0.114	0.517
ar.L2	-0.1159	0.211	-0.555	0.579	-0.530	0.298
ar.L3	0.0213	0.213	0.100	0.920	-0.397	0.439
ar.L4	-0.0078	0.185	-0.042	0.969	-0.371	0.355
ar.L5	0.1087	0.164	0.664	0.507	-0.212	0.430
ar.L6	-0.0427	0.147	-0.291	0.771	-0.330	0.245
ar.L7	0.2410	0.189	1.275	0.202	-0.129	0.611
ar.L8	-0.3288	0.231	-1.428	0.154	-0.781	0.123
sigma2	3.734e+07	0.100	3.74e+08	0.000	3.73e+07	3.73e+07

Ljung-Box (L1) (Q): 0.06 Jarque-Bera (JB): 18.75
Prob(Q): 0.80 Prob(JB): 0.00
Heteroskedasticity (H): 0.49 Skew: 1.21
Prob(H) (two-sided): 0.12 Kurtosis: 4.29

Lampiran 37 Model ARIMA (17,0,0)

```
In [15]: mymodel2 = ARIMA(dfmean.value, order=(17,0,0))
model_fit = mymodel2.fit()
model_fit.summary()

executed in 332ms, finished 15:13:06 2023-07-09

C:\Users\Zulqarnain\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequency M will be used.
warnings.warn("No frequency information was")
C:\Users\Zulqarnain\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequency M will be used.
warnings.warn("No frequency information was")
C:\Users\Zulqarnain\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:524: ValueWarning: No frequency information was provided, so inferred frequency M will be used.
warnings.warn("No frequency information was")
```

```
Out[15]: SARIMAX Results

Dep. Variable:          value      No. Observations:      60
Model:                ARIMA(17, 0, 0)      Log Likelihood:    602.933
Date:                Sun, 09 Jul 2023      AIC:              1243.866
Time:                15:13:06             BIC:              1283.659
Sample:              01-31-2017          HQIC:            1259.431
                  -12-31-2021

Covariance Type:      oppg

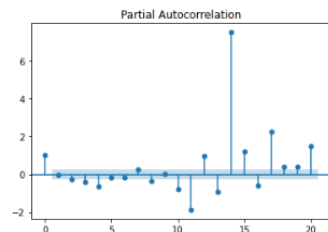
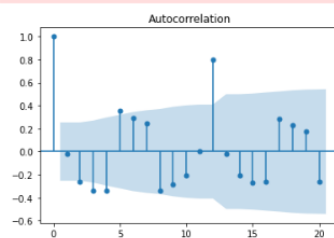
coef      std err      z      P>|z|      [0.025      0.975]
-----
const     1.197e+04    1218.908     9.823    0.000    9564.347    1.44e+04
ar.L1     0.1907         0.156     1.226    0.220    -0.114     0.496
ar.L2    -0.1613         0.213    -0.758    0.449    -0.578     0.256
ar.L3     0.0587         0.187     0.314    0.753    -0.307     0.425
ar.L4     0.0537         0.169     0.318    0.750    -0.277     0.384
ar.L5     0.0253         0.190     0.133    0.894    -0.347     0.398
ar.L6    -0.0083         0.173    -0.048    0.962    -0.348     0.330
ar.L7     0.2979         0.179     1.664    0.096    -0.053     0.649
ar.L8    -0.3524         0.227    -1.555    0.120    -0.796     0.092
ar.L9     0.1297         0.191     0.680    0.497    -0.244     0.504
ar.L10   -0.2795         0.230    -1.215    0.225    -0.731     0.172
ar.L11   -0.0562         0.159    -0.353    0.724    -0.369     0.256
ar.L12   0.0709         0.129     0.551    0.582    -0.181     0.323
ar.L13   -0.0765         0.210    -0.365    0.715    -0.488     0.335
ar.L14   -0.0646         0.337    -0.192    0.848    -0.725     0.596
ar.L15   0.0288         0.233     0.123    0.902    -0.429     0.486
ar.L16   -0.1501         0.154    -0.977    0.328    -0.451     0.151
ar.L17   0.3388         0.160     2.118    0.034    0.025     0.652
sigma2    2.62e+07         0.060    4.37e+08    0.000    2.62e+07    2.62e+07
```

Lampiran 38 Plot ACF dan PACF musiman

```
In [17]: from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
plot_acf(seasonal, lags =20)
plot_pacf(seasonal, lags =20)
plt.show()

executed in 280ms, finished 15:13:07 2023-07-09

C:\Users\Zulqarnain\anaconda3\lib\site-packages\statsmodels\regression\linear_model.py:1434: RuntimeWarning: invalid value encountered in sqrt
return rho, np.sqrt(sigmamq)
```



Lampiran 39 Model ARIMA musiman

```
In [33]: mod1 = sm.tsa.statespace.SARIMAX(dfmean,
                                         order=(17, 0, 0),
                                         seasonal_order=(0, 0, 0, 2),
                                         enforce_stationarity=False,
                                         enforce_invertibility=False)

results = mod1.fit()

print(results.summary())
```

executed in 132ms, finished 19:30:09 2023-07-09

```
=====
SARIMAX Results
=====
Dep. Variable:          value      No. Observations:          60
Model:                 SARIMAX(17, 0, 0)      Log Likelihood:         -429.729
Date:                  Sun, 09 Jul 2023      AIC:                   895.458
Time:                  19:30:09             BIC:                   927.160
Sample:                01-31-2017          HQIC:                  907.149
                    - 12-31-2021

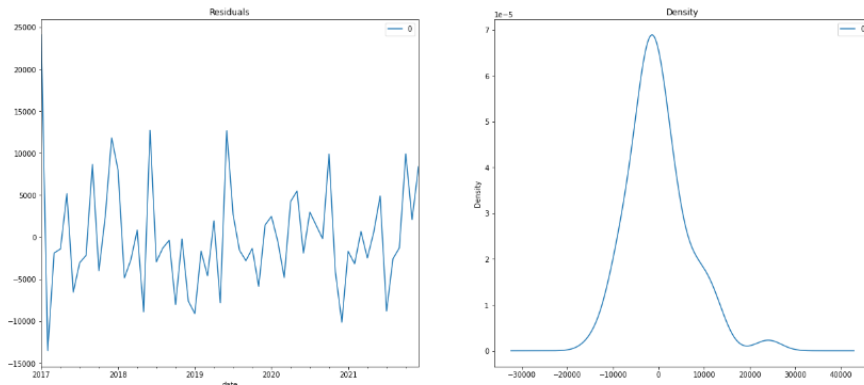
Covariance Type:      opg
=====
              coef      std err          z      P>|z|      [0.025      0.975]
-----
ar.L1         0.2659      0.234         1.135      0.257      -0.193      0.725
ar.L2        -0.0643      0.268        -0.240      0.811      -0.590      0.461
ar.L3         0.0643      0.530         0.121      0.903      -0.975      1.104
ar.L4         0.0202      0.213         0.095      0.924      -0.397      0.438
ar.L5         0.2155      0.203         1.064      0.287      -0.181      0.612
ar.L6         0.1106      0.171         0.647      0.518      -0.224      0.446
ar.L7         0.3172      0.199         1.592      0.111      -0.073      0.708
ar.L8        -0.3804      0.386        -0.986      0.324      -1.136      0.376
ar.L9         0.1508      0.280         0.539      0.590      -0.398      0.699
ar.L10        -0.1953      0.392        -0.498      0.618      -0.964      0.573
ar.L11        -0.1153      0.231        -0.500      0.617      -0.567      0.336
ar.L12         0.1017      0.194         0.523      0.601      -0.279      0.483
ar.L13        -0.0263      0.218        -0.121      0.904      -0.454      0.401
ar.L14         0.0590      0.327         0.180      0.857      -0.582      0.700
ar.L15         0.0329      0.329         0.100      0.920      -0.613      0.679
ar.L16         0.0048      0.196         0.024      0.981      -0.379      0.388
ar.L17         0.3836      0.167         2.301      0.021      0.057      0.710
sigma2        2.805e+07      1.19e-08      2.36e+15      0.000      2.81e+07      2.81e+07
=====
Ljung-Box (L1) (Q):          0.00      Jarque-Bera (JB):          2.40
Prob(Q):                    0.95      Prob(JB):                  0.30
Heteroskedasticity (H):     0.73      Skew:                      0.58
Prob(H) (two-sided):        0.57      Kurtosis:                  3.03
=====
```

Lampiran 40 Plot Residual dan Plot Density

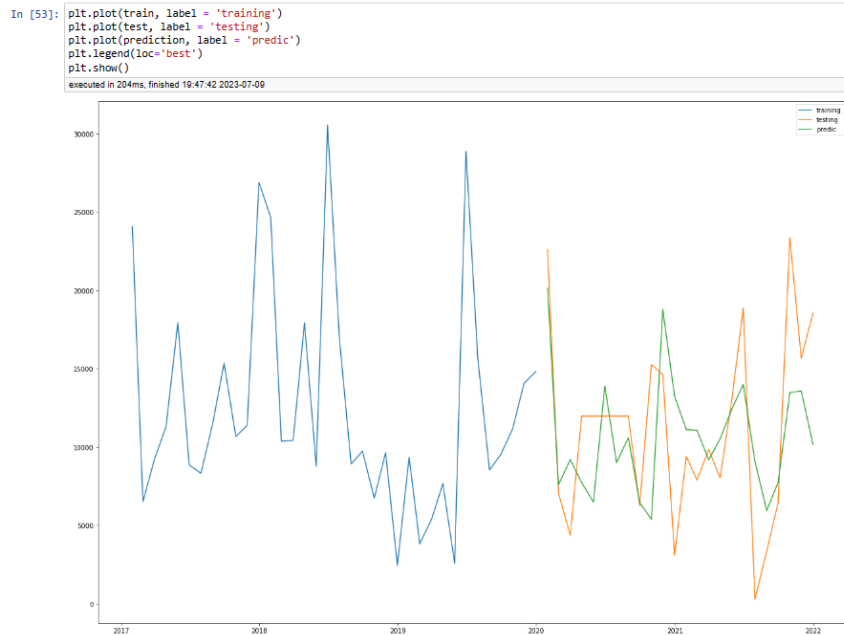
```
In [43]: mymodel = SARIMAX(dfmean.value, order=(17,0,0), seasonal_order=(0,0,0,2))
result = mymodel.fit()

myresiduals = pd.DataFrame(result.resid)
fig, ax = plt.subplots(1,2)
myresiduals.plot(title = "Residuals", ax = ax[0])
myresiduals.plot(kind = 'kde', title = 'Density', ax = ax[1])
plt.rcParams.update({'figure.figsize': (21,15)})
plt.show()
```

executed in 581ms, finished 19:37:14 2023-07-09



Lampiran 41 Grafik Prediksi Data Testing



Lampiran 42 Uji Nilai Error

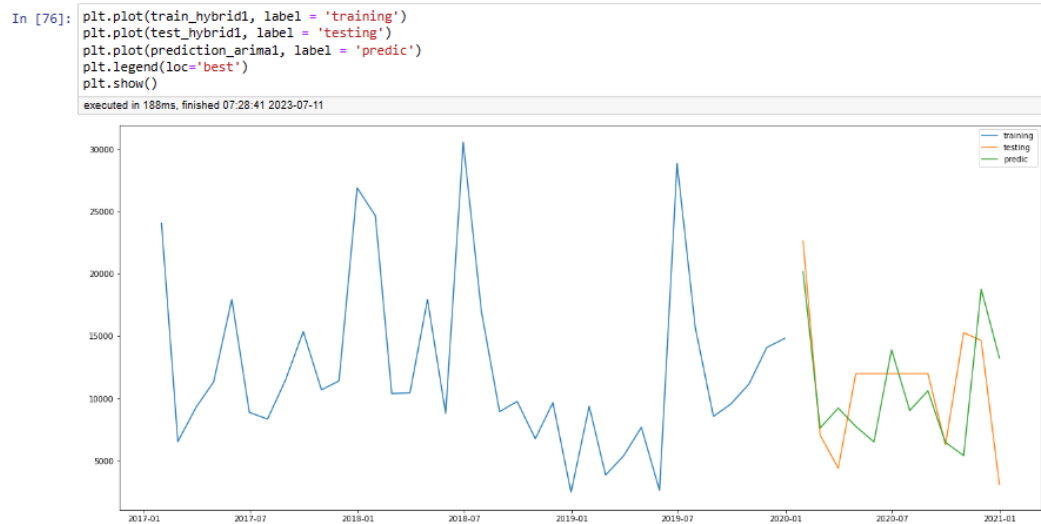
```
In [54]: test_act = test
test_fc = prediction

mape = np.mean(np.abs(test_fc - test_act)/np.abs(test_act))
mae = np.mean(np.abs(test_fc - test_act))
mpe = np.mean(test_fc - test_act)
rmse = np.mean((test_fc - test_act)**2)**.5

import pprint
pprint.pprint({'mape':mape,'mae':mae, 'mpe':mpe,
              'rmse':rmse})
executed in 142ms, finished 19:47:45 2023-07-09
```

```
{'mae': value 3947.359014
 dtype: float64,
 'mape': value 1.692819
 dtype: float64,
 'mpe': value -462.652341
 dtype: float64,
 'rmse': value 5047.593347
 dtype: float64}
```

Lampiran 43 Grafik Peramalan Data Testing ARIMA Musiman



Lampiran 44 Nilai Peramalan Data Testing ARIMA Musiman

```
In [75]: prediction_arima1 = pd.DataFrame(result.predict(n_periods = 12), index = test_hybrid1.index)
prediction_arima1.columns = ['value']
prediction_arima1
executed in 16ms, finished 07:28:30 2023-07-11
```

Out[75]:

	value
2020-01-31	20153.523618
2020-02-29	7595.957080
2020-03-31	9197.152310
2020-04-30	7726.823822
2020-05-31	6483.667303
2020-06-30	13885.633106
2020-07-31	9001.929242
2020-08-31	10589.172789
2020-09-30	6446.173154
2020-10-31	5382.405019
2020-11-30	18770.348433
2020-12-31	13235.924281

Lampiran 45 Pembaruan Dataset Peramalan Data Testing

```
In [64]: df_test_hybrid = pd.read_excel('datasetbaru1-rev.xlsx')
df_test_hybrid['date'] = pd.to_datetime(df_test_hybrid['date'], infer_datetime_format=True)
df_test_hybrid = df_test_hybrid.set_index(['date'])
df_test_hybrid.tail(12)
executed in 47ms, finished 07:22:53 2023-07-11
```

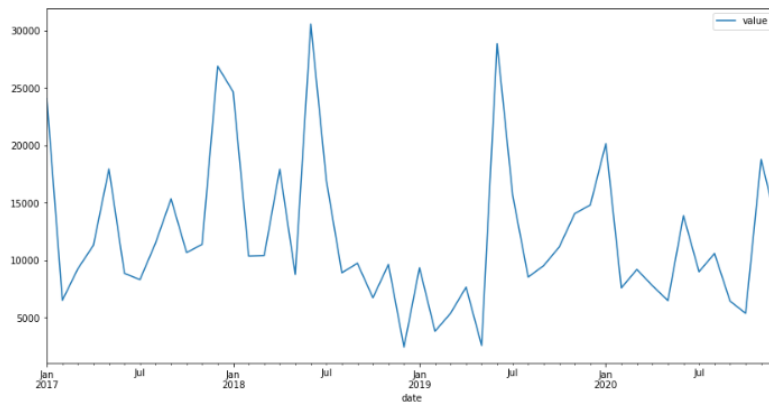
```
Out[64]:
```

date	value
2020-01-31	20153
2020-02-29	7595
2020-03-31	9197
2020-04-30	7788
2020-05-31	6483
2020-06-30	13885
2020-07-31	9001
2020-08-31	10589
2020-09-30	6446
2020-10-31	5382
2020-11-30	18770
2020-12-31	13235

Lampiran 46 Grafik Dataset Baru

```
In [65]: df_test_hybrid.plot(figsize = (14,7))
executed in 219ms, finished 07:23:33 2023-07-11
```

```
Out[65]: <AxesSubplot: xlabel='date'>
```



Lampiran 47 Pembuatan Model FFNN

```
In [81]: scaler_test_hybrid = MinMaxScaler(feature_range=(0, 1))
#scaler.fit(train)
dfHybrid = scaler_test_hybrid.fit_transform(df_test_hybrid)

trainScaled = scaler_test_hybrid.transform(train_hybrid)
testScaled = scaler_test_hybrid.transform(test_hybrid)

trainScaled.shape
executed in 31ms, finished 07:31:51 2023-07-11
```

```
Out[81]: (36, 1)
```

Lampiran 48 Pembuatan Dimensi FFNN

```
In [114]: def to_sequences(df_test_hybrid, seq_size=1):
x = []
y = []

for i in range(len(df_test_hybrid)-seq_size-1):
    #print(i)
    window = df_test_hybrid[i:(i+seq_size), 0]
    x.append(window)
    y.append(df_test_hybrid[i+seq_size, 0])

return np.array(x),np.array(y)
executed in 15ms, finished 07:41:11 2023-07-11
```

```
In [118]: seq_size = 6
x_train, y_train = to_sequences(trainScaled,seq_size)
x_test, y_test = to_sequences(testScaled, seq_size)

x_train_ = np.expand_dims(x_train, axis=2)
y_train_ = np.expand_dims(y_train, axis=1)
x_test_ = np.expand_dims(x_test, axis=2)
y_test_ = np.expand_dims(y_test, axis=1)
executed in 14ms, finished 07:41:35 2023-07-11
```

```
In [119]: print(x_train.shape, y_train.shape, x_test.shape, y_test.shape, x_train_.shape, x_test_.shape, y_train_.shape, y_test_.shape)
executed in 10ms, finished 07:41:38 2023-07-11
(29, 6) (29,) (5, 6) (5,) (29, 6, 1) (5, 6, 1) (29, 1) (5, 1)
```

Lampiran 49 Nilai Peramalan ARIMA musiman tahun 2022 Menggunakan Dataset Rata-Rata

```
In [67]: pred_uc.predicted_mean
executed in 20ms, finished 07:51:22 2023-07-10
```

```
Out[67]: 2022-01-31    15081.568492
2022-02-28    4783.599285
2022-03-31    13617.335815
2022-04-30    13355.841224
2022-05-31    15844.942240
2022-06-30    11657.297092
2022-07-31    12518.753858
2022-08-31     7533.816604
2022-09-30     5829.320054
2022-10-31    12891.863302
2022-11-30    13494.271114
2022-12-31    10106.198820
Freq: M, Name: predicted_mean, dtype: float64
```

```
In [68]: print(round(pred_uc.predicted_mean))
executed in 9ms, finished 07:51:25 2023-07-10
```

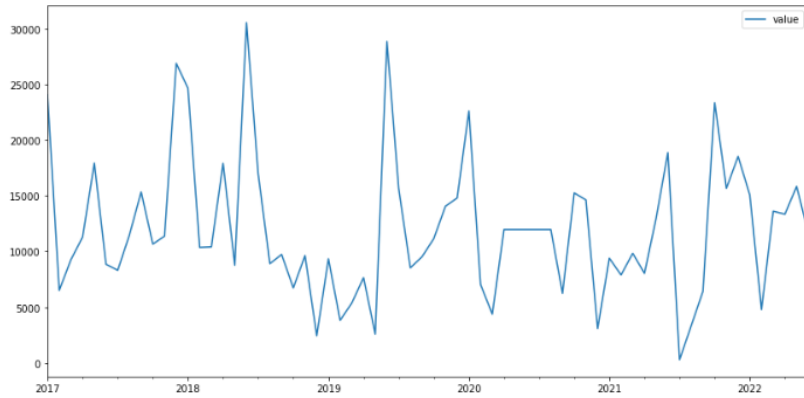
```
2022-01-31    15082.0
2022-02-28     4784.0
2022-03-31    13617.0
2022-04-30    13356.0
2022-05-31    15845.0
2022-06-30    11657.0
2022-07-31    12519.0
2022-08-31     7534.0
2022-09-30     5829.0
2022-10-31    12892.0
2022-11-30    13494.0
2022-12-31    10106.0
Freq: M, Name: predicted_mean, dtype: float64
```

Lampiran 50 Pembaruan Dataset Peramalan Hybrid

```
In [239]: df_fc_hybrid.plot(figsize = (14,7))
```

```
executed in 172ms, finished 13:31:09 2023-07-11
```

```
Out[239]: <AxesSubplot: xlabel='date'>
```



Lampiran 51 Pemberian Index Tanggal Pada Nilai Peramalan Hybrid Model FFNN

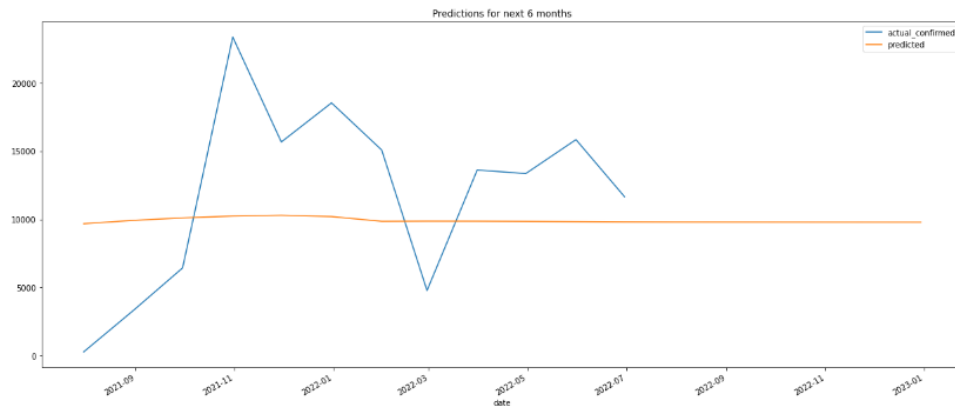
```
In [271]: fc_hybrid = pd.DataFrame(columns=["actual_confirmed", "predicted"], index=time_series_array_)
```

```
fc_hybrid.loc[:, "predicted"] = rescaled_prediction_new[:, 0]
fc_hybrid.loc[:, "actual_confirmed"] = test_fc_hybrid["value"] # In[ ]:
executed in 47ms, finished 13:53:42 2023-07-11
```

```
In [272]: fc_hybrid.plot(title="Predictions for next 6 months", figsize=(21,9))
```

```
executed in 170ms, finished 13:53:44 2023-07-11
```

```
Out[272]: <AxesSubplot: title={'center': 'Predictions for next 6 months'}, xlabel='date'>
```



Lampiran 52 Nilai Peramalan Hybrid Model FFNN

In [273]: fc_hybrid

executed in 31ms, finished 13:53:49 2023-07-11

Out[273]:

date	actual_confirmed	predicted
2021-07-31	288.0	9079.905404
2021-08-31	3358.0	9924.151115
2021-09-30	6435.0	10107.883881
2021-10-31	23362.0	10238.756009
2021-11-30	15671.0	10293.530051
2021-12-31	18539.0	10201.805788
2022-01-31	15082.0	9853.205432
2022-02-28	4784.0	9884.446969
2022-03-31	13617.0	9881.050609
2022-04-30	13356.0	9847.549041
2022-05-31	15845.0	9827.887851
2022-06-30	11657.0	9807.814517
2022-07-31	NaN	9798.384793
2022-08-28	NaN	9795.980466
2022-09-30	NaN	9792.998251
2022-10-30	NaN	9790.185191
2022-11-30	NaN	9788.022559
2022-12-30	NaN	9788.708568

Lampiran 53 Link Lampiran Online

<https://drive.google.com/drive/folders/1YUNJ4aCTZvhscQluAZ9TEz9E67unL10p?usp=sharing>