TUGAS AKHIR – IS184853

PERAMALAN JUMLAH KASUS VIRUS CORONA DI INDONESIA MENGGUNAKAN ARIMA-LSTM

FORECASTING CORONAVIRUS CASES IN INDONESIA USING ARIMA-LSTM

LEE YE CHAN
NRP 05211740000094

Dosen Pembimbing :
Raras Tyasnurita, S.Kom., M.BA., Ph.D.
Ahmad Mukhlason, S.Kom., M.Sc., Ph.D.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Elekro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

TUGAS AKHIR – IS184853

PERAMALAN JUMLAH KASUS VIRUS CORONA DI INDONESIA MENGGUNAKAN ARIMA-LSTM

LEE YE CHAN
NRP 05211740000094

Dosen Pembimbing :
Raras Tyasnurita, S.Kom., M.BA., Ph.D
Ahmad Mukhlason, S.Kom., M.Sc., Ph.D.

DEPARTEMEN SISTEM INFORMASI
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

TUGAS AKHIR – IS184853

FORECASTING CORONAVIRUS CASES IN INDONESIA USING ARIMA-LSTM

LEE YE CHAN
NRP 05211740000094

SUPERVISOR :
Raras Tyasnurita, S.Kom., M.BA., Ph.D
Ahmad Mukhlason, S.Kom., M.Sc., Ph.D.

DEPARTMENT OF INFORMATION SYSTEMS
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2021

PERAMALAN JUMLAH KASUS VIRUS CORONA DI INDONESIA MENGGUNAKAN ARIMA-LSTM

LP/P/21/059

LEMBAR PENGESAHAN

PERAMALAN JUMLAH KASUS VIRUS CORONA DI INDONESIA MENGGUNAKAN ARIMA-LSTM

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer (S.Kom)
pada
Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas (ELECTICS)
Institut Teknologi Sepuluh Nopember

Oleh

Lee Ye Chan
05211740000094

Surabaya, 12 Agustus 2021

Kepala Departemen Sistem Informasi

Dr. Madjahidin, ST., MT.
NIP. 197010102003121001

LEMBAR PERSETUJUAN

PERAMALAN JUMLAH KASUS VIRUS CORONA DI INDONESIA MENGGUNAKAN ARIMA-LSTM

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Oleh :

Lee Ye Chan
NRP: 05211740000094

Disetujui Tim Penguji: Tanggal Ujian: 30 July 2021
Periode Wisudas: September 2021

Raras Tyasnurita, S.Kom, MBA, Ph.D (Pembimbing 1)

Ahmad Mukhlason, S.Kom, M.Sc, Ph.D (Pembimbing 2)

Dr. Retno Aulia Vinarti, S.Kom, M.Kom (Penguji 1)

Edwin Riksakomara, S.Kom, M.T (Penguji 2)

Nama Mahasiswa : Lee Ye Chan
NRP : 05211740000094
Departemen : Sistem Informasi FTEIC-ITS
Pembimbing 1 : Raras Tyasnurita, S.Kom., M.BA., Ph.D
Pembimbing 2 : Ahmad Muklason, S.Kom., M.Sc., Ph.D.

ABSTRAK

Virus adalah salah satu jenis penyebab penyakit yang bahaya dan mudah untuk menyebar. Virus corona atau juga dikenal sebagai COVID-19 adalah virus yang baru saja ditemukan pada Desember tahun 2019 dan menjadi ancaman kesehatan terbesar di seluruh dunia. Gejala penyakit virus corona dapat muncul dengan berbagai macam cara kepada setiap orang. Gejala umum dari penyakit virus corona adalah demam tinggi, batuk kering, kelelahan, nyeri otot, peradangan tenggorokan, diare, konjungtivitis, sakit kepala, hilangnya kemampuan mengecap rasa atau mencium bau, dan ruam di kulit. Gejala-gejala tersebut pada umumnya muncul dalam waktu 5-6 hari setelah penderita terpapar virus corona namun bisa saja membutuh hingga 14 hari untuk gejalanya muncul. Indonesia merupakan salah satu negara dengan jumlah kasus virus corona terbanyak di dunia. Jumlah kasus virus corona terus bertambah setiap hari di Indonesia dan juga menyebabkan kenaikan kematian yang disebabkan oleh virus corona. Dengan bertambahnya jumlah kasus virus corona dan jumlah kematian virus corona di Indonesia maka dibutuhkan penanganan yang cepat dan efektif.

Penelitian ini bertujuan untuk melakukan peramalan jumlah kasus virus corona di Indonesia untuk mempersiapkan fasilitas kesehatan dan juga untuk membantu pemerintah untuk melakukan usaha penanganan virus corona. Pada penelitian ini akan menggunakan metode hybrid Autoregressive Integrated Moving Average-Long Short Term Memory (hybrid ARIMA-

LSTM) dalam meramalkan jumlah kasus virus corona. Metode hybrid ARIMA-LSTM dipilih karena dapat melakukan peramalan pada tiga pola data yaitu trend, seasonal dan residual. Pola trend akan dihitungkan oleh model ARIMA dan pola seasonal dan residual akan dihitungkan oleh model LSTM. Metode hybrid ARIMA-LSTM diharapkan memberikan hasil peramalan yang lebih akurat dibandingkan dengan metode ARIMA atau LSTM. Hasil dari penelitian ini akan dibandingkan dengan penelitian sebelum yang pernah diteliti untuk mencari metode peramalan terbaik untuk kasus virus corona di Indonesia.

Hasil dari performa model peramalan virus corona di Indonesia menggunakan data periode Maret 2020 hingga Desember 2020 menunjukkan bahwa LSTM menghasilkan nilai akurasi yang lebih baik dibandingkan dengan ARIMA dan Hybrid ARIMA-LSTM. Model LSTM menghasilkan nilai MAPE 6,0197%, dan RMSE 473.4757. Sementara untuk peramalan virus corona di Indonesia menggunakan data periode Maret 2020 hingga Juli 2021 model Hybrid ARIMA-LSTM menghasilkan nilai akurasi yang lebih baik dibandingkan ARIMA dan LSTM. Model Hybrid ARIMA-LSTM menghasilkan nilai MAPE 5,7135%, dan RMSE 1212.3503. Hasil penelitian menunjukkan bahwa penggunaan data yang lebih banyak untuk model Hybrid ARIMA-LSTM menghasilkan performa akurasi yang lebih baik.

Kata Kunci: Virus Corona, Peramalan, Hybrid ARIMA-LSTM

FORECASTING CORONAVIRUS CASES IN INDONESIA USING ARIMA-LSTM

Name	**: Lee Ye Chan**
NRP	**: 05211740000094**
Department	**: Information Systems ELECTICS-ITS**
Supervisor 1	**: Raras Tyasnurita, S.Kom., M.BA., Ph.D**
Supervisor 2	**: Ahmad Muklason, S.Kom., M.Sc., Ph.D.**

ABSTRACT

Virus is kind of disease which is dangerous and easy to spread. Corona viurs or also known as COVID19 is a new virus found in December 2019 and became one of biggest threat to health worldwide. Symptoms of corona virus disease appear in many different way to different kind of people. Common symptom of corona virus is high fever, dry cough, fatigue, muscle ache, sore throat, diarrhea, congestion, headache, loss of tatse and smell. These symptoms generally appears between 5 to 6 days after being exposed to corona virus, but it could also take up to 14 days for the symptoms to appear in some people. Indonesia is one of the country with highest corona virus around world. Increase in corona virus cases in Indonesia happens everyday and it also causes more deaths due to corona virus. With the increasing number of corona virus cases and deaths due to corona virus in Indonesia, solutions are needed fast and effectively.

This research is purposed to forecast corona virus cases in Indonesia in order to prepare more health facilities and also to help the government to take actions opposing the corona virus. In this research, hybrid Autoregressive Integrated Moving Average-Long Short Term Memory (hybrid ARIMA-LSTM) model is going to be used to forecast the corona virus cases. Hybrid ARIMA-LSTM method is used because some past

researches manage to perform better compared to ARIMA method and LSTM method. Hybrid ARIMA-LSTM method is hoped to find more accurate result in comparison to ARIMA method or LSTM method. Result of this research will be compared to past researches to find the better model to forecast corona virus cases in Indonesia.

Result from predicting corona virus in Indonesia using data from March 2020 until Desember 2020 shows that LSTM model performs better than ARIMA and Hybrid ARIMA-LSTM. LSTM model resulted in MAPE value of 6,0197%, and RMSE value of 473.4757. Mean while predicting corona virus in Indonesia using data from March 2020 until July 2021 shows that Hybrid ARIMA-LSTM could perform better than ARIMA and LSTM. Hybrid ARIMA-LSTM model resulted MAPE value of 5,7135%, and RMSE value of 1212.3503. Result from this research shows that using more data for Hybrid ARIMA-LSTM model could result in better performance.

Keywords: Corona Virus, Forecasting, Hybrid ARIMA-LSTM

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertandatangan di bawah ini:

Nama	: Lee Ye Chan
NRP	: 05211740000094
Tempat / Tanggal lahir	: Busan, Korea Selatan / 20 Oktober 1998
Fakultas / Departemen	: FTEIC / Sistem Informasi
Nomor Telp / Hp / Email	: 082187387682 / lyc98yc@gmail.com

Dengan ini menyatakan dengan sesungguhnya bahwa penelitian/makalah/tugas akhir saya yang berjudul:

PERAMALAN JUMLAH KASUS VIRUS CORONA DI INDONESIA MENGGUNAKAN ARIMA-LSTM

Bebas Dari Plagiarisme Dan Bukan Hasil Karya Orang Lain.

Apabila dikemudian hari ditemukan seluruh atau sebagian penelitian/makalah/tugas akhir tersebut terdapat indikasi plagiarisme, maka saya bersedia menerima sanksi sesuai peraturan dan ketentuan yang berlaku. Demikian surat pernyataan ini saya buat dengan sesungguhnya dan untuk dipergunakan sebagaimana mestinya.

Surabaya, 30 Juli 2021

Lee Ye Chan
NRP.05211740000094

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan berkat dan rahmat-Nya kepada penulis sehingga penulis dapat menyelesaikan buku tugas akhir dengan judul:

PERAMALAN JUMLAH KASUS VIRUS CORONA DI INDONESIA MENGGUNAKAN ARIMA-LSTM

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa material maupun spiritual demi tercapainya tujuan pembuatan tugas akhir ini. Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari berbagai pihak yang sudah meluangkan waktu, tenaga dan pikirannya. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada :

1. Tuhan Yang Maha Esa yang telah memberi segala rahmat dan pencerahan untuk dapat menyelesaikan buku tugas akhir.
2. Bapak Lee Jae Won dan Ibu Ahn Young selaku kedua orang tua dari penulis serta Mas Lee Sac dan Lee Ye Jun selaku saudara kandung yang selalu memberi dukungan.
3. Ibu Raras Tyasnurita, S.Kom., M.BA., Ph.D. dan Bapak Ahmad Mukhlason, S.Kom., M.Sc., Ph.D. selaku dosen pembimbing yang senantiasa meluangkan waktu, memberikan ilmu dan petunjuk, serta memotivasi untuk kelancaran tugas akhir penulis.
4. Ibu Dr. Retno Aulia Vinarti S.Kom., M.Kom dan Bapak Edwin Riksakomara, S.Kom., MT. selaku dosen penguji yang telah memberikan *feedback* berupa kritik dan saran untuk kebaikan pengerjaan tugas akhir.
5. Bapak Edwin Riksakomara, S.Kom., MT. selaku dosen wali yang telah memberikan banyak saran dan dukungan selama masa perkuliahan.

6. Segenap dosen dan karyawan Departemen Sistem Informasi FTEIC ITS yang telah membantu serta memberikan ilmu yang bermanfaat.
7. Teman-teman grup ‘FIFA 2069’ yaitu Ian, Bram, Firwam, Yoel dan Imago yang merupakan *circle* terdekat penulis yang selalu menemani dan mewarnai hari-hari penulis selama perkuliahan.
8. Teman-teman DHISTAKARNA yang telah membersamai penulis dan memberikan banyak kenangan selama perkuliahan.
9. Berbagai pihak yang tidak bisa disebutkan satu persatu yang turut serta mendukung penulis menyelesaikan Studi.

Penyusunan laporan ini masih jauh dari kata sempurna sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, 21 Juli 2021

Penulis

DAFTAR ISI

ABSTRAK v
ABSTRACT vii
SURAT PERNYATAAN BEBAS PLAGIARISME ix
KATA PENGANTAR x
DAFTAR GAMBAR xv
DAFTAR TABEL xvii
DAFTAR KODE PROGRAM xix
BAB I. PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Perumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan 4
1.5 Manfaat 4
1.6 Relevansi 4
BAB II. TINJAUAN PUSTAKA 6
2.1 Penelitian Sebelumnya 6
2.2 Dasar Teori 9
2.2.1 Virus Corona 9
2.2.2 Peramalan (*Forecasting*) 9
2.2.3 Autoregressive Integrated Moving Average (ARIMA) 10
2.2.4 Artificial Neural Network (ANN) 11
2.2.5 Long Short Term Memory (LSTM) 13
2.2.6 *Hybrid Autoregressive Integrated Moving Average – Long Short Term Memory* (Hybrid ARIMA-LSTM) 15
2.2.7 Performa Peramalan 17
BAB III. METODOLOGI PENELITIAN 19
3.1 Tahapan Pengerjaan Tugas Akhir 19
3.2 Uraian Metodologi 20
3.2.1 Studi Literatur 20
3.2.2 Pra-Proses Data 20
3.2.3 Pembuatan Model Hybrid ARIMA-LSTM 20
3.2.4 Melakukan Peramalan pada Data Test Set 21
3.2.5 Evaluasi Model Peramalan 21

3.2.6 Penyusunan Buku Tugas Akhir 21
BAB IV. PERANCANGAN 22
4.1 Rancangan Pengumpulan Data 22
4.2 Rancangan Pra-Proses Data 22
4.2.1 Penentuan Variabel 22
4.2.2 Menghilangkan Missing Value 22
4.2.3 Pembagian Data 23
4.2.4 Normalisasi Data 23
4.3 Pembentukan Model 23
4.3.1 Pembentukan Model ARIMA 23
4.3.2 Pembentukan Model LSTM 25
4.3.3 Pembentukan Model Hybrid ARIMA-LSTM 27
BAB V. IMPLEMENTASI 29
5.1 Lingkungan Implementasi 29
5.2 Pengumpulan Data 30
5.3 Pra-Proses Data 30
5.3.1 Penentuan Variabel 30
5.3.2 Menghilangkan *Missing Value* 30
5.3.3 Pembagian Data 31
5.3.4 Normalisasi Data 32
5.4 Pembentukan Model 32
5.4.2 Pembentukan Model LSTM 36
5.4.3 Pembentukan Model Hybrid ARIMA-LSTM 39
BAB VI. HASIL DAN PEMBAHASAN 46
6.1 Hasil Pengumpulan Data 46
6.1.1 Hasil Pengumpulan Data 46
6.1.2 Visualisasi Data 47
6.2 Hasil Pra-Proses Data 47
6.2.1 Hasil Penentuan Variabel 47
6.2.2 Hasil Menghilangkan Missing Value 48
6.2.3 Hasil Pembagian Data 50
6.2.4 Hasil Normalisasi Data 51
6.3 Hasil Pembentukan Model 52
6.3.1 Hasil Pembentukan Model ARIMA 52
6.3.2 Hasil Pembentukan Model LSTM 59
6.3.3 Hasil Pembentukan Model Hybrid ARIMA-LSTM 68

6.4 Hasil Perbandingan Model Peramalan 87
6.5 Hasil *Forecasting* Data Masa Depan Dengan Data 2020 88
6.6 Hasil Uji Skenario Dengan Data 2021 89
6.7 Hasil *Forecasting* Data Masa Depan Dengan Data 2020-2021 91
6.7.1 Saran Penanganan Virus Corona 92
6.8 Hasil Uji Skenario Dengan Data Korea Selatan 93
6.9 Rangkuman Hasil dan Pembahasan 94
BAB VII. KESIMPULAN DAN SARAN 96
7.1 Kesimpulan 96
7.2 Saran 96
DAFTAR PUSTAKA 98
BIODATA PENULIS 101
LAMPIRAN A 102
LAMPIRAN B 107
LAMPIRAN C 108
LAMPIRAN D 109
LAMPIRAN E 110

DAFTAR GAMBAR

Gambar 1.1 Roadmap Penelitian RDIB 5
Gambar 2.1 Arsitektur ANN 12
Gambar 2.2 Arsitektur LSTM Sederhana..... 13
Gambar 2.3 Struktur LSTM Sederhana [8] 14
Gambar 2.4 Model Hybrid ARIMA-LSTM [8] 16
Gambar 3.1 Tahapan Pengerjaan Tugas Akhir 19
Gambar 4.1 Hasil Dekomposisi Data 28
Gambar 5.1 Augmented Dickey-Fuller Test 34
Gambar 5.2 Hasil Auto-ARIMA 35
Gambar 5.3 Parameter ARIMA Untuk Komponen Trend 41
Gambar 6.1 Visualisasi Data Virus Corona Yang Terjadi Di Indonesia 47
Gambar 6.2 Hasil Normalisasi Data 52
Gambar 6.3 Perbandingan AIC Model ARIMA 53
Gambar 6.4 Perbandingan AIC Model SARIMA 55
Gambar 6.5 Grafik Perbandingan Model ARIMA dan SARIMA Dengan Data Aktual 56
Gambar 6.6 Perbandingan Nilai Peramalan Dengan Data Aktual Model SARIMA(0,1,1)(2,0,2)[13] Dan Model SARIMA(0,1,1)(1,0,1)[7] 59
Gambar 6.7 Hasil Perbandingan Peramalan LSTM Dengan Data Aktual 66
Gambar 6.8 Komponen Trend 69
Gambar 6.9 Komponen Seasonal 69
Gambar 6.10 Komponen Residual 70
Gambar 6.11 Hasil Fungsi auto_arima Terhadap Komponen Trend 71
Gambar 6.12 Hasil Perbandingan Peramalan Model ARIMA Terhadap Komponen Trend Dengan Data Trend Aktual 72
Gambar 6.13 Hasil Perbandingan Peramalan Model LSTM Terhadap Komponen Seasonal Dengan Data Seasonal Aktual 79
Gambar 6.14 Hasil Perbandingan Peramalan Model LSTM Terhadap Komponen Residual Dengan Data Residual Aktual 86

Gambar 6.15 Hasil Perbandingan Model Hybrid ARIMA-LSTM Dengan Data Aktual 87
Gambar 6.16 Perbandingan Model Peramalan Dalam Bentuk Grafik 88
Gambar 6.17 Perbandingan Hasil Forecast dan Data Aktual Dengan Model LSTM 89
Gambar 6.18 Grafik Perbandingan Model Peramalan 90
Gambar 6.19 Hasil Forecast Masa Depan.....91
Gambar 6.20 Percobaan Model Hybrid ARIMA-LSTM Pada Data Korea Selatan.....93

DAFTAR TABEL

Tabel 2.1 Rangkuman Penelitian Sebelumnya.....6
Tabel 2.2 Standar Penilaian Akurasi MAPE.....18
Tabel 4.1 Parameter Yang Digunakan 25
Tabel 4.2 Penjelasan Parameter 25
Tabel 5.1 Spesifikasi Perangkat Pengerjaan Tugas Akhir 29
Tabel 5.2 Daftar Library.....29
Tabel 5.3 Pengisian Missing Value.....31
Tabel 6.1 Data Yang Diperoleh Dari WHO.....46
Tabel 6.2 Hasil Penentuan Variabel.....48
Tabel 6.3 Missing Value 49
Tabel 6.4 Hasil Menghilangkan Missing Value.....50
Tabel 6.5 Pembagian Data Training Dan Data Testing.....51
Tabel 6.6 Tabel Perbandingan Validasi Model.....56
Tabel 6.7 Perbandingan Faktor M Pada Model SARIMA 57
Tabel 6.8 Hasil Tuning Parameter Timestep 1.....60
Tabel 6.9 Hasil Tuning Parameter Timestep 2.....60
Tabel 6.10 Hasil Tuning Parameter Timestep 3.....61
Tabel 6.11 Hasil Tuning Parameter Timestep 4.....62
Tabel 6.12 Hasil Tuning Parameter Timestep 5.....62
Tabel 6.13 Hasil Tuning Parameter Timestep 6.....63
Tabel 6.14 Hasil Tuning Parameter Timestep 7.....64
Tabel 6.15 Hasil Tuning Parameter Timestep 8.....64
Tabel 6.16 Hasil Tuning Parameter Timestep 9.....65
Tabel 6.17 Hasil Tuning Parameter Timestep 10.....65
Tabel 6.18 Perbandingan MAPE Terendah Dari Setiap Timestep.....67
Tabel 6.19 Perbandingan RMSE Terendah Dari Setiap Timestep.....68
Tabel 6.20 Hasil Tuning Parameter Komponen Seasonal Timestep 1.....73
Tabel 6.21 Hasil Tuning Parameter Komponen Seasonal Timestep 2.....73
Tabel 6.22 Hasil Tuning Parameter Komponen Seasonal Timestep 3.....74

Tabel 6.23 Hasil Tuning Parameter Komponen Seasonal Timestep 4 74
Tabel 6.24 Hasil Tuning Parameter Komponen Seasonal Timestep 5 75
Tabel 6.25 Hasil Tuning Parameter Komponen Seasonal Timestep 6 76
Tabel 6.26 Hasil Tuning Parameter Komponen Seasonal Timestep 7 76
Tabel 6.27 Hasil Tuning Parameter Komponen Seasonal Timestep 8 77
Tabel 6.28 Hasil Tuning Parameter Komponen Seasonal Timestep 9 78
Tabel 6.29 Hasil Tuning Parameter Komponen Seasonal Timestep 10 78
Tabel 6.30 Hasil Tuning Parameter Komponen Residual Timestep 1 80
Tabel 6.31 Hasil Tuning Parameter Komponen Residual Timestep 2 80
Tabel 6.32 Hasil Tuning Parameter Komponen Residual Timestep 3 81
Tabel 6.33 Hasil Tuning Parameter Komponen Residual Timestep 4 81
Tabel 6.34 Hasil Tuning Parameter Komponen Residual Timestep 5 82
Tabel 6.35 Hasil Tuning Parameter Komponen Residual Timestep 6 83
Tabel 6.36 Hasil Tuning Parameter Komponen Residual Timestep 7 83
Tabel 6.37 Hasil Tuning Parameter Komponen Residual Timestep 8 84
Tabel 6.38 Hasil Tuning Parameter Komponen Residual Timestep 9 85
Tabel 6.39 Hasil Tuning Parameter Komponen Residual Timestep 10 85
Tabel 6.40 Perbandingan Model Peramalan 87
Tabel 6.41 Hasil Perbandingan Model Terbaik Data Terbaru 90

DAFTAR KODE PROGRAM

Kode Program 5.1 Pembagian Data 32
Kode Program 5.2 Mengimport Fungsi Normalisasi 32
Kode Program 5.3 Library Yang Digunakan Untuk Model ARIMA 33
Kode Program 5.4 Praproses Data 33
Kode Program 5.5 Penentuan Parameter ARIMA 34
Kode Program 5.6 Pembentukkan Model ARIMA 36
Kode Program 5.7 Pengukuran Akurasi Model 36
Kode Program 5.8 Library Yang Digunakan Untuk Model ARIMA 37
Kode Program 5.9 Praproses Data Untuk Model LSTM 37
Kode Program 5.10 Pelatihan Model LSTM.....38
Kode Program 5.11 Peramalan Model LSTM.....38
Kode Program 5.12 Normalisasi Data Dan Plotting 39
Kode Program 5.13 Library Yang Digunakan Untuk Model Hybrid ARIMA-LSTM 39
Kode Program 5.14 Dekomposisi 40
Kode Program 5.15 Auto_arima Pada Komponen Trend 40
Kode Program 5.16 Pembagian Komponen Trend 41
Kode Program 5.17 Pelatihan Model ARIMA Untuk Komponen Trend 41
Kode Program 5.18 Peramalan Model ARIMA Untuk Komponen Trend 42
Kode Program 5.19 Pembagian Komponen Seasonal.....42
Kode Program 5.20 Normalisasi Data Dan Penentuan Parameter.....43
Kode Program 5.21 Pelatihan Model LSTM Untuk Komponen Nonlinier 43
Kode Program 5.22 Peramalan Model LSTM Untuk Komponen Nonlinier 44
Kode Program 5.23 Normalisasi Data Dan Plotting Perbandingan Untuk Komponen Nonlinier.....44
Kode Program 5.24 Pemodelan Hybrid ARIMA-LSTM 45

BAB I.
PENDAHULUAN

Pada bagian ini, akan dijelaskan mengenai latar belakang permasalahan, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan relevansi hasil tugas akhir terhadap ilmu yang didapatkan dari jurusan dan laboratorium.

Berdasarkan uraian bagian ini, diharapkan dapat memberi gambaran umum mengenai tugas akhir yang dikerjakan.

1.1 Latar Belakang

Penyakit-penyakit dapat menular dengan banyak cara dan salah satunya adalah melalui virus. Virus adalah sebuah patogen yang menyebabkan penyakit dan membutuhkan inang untuk bereplikasi. Virus terdiri atas berbagai jenis yang dapat menyerang tubuh manusia yang memiliki sistem imun lemah. Salah satu jenis virus yang saat ini paling berbahaya dan mudah sekali menyebar adalah virus corona. Menurut *World Health Organization (WHO)*, virus corona adalah penyakit menular yang disebabkan oleh jenis virus baru yang ditemukan. Virus corona menyebar terutama melalui tetesan air liur atau keluarnya cairan dari hidung saat orang yang terinfeksi batuk atau bersin [1]. Virus *corona* pertama kali ditemui di Wuhan di Cina pada Desember tahun 2019 dan telah menyebar ke seluruh dunia [2]. Di Indonesia kasus pertama penyakit virus corona dilaporkan pada tanggal 2 Maret 2020 dan menyebar dengan sangat cepat. Sekarang Indonesia merupakan salah satu negara yang memiliki kasus virus corona yang tinggi di dunia. Dari data yang diperoleh dari *WHO*, persentase orang yang menderita virus corona di Indonesia berpersentase 0,2737% (735,124 kasus) dan persentase kematiannya 0,008% (21,944 kasus) dari total penduduk Indonesia (per tanggal 31 Desember 2020). Dari data yang dikumpulkan sementara angka kematian virus corona adalah di antara 3-4% [3].

Salah satu tindakan yang diambil oleh pemerintah Indonesia terhadap kondisi ini adalah mengeluarkan kebijakan untuk

mencegah penularan virus corona. Kebijakan tersebut adalah menetapkan peraturan tentang Pembatasan Sosial Berskala Besar (PSBB) melalui Peraturan Pemerintah (PP) Nomor 21 Tahun 2020 tentang PSBB dan mulai berlaku sejak 1 April 2020 [4]. Tindakan ini diharapkan untuk mencegah penularan penyakit virus corona dengan efektif karena diberikan sanksi bagi yang melanggar. Akan tetapi dengan adanya PSBB perekonomian Indonesia menurun dan menjadi minus sebesar -5.32 persen (*year-on-year*) [5]. Meskipun berbagai upaya sudah dilakukan oleh pemerintah, penularan virus corona tetap menjadi tantangan besar bagi Indonesia. Banyak faktor yang menyebabkan penyebaran virus ini sulit untuk dihindari, diantaranya adalah kurangnya kesadaran masyarakat untuk mematuhi protokol kesehatan dan vaksin yang belum ditemukan hingga saat ini. Kondisi ini semakin parah dengan kurangnya fasilitas kesehatan yang memadai sehingga penanganan terhadap pasien masih kurang maksimal. Kurangnya pengetahuan masyarakat terhadap virus corona juga menjadi salah satu alasan mengapa virus ini lebih cepat menyebar.

Berdasarkan kondisi tersebut, maka diperlukan upaya untuk mengantisipasi terjadinya hal yang lebih membahayakan. Beberapa penelitian telah membahas berbagai macam upaya untuk mengantisipasi hal tersebut. Diantaranya ada penelitiannya Pakan [6] yang meneliti tentang peramalan kasus positif virus corona di Indonesia dengan menggunakan metode *Long Short Term Memory* (LSTM) *network.* Penelitian lain oleh Fadly [7] menemukan jumlah korban virus corona di Jakarta dengan menggunakan metode *Autoregressive Integrated Moving Average* (ARIMA).

Setiap metode memiliki kelebihan dan kekurangan masing-masing, termasuk metode ARIMA dan LSTM. Salah satu kekurangan dari metode ARIMA adalah termasuk dalam model univariat di mana model ini hanya dapat melakukan peramalan dengan satu variabel. Sedangkan salah satu kekurangan dari

metode LSTM adalah tidak bisa meramalkan jika data yang tersedia berskala kecil. Saat ini masih belum ada penelitian yang membahas penerapan metode *hybrid* ARIMA-LSTM dalam peramalan kasus virus corona di Indonesia, padahal metode ini dapat memenuhi kekurangan dari kedua metode yang telah disebutkan. Hal ini dibuktikan pada penelitian Ketu [8] yang membandingkan metode ARIMA, LSTM dan *hybrid* ARIMA-LSTM untuk meramalkan penjualan rumah di Turki. Selain itu juga masih sedikit yang meramalkan kasus virus corona di Indonesia.

Pada penelitian tugas akhir ini akan menggunakan metode *hybrid* ARIMA-LSTM untuk meramalkan jumlah kasus virus corona di Indonesia. Metode ini digunakan karena dapat memenuhi kekurangan dari metode ARIMA dan LSTM. Pada metode *hybrid* ARIMA-LSTM, metode ARIMA akan digunakan untuk sebagai model linier dan metode LSTM akan digunakan model non-linier. Luaran dari penelitian ini adalah hasil peramalan jumlah kasus virus corona di Indonesia.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka rumusan masalah dari penelitian tugas akhir ini adalah sebagai berikut:

1. Bagaimana membangun model peramalan jumlah kasus virus corona menggunakan metode *hybrid* ARIMA-LSTM.
2. Bagaimana performa dari model peramalan jumlah kasus virus corona dengan menggunakan metode *hybrid* ARIMA-LSTM.

1.3 Batasan Masalah

Berdasarkan rumusan permasalahan yang telah dijelaskan di atas, maka batasan masalah dari penelitian ini adalah sebagai berikut:

1. Data yang digunakan untuk model peramalan adalah data historis harian jumlah kasus virus corona di Indonesia dari periode Maret 2020-Desember 2020. Data didapatkan dari situs *who.int.*
2. Pembuatan model peramalan metode *hybrid* ARIMA-LSTM mereferensikan pembuatan model pada penelitian Ketu [8].
3. *Tools* yang digunakan dalam penelitian tugas akhir ini adalah *Jupyter Notebook*.

1.4 Tujuan

Tujuan dari penelitian tugas akhir ini adalah sebagai berikut:

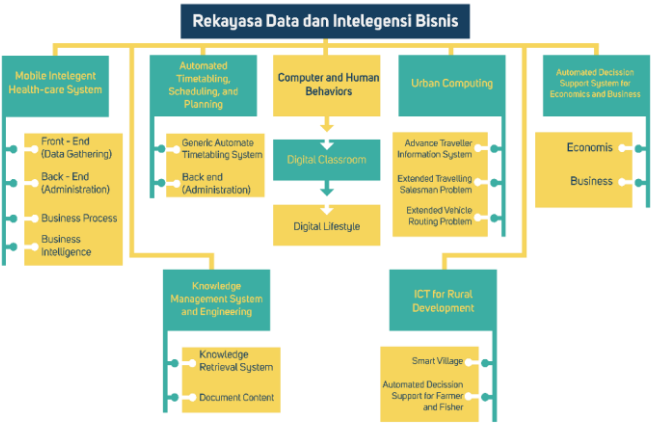
1. Membangun model peramalan jumlah kasus virus corona di Indonesia dengan menggunakan metode *hybrid* ARIMA-LSTM.
2. Mengetahui performa model peramalan jumlah kasus virus corona di Indonesia dengan menggunakan metode *hybrid* ARIMA-LSTM.

1.5 Manfaat

Manfaat yang diharapkan dari penelitian tugas akhir ini adalah dapat memberikan manfaat dalam memprediksi jumlah kasus virus corona di Indonesia untuk membantu pihak Dinas Kesehatan dalam menyiapkan keputusan yang terbaik untuk menanggulangi jumlah kasus virus corona di Indonesia.

1.6 Relevansi

Penelitian tugas akhir ini memiliki relevansi dengan salah satu *roadmap* penelitian Laboratorium Rekayasa Data dan Intelegensi Bisnis (RDIB) yaitu *Mobile Intelegent Health-care System*. Penelitian termasuk dalam kategori *Business Intelligence*. *Roadmap* laboratorium RDIB dapat dilihat pada Gambar 1.1.



Gambar 1.1 Roadmap Penelitian RDIB

Penelitian tugas akhir ini juga relevan karena melakukan implementasi mata kuliah yang dipelajari Departemen Sistem Informasi, adalah:

1. Teknik Peramalan
2. Penggalian Data

BAB II.
TINJAUAN PUSTAKA

Pada bagian ini, akan dijelaskan mengenai penelitian sebelumnya yang dijadikan referensi tugas akhir ini dan dasar teori yang dijadikan sebagai landasan dalam pengerjaan tugas akhir ini.

2.1 Penelitian Sebelumnya

Terdapat banyak penelitian sebelumnya yang berkaitan dengan penelitian tugas akhir ini. Beberapa penelitian yang dijadikan referensi dalam pengerjaan tugas akhir dapat dilihat pada Tabel 2.1.

Tabel 2.1 Rangkuman Penelitian Sebelumnya

Judul Paper	*A Hybrid Deep Learning Model for COVID-19 Prediction and Current Status of Clinical Trials Worldwide* [8]
Penulis; Tahun	Shwet Ketu, Pramod Kumar Mishra; 2020
Jurnal	*Tech Science Press*
Metode	*Autoregressive Integrated Moving Average* (ARIMA) *Long Short Term Memory* (LSTM) *Hybrid* ARIMA-LSTM
Deskripsi Umum Penelitian	Penelitian ini dilakukan untuk menghadapi wabah virus corona. Penelitian ini melakukan peramalan virus corona di seluruh dunia dengan menggunakan metode *Autoregressive Integrated Moving Average* (ARIMA), *Long Short Term Memory* (LSTM) dan *hybrid* ARIMA-LSTM. Hasil dari penelitian membuktikan bahwa metode *hybrid* ARIMA-LSTM menghasilkan hasil yang paling akurat dari *Mean Absolute Percentage Error* (MAPE), *Coefficient of determination* (R2) dan *Root Mean Square Error* (RMSE).
Kelebihan	Model yang digunakan pada penelitian ini ada tiga dan setiap model diperbandingkan *error*nya

	untuk mencari yang *error* paling rendah. Ada tiga pengukur *error* sehingga hasil bisa lebih akurat.
Keterkaitan Penelitian	Penelitian ini menjadi referensi utama dalam penelitian tugas akhir ini. Metode yang digunakan pada penelitiannya sama dengan metode yang digunakan pada penelitian tugas akhir ini. Sehingga dapat menjadi acuan saat membangun model metode.
Judul Paper	Time series analysis of COVID-19 infection curve: A change-point perspective [9]
Penulis; Tahun	Feiyu Jiang, Zifeng Zhao, Xiaofeng Shao; 2020
Jurnal	*Journal of Econometrics*
Metode	Self Normalization-based method
Deskripsi Umum Penelitian	Penelitian ini memodelkan lintasan dari kasus kumulatif positif virus corona dan kematian pasien virus corona dengan menggunakan model linier tren. Model tersebut akan menangkap fase transisi dari tingkat pertembuhan virus corona. Metodologi yang digunakan pada penelitian ini adalah Self Normalization-based method.
Kelebihan	Penelitian ini membuktikan bahwa kasus virus corona dapat diramalkan dengan menggunakan model *time series*.
Kekurangan	Tidak disebutkan pada paper.
Keterkaitan Penelitian	Penelitian ini menjadi referensi utama dalam penelitian tugas akhir ini. Penelitian ini dapat membantu membuktikan bahwa kasus virus corona dapat diramalkan dengan menggunakan model *time series*.
Judul Paper	PERAMALAN KASUS POSITIF COVID19 DI INDONESIA MENGGUNAKAN LSTM [6]
Penulis; Tahun	Prisca Deviani Pakan; 2020
Jurnal	Jurnal Ilmiah Flash

Metode	*Long Short Term Memory* (LSTM)
Deskripsi Umum Penelitian	Penelitian ini dilakukan untuk menghadapi wabah virus corona. Penelitian ini melakukan peramalan virus corona di Indonesia dengan menggunakan metode *Long Short Term Memory* (LSTM). Pengukur performa model dilakukan dengan *Mean Absolute Error* (MAE), *Mean Absolute Percentage Error* (MAPE) dan *Root Mean Square Error* (RMSE).
Kelebihan	Pembandingan *error* dilakukan dengan 3 pengukur sehingga hasil bisa lebih akurat.
Keterkaitan Penelitian	Penelitian ini menjadi referensi utama dalam penelitian tugas akhir ini. Metode yang digunakan mempunyai keterkaitan dengan metode yang digunakan pada tugas akhir ini dan data yang digunakan serupa dengan tugas akhir ini.
Judul Paper	*An Approach to Measure the Death Impact of Covid-19 in Jakarta using Autoregressive Integrated Moving Average* (ARIMA) [7]
Penulis; Tahun	Ferdian Fadly, Erika Sari; 2020
Jurnal	*Unnes Journal of Public Health*
Metode	*Autoregressive Integrated Moving Average* (ARIMA)
Deskripsi Umum Penelitian	Penelitian ini dilakukan untuk mengetahui kasus kematian virus corona di Indonesia. Pada penelitian ini metode yang digunakan adalah *Autoregressive Integrated Moving Average* (ARIMA).
Kelebihan	Pada penelitian ini diperkirakan jumlah kematian dari pasien virus corona yang tidak memiliki gejala apapun dan membandingkannya dengan jumlah kematian warga Indonesia pada periode sebelumnya.
Kekurangan	Tidak disebutkan pada paper.

Keterkaitan Penelitian	Penelitian ini menjadi referensi utama dalam penelitian tugas akhir ini. Metode yang digunakan mempunyai keterkaitan dengan metode yang digunakan pada tugas akhir ini dan data yang digunakan sama dengan tugas akhir ini.

2.2 Dasar Teori

Sub bab dari dasar teori merupakan teori-teori yang mendukung pengerjaan penelitian tugas akhir ini.

2.2.1 Virus Corona

Penyakit virus corona atau disebut juga sebagai COVID-19 adalah penyakit menular yang disebabkan oleh virus corona yang baru muncul pada Desember tahun 2019. Penyakit virus corona merupakan salah satu masalah kesehatan terbesar di dunia pada tahun 2020. Gejala penyakit virus corona dapat muncul dengan berbagai macam cara kepada setiap orang. Gejala umum dari penyakit virus corona adalah demam tinggi, batuk kering, kelelahan, nyeri otot, peradangan tenggorokan, diare, konjungtivitis, sakit kepala, hilangnya kemampuan mengecap rasa atau mencium bau, dan ruam di kulit. Gejala-gejala tersebut pada umumya muncul dalam waktu 5-6 hari setelah penderita terpapar virus corona namun bisa saja membutuh hingga 14 hari untuk gejalanya muncul. Virus corona bisa ditularkan antar manusia dengan mudah, melalui percikan ludah yang keluar saat penderita penyakit virus corona batuk atau bersin. Dengan bertambahnya pasien virus corona maka kontak antara pasien virus corona dengan orang lain akan bertambah sehingga meningkatkan persebaran virus corona. Menurut *World Health Organization* (WHO), persentase orang yang menderita virus corona di Indonesia berpersentase 0,2737% (735,124 kasus) dan persentase kematiannya 0,008% (21,944 kasus) dari total penduduk Indonesia (per tanggal 31 Desember 2020) [1].

2.2.2 Peramalan (*Forecasting*)

Menurut buku *Introduction to Time Series Analysis and Forecasting* [10], “peramalan adalah memprediksikan beberapa

peristiwa yang akan mendatang." Peramalan adalah teknik yang sangat penting dan bisa digunakan pada berbagai bidang studi termasuk bisnis dan industri, pemerintahan, ekonomi, ilmu lingkungan, kedokteran, ilmu sosial, politik dan keuangan. Pada umumya permasalahan peramalan diklasifikasikan menjadi *short-term*, *medium-term*, dan *long-term*. Peramalan *short-term* adalah memprediksikan peristiwa yang hanya masa periode yang dekat (harian, mingguan, bulanan). Peramalan *medium-term* adalah memprediksikan peristiwa yang berperiode satu atau dua tahun ke depannya. Peramalan *long-term* adalah memprediksikan peristiwa yang berperiode lebih dari dua tahun. Pada umumnya peramalan long-term digunakan untuk perencanaan strategis sedangkan *short-term* dan *medium-term* digunakan untuk manajemen operasi dan penganggaran atau untuk memulai proyek penelitian dan pengembangan yang baru. Pada penelitian tugas akhir ini akan dilakukan peramalan *short-term*.

2.2.3 Autoregressive Integrated Moving Average (ARIMA)

Autoregressive Integrated Moving Average (ARIMA) atau juga dikenal sebagai *Box Jenkins models* adalah metode peramalan yang banyak digunakan untuk melakukan peramalan *time series model*. Metode ARIMA pertama kali dikenalkan pada tahun 1976 oleh Box dan Jenkins. Metode ARIMA berkinerja sangat baik jika digunakan untuk meramalkan dengan *short-term* tetapi kurang akurat ketika digunakan untuk meramalkan dengan *long-term* [11]. Metode ARIMA memiliki tiga proses utama, yaitu adalah pemeriksaan diagnostik, identifikasi dan prediksi. Pada proses pertama yang disebut pemeriksaan diagnostik akan dilakukan pengendalian stasioneritas pada data *time series.* Stasioneritas sangat penting untuk memodelkan ARIMA karena dapat membuat prediksi menjadi sangat praktis dan bermanfaat. Untuk mengonversikan data *time series* yang non-stasioner menjadi stasioner, *differencing* (d) pada tingkat yang sesuai dilakukan. (d) diulangi terus sampai data *time series* menjadi stasioner [8]. Metode ARIMA terdiri atas 3 aspek yang

fundamental bermanfaat. Tiga aspek tersebut adalah sebagai berikut [8]:

- *Autoregressive* (AR) – Bagian yang bertanggung jawab atas penyimpanan dan pengambilan informasi masa lalu dari proses.
- *Integrated* (I) – Bagian yang bertanggung jawab atas pengonversian *time series* non-stasioner menjadi *time series* stasioner.
- *Moving Average* (MA) – Bagian yang bertanggung jawab atas pengaturan informasi masa lalu yang *noise-related* dari proses.

Rumus matematis dari model ARIMA dapat dirumuskan dengan menggunakan tiga aspek tersebut. Nilai dari AR ditentukan oleh parameter p dan nilai dari MA ditentukan oleh parameter q. Pada tahun 1983, Newbold mendefinisikan model ARIMA dengan menggunakan ARIMA (p,d,q), di mana p adalah *degree* dari AR, d adalah *differencing degree*, dan q adalah *degree* dari MA. Rumusan matematis dapat dilihat pada Persamaan 2.1.

$$y_t = \alpha_1 w_{t-1} + \alpha_2 w_{t-2} + \cdots + \alpha_p w_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}$$

Persamaan 2.1 Rumusan Matematis Model ARIMA

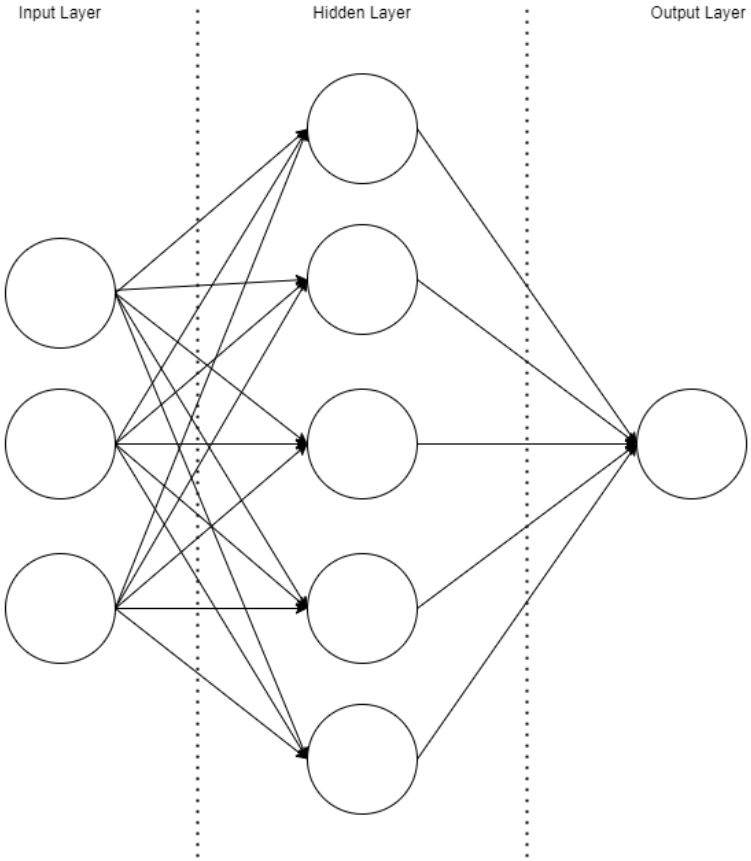
Di mana:

- $\alpha_1, \alpha_2, \ldots, \alpha_p$ = Parameter AR
- $\theta_1, \theta_2, \ldots, \theta_q$ = Parameter MA
- t = Waktu
- $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_t$ = *Error*
- $w_{t-1}, w_{t-2}, \ldots, w_{t-p}$ = Data yang diobservasi
- $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$ = *Error* data

2.2.4 Artificial Neural Network (ANN)

Artificial Neural Network (ANN) adalah model matematis yang terinspirasi oleh organisasi dan berfungsinya neuron biologis.

ANN diimplementasikan ke berbagai bidang keilmuan untuk melakukan peramalan atau memodelkan keputusan permasalahan. Pada dasarnya arsitektur ANN terdiri atas 3 layer yang disebut *Input Layer*, *Hidden Layer* dan *Output Layer*. Karena arsitekturnya terdiri atas beberapa layer sehingga dipanggil juga sebagai *Multi Layer Perceptron* (MLP). Contoh arsitektur ANN dapat dilihat pada Gambar 2.1. [12]

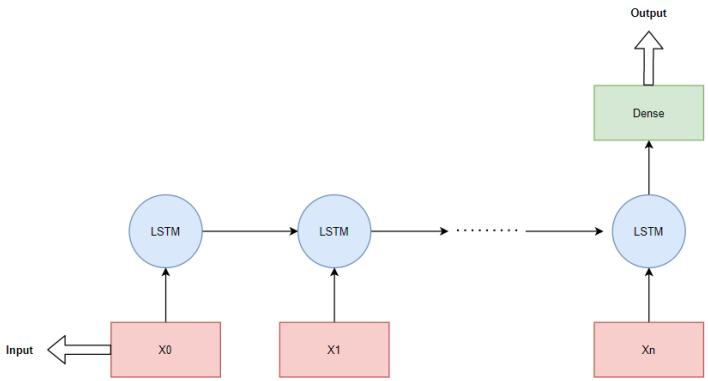


Gambar 2.1 Arsitektur ANN

Long Short Term Memory adalah salah satu jenis ANN yang digunakan untuk melakukan peramalan.

2.2.5 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) adalah salah satu tipe model peramalan dari *Recurrent Neural Network* (RNN) yang dibangun untuk mempelajari ketergantungan (*dependencies*) yang *long-term*. Bedanya LSTM dari RNN yang lain adalah LSTM memiliki *memory cell* yang dapat digunakan untuk

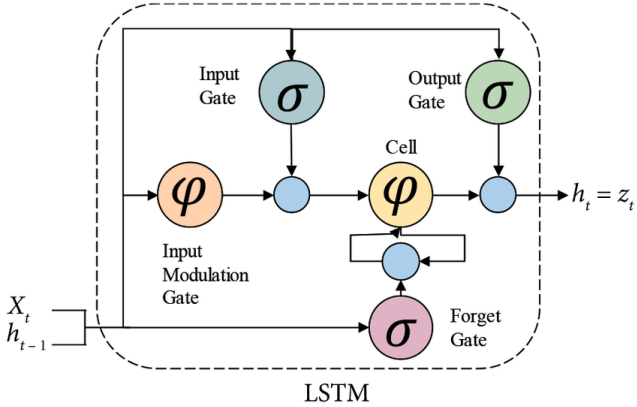


mempertahankan informasi untuk waktu yang lama. Model LSTM pertama kali dibangunkan oleh Hochreiter et al. pada tahun 1997. LSTM adalah struktur yang kompleks yang berada di *hidden layer*. Pada akhir-akhir ini LSTM adalah model *deep learning* yang sering digunakan pada berbagai bidang. Arsitektur LSTM yang sederhana dapat dilihat pada Gambar 2.2.

Gambar 2.2 Arsitektur LSTM Sederhana

Di dalam struktur LSTM ada sebuah sel memori yang berada dengan sel RNN. Sel memori ini bermanfaat untuk menyimpan informasi dan juga memanggil informasi dari masa lalu. Sel memori ini membantu memanggil informasi dari masa lalu dan mengirimkan informasi ke tingkat berikut. Model akan memilih informasi masa lalu berdasarkan *training*. Mengingatkan

informasi yang penting selama periode yang lama adalah karakteristik dari *network* LSTM [9].



Gambar 2.3 Struktur LSTM Sederhana *[8]*

Pada Gambar 2.3, X_t merupakan data input pada waktu t dan juga sebagai data output dari unit sebelumnya, h_t merupakan unit *hidden output*, dan h_{t-1} merupakan output dari sebelum. Struktur LSTM mempunyai berbagai *gate* seperti *input gate*, *output gate*, *input modulation gate* dan *forget gate*. Setiap *gate* memiliki rumusan matematis sendiri. Rumusan matematis dari setiap *gate* dapat dilihat pada Persamaan 2.2.

$$\text{Input Gate } (I_t^j) = \sigma(W_I x_t + W_I h_{t-1} + b_I)^J$$

$$\text{Forget Gate } (F_t^j) = \sigma(W_F x_t + W_F h_{t-1} + b_F)^J$$

$$\text{Output Gate } (O_t^j) = \sigma(W_O x_t + W_O h_{t-1} + b_O)^J$$

Persamaan 2.2 Rumusan Matematis Setiap Gate

Di mana:

- σ = Sigmoid Function
- b = Voltage Factor

- W = Weight matrices

Pada model LSTM, memory disimpan pada waktu t, dan memori yang diperbarui diperhitungkan dengan menggunakan fungsi (C_t^j). Rumus fungsi (C_t^j) dapat dilihat pada Persamaan 2.3.

$$C_t^j = F_t^j C_{t-1}^j + I_t^j C_t^j$$

Persamaan 2.3 Rumusan Fungsi (C_t^j)

Memori baru yang diperbarui akan dideterminasikan dengan menggunakan Persamaan 2.4.

$$C_t^j = \tanh(WCX_t + WCh_{t-1} + b_C)^j$$

Persamaan 2.4 Rumusan Determinasi Memori Baru

Output dari model LSTM di estimasikan dengan menggunakan Persamaan 2.5

$$h_t^j = O_t^j \tanh C_t^j$$

Persamaan 2.5 Rumus Fungsi h_t^j

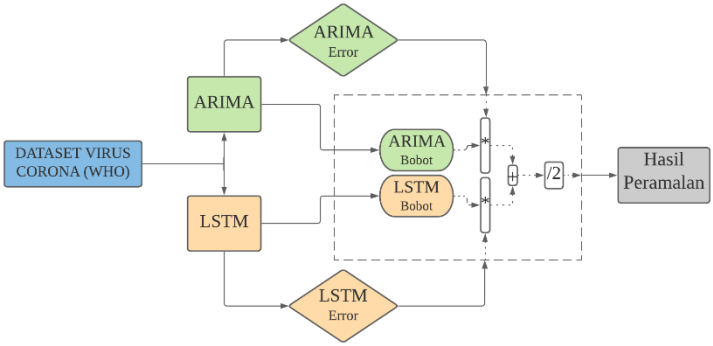
Training pada LSTM dilakukan melalui *epoch* seperti pada ANN yang lain. *Epoch* adalah jumlah iterasi yang dilakukan saat melakukan *training* pada sebuah dataset untuk menghitungkan bobot dari networknya (w). *Epoch* adalah bukti bahwa seluruh dataset sudah dikirimkan kepada layer berikutnya.

2.2.6 *Hybrid Autoregressive Integrated Moving Average – Long Short Term Memory* (Hybrid ARIMA-LSTM)

Berbagai model peramalan *time series* menggunakan hubungan linier dan non-linier. Model ARIMA sangat efisien ketika memodelkan hubungan linier pada data *time series*. Akan tetapi model ARIMA tidak dapat memodelkan ketika menggunakan

hubungan non-linier. Sedangkan model LSTM dapat memodelkan untuk hubungan linier dan non-linier tetapi tidak bisa menghasilkan hasil yang sama untuk setiap dataset. Untuk memenuhi kekurangan dari model ARIMA dan model LSTM maka kedua model akan digabungkan untuk menghasilkan hasil peramalan yang terbaik [9].

Sudah ada banyak penelitian sebelumnya yang menggunakan model *hybrid* untuk melakukan peramalan kasus lain. Pada penelitian tugas akhir ini akan menggunakan model *hybrid* ARIMA-LSTM untuk meramalkan jumlah kasus virus corona di Indonesia. Model *hybrid* ARIMA-LSTM yang digunakan ditunjukkan pada Gambar 2.4.



Gambar 2.4 Model Hybrid ARIMA-LSTM *[8]*

Pada penelitian tugas akhir ini komponen linier akan dihitungkan dengan menggunakan model ARIMA dan komponen nonlinier akan dihitungkan dengan menggunakan model LSTM. Komponen linier dari dataset virus corona adalah bagian trend dari datanya sedangkan komponen nonlinier dari dataset virus corona adalah bagian seasonality dan residual dari datanya. Setelah komponen linier dan non-linier sudah dihitungkan akan digabungkan dengan menggunakan Persamaan 2.6. [13]

$$Hybrid_{Ramalan} = TREND_{Ramalan}[i] + SEASONAL_{Ramalan}[i] + RESIDUAL_{Ramalan}[i]$$

Persamaan 2.6 Rumus Hybrid ARIMA-LSTM

2.2.7 Performa Peramalan

Untuk mengukur kinerja dari metode peramalan yang digunakan, maka diperlukan sebuah pengukur untuk mengukur tingkat akurasi metode peramalan. Pengukuran performa metode peramalan dalam penelitian ini akan diukur dengan menggunakan *Mean Absolute Percentage Error* (MAPE) dan *Root Mean Squared Error* (RMSE). Pengukur tersebut akan digunakan untuk membandingkan hasil peramalan dan aktualnya untuk mengetahui akurasi dari metode peramalan.

a. Mean Absolute Percentage Error (MAPE)

Mean Absolute Percentage Error (MAPE) adalah rata-rata dari sebuah persentase *error* yang mutlak dari sebuah peramalan. *Error* didefinisikan dengan nilai aktual diminus nilai ramalan. Persentase *error* dijumlahkan untuk mencari MAPE. Pengukuran MAPE mudah untuk dipahami karena menggunakan persentase. Nilai MAPE yang kecil membuktikan bahwa peramalannya bagus. Persamaan MAPE dapat dilihat pada Persamaan 2.7. [14]

$$\text{MAPE} = \left(\sum_{t=1}^{n} \left|\frac{y_t - \hat{y}_t}{yt}\right|\right) \frac{100}{n}$$

Persamaan 2.7 Rumus MAPE

Di mana:

- y_t = Nilai Aktual pada periode t
- $\hat{y}_t$ = Nilai prediksi pada periode t
- n = Jumlah Data

MAPE memiliki standar yang dapat digunakan untuk mengetahui kinerja dari model yang telah dibuat. Standar MAPE dapat dilihat pada Tabel 2.2.

Tabel 2.2 Standar Penilaian Akurasi MAPE

MAPE	Kekuatan Peramalan
<10%	Akurasi peramalan sangat baik
10%~20%	Akurasi peramalan baik
20%~50%	Akurasi peramalan masih dalam batas wajar
>50%	Akurasi peramalan tidak akurat

b. *Root Mean Square Error* (RMSE)

Root Mean Square Error (RMSE) adalah metrik model evaluasi yang sering digunakan dengan model regresi. Persamaan RMSE dapat dilihat pada Persamaan 2.8.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$$

Persamaan 2.8 Rumus RMSE

Di mana:

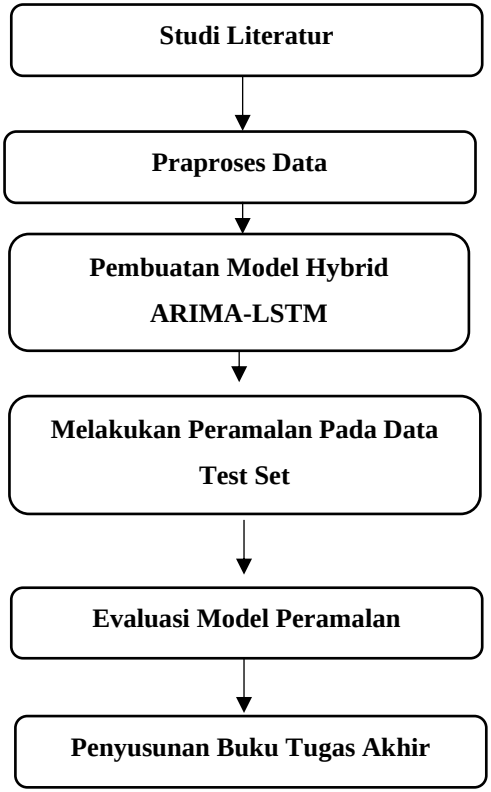
- y_t = Nilai aktual pada periode t
- $\hat{y}_t$ = Nilai prediksi pada periode t
- n = Jumlah data

BAB III.
METODOLOGI PENELITIAN

Pada bagian ini, akan dijelaskan mengenai tahapan-tahapan proses pengerjaan tugas akhir ini.

3.1 Tahapan Pengerjaan Tugas Akhir

Pada bagian ini akan dijelaskan mengenai metodologi dalam pengerjaan tugas akhir. Langkah-langkah pengerjaan tugas akhir ini ditampilkan pada Gambar 3.1.



Gambar 3.1 Tahapan Pengerjaan Tugas Akhir

3.2 Uraian Metodologi

Pada bagian ini akan dijelaskan secara detil mengenai tahapan-tahapan pengerjaan tugas akhir ini.

3.2.1 Studi Literatur

Tahap studi literatur merupakan tahapan dimana dilakukan pengumpulan referensi seperti penelitian sebelumnya atau buku pustaka yang dapat mendukung pengerjaan tugas akhir ini. Tahapan ini diawali dengan mencari informasi mengenai kasus virus corona, khususnya yang terjadi di Indonesia. Setelah informasi dikumpulkan dan dipahami, dilakukan pencarian data dan informasi pada buku atau penelitian sebelumnya mengenai penggunaan metode peramalan *hybrid Autoregressive Integrated Moving Average-Long Short Term Memory*. Buku atau penelitian sebelumnya yang didapatkan akan dijadikan sebagai landasan teori dalam pengerjaan tugas akhir ini.

3.2.2 Pra-Proses Data

Tahap pra-proses data merupakan tahapan di mana data akan dikumpulkan dan dipersiapkan sebelum memulai melakukan analisis. Data yang akan digunakan pada tugas akhir ini adalah data historis jumlah kasus virus corona di Indonesia. Data historis jumlah kasus virus corona di Indonesia akan diperoleh dari website resmi *World Health Organization* (https://covid19.who.int/). Data terdapat dalam format *Excel* (xls) dan berisikan dengan variabel tanggal, Negara, Jumlah kasus baru, Jumlah kasus kumulatif, Jumlah kematian baru, dan Jumlah kematian kumulatif. Data yang diambil adalah periode harian dari bulan Maret 2020 hingga Desember 2020. Untuk pembagian data *training* dan data *testing* akan dibagikan dengan 12 hari terakhir dari data akan dijadikan sebagai data *testing* dan data sisanya akan dijadikan data *training*.

3.2.3 Pembuatan Model Hybrid ARIMA-LSTM

Pada tahap pembuatan model *hybrid* ARIMA-LSTM akan dimodelkan model peramalan *hybrid* ARIMA-LSTM. Penyusunan model *Hybrid* ARIMA-LSTM memerlukan pemodelan model ARIMA dan LSTM. Pada tahap ini data

training set akan digunakan dalam pemodelan model *hybrid* ARIMA-LSTM. Hasil dari tahap ini akan merupakan model peramalan. Pertama model ARIMA akan dimodelkan untuk untuk menghitung komponen linier. Setelah komponen linier dihitungkan akan memodelkan model LSTM untuk menghitung komponen non-linier. Setelah perhitungan kedua model selesai akan dilakukan perhitungan *error*. Pada kasus virus corona, komponen linier adalah secara keseluruhan dan komponen non-linier adalah residual setelah perhitungan keseluruhannya.

3.2.4 Melakukan Peramalan pada Data Test Set

Pada tahap ini model peramalan yang dimodelkan pada tahap sebelumnya akan digunakan untuk meramalkan jumlah kasus virus corona di Indonesia. Hasil data peramalan akan dibandingkan dengan data aktual untuk mengetahui performa model peramalannya. Data yang digunakan pada tahap ini adalah data *test set*.

3.2.5 Evaluasi Model Peramalan

Pada tahap ini performa peramalan akan diukur dengan menggunakan pengukuran *Mean Absolute Percentage Error* (MAPE) dan *Root Mean Square Error* (RMSE). Nilai MAPE dan RMSE akan didokumentasikan. Selain itu, performa dari alogritma hybrid ARIMA-LSTM akan dibandingkan dengan performa dari alogritma ARIMA dan LSTM.

3.2.6 Penyusunan Buku Tugas Akhir

Tahap penyusunan buku tugas akhir merupakan tahapan yang dilakukan setelah semua langkah diselesaikan. Pada tahapan ini laporan tugas akhir akan disusun dengan mendokumentasikan setiap tahapan yang telah dilakukan. Laporan tugas akhir ini berisi semua langkah-langkah dalam pengerjaan tugas akhir secara detil dan kesimpulan.

BAB IV.
PERANCANGAN

Pada bagian ini, akan dijelaskan mengenai rancangan penelitian tugas akhir yang terdiri dari pengumpulan data dan pra proses data, pembentukan model peramalan, melakukan peramalan, dan evaluasi model peramalan yang telah dibuat.

4.1 Rancangan Pengumpulan Data

Data yang digunakan pada penelitian tugas akhir ini adalah data historis jumlah kasus virus corona di Indonesia. Data historis jumlah kasus virus corona di Indonesia akan diperoleh dari situs resmi *World Health Organization* (https://covid19.who.int/). Data akan digunakan dalam periode harian dan data akan dibagi menjadi data *training* dan *testing*. Untuk pembagian data *training* dan *testing* akan dibagikan dengan 12 hari terakhir dari 277 hari data akan dijadikan sebagai data *testing* dan data sisanya akan dijadikan data *training*.

4.2 Rancangan Pra-Proses Data

Pada tahap ini akan dilakukan pra-proses data, dimana data akan diproses sebelum digunakan. Proses ini terdiri atas penentuan variabel, memastikan tidak ada *missing value*, dan pembagian data. Grafik kasus virus corona di Indonesia dapat dilihat pada Lampiran.

4.2.1 Penentuan Variabel

Pada proses ini dilakukan penentuan variabel yang akan digunakan untuk pembangunan model peramalan. Pada penelitian tugas akhir ini variabel yang akan digunakan adalah kasus baru yang bertambah per hari.

4.2.2 Menghilangkan Missing Value

Untuk pembuatan model peramalan harus memastikan bahwa didalam dataset tidak ada *missing value* (nilai kosong/null). Proses ini dilakukan karena model peramalan yang telah dibangun dengan bahasa pemrograman *python* tidak bisa digunakan ketika ada nilai kosong/null. Untuk dataset yang

digunakan pada penelitian tugas akhir ini tidak ada nilai kosong/null tetapi dataset memiliki kesalahan saat input. Dataset ini memiliki kesalah input dimana data untuk dua periode dimasukkan dalam satu periode. Untuk menghilangkan kesalahan periode dimana ada salah input akan dibagi dua dan dimasukan kedalam periode sebelum atau sesudah yang kosong.

4.2.3 Pembagian Data

Pembagian data pada penelitian tugas akhir akan dibagikan menjadi dua. Data akan dibagi menjadi data *training* dan data *testing*. Data 12 hari terakhir akan dijadikan sebagai data *testing* dan 265 hari pertama akan dijadikan sebagai data *training*.

4.2.4 Normalisasi Data

Untuk melakukan pembentukan model LSTM dibutuhkan normalisasi data terlebih dahulu. Normalisasi data dilakukan dengan mengubah data menjadi rentang nilai 0 sampai 1. Untuk proses normalisasi data metode yang digunakan adalah *Min Max Scaler* dari library *scikit-learn.*

4.3 Pembentukan Model

Tahap ini adalah rancangan membentuk model peramalan yang akan digunakan untuk peramalan. Dalam pembentukan model peramalan hybrid ARIMA-LSTM, diperlukan lebih dahulu membentuk model ARIMA dan model LSTM.

4.3.1 Pembentukan Model ARIMA

Pemodelan model ARIMA akan dibangunkan dengan menggunakan library pmdarima. Pmdarima adalah sebuah library statistik yang dibuat untuk membantu menganalisis model time series pada Python.

Tahap pertama pada pemodelan model ARIMA adalah mengimport library-library yang akan membantu dalam pembangunan model ARIMA. Setelah library-library yang dibutuhkan diimport, dilanjutkan dengan mengimport data yang akan digunakan. Tipe data akan dikonversikan sesuai kebutuhan dan menentukan variable yang akan dijadikan index.

Tahap selanjutnya adalah melakukan uji stasioneritas rataan. Untuk melakukan peramalan dengan model ARIMA, dataset yang digunakan harus stasioner. Pengujian stasioneritas rataan dilakukan dengan library yang tersedia oleh Python yang dapat menunjukkan pvalue, usedlag, jumlah observasi dan critical values dari dataset. Data dapat dikatakan stasioner jika pvalue dibawah 0,05.

Setelah pengujian stasioneritas rataan dilakukan, dilanjutkan dengan differensiasi data jika dibutuhkan. Differensiasi data dibutuhkan apabila data yang digunakan belum stasioner. Pada tahap ini, apabila pvalue dibawah 0,05 maka data dapat dianggap telah stasioner. Differensiasi data harus dilakukan sampai pvalue dibawah 0,05. Jika setelah differensiasi data pertama data belum stasioner maka perlu dilakukan differensiasi hingga data menjadi stasioner.

Tahap selanjutnya adalah estimasi parameter yang dapat membantu menentukan model yang digunakan ARIMA. Tahap ini dapat dilakukan dengan melihat lag dari ACF dan PCF. Pada penelitian ini akan menggunakan library auto_arima yang dapat digunakan untuk menentukan model ARIMA yang terbaik dengan mencari model dengan AIC yang terkecil. Penentuan model ARIMA yang terbaik dapat dilihat dari AIC yang terkecil.

AIC adalah *Akaike's Information Criterion*. AIC adalah teknik yang berdasarkan in-sample fit untuk mengestimasikan model terbaik. Model dikatakan baik ketika nilai AIC semakin kecil [15].

Setelah model ARIMA dengan AIC terendah dipilih maka dilakukan pembagian data *training* dan data *testing*. Untuk pembagian data *training* dan *testing* akan dibagikan dengan 12 hari terakhir dari data akan dijadikan sebagai data *testing* dan data sisanya akan dijadikan data *training*. Setelah data di bagi dilakukan pengujian model ARIMA. Data *training* akan digunakan untuk menguji model.

Tahap selanjutnya adalah pengukuran tingkat akurasi dari model ARIMA yang telah dibangun dengan menggunakan MAPE dan RMSE. Hasil dengan MAPE dan RMSE yang lebih kecil merupakan model yang lebih akurat. Tahap ini dapat dilakukan dengan menggunakan salah satu library Python.

4.3.2 Pembentukan Model LSTM

Pembentukan model LSTM akan dilakukan dengan menggunakan library keras yang tersedia oleh Python.

Sebelum data *training* dan data *testing* digunakan dalam model peramalan terlebih dahulu dinornmalisasikan dengan menggunakan library Sklearn. Library sklearn memiliki fungsi MinMaxScaler yang dapat normalisasi data menjadi antara rentang nilai 0 sampai dengan 1 agar data yang diproses dapat digunakan dalam model LSTM.

Pembuatan base model LSTM akan menggunakan 10 *timestep* dan 1 fitur. *Parameter* yang digunakan dalam pembentukan model dapat dilihat pada Tabel 4.1.

Tabel 4.1 Parameter Yang Digunakan

Parameter	Nilai
Jumlah *Neuron*	50, 100
Epoch	500
Batch Size	32, 64, 128
Fungsi Aktivasi	ReLU, tanh
Optimizer	Adam, RMSprop, SGD

Penjelasan parameter yang digunakan dalam pembangunan model dapat dilihat pada Tabel 4.2.

Tabel 4.2 Penjelasan Parameter

Parameter	Nilai
Jumlah *Neuron*	Jumlah *Neuron* adalah jumlah *hidden node* yang berada pada *hidden layer*. Pada penelitian tugas akhir ini jumlah *neuron* yang digunakan adalah 50 dan 100
Epoch	*Epoch* adalah jumlah iterasi yang dilakukan untuk melakukan proses *training* model. Jumlah iterasi

	yang digunakan untuk penelitian tugas akhir ini adalah 500.
Batch size	*Batch size* adalah jumlah sampel yang digunakan dan diperbarui saat melakukan *training* model. *Batch size* yang digunakan untuk penelitian tugas akhir ini adalah 32, 64, dan 128.
Fungsi Aktivasi	Fungsi aktivasi menunjukkan bagaimana *parameter input* akan digabungkan dan ditransformasikan menjadi *output*. Fungsi aktivasi yang digunakan pada penelitian tugas akhir ini adalah ReLU dan tanh. ReLU (*Rectified Linear Unit*) adalah fungsi aktivasi yang digunakan untuk memisahkan eksitasi spesifik dan inhibisi tidak spesifik dalam *neural abstraction pyramid* [16]. Tanh (*Hyperbolic Tangent*) adalah fungsi aktivasi yang menerima nilai aktual sebagai *input* dan *output* sebagai rentang nilai dari -1 hingga 1. Nilai *input* yang lebih besar akan lebih dekat dengan 1 [16].
Optimizer	Optimizer adalah sebuah algoritma atau argumen yang digunakan untuk *compile* atau menyusun parameter yang sudah dipilih. Optimizer yang digunakan untuk penelitian tugas akhir ini adalah adam, RMSprop dan SGD. Adam (*Adaptive Moment Estimation*) adalah metode *stochastic gradient descent* yang didasarkan pada estimasi adaptif yang orde pertama dan orde kedua. Metode adam efisien dalam komputasi, tidak membutuhkan banyak memori, dan cocok untuk permasalahan dengan banyak data atau *parameter* [16]. RMSprop (*Root Mean Squared Propagation*) adalah metode optimasi berbasis gradien yang dikembangkan sebagai teknik skolastik. RMSprop digunakan untuk mempertahankan *moving average* dari kuadrat dari gradien. RMSprop digunakan juga untuk membagi gradien dengan akar dari rata-rata [16].

	SGD (*Stochastic Gradient Descent*) adalah metode iteratif untuk mengoptimasi fungsi obyektif dengan properti kelancaran [16].

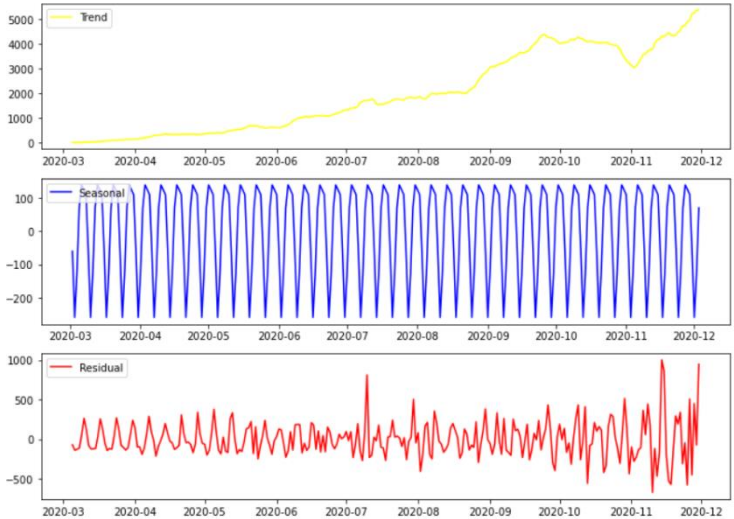
Setelah *parameter* model ditentukan, dilakukan pembuatan model peramalan dengan menggunakan *library keras* yang tersedia oleh *Python*. *Library keras* adalah sebuah *platform opensource* yang tersedia oleh *Python* untuk *machine learning*. *Keras* akan membantu membangun model LSTM.

Pembentukan model LSTM akan dilakukan pada data *training*. Jumlah nilai dari parameter model akan mempengaruhi nilai keluaran dari modelnya. Setelah model LSTM dibangun menggunakan data *training*, hasilnya akan didenormalisasikan kembali. Alasan data-data tersebut didenormalisasikan kembali adalah karena data masih berupa nilai rentang antara 0 dan 1 sehingga data prediksi dapat diperbandingkan dengan data aktual.

Tahap selanjutnya adalah pengukuran tingkat akurasi dari model LSTM yang telah dibangun dengan menggunakan MAPE dan RMSE. Hasil dengan MAPE dan RMSE yang lebih kecil merupakan model yang lebih akurat. Tahap ini dapat dilakukan dengan menggunakan salah satu library Python.

4.3.3 Pembentukan Model Hybrid ARIMA-LSTM

Pembentukan model Hybrid ARIMA-LSTM akan dilakukan dengan menggabungkan model ARIMA dan model LSTM. Pada pembentukan model Hybrid ARIMA-LSTM, library statsmodels akan digunakan untuk menggunakan fungsi *seasonal decomposition*. Dengan menggunakan fungsi *seasonal decomposition* data *time series* dapat dibagikan menjadi 3 komponen yaitu adalah *trend*, *seasonality* dan *residuals*. Gambar 4.1 menunjukkan data setelah melakukan dekomposisi. Garis kuning menunjukkan *trend* dari data, garis biru menunjukkan *seasonality* dari data dan garis merah menunjukkan *residual* dari data.



Gambar 4.1 Hasil Dekomposisi Data

Model ARIMA akan digunakan untuk menghitungkan komponen *trend* dan model LSTM akan digunakan untuk menghitungkan komponen *seasonality* dan *residuals*. Kedua model akan digabungkan dengan rumusan matematika yang dapat dilihat pada Persamaan 4.1.

$$Hybrid_{Ramalan} = TREND_{Ramalan}[i] + SEASONAL_{Ramalan}[i] + RESIDUAL_{Ramalan}[i]$$

Persamaan 4.1 Rumus Hybrid ARIMA-LSTM

Setelah mendapatkan model hybrid ARIMA-LSTM yang terbaik maka dilakukan analisis hasil peramalan untuk mengetahui keakuratan model peramalan dengan membandingkan data prediksi dan data aktual. Perhitungan nilai akurasi dari model hybrid ARIMA -LSTM akan dilakukan dengan menggunakan MAPE dan RMSE. Hasil dengan MAPE dan RMSE yang lebih kecil merupakan model yang lebih akurat. Tahap ini dapat dilakukan dengan menggunakan salah satu library *Python* [13].

BAB V.
IMPLEMENTASI

Pada bagian ini, akan dijelaskan mengenai proses implementasi rancangan yang telah dirancangkan sesuai dengan metode peramalan. Bagian ini akan dijelaskan mengenai lingkungan implementasi, pra-proses data, pemodelan model peramalan dan uji coba peramalan.

5.1 Lingkungan Implementasi

Penelitian tugas akhir ini menggunakan perangkat keras dan lunak dalam proses pengerjaan penelitiannya. Spesifikasi dari perangkat keras yang digunakan dalam penelitian tugas akhir dapat dilihat pada Tabel 5.1.

Tabel 5.1 Spesifikasi Perangkat Pengerjaan Tugas Akhir

Jenis Perangkat	Laptop
Processor Perangkat	Intel Core i5-7300HQ CPU @ 2.50GHz
Memory	12.0 GB
Sistem Operasi	*Windows 10 Home 64-bit operating system, x64-based processor*

Aplikasi yang digunakan dalam proses pembuatan model peramalan adalah Jupyter Notebook dan spesifikasi dari aplikasinya dapat dilihat pada Tabel 5.2.

Tabel 5.2 Daftar *Library*

Bahasa Pemrograman	Python
Antarmuka Bahasa Pemrograman	Jupyter Notebook
Database	Excel
Library	Numpy Pandas Matplotlib Pmdarima Statsmodels Sklearn

	Keras

5.2 Pengumpulan Data

Data kasus virus corona yang terjadi di Indonesia diunduh dari situs resmi *World Health Organization* (WHO) (https://covid19.who.int/). Data yang diunduh dari situs WHO disimpan dalam bentuk file excel.

5.3 Pra-Proses Data

Pada tahap ini akan dilakukan persiapan data sebelum dapat digunakan dalam pemodelan model peramalan. Terdapat beberapa langkah dalam proses ini seperti penentuan variabel, memastikan tidak ada *missing value*, dan pembagian data.

5.3.1 Penentuan Variabel

Pada data kasus virus corona yang diunduh dari situs resmi WHO mempunyai delapan variabel yaitu *Date_reported*, *Country_code*, *Country*, *WHO_region*, *New_cases*, *Cumulative_cases*, *New_deaths*, dan *Cumulative_deaths*. Pada penelitian tugas akhir ini, hanya akan menggunakan variabel *Date_reported* dan *New_cases* dalam pemodelan model peramalan. Data kasus virus corona yang terjadi di negara lain dihapus dari file excel karena hanya membutuhkan data kasus virus corona yang terjadi di Indonesia. Setelah data kasus virus corona yang terjadi di negara lain dihapus, kolom yang tidak digunakan dihapus juga untuk mempersiapkan sebelum melakukan pemodelan model peramalan.

5.3.2 Menghilangkan *Missing Value*

Pada data kasus virus corona yang diunduh dari situs resmi WHO, ada beberapa tanggal dimana data kasus virus corona kosong. Ini sebabkan karena data hari sebelum atau sesudah dijadikan satu seperti pada Tabel 5.3. Seperti yang dapat dilihat pada gambar tanggal 2020-11-18 memiliki angka yang sangat tinggi dibandingkan dengan tanggal yang lain. Tanggal 2020-11-18 merupakan jumlah kasus virus corona tanggal 2020-11-17 ditambah dengan 2020-11-8. Sehingga jumlah kasus virus corona yang kosong akan diisikan dengan membagi tanggal

sebelum atau sesudah yang terlalu tinggi dengan dua. Hal ini terjadi karena adanya kesalahan saat input sehingga satu periode mengandung jumlah kasus virus corona untuk dua periode.

Tabel 5.3 Pengisian Missing Value

Tanggal	Kasus
2020-11-09	3880
2020-11-10	2853
2020-11-11	3779
2020-11-12	3770
2020-11-13	4173
2020-11-14	5444
2020-11-15	5272
2020-11-16	4106
2020-11-17	0
2020-11-18	7342
2020-11-19	4265
2020-11-20	4798
2020-11-21	4792
2020-11-22	4998
2020-11-23	4360
2020-11-24	4442
2020-11-25	4192
2020-11-26	5534

5.3.3 Pembagian Data

Untuk pembagian data akan dibagikan dengan 12 hari terakhir sebagai data testing dan sisa data sebagai data *training*. Pembagian data data dilakukan dengan menggunakan

```
train = df.iloc[:len(df)-12]
test = df.iloc[len(df)-12:]
```

Kode Program *5.1*.

```
train = df.iloc[:len(df)-12]
test = df.iloc[len(df)-12:]
```

Kode Program 5.1 Pembagian Data

5.3.4 Normalisasi Data

Untuk melakukan peramalan menggunakan model LSTM dibutuhkan data yang sudah dinormalisasi. Normalisasi data yang digunakan pada penelitian tugas akhir ini dilakukan dengan menggunakan fungsi *Min Max Scaler* dari library *scikit-learn.* Fungsi ini akan diimport kedalam kodingan dengan menggunakan Kode Program 5.2.

```
from sklearn.preprocessing import MinMaxScaler
```

Kode Program 5.2 Mengimport Fungsi Normalisasi

5.4 Pembentukan Model

Setelah tahap pra-proses data selesai maka dilanjutkan dengan pembentukan model peramalan. Dalam melakukan pembentukan model hybrid ARIMA-LSTM menggunakan pemrograman python, model ARIMA dan model LSTM harus dibentukan terlebih dahulu.

a. Pembentukan Model ARIMA

Sebelum memulai pembentukan model ARIMA, library-library yang dibutuhkan harus diimport terlebih dahulu. Library yang akan digunakan pada pembentukan model ARIMA dan kode mengimportnya dapat dilihat pada Kode Program 5.3.

```
import numpy as np
import pandas as pd
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.seasonal import seasonal_decompose
from pmdarima import auto_arima
from sklearn.metrics import mean_squared_error,
mean_absolute_error
from math import sqrt
```

Kode Program 5.3 Library Yang Digunakan Untuk Model ARIMA

Setelah mengimport library-library yang akan digunakan, dilakukan pengimportan data dan tipe data yang diimport akan dikonversikan sesuai kebutuhan dan variabel index akan ditentukan. Setelah praproses data selesai, data akan diplotting untuk melihat bagaimana pola data kasus pasien baru di Indonesia. Kodingan untuk praproses data dapat dilihat pada Kode Program 5.4.

```
df = pd.read_csv("Indonesia COVID19 Data.csv",
index_col = 'Tanggal', parse_dates = True)
df.columns = ['Kasus']
df.plot(figsize = (12,8))
```

Kode Program 5.4 Praproses Data

Tahap selanjutnya adalah melakukan uji stasioneritas rataan. Untuk memodelkan model ARIMA dibutuhkan dataset yang stasioner. Pengujian dilakukan dengan menggunakan fungsi *Augmented Dickey-Fuller(ADF) test* yang tersedia pada library *Statsmodels* dari *Python*. Dengan melakukan ADF test, dapat diketahui pvalue, usedlags, jumlah observasi dan critical values dari dataset yang digunakan. Hasil dari ADF test dapat dilihat pada Gambar 5.1.

```
Augmented Dickey-Fuller Test:
ADF test statistic      0.673028
p-value                 0.989284
# lags used             8.000000
#observations         268.000000
Weak evidence against the null hypothesis
Fail to reject the null hypothesis
Data has a unit root and is non-stationary
```

Gambar 5.1 Augmented Dickey-Fuller Test

Dari hasil ADF test dapat diketahui bawah dataset tersebut belum stasioner karena p-value diatas 0,05 sehingga perlu melakukan differensiasi. Differensiasi data akan dilakukan sampai p-value dibawah 0,05. Jika setelah differensiasi data pertama data belum stasioner maka perlu dilakukan differensiasi hingga data menjadi stasioner.

Tahap selanjutnya adalah mengestimasikan parameter yang akan digunakan pada model ARIMA. Tahap ini dapat dilakukan dengan melihat lag dari grafik ACF dan PCF. Pada penelitian tugas akhir ini akan menggunakan library pmdarima untuk mencari parameter yang terbaik untuk dataset. Fungsi autoarima dari library pmdarima membantu menentukan model ARIMA yang terbaik untuk dataset dengan mencari AIC yang terkecil. Kodingan untuk mencari parameter dapat dilihat pada Kode Program 5.5.

```
auto_arima(df['Kasus'], seasonal = True, m =9, trace =
True).summary()
```

Kode Program 5.5 Penentuan Parameter ARIMA

Hasil dari fungsi auto-arima dapat dilihat pada Gambar 5.2. Sepertinya yang dapat dilihat, auto-arima akan mencari AIC dengan nilai yang terendah untuk dijadikan sebagai parameter model ARIMA.

```
ARIMA(2,1,1)(1,0,0)[7]              : AIC=3937.362, Time=0.21 sec
ARIMA(2,1,1)(2,0,1)[7]              : AIC=3905.852, Time=0.71 sec
ARIMA(2,1,1)(1,0,2)[7]              : AIC=3905.800, Time=0.64 sec
ARIMA(2,1,1)(0,0,0)[7]              : AIC=3974.774, Time=0.13 sec
ARIMA(2,1,1)(0,0,2)[7]              : AIC=3945.765, Time=0.37 sec
ARIMA(2,1,1)(2,0,0)[7]              : AIC=3929.058, Time=0.36 sec
ARIMA(2,1,1)(2,0,2)[7]              : AIC=inf, Time=0.83 sec
ARIMA(1,1,1)(1,0,1)[7]              : AIC=3903.567, Time=0.35 sec
ARIMA(1,1,1)(0,0,1)[7]              : AIC=3947.970, Time=0.19 sec
ARIMA(1,1,1)(1,0,0)[7]              : AIC=3936.948, Time=0.16 sec
ARIMA(1,1,1)(2,0,1)[7]              : AIC=3905.439, Time=0.63 sec
ARIMA(1,1,1)(1,0,2)[7]              : AIC=3905.416, Time=0.58 sec
ARIMA(1,1,1)(0,0,0)[7]              : AIC=3974.867, Time=0.11 sec
ARIMA(1,1,1)(0,0,2)[7]              : AIC=3944.078, Time=0.32 sec
ARIMA(1,1,1)(2,0,0)[7]              : AIC=3928.983, Time=0.29 sec
ARIMA(1,1,1)(2,0,2)[7]              : AIC=inf, Time=0.91 sec
ARIMA(0,1,1)(1,0,1)[7]              : AIC=3902.505, Time=0.27 sec
ARIMA(0,1,1)(0,0,1)[7]              : AIC=3959.211, Time=0.11 sec
ARIMA(0,1,1)(1,0,0)[7]              : AIC=3942.738, Time=0.09 sec
ARIMA(0,1,1)(2,0,1)[7]              : AIC=3904.429, Time=0.46 sec
ARIMA(0,1,1)(1,0,2)[7]              : AIC=3904.413, Time=0.61 sec
ARIMA(0,1,1)(0,0,0)[7]              : AIC=3994.962, Time=0.05 sec
ARIMA(0,1,1)(0,0,2)[7]              : AIC=3953.340, Time=0.19 sec
ARIMA(0,1,1)(2,0,0)[7]              : AIC=3931.346, Time=0.17 sec
ARIMA(0,1,1)(2,0,2)[7]              : AIC=inf, Time=0.88 sec
ARIMA(0,1,0)(1,0,1)[7]              : AIC=3962.147, Time=0.14 sec
ARIMA(0,1,2)(1,0,1)[7]              : AIC=3903.848, Time=0.32 sec
ARIMA(1,1,0)(1,0,1)[7]              : AIC=3916.633, Time=0.29 sec
ARIMA(1,1,2)(1,0,1)[7]              : AIC=3904.283, Time=0.48 sec
ARIMA(0,1,1)(1,0,1)[7] intercept    : AIC=3904.154, Time=0.51 sec

Best model:  ARIMA(0,1,1)(1,0,1)[7]
Total fit time: 39.508 seconds
```

Gambar 5.2 Hasil Auto-ARIMA

Setelah model ARIMA dengan AIC yang terendah dipilih maka dilanjutkan dengan pembagian data *training* dan data *testing*. Data *training* yang sudah dibagi akan digunakan dalam model ARIMA dengan AIC yang terendah untuk melakukan peramalan. Peramalan tersebut akan dibandingkan dengan data testing untuk mengetahui keakuratan dari model yang telah dibuat. Kodingan dari perbandingan hasil peramalan dan data aktual dapat dilihat pada Kode Program 5.6.Kode **Program *5.6***

```
model = SARIMAX(train['Kasus'], order = (0, 1, 1),
seasonal_order = (1, 0, 1, 7))
results = model.fit()
start = len(train)
end = len(train) + len(test) – 1
predictions = results.predict(start = start, end = end, typ =
'levels').rename('SARIMA Predictions')
test['Kasus'].plot(figsize = (12,8), legend = True)
predictions.plot(legend = True)
```

Kode Program 5.6 Pembentukkan Model ARIMA

Tahap selanjutnya adalah pengukuran tingkat akurasi dari ARIMA yang telah dibangunkan dengan menggunakan MAPE dan RMSE. Pengukuran akurasi dapat dilakukan dengan menggunakan Kode Program 5.7.

```
def eval_metrics(y_true, y_pred):
   MAPE = np.mean(np.abs((y_true - y_pred) / y_true)) *
100
   MSE = mean_squared_error(y_true, y_pred)
   RMSE = sqrt(MSE)

   print("MAPE: ", MAPE)
   print("RMSE: ", RMSE)
   print(RMSE/test['Kasus'].mean())
eval_metrics(test['Kasus'], test['predictions'])
```

Kode Program 5.7 Pengukuran Akurasi Model

5.4.2 Pembentukan Model LSTM

Sebelum memulai pembentukan model LSTM, library-library yang dibutuhkan harus diimport terlebih dahulu. Library yang akan digunakan pada pembentukan model LSTM dan kode mengimportnya dapat dilihat pada Kode Program 5.8.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from keras.preprocessing.sequence import
TimeseriesGenerator
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from math import sqrt
from keras.models import load_model
```

Kode Program 5.8 Library Yang Digunakan Untuk Model LSTM

Setelah mengimport library-library yang akan digunakan, dilakukan pengimportan data dan tipe data yang diimport akan dikonversikan sesuai kebutuhan, variabel index akan ditentukan dan data akan dibagi menjadi data *training* dan data *testing*. Setelah praproses data selesai, data akan dinormalisasi untuk dapat digunakan pada model LSTM. Kodingan untuk praproses data dan normalisasi dapat dilihat pada Kode Program 5.9.

```
df = pd.read_csv("Indonesia COVID19 Data.csv",
index_col = 'Tanggal', parse_dates = True)
df.columns = ['Kasus']
train = df.iloc[:len(df)-12]
test = df.iloc[len(df)-12:]
scaler = MinMaxScaler()
scaler.fit(train)
scaled_train = scaler.transform(train)
scaled_test = scaler.transform(test)
```

Kode Program 5.9 Praproses Data Untuk Model LSTM

Setelah pra-proses data selesai akan dilanjutkan dengan pembangunan model LSTM. Tahap pertama pada pembangunan model LSTM adalah melatih model dengan menggunakan data *training*. Parameter pada pelatihan model

dapat dilihat pada Tabel 4.1. Dengan menggunakan parameter tersebut pelatihan model akan dilakukan dengan menggunakan Kode Program 5.10.

```
n_input = 10
n_features = 1
train_generator = TimeseriesGenerator(scaled_train,
scaled_train, length = n_input, batch_size = 32)
model = Sequential()
model.add(LSTM(50, activation = 'ReLU', input_shape =
(n_input, n_features)))
model.add(Dense(1))
model.compile(optimizer = 'adam', loss = 'mse')
```

Kode Program 5.10 Pelatihan Model LSTM

Setelah pelatihan model selesai akan dilanjutkan dengan peramalan model LSTM dengan menggunakan data test. Peramalan model LSTM dengan menggunakan data test dapat dilakukan dengan menggunakan Kode Program 5.11.

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

Kode Program 5.11 Peramalan Model LSTM

Setelah peramalan model LSTM selesai, data yang digunakan akan dinormalisasikan kembali untuk dapat melihat hasil dengan angka sebenarnya. Hasil tersebut akan diplotting dan diperbandingkan dengan data aktual untuk melihat perbandingannya. Tahap ini dapat dilakukan dengan menggunakan Kode Program 5.12.

```
true_predictions =
scaler.inverse_transform(test_predictions)
test['predictions'] = true_predictions
test.plot()
```

Kode Program 5.12 Normalisasi Data Dan Plotting

Tahap selanjutnya adalah pengukuran tingkat akurasi dari model LSTM yang telah dibangunkan dengan menggunakan MAPE dan RMSE. Pengukuran akurasi dapat dilakukan dengan menggunakan Kode Program 5.7.

5.4.3 Pembentukan Model Hybrid ARIMA-LSTM

Sebelum memulai pembentukan model hybrid ARIMA-LSTM, library-library yang dibutuhkan harus diimport terlebih dahulu. Library yang akan digunakan pada pembentukan model hybrid ARIMA-LSTM dan kode mengimportnya dapat dilihat pada Kode Program 5.13.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pmdarima import auto_arima
from statsmodels.tsa.statespace.sarimax import
SARIMAX
from statsmodels.tsa.seasonal import seasonal_decompose
from sklearn.preprocessing import MinMaxScaler
from keras.preprocessing.sequence import
TimeseriesGenerator
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from sklearn.metrics import mean_squared_error,
mean_absolute_error
from math import sqrt
from keras.models import load_model
```

Kode Program 5.13 Library Yang Digunakan Untuk Model Hybrid ARIMA-LSTM

Setelah mengimport library-library yang akan digunakan, dilakukan pengimportan data dan tipe data yang diimport akan dikonversikan sesuai kebutuhan dan variabel index akan ditentukan. Setelah praproses data selesai, data akan diplotting untuk melihat bagaimana pola data kasus pasien baru di Indonesia. Kodingan untuk praproses data dapat dilihat pada Kode Program 5.4.

Pada model hybrid ARIMA-LSTM data yang digunakan akan didekomposisi menjadi 3 bagian yaitu, seasonal, trend dan residuals. Tahap ini dilakukan karena komponen linier (Trend) akan dihitungkan menggunakan model ARIMA dan komponen nonlinier (Seasonal dan residuals) akan dihitungkan menggunakan model LSTM. Dekomposisi dapat dilakukan dengan menggunakan Kode Program 5.14.

```
decompose = seasonal_decompose(df['Kasus'])
df['Seasonal'] = decompose.seasonal
df['Trend'] = decompose.trend
df['Residuals'] = decompose.resid
```

Kode Program 5.14 Dekomposisi

Setelah dekomposisi data selesai, komponen trend akan dihitungkan terlebih dahulu. Parameter dari model ARIMA untuk komponen trend akan ditentukan dengan menggunakan fungsi pmdarima. Fungsi autoarima dari library pmdarima membantu menentukan model ARIMA yang terbaik untuk dataset dengan mencari AIC yang terkecil. Kodingan untuk mencari parameter dapat dilihat pada Kode Program 5.15.

```
df_Trend = df['Trend']
auto_arima(df_Trend, seasonal = False, trace =
True).summary()
```

Kode Program 5.15 Auto_arima Pada Komponen Trend

Hasil dari fungsi auto-arima dapat dilihat pada Gambar 5.3. Sepertinya yang dapat dilihat, auto-arima akan mencari AIC dengan nilai yang terendah untuk dijadikan sebagai parameter model ARIMA. Sehingga parameter yang akan digunakan adalah (2,1,5).

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=2865.411, Time=0.30 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=2956.421, Time=0.01 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=2877.749, Time=0.05 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=2908.087, Time=0.10 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=2985.235, Time=0.01 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=2866.674, Time=0.16 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=2866.643, Time=0.17 sec
 ARIMA(3,1,2)(0,0,0)[0] intercept   : AIC=2867.363, Time=0.36 sec
 ARIMA(2,1,3)(0,0,0)[0] intercept   : AIC=2855.593, Time=0.45 sec
 ARIMA(1,1,3)(0,0,0)[0] intercept   : AIC=2868.665, Time=0.26 sec
 ARIMA(3,1,3)(0,0,0)[0] intercept   : AIC=inf, Time=0.54 sec
 ARIMA(2,1,4)(0,0,0)[0] intercept   : AIC=inf, Time=0.60 sec
 ARIMA(1,1,4)(0,0,0)[0] intercept   : AIC=2863.297, Time=0.46 sec
 ARIMA(3,1,4)(0,0,0)[0] intercept   : AIC=2840.190, Time=0.63 sec
 ARIMA(4,1,4)(0,0,0)[0] intercept   : AIC=inf, Time=0.63 sec
 ARIMA(3,1,5)(0,0,0)[0] intercept   : AIC=2821.932, Time=0.74 sec
 ARIMA(2,1,5)(0,0,0)[0] intercept   : AIC=2821.885, Time=0.68 sec
 ARIMA(1,1,5)(0,0,0)[0] intercept   : AIC=2848.165, Time=0.54 sec
 ARIMA(2,1,5)(0,0,0)[0]             : AIC=2825.012, Time=0.62 sec

Best model:  ARIMA(2,1,5)(0,0,0)[0] intercept
Total fit time: 7.347 seconds
```

Gambar 5.3 Parameter ARIMA Untuk Komponen Trend

Setelah parameter ARIMA diketahui, maka dilanjutkan dengan pembagian data untuk komponen trend. Pembagian data dapat dilakukan dengan menggunakan Kode Program 5.16.

```
train_trend = df_Trend.iloc[:len(df_Trend)-12]
test_trend = df_Trend.iloc[len(df_Trend)-12:]
```

Kode Program 5.16 Pembagian Komponen Trend

Setelah data dibagi menjadi data testing dan data *training*, model ARIMA akan dibangunkan dengan parameter yang sudah diketahui. Model ARIMA akan dilatih dulu dengan menggunakan data *training*. Tahap pelatihan model ARIMA dapat dilihat pada Kode Program 5.17.

```
ARIMA_model = SARIMAX(train_trend, order = (2, 1,
5))
ARIMA_results = ARIMA_model.fit()
```

Kode Program 5.17 Pelatihan Model ARIMA Untuk Komponen Trend

Setelah pelatihan model selesai, akan dilanjutkan dengan peramalan komponen trend dengan menggunakan model ARIMA yang telah dilatih. Peramalan komponen trend dan perbandingan dengan trend aktual dapat dilakukan dengan menggunakan Kode Program 5.18.

```
start = len(train_trend)
end = len(train_trend) + len(test_trend) – 1
ARIMA_predictions = ARIMA_results.predict(start =
start, end = end, typ='levels').rename('ARIMA Predictions')
plt1Trend, = plt.plot(test_trend);
plt2Trend, = plt.plot(ARIMA_predictions)
plt.legend([plt1Trend, plt2Trend], ['True trend',
'Predictions'])
```

Kode Program 5.18 Peramalan Model ARIMA Untuk Komponen Trend

Tahap selanjutnya adalah pengukuran tingkat akurasi dari model ARIMA untuk komponen trend yang telah dibangunkan dengan menggunakan MAPE dan RMSE. Pengukuran akurasi dapat dilakukan dengan menggunakan Kode Program 5.7.

Setelah penghitungan komponen linier selesai, maka dilanjutkan dengan penghitungan komponen nonlinier dengan menggunakan model LSTM. Komponen seasonal akan dihitungkan terlebih dahulu. Tahap pertama adalah pembagian data pada komponen seasonal. Tahap ini dapat dilakukan dengan menggunakan Kode Program 5.19.

```
ARIMA_model = SARIMAX(train_trend, order = (2, 1,
5))
ARIMA_results = ARIMA_model.fit()
```

Kode Program 5.19 Pembagian Komponen Seasonal

Setelah data dibagi akan dilanjutkan dengan normalisasi data dan penentuan parameter. Tahap ini dapat dilakukan dengan menggunakan Kode Program 5.20.

```
LSTMscaler = MinMaxScaler()
LSTMscaler.fit(train_LSTM)
LSTMscaled_train = LSTMscaler.transform(train_LSTM)
LSTMscaled_test = LSTMscaler.transform(test_LSTM)
LSTMn_input = 10
LSTMn_features = 1
```

Kode Program 5.20 Normalisasi Data Dan Penentuan Parameter

Setelah parameter ditentukan, akan dilakukan pelatihan model LSTM dengan menggunakan data *training* dari komponen seasonal. Tahap ini dapat dilakukan dengan menggunakan Kode Program 5.21.

```
LSTMgenerator = TimeseriesGenerator(LSTMscaled_train,
LSTMscaled_train, length = LSTMn_input, batch_size =
32)
LSTMmodel = Sequential()
LSTMmodel.add(LSTM(50, activation = 'ReLU',
input_shape = (LSTMn_input, LSTMn_features)))
LSTMmodel.add(Dense(1))
LSTMmodel.compile(optimizer = 'adam', loss = 'mse')
```

Kode Program 5.21 Pelatihan Model LSTM Untuk Komponen Nonlinier

Setelah model sudah dilatih akan dilanjutkan dengan peramalan komponen seasonal dengan menggunakan data testing. Tahap ini dapat dilakukan dengan menggunakan Kode Program 5.22.

```
LSTMtest_predictions = []
LSTMfirst_eval_batch = LSTMscaled_train[-
LSTMn_input:]
LSTMcurrent_batch = LSTMfirst_eval_batch.reshape((1,
LSTMn_input, LSTMn_features))
for i in range(len(test_LSTM)):
    LSTMcurrent_pred =
LSTMmodel.predict(LSTMcurrent_batch)[0]
    LSTMtest_predictions.append(LSTMcurrent_pred)
    LSTMcurrent_batch = np.append(LSTMcurrent_batch[:,
1:, :], [[LSTMcurrent_pred]], axis = 1)
```

Kode Program 5.22 Peramalan Model LSTM Untuk Komponen Nonlinier

Setelah berhasil melakukan peramalan dengan komponen seasonal, data yang digunakan akan dinormalisasikan kembali untuk diperbandingkan dengan data aktual. Tahap ini dapat dilakukan dengan menggunakan Kode Program 5.23.

```
LSTMtrue_predictions =
LSTMscaler.inverse_transform(LSTMtest_predictions)
test_LSTM['Predictions'] = LSTMtrue_predictions
test_LSTM.plot()
```

Kode Program 5.23 Normalisasi Data Dan Plotting Perbandingan Untuk Komponen Nonlinier

Tahap selanjutnya adalah pengukuran tingkat akurasi dari model LSTM untuk komponen seasonal yang telah dibangunkan dengan menggunakan MAPE dan RMSE. Pengukuran akurasi dapat dilakukan dengan menggunakan Kode Program 5.7.

Setelah semua komponen diperhitungkan, semua hasil peramalan akan digabungkan dengan menggunakan Persamaan 4.1. Pergabungan dapat dilakukan dengan menggunakan Kode Program 5.24.

```
hybrid_test = test_LSTM['Predictions'] +
ARIMA_predictions + test_residual['Predictions']
```

Kode Program 5.24 Pemodelan Hybrid ARIMA-LSTM

Pada Kode Program 5.24 hasil perhitungan komponen *trend* dengan model ARIMA dan hasil perhitungan komponen *seasonal* dan *residual* akan dijadikan satu dengan menggunakan Persamaan 4.1.

BAB VI.
HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil uji coba dan analisis terhadap hasil yang diperoleh dalam melakukan pelatihan model hybrid ARIMA-LSTM.

6.1 Hasil Pengumpulan Data

Pengumpulan data dilakukan dengan mengunduh data dari situs *World Health Organization* (WHO) (https://covid19.who.int/). Untuk mempermudah analisis, dilakukan visualisasi terhadap masing-masing data.

6.1.1 Hasil Pengumpulan Data

Tabel 6.1 merupakan sebagian data yang diperoleh dari WHO. Untuk mengetahui performa model terhadap jumlah kasus baru yang terjadi, penelitian tugas akhir ini menggunakan variabel *New_cases*. Untuk keseluruhan data dapat dilihat pada LAMPIRAN A.

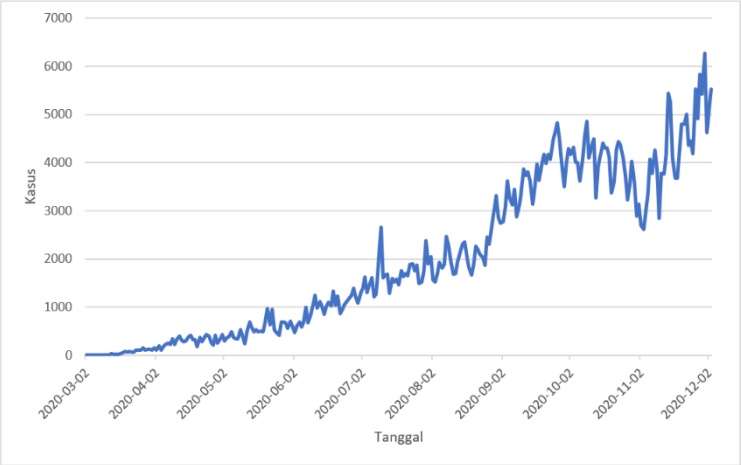
Tabel 6.1 Data Yang Diperoleh Dari WHO

Date_reported	Country_c	Country	WHO_region	New_case	Cumulative_cases	New_deaths	Cumulative_deaths
2020-09-07	ID	Indonesia	SEARO	3444	194109	85	8025
2020-09-08	ID	Indonesia	SEARO	2880	196989	105	8130
2020-09-09	ID	Indonesia	SEARO	3046	200035	100	8230
2020-09-10	ID	Indonesia	SEARO	3307	203342	106	8336
2020-09-11	ID	Indonesia	SEARO	3861	207203	120	8456
2020-09-12	ID	Indonesia	SEARO	3737	210940	88	8544
2020-09-13	ID	Indonesia	SEARO	3806	214746	106	8650
2020-09-14	ID	Indonesia	SEARO	3636	218382	73	8723

Dari data yang diperoleh variabel Date_reported menunjukkan tanggal terjadinya virus corona, variabel Country_code menunjukkan kode negara (Kode Negara Indonesia adalah ID), variabel Country yang menunjukkan nama negara, variabel WHO_region yang menunjukkan daerah dari negara (WHO_region Indonesia adalah SEARO(South East Asia Region), variabel New_cases yang menunjukkan kasus virus corona yang baru ditambah, variabel Cumulative_cases yang menunjukkan kasus kumulatif, variabel New_deaths yang menunjukkan kematian baru yang terjadi, dan terakhir variabel Cumulative_deaths yang menunjukkan kematian kumulatif.

6.1.2 Visualisasi Data

Untuk mempermudah dalam menganalisis karakteristik data, dilakukan visualisasi data dalam bentuk grafik. Visualisasi dari data yang diperoleh dapat dilihat pada Gambar 6.1. Gambar 6.1 merupakan *line chart* yang menggunakan jumlah kasus baru per tanggal. Data kasus virus corona di Indonesia memiliki karakteristik mengalami peningkatan selama sepuluh bulan.



Gambar 6.1 Visualisasi Data Virus Corona Yang Terjadi Di Indonesia

6.2 Hasil Pra-Proses Data

Hasil implementasi pra-proses data menghasilkan penentuan variabel, menghilangkan *missing value*, pembagian data dan normalisasi data.

6.2.1 Hasil Penentuan Variabel

Hasil dari penentuan variabel adalah *file excel* dengan hanya variabel-variabel yang akan digunakan dalam pembentukan model peramalan. Berdasarkan **Error! Reference source not found.** data virus corona memiliki 8 variabel. Namun tidak semua 8 variabel tersebut akan digunakan dalam pembentukan model. Sehingga variabel-variabel yang tidak digunakan akan dihilangkan dari file excel dan juga negara-negara yang bukan Indonesia akan dihilangkan.

Variabel yang akan digunakan dalam penelitian tugas akhir ini adalah variabel "*Date_reported*" dan "*New_cases*" dimana berisi informasi tanggal dan kasus virus corona yang bertambah. Hasil data setelah penentuan variabel dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Penentuan Variabel

Tanggal	Kasus
2020-09-07	3444
2020-09-08	2880
2020-09-09	3046
2020-09-10	3307
2020-09-11	3861
2020-09-12	3737
2020-09-13	3806
2020-09-14	3636

6.2.2 Hasil Menghilangkan Missing Value

Pada data yang dioleh dari WHO data beberapa tanggal dimana jumlah kasus pada hari tersebut tidak dicatatkan seperti pada Tabel 6.3. Ini terjadi karena ada kesalahan saat input oleh WHO.

Tabel 6.3 Missing Value

Tanggal	Kasus
2020-11-09	3880
2020-11-10	2853
2020-11-11	3779
2020-11-12	3770
2020-11-13	4173
2020-11-14	5444
2020-11-15	5272
2020-11-16	4106
2020-11-17	0
2020-11-18	7342
2020-11-19	4265
2020-11-20	4798
2020-11-21	4792
2020-11-22	4998
2020-11-23	4360
2020-11-24	4442
2020-11-25	4192
2020-11-26	5534

Seperti yang dapat dilihat pada Tabel 6.3, data pada tanggal sesudah missing value lebih besar jika dibandingkan dengan hari sebelum atau sesudahnya. Hal ini terjadi karena data hari sebelumnya dijumlahkan sehingga data hari sebelumnya 0. Untuk pembentukan model yang lebih akurat, hari dengan data yang tinggi dibandingkan hari yang lain akan dibagi menjadi dua dan dimasukkan pada hari sebelumnya. Hasil dari menghilangkan missing value dapat dilihat pada Tabel 6.4.

Tabel 6.4 Hasil Menghilangkan Missing Value

Tanggal	Kasus
2020-11-16	4106
2020-11-17	3671
2020-11-18	3671
2020-11-19	4265

Menghilangkan missing value akan dilakukan untuk semua hari dengan nilai 0.

6.2.3 Hasil Pembagian Data

Langkah selanjutnya adalah membagi data menjadi data *training* dan data *testing*. Data dibagi menjadi 12 hari terakhir sebagai data *testing* dan sisa data untuk data *training*. Data *training* akan digunakan untuk pembangunan model sedangkan data *testing* akan digunakan untuk menguji performa model. Hasil dari pembagian data dapat dilihat pada Tabel 6.5.

Tabel 6.5 Pembagian Data Training Dan Data Testing

```
train
```

Tanggal	Kasus
2020-03-02	6
2020-03-03	0
2020-03-04	0
2020-03-05	0
2020-03-06	2
...	...
2020-11-17	3671
2020-11-18	3671
2020-11-19	4265
2020-11-20	4798
2020-11-21	4792

265 rows × 1 columns

```
test
```

Tanggal	Kasus
2020-11-22	4998
2020-11-23	4360
2020-11-24	4442
2020-11-25	4192
2020-11-26	5534
2020-11-27	4917
2020-11-28	5828
2020-11-29	5418
2020-11-30	6267
2020-12-01	4617
2020-12-02	5092
2020-12-03	5533

6.2.4 Hasil Normalisasi Data

Untuk peramalan model LSTM dan hybrid ARIMA-LSTM, data yang digunakan harus dinormalisasi. Normalisasi yang dilakukan dalam penelitian tugas akhir ini menggunakan fungsi *min-max scaler*. Hasil dari data yang dinormalisasi dapat dilihat pada Gambar 6.2.

```
array([[1.10213079e-03],
       [0.00000000e+00],
       [0.00000000e+00],
       [0.00000000e+00],
       [3.67376929e-04],
       [0.00000000e+00],
       [3.67376929e-04],
       [0.00000000e+00],
       [2.38795004e-03],
       [1.46950771e-03],
       [1.28581925e-03],
       [6.42909625e-03],
       [4.95958854e-03],
       [3.85745775e-03],
       [3.12270389e-03],
       [6.98016165e-03],
       [1.01028655e-02],
       [1.43277002e-02],
       [1.10213079e-02],
```

Gambar 6.2 Hasil Normalisasi Data

6.3 Hasil Pembentukan Model

Subbab ini berisi hasil dari pembentukkan model peramalan.

Untuk hasil pembentukkan model ARIMA dapat dilihat pada 6.3.1.

Untuk hasil pembentukkan model LSTM dapat dilihat pada 6.3.2.

Untuk hasil pembenttukkan model hybrid ARIMA-LSTM dapat dilihat pada 6.3.3.

6.3.1 Hasil Pembentukan Model ARIMA

Pada bagian ini akan diuraikan hasil dari proses permodelan ARIMA dengan data virus corona di Indonesia. Karena ada beberapa penelitian yang menggunakan ARIMA yang *nonseasonal* [8] dan ada juga penelitian yang mengatakan bawah virus corona adalah *seasonal* [17], penelitian tugas akhir ini akan mencoba ARIMA dan SARIMA untuk mendapatkan hasil yang akurat.

Pada penelitian tugas akhir ini auto_arima digunakan untuk mencari model ARIMA yang terbaik dengan meminimalisir angka AIC.

Hasil dari fungsi auto_arima dari model ARIMA *non-seasonal* dapat dilihat pada Gambar 6.3.

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=inf, Time=0.36 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=4011.664, Time=0.01 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=4001.849, Time=0.04 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=3994.286, Time=0.12 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=4010.597, Time=0.01 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=3968.622, Time=0.26 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=3968.353, Time=0.29 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=3999.964, Time=0.03 sec
 ARIMA(3,1,1)(0,0,0)[0] intercept   : AIC=3956.586, Time=0.35 sec
 ARIMA(3,1,0)(0,0,0)[0] intercept   : AIC=3983.932, Time=0.05 sec
 ARIMA(4,1,1)(0,0,0)[0] intercept   : AIC=3952.356, Time=0.43 sec
 ARIMA(4,1,0)(0,0,0)[0] intercept   : AIC=3973.064, Time=0.07 sec
 ARIMA(5,1,1)(0,0,0)[0] intercept   : AIC=3931.555, Time=0.33 sec
 ARIMA(5,1,0)(0,0,0)[0] intercept   : AIC=3932.538, Time=0.09 sec
 ARIMA(5,1,2)(0,0,0)[0] intercept   : AIC=3907.878, Time=0.60 sec
 ARIMA(4,1,2)(0,0,0)[0] intercept   : AIC=3917.097, Time=0.60 sec
 ARIMA(5,1,3)(0,0,0)[0] intercept   : AIC=3927.502, Time=0.73 sec
 ARIMA(4,1,3)(0,0,0)[0] intercept   : AIC=3925.395, Time=0.63 sec
 ARIMA(5,1,2)(0,0,0)[0]             : AIC=3906.668, Time=0.55 sec
 ARIMA(4,1,2)(0,0,0)[0]             : AIC=3913.265, Time=0.44 sec
 ARIMA(5,1,1)(0,0,0)[0]             : AIC=3940.844, Time=0.20 sec
 ARIMA(5,1,3)(0,0,0)[0]             : AIC=3907.547, Time=0.59 sec
 ARIMA(4,1,1)(0,0,0)[0]             : AIC=3960.058, Time=0.14 sec
 ARIMA(4,1,3)(0,0,0)[0]             : AIC=3927.332, Time=0.52 sec

Best model:  ARIMA(5,1,2)(0,0,0)[0]
Total fit time: 7.478 seconds
```

Gambar 6.3 Perbandingan AIC Model ARIMA

Pada fungsi auto_ARIMA, Uji stasionaritas dan uji differensiasi sudah termasuk dalam perhitungannya. Sehingga Pada hasil fungsi auto_arima yang dapat dilihat pada Gambar 6.3 sudah termasuk hasil dari uji stasionaritas dan uji differensiasi sehingga parameter d adalah 1.

Seperti yang dapat dilihat pada Gambar 6.3 model ARIMA dengan AIC terendah adalah ARIMA(5,1,2) sehingga model tersebut akan dipilih sebagai model.

Hasil dari fungsi auto_arima dari model ARIMA *seasonal* dapat dilihat pada Gambar 6.4.

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(1,0,1)[7] intercept   : AIC=inf, Time=0.88 sec
 ARIMA(0,1,0)(0,0,0)[7] intercept   : AIC=4011.664, Time=0.09 sec
 ARIMA(1,1,0)(1,0,0)[7] intercept   : AIC=3957.963, Time=0.16 sec
 ARIMA(0,1,1)(0,0,1)[7] intercept   : AIC=3958.748, Time=0.24 sec
 ARIMA(0,1,0)(0,0,0)[7]             : AIC=4010.597, Time=0.01 sec
 ARIMA(1,1,0)(0,0,0)[7] intercept   : AIC=4001.849, Time=0.06 sec
 ARIMA(1,1,0)(2,0,0)[7] intercept   : AIC=3948.488, Time=0.27 sec
 ARIMA(1,1,0)(2,0,1)[7] intercept   : AIC=3920.411, Time=0.75 sec
 ARIMA(1,1,0)(1,0,1)[7] intercept   : AIC=3918.505, Time=0.43 sec
 ARIMA(1,1,0)(0,0,1)[7] intercept   : AIC=3971.795, Time=0.15 sec
 ARIMA(1,1,0)(1,0,2)[7] intercept   : AIC=3920.400, Time=0.72 sec
 ARIMA(1,1,0)(0,0,2)[7] intercept   : AIC=3967.453, Time=0.27 sec
 ARIMA(1,1,0)(2,0,2)[7] intercept   : AIC=inf, Time=0.80 sec
 ARIMA(0,1,0)(1,0,1)[7] intercept   : AIC=3964.054, Time=0.27 sec
 ARIMA(2,1,0)(1,0,1)[7] intercept   : AIC=3915.326, Time=0.55 sec
 ARIMA(2,1,0)(0,0,1)[7] intercept   : AIC=3970.175, Time=0.19 sec
 ARIMA(2,1,0)(1,0,0)[7] intercept   : AIC=3955.777, Time=0.21 sec
 ARIMA(2,1,0)(2,0,1)[7] intercept   : AIC=3917.276, Time=0.98 sec
 ARIMA(2,1,0)(1,0,2)[7] intercept   : AIC=3917.263, Time=0.76 sec
 ARIMA(2,1,0)(0,0,0)[7] intercept   : AIC=3999.964, Time=0.03 sec
 ARIMA(2,1,0)(0,0,2)[7] intercept   : AIC=3965.705, Time=0.31 sec
 ARIMA(2,1,0)(2,0,0)[7] intercept   : AIC=3945.542, Time=0.57 sec
 ARIMA(2,1,0)(2,0,2)[7] intercept   : AIC=inf, Time=1.14 sec
 ARIMA(3,1,0)(1,0,1)[7] intercept   : AIC=3908.566, Time=0.65 sec
 ARIMA(3,1,0)(0,0,1)[7] intercept   : AIC=3959.987, Time=0.25 sec
 ARIMA(3,1,0)(1,0,0)[7] intercept   : AIC=3947.416, Time=0.26 sec
 ARIMA(3,1,0)(2,0,1)[7] intercept   : AIC=3910.934, Time=1.02 sec
 ARIMA(3,1,0)(1,0,2)[7] intercept   : AIC=3910.595, Time=0.98 sec
 ARIMA(3,1,0)(0,0,0)[7] intercept   : AIC=3983.932, Time=0.05 sec
 ARIMA(3,1,0)(0,0,2)[7] intercept   : AIC=3954.613, Time=0.35 sec
 ARIMA(3,1,0)(2,0,0)[7] intercept   : AIC=3936.929, Time=0.65 sec
 ARIMA(3,1,0)(2,0,2)[7] intercept   : AIC=inf, Time=1.25 sec
 ARIMA(4,1,0)(1,0,1)[7] intercept   : AIC=3909.610, Time=0.78 sec
 ARIMA(3,1,1)(1,0,1)[7] intercept   : AIC=inf, Time=0.71 sec
 ARIMA(2,1,1)(1,0,1)[7] intercept   : AIC=inf, Time=0.73 sec
 ARIMA(4,1,1)(1,0,1)[7] intercept   : AIC=inf, Time=0.81 sec
 ARIMA(3,1,0)(1,0,1)[7]             : AIC=3906.830, Time=0.50 sec
 ARIMA(3,1,0)(0,0,1)[7]             : AIC=3960.020, Time=0.16 sec
 ARIMA(3,1,0)(1,0,0)[7]             : AIC=3946.721, Time=0.16 sec
```

```
 ARIMA(3,1,0)(2,0,1)[7]              : AIC=3908.738, Time=0.65 sec
 ARIMA(3,1,0)(1,0,2)[7]              : AIC=3908.720, Time=0.61 sec
 ARIMA(3,1,0)(0,0,0)[7]              : AIC=3984.862, Time=0.03 sec
 ARIMA(3,1,0)(0,0,2)[7]              : AIC=3954.449, Time=0.26 sec
 ARIMA(3,1,0)(2,0,0)[7]              : AIC=3936.006, Time=0.26 sec
 ARIMA(3,1,0)(2,0,2)[7]              : AIC=inf, Time=1.13 sec
 ARIMA(2,1,0)(1,0,1)[7]              : AIC=3913.486, Time=0.35 sec
 ARIMA(4,1,0)(1,0,1)[7]              : AIC=3907.727, Time=0.55 sec
 ARIMA(3,1,1)(1,0,1)[7]              : AIC=3904.455, Time=0.57 sec
 ARIMA(3,1,1)(0,0,1)[7]              : AIC=3945.604, Time=0.25 sec
 ARIMA(3,1,1)(1,0,0)[7]              : AIC=3936.820, Time=0.27 sec
 ARIMA(3,1,1)(2,0,1)[7]              : AIC=3906.257, Time=0.88 sec
 ARIMA(3,1,1)(1,0,2)[7]              : AIC=3906.227, Time=0.76 sec
 ARIMA(3,1,1)(0,0,0)[7]              : AIC=3963.739, Time=0.14 sec
 ARIMA(3,1,1)(0,0,2)[7]              : AIC=3942.324, Time=0.36 sec
 ARIMA(3,1,1)(2,0,0)[7]              : AIC=3929.167, Time=0.45 sec
 ARIMA(3,1,1)(2,0,2)[7]              : AIC=inf, Time=1.12 sec
 ARIMA(2,1,1)(1,0,1)[7]              : AIC=3904.144, Time=0.39 sec
 ARIMA(2,1,1)(0,0,1)[7]              : AIC=3949.865, Time=0.22 sec
 ARIMA(2,1,1)(1,0,0)[7]              : AIC=3937.362, Time=0.21 sec
 ARIMA(2,1,1)(2,0,1)[7]              : AIC=3905.852, Time=0.74 sec
 ARIMA(2,1,1)(1,0,2)[7]              : AIC=3905.800, Time=0.64 sec
 ARIMA(2,1,1)(0,0,0)[7]              : AIC=3974.774, Time=0.13 sec
 ARIMA(2,1,1)(0,0,2)[7]              : AIC=3945.765, Time=0.37 sec
 ARIMA(2,1,1)(2,0,0)[7]              : AIC=3929.058, Time=0.35 sec
 ARIMA(2,1,1)(2,0,2)[7]              : AIC=inf, Time=0.88 sec
 ARIMA(1,1,1)(1,0,1)[7]              : AIC=3903.567, Time=0.37 sec
 ARIMA(1,1,1)(0,0,1)[7]              : AIC=3947.970, Time=0.19 sec
 ARIMA(1,1,1)(1,0,0)[7]              : AIC=3936.948, Time=0.16 sec
 ARIMA(1,1,1)(2,0,1)[7]              : AIC=3905.439, Time=0.66 sec
 ARIMA(1,1,1)(1,0,2)[7]              : AIC=3905.416, Time=0.62 sec
 ARIMA(1,1,1)(0,0,0)[7]              : AIC=3974.867, Time=0.11 sec
 ARIMA(1,1,1)(0,0,2)[7]              : AIC=3944.078, Time=0.33 sec
 ARIMA(1,1,1)(2,0,0)[7]              : AIC=3928.983, Time=0.30 sec
 ARIMA(1,1,1)(2,0,2)[7]              : AIC=inf, Time=0.94 sec
 ARIMA(0,1,1)(1,0,1)[7]              : AIC=3902.505, Time=0.28 sec
 ARIMA(0,1,1)(0,0,1)[7]              : AIC=3959.211, Time=0.10 sec
 ARIMA(0,1,1)(1,0,0)[7]              : AIC=3942.738, Time=0.22 sec
 ARIMA(0,1,1)(2,0,1)[7]              : AIC=3904.429, Time=0.46 sec
 ARIMA(0,1,1)(1,0,2)[7]              : AIC=3904.413, Time=0.63 sec
 ARIMA(0,1,1)(0,0,0)[7]              : AIC=3994.962, Time=0.05 sec
 ARIMA(0,1,1)(0,0,2)[7]              : AIC=3953.340, Time=0.20 sec
 ARIMA(0,1,1)(2,0,0)[7]              : AIC=3931.346, Time=0.16 sec
 ARIMA(0,1,1)(2,0,2)[7]              : AIC=inf, Time=0.97 sec
 ARIMA(0,1,0)(1,0,1)[7]              : AIC=3962.147, Time=0.15 sec
 ARIMA(0,1,2)(1,0,1)[7]              : AIC=3903.848, Time=0.33 sec
 ARIMA(1,1,0)(1,0,1)[7]              : AIC=3916.633, Time=0.29 sec
 ARIMA(1,1,2)(1,0,1)[7]              : AIC=3904.283, Time=0.50 sec
 ARIMA(0,1,1)(1,0,1)[7] intercept    : AIC=3904.154, Time=0.56 sec

Best model:  ARIMA(0,1,1)(1,0,1)[7]
Total fit time: 40.330 seconds
```

Gambar 6.4 Perbandingan AIC Model SARIMA

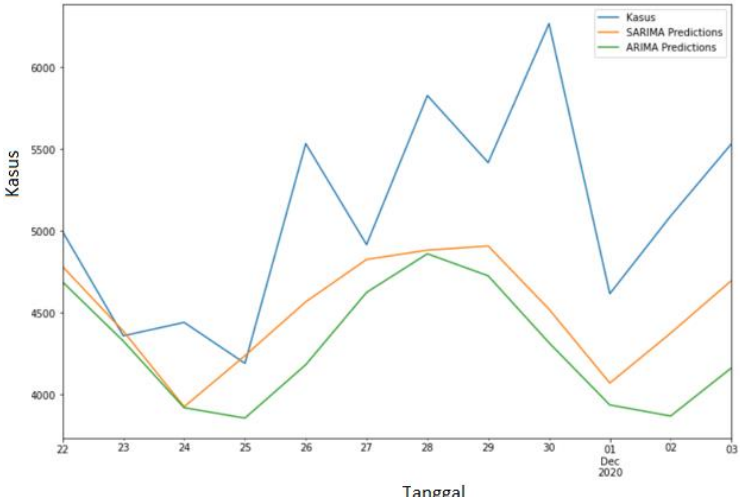
Seperti yang dapat dilihat pada Gambar 6.4 model SARIMA dengan AIC terendah adalah SARIMA(0,1,1)(1,0,1)[7] sehingga model tersebut akan dipilih sebagai model.

Setelah parameter model ARIMA dan SARIMA diketahui, kedua model akan divalidasi untuk mengetahui keakuratan dari model yang telah pilih. Hasil dari perbandingan validasi model ARIMA dan SARIMA dapat dilihat pada Tabel 6.6.

Tabel 6.6 Tabel Perbandingan Validasi Model

Model	MAPE	RMSE
ARIMA(5,1,2)	15.0719%	975.9489
SARIMA(0,1,1)(1,0,1)[7]	10.9468%	761.4039

Hasil dari perbandingan model ARIMA(5,1,2) dan SARIMA(0,1,1)(1,0,1)[7] dengan data aktual dapat dilihat pada Gambar 6.5.



Gambar 6.5 Grafik Perbandingan Model ARIMA dan SARIMA Dengan Data Aktual

Seperti yang dapat dilihat pada Tabel 6.6, model SARIMA menghasilkan hasil yang lebih baik dibandingkan model ARIMA sehingga model SARIMA akan digunakan sebagai model akhir.

Pada model SARIMA ada parameter (p,d,q)(P,D,Q)[M]. Dari parameter (p,d,q) p adalah *autoregression order* untuk komponen *trend*, d adalah *difference order* untuk komponen *trend* dan q adalah *moving average order* untuk komponen *trend*. Sedangkan untuk parameter (P,D,Q) P adalah *autoregression order* untuk komponen *seasonal*, d adalah *difference order* untuk komponen *seasonal* dan q adalah *moving average order* untuk komponen *seasonal*. Parameter M adalah jumlah *time step*. Parameter M juga akan diperbandingkan dengan nilai M yang berbeda untuk mencari validasi model yang terbaik. Hasil perbandingan faktor M dapat dilihat pada Tabel 6.7.

Tabel 6.7 Perbandingan Faktor M Pada Model SARIMA

Model SARIMA	MAPE	RMSE
SARIMA(1,1,1)(1,0,1)[2]	12.7217	883.6512
SARIMA(2,1,2)(2,0,2)[3]	14.4931	955.6770
SARIMA(0,1,1)(2,0,2)[4]	12.4773	893.9856
SARIMA(0,1,1)(2,0,2)[5]	10.3083	748.5865
SARIMA(0,1,1)(2,0,2)[6]	10.8441	741.0184
SARIMA(0,1,1)(1,0,1)[7]	10.9468	761.4039
SARIMA(1,1,1)(0,0,1)[8]	12.0883	840.8154
SARIMA(0,1,2)(2,0,2)[9]	17.0429	1107.5626
SARIMA(1,1,1)(1,0,0)[10]	13.3501	924.2190
SARIMA(1,1,1)(2,0,0)[11]	15.7594	1005.2891
SARIMA(1,1,1)(2,0,0)[12]	13.7584	886.6167
SARIMA(0,1,1)(2,0,2)[13]	**9.2602**	**694.5913**
SARIMA(1,1,3)(1,0,1)[14]	12.0131	799.4301
SARIMA(1,1,1)(0,0,1)[15]	11.4094	780.6342
SARIMA(1,1,1)(2,0,0)[16]	11.1016	818.3917
SARIMA(1,1,1)(1,0,0)[17]	13.5318	935.4495
SARIMA(1,1,1)(1,0,0)[18]	15.1571	1014.9281
SARIMA(1,1,1)(2,0,0)[19]	14.0710	997.0898
SARIMA(1,1,1)(0,0,2)[20]	11.6554	831.1424
SARIMA(5,1,0)(1,0,1)[21]	11.3014	785.8062
SARIMA(1,1,1)(0,0,1)[22]	13.1642	886.1101
SARIMA(1,1,1)(2,0,0)[23]	13.0036	865.4915
SARIMA(1,1,1)(0,0,1)[24]	12.6579	839.1276

SARIMA(1,1,1)(2,0,0)[25]	11.4269	798.8808
SARIMA(1,1,1)(0,0,1)[26]	11.5422	794.4248
SARIMA(1,1,1)(0,0,1)[27]	15.5610	1052.4673
SARIMA(3,1,1)(0,0,2)[28]	16.5258	1082.1911

Seperti yang bisa dilihat pada Tabel 6.7, setelah nilai 17 nilai validasi hasil peramalan tidak mengaruh besar, hanya naik turun diatas 10 sehingga tidak dilanjutkan lagi. Sehingga model SARIMA yang terpilih menjadi model peramalan adalah SARIMA(0,1,1)(2,0,2)[13] dimana nilai MAPE yang dihasilkan adalah 9.2602% dan nilai RMSE yang dihasilkan adalah 694.5913. Berdasarkan standar penilaian akurasi MAPE yang dapat dilihat pada Tabel 2.2, model SARIMA(0,1,1)(2,0,2)[13] adalah model dengan peramalan yang sangat baik.

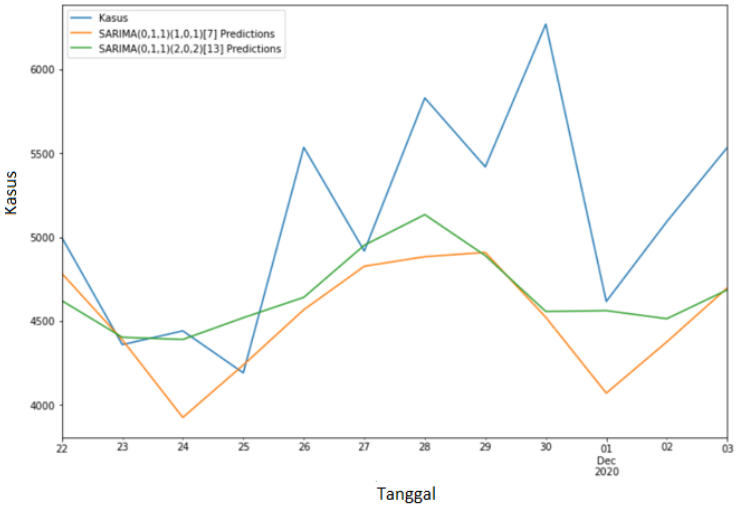
Dari model SARIMA(0,1,1)(2,0,2)[13], berdasarkan *rules for identifying ARIMA models* [18], d dari model SARIMA adalah 1 berdasarkan rumusan matematis yang dapat dilihat pada

$$d = 1: y_t = Y_t - Y_{t-1}$$

Persamaan 6.1 Rumusan Matematis Parameter d

Dimana Yt adalah model ARIMA. Parameter p adalah 0 karena model ARIMA tersebut adalah model tanpa konstan. Sedangkan parameter q adalah 1 karena model ARIMA tersebut memiliki 1 *random event* yang terjadi pada grafik ACF. Sehingga model ARIMA(0,1,1) adalah model tanpa konstan/*simple exponential smoothing*. Untuk parameter P adalah 2 karena parameter SAR bergantung pada *season* sebelum dari data *time series*. *Season* yang terbaru akan tergantung pada *season* sebelumnya. Parameter D adalah 0 karena struktur *time series* sudah *stationary*. Parameter Q adalah 2 berdasarkan moving average yang berlintas *season*.

Sehingga model SARIMA yang dipilih menjadi model akhir adalah SARIMA(0,1,1)(2,0,2)[13]. Perbandingan hasil peramalan dengan data aktual model SARIMA(0,1,1)(2,0,2)[13] dan model SARIMA(0,1,1)(1,0,1)[7] dapat dilihat pada Gambar 6.6.



Gambar 6.6 Perbandingan Nilai Peramalan Dengan Data Aktual Model SARIMA(0,1,1)(2,0,2)[13] Dan Model SARIMA(0,1,1)(1,0,1)[7]

6.3.2 Hasil Pembentukan Model LSTM

Pada bagian ini akan diuraikan hasil dari proses permodelan LSTM dengan data virus corona di Indonesia. Pemilihan model LSTM terbaik akan ditentukan dari nilai RMSE yang terendah. Untuk jumlah *timestep* yang digunakan pada penelitian tugas akhir ini adalah 10. Setiap *timestep* memiliki jumlah skenario parameter yang berbeda karena parameter yang berbeda dapat menghasilkan hasil yang berbeda.

a. Timestep 1

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.8 merupakan 10 model dengan MAPE dan RMSE yang tertinggi setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 1, model terbaik

didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* Adam dan *batch size* sebanyak 128. Nilai MAPE yang dihasilkan adalah 9.2406% dan nilai RMSE yang dihasilkan adalah 551.6970.

Tabel 6.8 Hasil Tuning Parameter Timestep 1

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	Adam	ReLU	128	9.2406	551.6970
50	Adam	ReLU	128	10.4338	581.7575
100	Adam	ReLU	64	9.9284	627.6154
50	Adam	tanh	128	10.1704	643.6723
100	RMSprop	tanh	128	10.9209	704.5965
50	RMSprop	ReLU	128	11.2265	730.8023
50	Adam	ReLU	64	12.9325	751.8390
100	RMSprop	ReLU	128	11.4839	757.2217
50	RMSprop	tanh	128	11.8173	785.6661
100	Adam	tanh	128	13.8402	919.3715

b. Timestep 2

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.9 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 2, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi tanh, *optimizer* RMSprop dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 12.0158% dan nilai RMSE yang dihasilkan adalah 826.5363.

Tabel 6.9 Hasil Tuning Parameter Timestep 2

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	RMSprop	tanh	32	12.0158	826.5363
100	RMSprop	ReLU	128	11.6818	827.8603
100	RMSprop	tanh	128	12.3057	855.8018
100	SGD	tanh	32	13.3979	877.5782
100	RMSprop	ReLU	64	12.957	896.2786
50	SGD	tanh	32	13.7443	901.7472
100	SGD	ReLU	32	15.0757	997.9482

100	adam	ReLU	128	15.6156	1017.9404
50	SGD	ReLU	32	15.4802	1050.0451
50	RMSprop	tanh	64	15.7321	1080.4593

c. Timestep 3

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.10 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 3, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* RMSprop dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 11.4968% dan nilai RMSE yang dihasilkan adalah 799.6381.

Tabel 6.10 Hasil Tuning Parameter Timestep 3

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	RMSprop	ReLU	32	11.4968	799.6381
50	SGD	tanh	32	14.1479	930.8499
100	RMSprop	ReLU	64	13.6604	934.4519
100	SGD	tanh	32	14.2149	935.0073
50	RMSprop	ReLU	128	14.5708	966.6401
100	adam	ReLU	128	14.9622	983.1447
100	adam	ReLU	64	15.5424	1012.5463
100	adam	tanh	128	15.5011	1043.1284
100	adam	tanh	64	15.9639	1068.7144
100	SGD	ReLU	32	16.2697	1095.6449

d. Timestep 4

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.11 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 4, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi *Tanh*, *optimizer* RMSprop dan *batch size* sebanyak 64. Nilai MAPE yang dihasilkan adalah 9.0673% dan nilai RMSE yang dihasilkan adalah 532.2553.

Tabel 6.11 Hasil Tuning Parameter Timestep 4

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	RMSprop	tanh	64	9.0673	532.2553
100	adam	tanh	128	9.6467	614.0574
100	adam	ReLU	128	10.4285	709.6912
100	RMSprop	ReLU	32	10.9843	730.2544
100	RMSprop	tanh	128	11.0011	616.7681
50	adam	ReLU	128	11.3372	799.6615
50	adam	tanh	64	12.0660	849.0953
100	SGD	tanh	32	12.4878	829.5344
50	RMSprop	ReLU	32	12.5421	878.2406
50	adam	ReLU	64	12.9453	905.5147

e. Timestep 5

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.12 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 5, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi *Tanh*, *optimizer* RMSprop dan *batch size* sebanyak 128. Nilai MAPE yang dihasilkan adalah 8.4591% dan nilai RMSE yang dihasilkan adalah 526.9309.

Tabel 6.12 Hasil Tuning Parameter Timestep 5

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	RMSprop	tanh	128	8.4591	526.9309
50	RMSprop	ReLU	32	8.4971	542.9815
50	SGD	tanh	128	10.3362	670.2256
50	SGD	ReLU	128	10.5814	689.5983
50	SGD	ReLU	64	10.7234	703.5132
50	SGD	tanh	64	10.9516	724.853
100	SGD	tanh	32	10.95	725.3973
100	SGD	tanh	128	10.9896	727.274
50	adam	ReLU	128	10.7566	770.443
100	RMSprop	tanh	32	10.6274	772.1096

f. Timestep 6

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.13 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 6, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi tanh, *optimizer* SGD dan *batch size* sebanyak 128. Nilai MAPE yang dihasilkan adalah 8.9501% dan nilai RMSE yang dihasilkan adalah 532.3749.

Tabel 6.13 Hasil Tuning Parameter Timestep 6

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	SGD	tanh	128	8.9501	532.3749
50	SGD	ReLU	64	9.3024	534.0982
100	SGD	ReLU	64	9.0100	552.3009
50	RMSprop	ReLU	64	7.9650	572.3114
100	SGD	tanh	64	8.9711	577.9262
50	RMSprop	tanh	64	8.1078	587.4965
50	SGD	tanh	64	9.2698	605.5463
50	RMSprop	tanh	128	8.2304	611.1886
50	SGD	ReLU	32	9.3852	613.7512
50	RMSprop	tanh	32	8.8401	665.1353

g. Timestep 7

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.14 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 7, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* SGD dan *batch size* sebanyak 64. Nilai MAPE yang dihasilkan adalah 8.9236% dan nilai RMSE yang dihasilkan adalah 519.465.

Tabel 6.14 Hasil Tuning Parameter Timestep 7

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	SGD	ReLU	64	8.9236	519.4650
50	SGD	ReLU	64	8.7519	520.1356
100	SGD	tanh	64	8.7050	522.3864
50	SGD	ReLU	32	8.5869	536.2565
50	SGD	tanh	64	8.6279	538.4196
50	adam	tanh	64	7.8371	561.9617
100	adam	ReLU	32	7.7258	570.6021
50	adam	ReLU	128	7.8505	583.4873
50	RMSprop	tanh	128	7.7901	584.8271
100	SGD	ReLU	32	8.8884	586.3254

h. Timestep 8

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.15 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 8, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* RMSprop dan *batch size* sebanyak 64. Nilai MAPE yang dihasilkan adalah 8.3297% dan nilai RMSE yang dihasilkan adalah 503.4190.

Tabel 6.15 Hasil Tuning Parameter Timestep 8

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	RMSprop	ReLU	64	8.3297	503.4190
50	SGD	ReLU	64	8.7021	517.7165
50	SGD	tanh	128	8.6748	518.5509
100	SGD	ReLU	64	8.6684	520.7273
100	SGD	tanh	64	8.6355	536.8711
100	SGD	ReLU	32	8.6009	544.1428
50	RMSprop	ReLU	64	8.1358	548.3893
50	SGD	ReLU	32	8.7065	564.6066
50	SGD	tanh	64	8.7377	570.7608
50	adam	tanh	128	8.5607	593.9690

i. Timestep 9

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.16 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 9, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* RMSprop dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 6.0197% dan nilai RMSE yang dihasilkan adalah 473.5747.

Tabel 6.16 Hasil Tuning Parameter Timestep 9

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	RMSprop	ReLU	32	6.0197	473.5747
50	RMSprop	tanh	64	8.2371	603.4113
100	SGD	tanh	32	11.2617	782.6268
100	SGD	tanh	64	11.4302	796.4792
50	SGD	ReLU	32	11.7683	818.6902
100	SGD	tanh	128	11.9261	830.8890
100	RMSprop	ReLU	128	11.9331	838.8575
50	SGD	tanh	64	12.3079	855.6851
50	SGD	tanh	32	12.3089	857.3750
50	SGD	tanh	128	12.8241	887.4132

j. Timestep 10

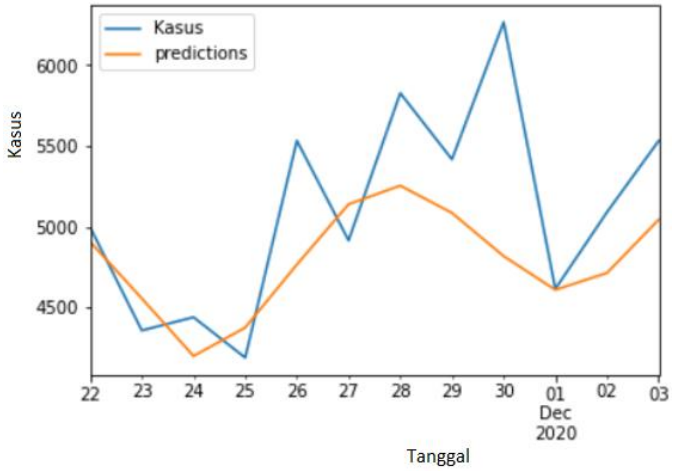
Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.17 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 10, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi ReLU, *optimizer* RMSprop dan *batch size* sebanyak 64. Nilai MAPE yang dihasilkan adalah 6.5866% dan nilai RMSE yang dihasilkan adalah 476.8011.

Tabel 6.17 Hasil Tuning Parameter Timestep 10

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	RMSprop	ReLU	64	6.5866	476.8011

100	RMSprop	ReLU	64	10.155	562.8096
50	RMSprop	tanh	128	8.7913	664.9616
100	SGD	tanh	128	11.5864	798.9943
100	SGD	tanh	64	11.6589	803.4482
50	SGD	tanh	64	11.8076	813.7690
100	SGD	tanh	32	12.0090	826.4481
50	SGD	tanh	32	12.1701	837.9545
100	SGD	ReLU	64	12.3432	849.6934
50	RMSprop	ReLU	128	12.3819	863.1836

Model LSTM dengan performa terbaik dilihat dari RMSE berada pada *timestep* 9 dengan nilai MAPE 6.0197 dan RMSE 473.4757. Gambar 6.7 merupakan hasil perbandingan hasil peramalan dengan data aktual dalam bentuk grafik.



Gambar 6.7 Hasil Perbandingan Peramalan LSTM Dengan Data Aktual

Hasil perbandingan MAPE terendah dari setiap *timestep* dapat dilihat pada

Tabel 6.18.

Tabel 6.18 Perbandingan MAPE Terendah Dari Setiap Timestep

Time step	LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
9	100	rmsprop	relu	32	6.0197	473.5747
10	50	rmsprop	relu	64	6.5866	476.8011
7	100	adam	relu	32	7.7258	570.6021
6	50	rmsprop	relu	64	7.9650	572.3114
8	50	rmsprop	relu	64	8.1358	548.3893
5	50	rmsprop	tanh	128	8.4591	526.9309
4	50	rmsprop	tanh	64	9.0673	532.2553
1	100	Adam	relu	128	9.2406	551.6970
3	100	rmsprop	relu	32	11.4968	799.6381
2	100	rmsprop	relu	128	11.6818	827.8603

Hasil perbandingan RMSE terendah dari setiap timestep dapat dilihat pada

Time step	LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
9	100	rmsprop	relu	32	6.0197	473.5747
10	50	rmsprop	relu	64	6.5866	476.8011
5	50	rmsprop	tanh	128	8.4591	526.9309
4	50	rmsprop	tanh	64	9.0673	532.2553
8	50	rmsprop	relu	64	8.1358	548.3893
1	100	Adam	relu	128	9.2406	551.6970
7	100	adam	relu	32	7.7258	570.6021
6	50	rmsprop	relu	64	7.9650	572.3114
3	100	rmsprop	relu	32	11.4968	799.6381
2	100	rmsprop	relu	128	11.6818	827.8603

Tabel *6.19*.

Tabel 6.19 Perbandingan RMSE Terendah Dari Setiap Timestep

Dari hasil perbandingan MAPE dan RMSE dari setiap *timestep* dapat diketahui bahwa model LSTM dapat performa baik dengan *optimizer* RMSprop dan adam. *Optimzier* SGD tidak cocok dalam perhitungan LSTM.

6.3.3 Hasil Pembentukan Model Hybrid ARIMA-LSTM

Pada bagian ini akan diuraikan hasil dari proses permodelan hybrid ARIMA-LSTM dengan data virus corona di Indonesia. Pada pembentukan model hybrid ARIMA-LSTM data akan dibagi menjadi 3 komponen, yaitu adalah *trend*, *seasonal* dan *residual*. Hasil dari Pembagian komponen dapat dilihat pada Gambar 6.8, Gambar 6.9 dan Gambar 6.10.

Time step	LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
9	100	rmsprop	relu	32	6.0197	473.5747
10	50	rmsprop	relu	64	6.5866	476.8011
5	50	rmsprop	tanh	128	8.4591	526.9309
4	50	rmsprop	tanh	64	9.0673	532.2553
8	50	rmsprop	relu	64	8.1358	548.3893
1	100	Adam	relu	128	9.2406	551.6970
7	100	adam	relu	32	7.7258	570.6021
6	50	rmsprop	relu	64	7.9650	572.3114
3	100	rmsprop	relu	32	11.4968	799.6381
2	100	rmsprop	relu	128	11.6818	827.8603

```
df['Trend']
```

```
Tanggal
2020-03-02            NaN
2020-03-03            NaN
2020-03-04            NaN
2020-03-05       1.428571
2020-03-06       0.571429
                 ...
2020-11-29    5381.857143
2020-11-30    5381.714286
2020-12-01            NaN
2020-12-02            NaN
2020-12-03            NaN
Name: Trend, Length: 277, dtype: float64
```

Gambar 6.8 Komponen *Trend*

```
df['Seasonal']
```

```
Tanggal
2020-03-02    -60.136096
2020-03-03   -259.264494
2020-03-04   -126.095321
2020-03-05     70.424344
2020-03-06    139.669765
                 ...
2020-11-29    109.951816
2020-11-30    -60.136096
2020-12-01   -259.264494
2020-12-02   -126.095321
2020-12-03     70.424344
Name: Seasonal, Length: 277, dtype: float64
```

Gambar 6.9 Komponen *Seasonal*

```
df['Residuals']
```

```
Tanggal
2020-03-02           NaN
2020-03-03           NaN
2020-03-04           NaN
2020-03-05    -71.852915
2020-03-06   -138.241194
                   ...
2020-11-29    -73.808959
2020-11-30    945.421810
2020-12-01           NaN
2020-12-02           NaN
2020-12-03           NaN
Name: Residuals, Length: 277, dtype: float64
```

Gambar 6.10 Komponen *Residual*

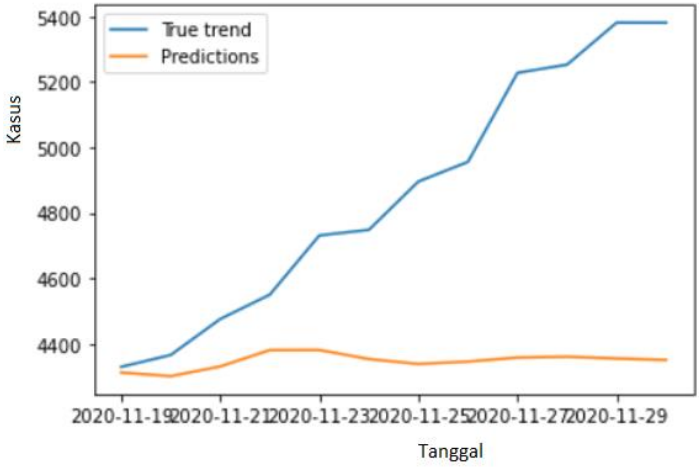
Pada penelitian tugas akhir ini auto_arima digunakan untuk mencari model ARIMA yang terbaik dengan meminimalisir angka AIC. SARIMA tidak digunakan pada model hybrid ARIMA-LSTM karena komponen *seasonal* didekomposisikan dan dihitungkan menggunakan model LSTM. Hasil dari fungsi auto_arima terhadap komponen *trend* dapat dilihat pada Gambar 6.11.

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=2865.411, Time=0.33 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=2956.421, Time=0.01 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=2877.749, Time=0.06 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=2908.087, Time=0.11 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=2985.235, Time=0.01 sec
 ARIMA(1,1,2)(0,0,0)[0] intercept   : AIC=2866.674, Time=0.17 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=2866.643, Time=0.18 sec
 ARIMA(3,1,2)(0,0,0)[0] intercept   : AIC=2867.363, Time=0.40 sec
 ARIMA(2,1,3)(0,0,0)[0] intercept   : AIC=2855.593, Time=0.47 sec
 ARIMA(1,1,3)(0,0,0)[0] intercept   : AIC=2868.665, Time=0.27 sec
 ARIMA(3,1,3)(0,0,0)[0] intercept   : AIC=inf, Time=0.58 sec
 ARIMA(2,1,4)(0,0,0)[0] intercept   : AIC=inf, Time=0.62 sec
 ARIMA(1,1,4)(0,0,0)[0] intercept   : AIC=2863.297, Time=0.51 sec
 ARIMA(3,1,4)(0,0,0)[0] intercept   : AIC=2840.190, Time=0.64 sec
 ARIMA(4,1,4)(0,0,0)[0] intercept   : AIC=inf, Time=0.66 sec
 ARIMA(3,1,5)(0,0,0)[0] intercept   : AIC=2821.932, Time=0.75 sec
 ARIMA(2,1,5)(0,0,0)[0] intercept   : AIC=2821.885, Time=0.71 sec
 ARIMA(1,1,5)(0,0,0)[0] intercept   : AIC=2848.165, Time=0.56 sec
 ARIMA(2,1,5)(0,0,0)[0]             : AIC=2825.012, Time=0.64 sec

Best model:  ARIMA(2,1,5)(0,0,0)[0] intercept
Total fit time: 7.701 seconds
```

Gambar 6.11 Hasil Fungsi auto_arima Terhadap Komponen Trend

Seperti yang dapat dilihat pada Gambar 6.11 model ARIMA dengan AIC terendah adalah ARIMA(2,1,5) sehingga model tersebut akan dipilih sebagai model. Hasil dari model ARIMA(2,1,5) terhadap komponen *trend* dengan perbandingan data komponen *trend* aktual dapat dilihat pada Gambar 6.12.



Gambar 6.12 Hasil Perbandingan Peramalan Model ARIMA Terhadap Komponen Trend Dengan Data Trend Aktual

Hasil validasi model ARIMA(2,1,5) terhadap komponen *trend* adalah MAPE dengan nilai 10.0217% dan RMSE dengan nilai 625.5786.

Setelah ARIMA dibangunkan model LSTM dibangunkan untuk menghitungkan komponen *seasonal* dan *residual*. Pada langkah ini model LSTM yang terbaik akan ditentukan dari nilai RMSE yang terendah. Untuk jumlah *timestep* yang digunakan pada penelitian tugas akhir ini adalah 10. Setiap *timestep* memiliki jumlah skenario parameter yang berbeda karena parameter yang berbeda dapat menghasilkan hasil yang berbeda.

a. Timestep 1

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.20 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 1, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi ReLU, *optimizer* SGD dan *batch size*

sebanyak 32. Nilai MAPE yang dihasilkan adalah 112.8276% dan nilai RMSE yang dihasilkan adalah 140.8594.

Tabel 6.20 Hasil Tuning Parameter Komponen Seasonal Timestep 1

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	SGD	ReLU	32	112.8276	140.8594
100	SGD	tanh	32	114.8621	143.2589
50	SGD	tanh	32	114.8256	143.2931
100	SGD	ReLU	32	114.7033	143.5697
100	SGD	ReLU	64	113.9949	145.2431
50	SGD	ReLU	64	114.3040	145.7924
50	SGD	tanh	64	115.4504	146.8471
100	SGD	tanh	64	116.1017	147.8008
50	RMSprop	ReLU	32	120.2065	155.3576
100	SGD	ReLU	128	120.3606	157.2647

b. Timestep 2

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.21 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 2, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* adam dan *batch size* sebanyak 64. Nilai MAPE yang dihasilkan adalah 18.9853% dan nilai RMSE yang dihasilkan adalah 29.2592.

Tabel 6.21 Hasil Tuning Parameter Komponen Seasonal Timestep 2

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	adam	ReLU	64	18.9853	29.2592
50	RMSprop	tanh	32	26.6703	33.6538
100	RMSprop	ReLU	128	16.7078	34.6246
100	adam	ReLU	128	24.2858	40.3116
100	RMSprop	tanh	128	24.2921	40.4248
50	RMSprop	tanh	128	35.2712	51.3061
50	adam	ReLU	32	35.3888	51.6591
50	RMSprop	ReLU	128	47.6214	61.9465
100	adam	ReLU	32	42.7548	64.8642
100	RMSprop	ReLU	64	49.7439	67.3503

c. Timestep 3

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.22 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 3, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi ReLU, *optimizer* Adam dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 1.0570% dan nilai RMSE yang dihasilkan adalah 1.138.

Tabel 6.22 Hasil Tuning Parameter Komponen Seasonal Timestep 3

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	adam	ReLU	32	1.0570	1.1380
100	adam	ReLU	32	3.5766	4.3779
100	RMSprop	ReLU	32	13.2686	15.1168
50	RMSprop	tanh	32	15.3881	21.3234
50	adam	tanh	32	23.6837	26.7128
100	RMSprop	tanh	64	17.8033	27.6400
50	RMSprop	ReLU	32	28.5851	32.0103
50	RMSprop	tanh	64	29.4187	33.7538
100	RMSprop	ReLU	64	33.8458	41.0171
50	RMSprop	tanh	128	35.04364	50.6556

d. Timestep 4

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.23 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 4, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* Adam dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 0.001% dan nilai RMSE yang dihasilkan adalah 0.001.

Tabel 6.23 Hasil Tuning Parameter Komponen Seasonal Timestep 4

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE

100	adam	ReLU	32	0.0010	0.0010
50	adam	ReLU	32	0.4205	0.5344
100	adam	tanh	32	3.9896	5.4146
100	adam	ReLU	64	6.5269	8.2955
50	adam	tanh	32	6.8047	10.4838
50	RMSprop	ReLU	32	11.7694	11.4637
100	RMSprop	ReLU	32	12.2017	14.1721
100	adam	tanh	64	9.9356	14.8732
100	RMSprop	tanh	32	13.0335	16.7409
50	RMSprop	tanh	32	11.9067	17.3393

e. Timestep 5

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.24 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 5, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* Adam dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 0.001639% dan nilai RMSE yang dihasilkan adalah 0.001906.

Tabel 6.24 Hasil Tuning Parameter Komponen Seasonal Timestep 5

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	adam	ReLU	32	0.00163	0.0019
100	adam	ReLU	64	0.00168	0.0021
50	adam	ReLU	64	5.2232	5.7208
100	adam	tanh	32	6.9048	7.3537
50	adam	tanh	32	5.8141	7.4157
50	adam	ReLU	32	7.4679	8.6780
100	adam	tanh	64	9.9398	11.6676
50	RMSprop	tanh	32	12.4127	14.0084
100	RMSprop	tanh	32	12.4597	14.2965
50	RMSprop	ReLU	32	13.4252	15.3971

f. Timestep 6

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.25 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 6, model terbaik

didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* Adam dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 0.0124% dan nilai RMSE yang dihasilkan adalah 0.0133.

Tabel 6.25 Hasil Tuning Parameter Komponen Seasonal Timestep 6

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	adam	ReLU	32	0.0124	0.0133
50	adam	ReLU	32	0.4471	0.4772
100	adam	ReLU	64	0.4098	0.4943
50	adam	ReLU	64	0.8306	1.0248
100	adam	tanh	32	0.9836	1.0420
50	adam	tanh	32	1.1721	1.2543
100	adam	tanh	64	4.6059	4.8306
50	RMSprop	tanh	32	16.5680	18.8380
100	RMSprop	ReLU	32	22.0027	22.0820
100	RMSprop	tanh	32	18.4710	22.4100

g. Timestep 7

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.26 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 7, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* Adam dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 0.0289% dan nilai RMSE yang dihasilkan adalah 0.0373.

Tabel 6.26 Hasil Tuning Parameter Komponen Seasonal Timestep 7

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	adam	ReLU	32	0.0289	0.0373
50	adam	ReLU	64	0.0497	0.0581
50	adam	ReLU	32	1.3065	1.4120
100	adam	ReLU	64	3.0855	3.4572
100	adam	tanh	32	2.4402	3.5174
50	adam	tanh	32	3.6487	4.4459
100	adam	tanh	64	6.2070	7.4472

100	RMSprop	tanh	32	11.3463	13.0778
50	adam	tanh	64	13.3283	13.9035
50	RMSprop	tanh	32	16.9405	25.5325

h. Timestep 8

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.27 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 8, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi ReLU, *optimizer* Adam dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 4.92E-05% dan nilai RMSE yang dihasilkan adalah 5.42E-05.

Tabel 6.27 Hasil Tuning Parameter Komponen Seasonal Timestep 8

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	adam	ReLU	32	4.92E-05	5.42E-05
100	adam	ReLU	64	0.0007	0.0011
100	adam	ReLU	32	0.0012	0.0020
50	adam	ReLU	64	1.8413	2.7193
100	adam	tanh	32	3.0354	4.0804
50	adam	tanh	32	4.2809	5.3569
100	adam	tanh	64	5.9400	7.1717
50	adam	tanh	64	7.8406	9.9166
100	RMSprop	tanh	32	19.8734	22.5845
100	RMSprop	ReLU	32	19.5424	27.0432

i. Timestep 9

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.28 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 9, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* Adam dan *batch size* sebanyak 64. Nilai MAPE yang dihasilkan adalah 0.000741% dan nilai RMSE yang dihasilkan adalah 0.000833.

Tabel 6.28 Hasil Tuning Parameter Komponen Seasonal Timestep 9

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	adam	ReLU	64	0.0007	0.0008
50	adam	ReLU	32	0.0605	0.0688
100	adam	ReLU	32	0.3981	0.4952
50	adam	tanh	32	2.1249	2.4756
100	adam	tanh	32	3.9276	4.7813
50	adam	ReLU	64	5.4680	8.6984
100	adam	tanh	64	7.4790	9.9729
100	RMSprop	tanh	32	11.7405	13.1142
50	RMSprop	tanh	64	12.7225	16.9525
50	RMSprop	tanh	32	13.9558	16.9953

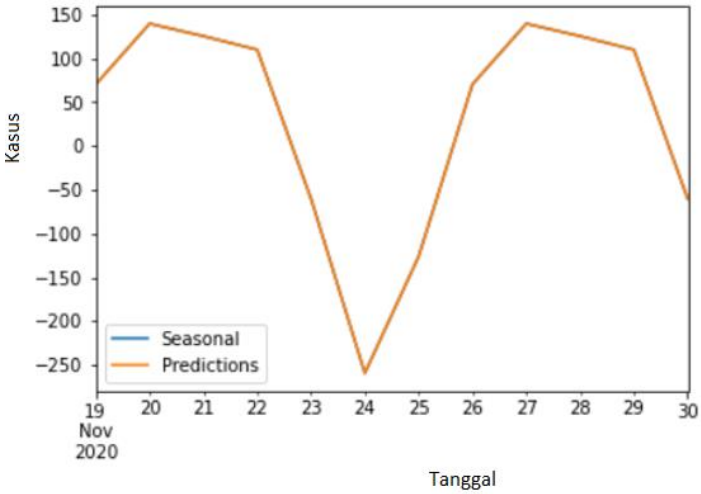
j. Timestep 10

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.29 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 10, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* Adam dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 0.6522% dan nilai RMSE yang dihasilkan adalah 0.7768.

Tabel 6.29 Hasil Tuning Parameter Komponen Seasonal Timestep 10

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	adam	ReLU	32	0.6522	0.7768
100	adam	ReLU	64	0.7171	1.0060
100	adam	tanh	32	2.9364	3.0989
100	adam	tanh	64	3.1668	4.0541
50	adam	tanh	32	4.0170	5.0929
50	adam	tanh	64	4.0538	5.6620
50	adam	ReLU	32	5.2654	6.5965
50	adam	ReLU	64	5.1943	7.9290
50	RMSprop	tanh	32	15.0927	17.2538
100	adam	ReLU	128	14.0850	20.9403

Model LSTM untuk komponen *seasonal* dengan performa terbaik dilihat dari RMSE berada pada *timestep* 8 dengan nilai MAPE 4.92E-05% dan RMSE 5.42E-05. Gambar 6.13 merupakan hasil perbandingan hasil peramalan dengan data aktual dalam bentuk grafik. Hasil peramalan sangat mendekati data aktual sehingga grafik beririsan.



Gambar 6.13 Hasil Perbandingan Peramalan Model LSTM Terhadap Komponen Seasonal Dengan Data Seasonal Aktual

Setelah LSTM untuk komponen seasonal dibangunkan, dilanjutkan dengan pembangunan LSTM untuk komponen residual.

k. Timestep 1

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.30 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 1, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi ReLU, *optimizer* SGD dan *batch size*

sebanyak 32. Nilai MAPE yang dihasilkan adalah 85.1714% dan nilai RMSE yang dihasilkan adalah 450.2969

Tabel 6.30 Hasil Tuning Parameter Komponen Residual Timestep 1

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	SGD	ReLU	32	85.1714	450.2969
100	SGD	ReLU	32	85.1610	450.5583
100	SGD	tanh	32	84.5358	451.0906
50	SGD	tanh	32	84.6211	451.1730
50	RMSprop	tanh	32	99.9747	462.3494
100	RMSprop	tanh	32	101.2208	462.9726
50	RMSprop	ReLU	32	121.7909	465.0193
50	adam	ReLU	32	119.4703	467.4827
50	SGD	ReLU	64	99.2145	469.2192
100	SGD	tanh	64	101.2286	470.1444

l. Timestep 2

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.31 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 2, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* adam dan *batch size* sebanyak 128. Nilai MAPE yang dihasilkan adalah 93.9870% dan nilai RMSE yang dihasilkan adalah 427.1049.

Tabel 6.31 Hasil Tuning Parameter Komponen Residual Timestep 2

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	adam	ReLU	128	93.9870	427.1049
100	RMSprop	ReLU	64	84.4117	435.3199
100	adam	ReLU	64	100.3754	445.8881
50	adam	ReLU	32	103.3364	447.0993
100	RMSprop	ReLU	32	101.9516	450.0644
100	SGD	ReLU	32	87.0688	454.6213
50	SGD	ReLU	32	86.8475	455.2832
100	SGD	tanh	32	90.7965	458.6965
50	SGD	tanh	32	92.5812	459.7218
50	RMSprop	ReLU	32	131.7452	471.7854

m. Timestep 3

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.32 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 3, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi ReLU, *optimizer* adam dan *batch size* sebanyak 128. Nilai MAPE yang dihasilkan adalah 76.3136% dan nilai RMSE yang dihasilkan adalah 415.4689.

Tabel 6.32 Hasil Tuning Parameter Komponen Residual Timestep 3

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	adam	ReLU	128	76.3136	415.4689
50	RMSprop	ReLU	64	76.4096	426.6150
100	RMSprop	ReLU	128	78.7626	428.5203
50	RMSprop	ReLU	32	75.6257	429.8043
50	RMSprop	ReLU	128	80.5086	430.0967
50	SGD	tanh	32	101.6482	432.6119
100	SGD	tanh	32	99.7759	432.7110
100	SGD	tanh	64	99.1417	433.0124
50	SGD	ReLU	32	99.3991	433.3114
50	RMSprop	tanh	128	89.8651	433.3998

n. Timestep 4

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.33 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 4, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi ReLU, *optimizer* RMSprop dan *batch size* sebanyak 128. Nilai MAPE yang dihasilkan adalah 97.1486% dan nilai RMSE yang dihasilkan adalah 424.1065.

Tabel 6.33 Hasil Tuning Parameter Komponen Residual Timestep 4

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	RMSprop	ReLU	128	97.1486	424.1065

100	SGD	tanh	32	97.2022	428.8920
50	SGD	tanh	32	97.4367	430.1092
100	SGD	tanh	64	98.0211	430.7037
50	SGD	ReLU	32	98.9928	431.4616
100	SGD	ReLU	32	98.5702	431.5629
50	SGD	tanh	128	99.0927	431.7787
100	SGD	tanh	128	99.3750	433.9879
100	SGD	ReLU	128	99.7973	434.3816
100	SGD	ReLU	64	99.3017	434.3969

o. Timestep 5

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.34 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 5, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi ReLU, *optimizer* RMSprop dan *batch size* sebanyak 128. Nilai MAPE yang dihasilkan adalah 138.0407% dan nilai RMSE yang dihasilkan adalah 426.2193.

Tabel 6.34 Hasil Tuning Parameter Komponen Residual Timestep 5

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	RMSprop	ReLU	128	138.0407	426.2193
100	SGD	tanh	32	95.9293	428.5711
100	SGD	tanh	64	95.9787	429.4419
50	SGD	tanh	32	98.5139	431.2632
50	SGD	ReLU	32	97.8931	432.6661
100	SGD	ReLU	64	98.9935	433.4806
100	SGD	tanh	128	99.0707	433.6745
100	SGD	ReLU	32	99.1538	433.7969
100	RMSprop	ReLU	128	145.2841	433.8346
50	SGD	ReLU	64	98.8643	434.1029

p. Timestep 6

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.35 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 6, model terbaik

didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi tanh, *optimizer* RMSprop dan *batch size* sebanyak 128. Nilai MAPE yang dihasilkan adalah 144.7323% dan nilai RMSE yang dihasilkan adalah 410.1815.

Tabel 6.35 Hasil Tuning Parameter Komponen Residual Timestep 6

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	RMSprop	tanh	128	144.7323	410.1815
100	adam	ReLU	32	119.7735	414.2871
50	RMSprop	tanh	128	149.8737	414.8153
100	adam	tanh	128	161.1379	416.6019
50	RMSprop	ReLU	128	151.1936	419.1996
50	adam	tanh	128	159.6520	420.3549
50	adam	tanh	64	166.2081	420.6422
100	adam	tanh	64	157.6542	421.0311
50	RMSprop	tanh	64	168.5821	422.6123
50	adam	tanh	32	165.9725	422.7845

q. Timestep 7

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.36 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 7, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi tanh, *optimizer* adam dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 129.4461% dan nilai RMSE yang dihasilkan adalah 391.5688.

Tabel 6.36 Hasil Tuning Parameter Komponen Residual Timestep 7

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	adam	tanh	32	129.4461	391.5688
50	RMSprop	tanh	64	138.1784	400.0728
100	RMSprop	tanh	128	152.7286	402.6002
100	adam	tanh	64	150.5867	403.0140
50	RMSprop	tanh	128	143.0847	404.4278
50	adam	tanh	32	159.6653	408.8129
50	adam	tanh	128	155.2869	409.6298
50	adam	tanh	64	156.4912	410.0423

100	adam	tanh	128	156.3730	413.6529
50	RMSprop	tanh	32	132.0749	423.1031

r. Timestep 8

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.37 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 8, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 100, fungsi aktivasi tanh, *optimizer* SGD dan *batch size* sebanyak 64. Nilai MAPE yang dihasilkan adalah 97.9456% dan nilai RMSE yang dihasilkan adalah 432.5769.

Tabel 6.37 Hasil Tuning Parameter Komponen Residual Timestep 8

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
100	SGD	tanh	64	97.9456	432.5769
50	SGD	ReLU	64	96.4434	433.0996
100	SGD	tanh	128	95.0865	433.2185
100	SGD	ReLU	32	98.1462	433.6134
50	SGD	tanh	128	96.8775	433.7152
100	SGD	ReLU	128	96.4684	434.3610
100	SGD	ReLU	64	95.4652	434.5313
50	SGD	ReLU	128	97.5946	434.7688
100	SGD	tanh	32	98.7812	433.8349
100	adam	ReLU	64	97.9238	490.9401

s. Timestep 9

Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan RMSE yang terendah. Tabel 6.38 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 9, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi ReLU, *optimizer* SGD dan *batch size* sebanyak 128. Nilai MAPE yang dihasilkan adalah 96.8453% dan nilai RMSE yang dihasilkan adalah 433.9982.

Tabel 6.38 Hasil Tuning Parameter Komponen Residual Timestep 9

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	SGD	ReLU	128	96.8453	433.9982
50	SGD	tanh	128	98.6435	432.5520
100	SGD	tanh	32	98.8612	432.6027
100	SGD	tanh	128	98.6207	433.4496
50	SGD	ReLU	32	98.7336	433.4666
100	SGD	tanh	64	99.0009	433.7500
100	SGD	ReLU	128	98.2777	433.7948
50	SGD	ReLU	64	98.0826	434.3238
100	SGD	ReLU	32	99.1969	434.4322
100	SGD	ReLU	64	97.7125	434.7459

t. Timestep 10

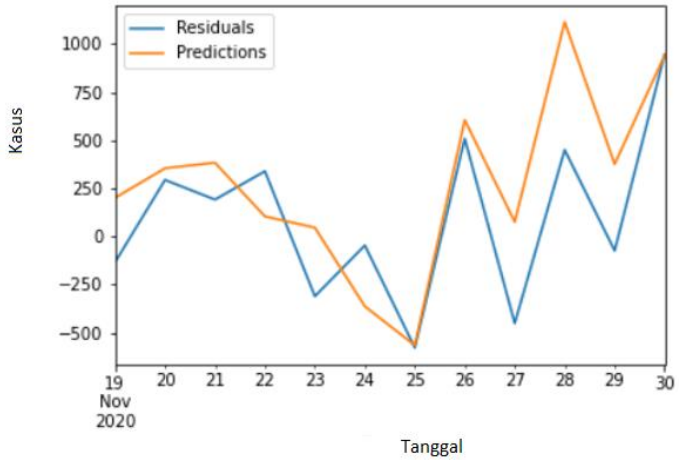
Jumlah unit LSTM, optimizer, activation dan batch size akan diganti udah mencari hasil dengan MAPE yang terendah. Tabel 6.39 merupakan 10 model dengan MAPE dan RMSE yang terendah setelah mencoba 36 kombinasi parameter. Dari ujicoba kombinasi parameter terhadap *timestep* 10, model terbaik didapatkan dari skenario parameter dengan jumlah unit LSTM 50, fungsi aktivasi tanh, *optimizer* SGD dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 100.1711% dan nilai RMSE yang dihasilkan adalah 432.9331.

Tabel 6.39 Hasil Tuning Parameter Komponen Residual Timestep 10

LSTM Unit	Optimizer	Activation	Batch Size	MAPE	RMSE
50	SGD	tanh	32	100.1711	432.9331
100	SGD	tanh	128	99.3561	433.4980
100	SGD	tanh	64	99.4700	433.5981
50	SGD	ReLU	128	98.8724	433.8744
100	SGD	ReLU	64	99.1279	433.9899
100	SGD	ReLU	128	98.7350	434.4868
50	SGD	ReLU	32	98.2539	434.5378
100	SGD	ReLU	32	98.1735	434.5778
50	SGD	ReLU	64	97.7613	434.9266
50	SGD	tanh	128	97.4836	436.0906

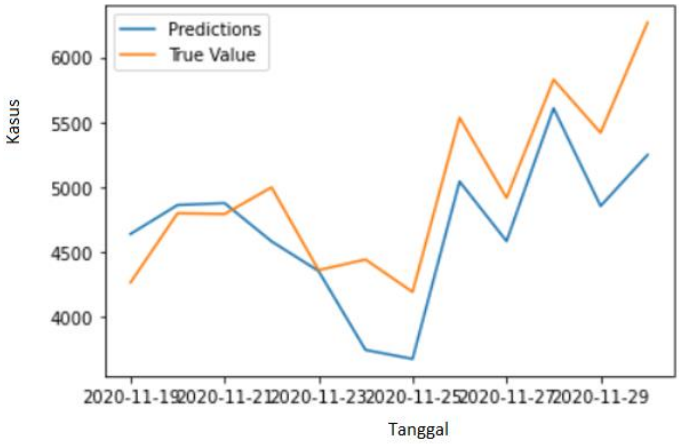
Model LSTM untuk komponen *residual* dengan performa terbaik dilihat dari RMSE berada pada *timestep* 3 dengan nilai MAPE 75.6257% dan RMSE 429.8043. Khusus untuk model LSTM untuk komponen *residual* hasil dari penelitian Emmanuel [13] lebih baik sehingga untuk komponen *residual* akan menggunakan tuning *parameter* yang sama dengan Emmanuel [13].

Parameter yang digunakan untuk komponen *residual* adalah *Timestep* 7 dengan *batch size* 1, LSTM unit 150, fungsi aktivasi ReLU dan *optimizer* adam. Hasil dari tuning tersebut adalah MAPE dengan nilai 177.3900% dan RMSE dengan nilai 337.1864. Gambar 6.14 merupakan hasil perbandingan hasil peramalan dengan data aktual dalam bentuk grafik.



Gambar 6.14 Hasil Perbandingan Peramalan Model LSTM Terhadap Komponen Residual Dengan Data Residual Aktual

Setelah semua komponen dekomposisi ditemukan, ketiga model akan digabungkan untuk membentuk hybrid ARIMA-LSTM. Hasil dari model hybrid ARIMA-LSTM dapat dilihat pada Gambar 6.15.



Gambar 6.15 Hasil Perbandingan Model Hybrid ARIMA-LSTM Dengan Data Aktual

Model Hybrid ARIMA-LSTM menghasilkan MAPE 7.8702% dan RMSE 485.9462.

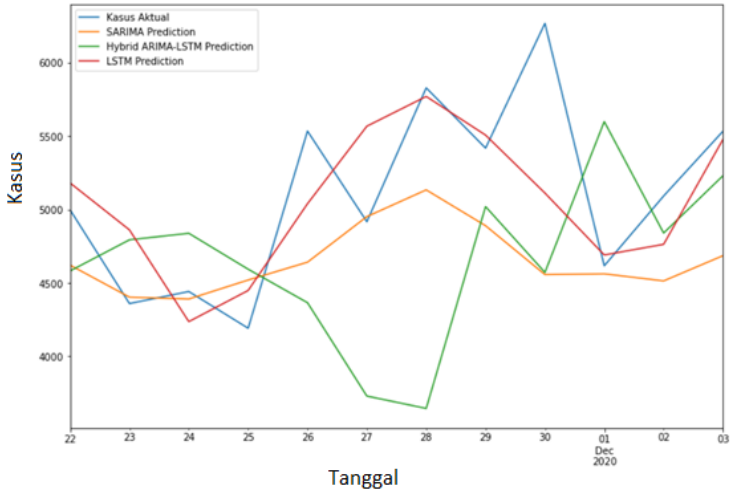
6.4 Hasil Perbandingan Model Peramalan

Hasil Perbandingan model peramalan terbaik dilihat dari RMSE dapat dilihat pada Tabel 6.40. Untuk melihat hasil pembentukkan model SARIMA dapat dilihat pada 6.3.1. Untuk melihat hasil pembentukkan model LSTM dapat dilihat pada 6.3.2. Untuk melihat hasil pembentukkan model Hybrid ARIMA-LSTM 6.3.3.

Tabel 6.40 Perbandingan Model Peramalan

Model	MAPE	RMSE
SARIMA(0,1,1)(2,0,2)[13]	9.2602%	694.5913
LSTM	6.0197%	473.4757
Hybrid ARIMA-LSTM	7.8702%	485.9462

Grafik perbandingan ketiga model dengan data aktual dapat dilihat pada Gambar 6.16.



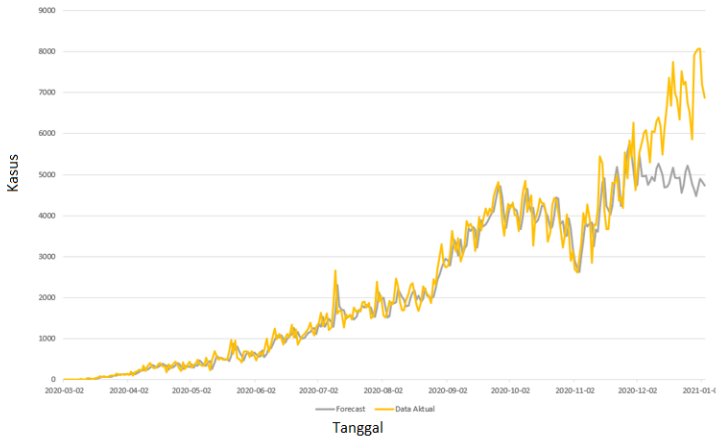
Gambar 6.16 Perbandingan Model Peramalan Dalam Bentuk Grafik

Dari hasil tersebut dapat diketahui bahwa model LSTM merupakan model yang terbaik dilihat dari RMSE dan MAPE. Model LSTM menghasilkan MAPE 6.0197% dan RMSE 473.4757.

6.5 Hasil *Forecasting* Data Masa Depan Dengan Data 2020

Pada bagian ini akan dijelaskan mengenai hasil *forecasting* masa depan dengan menggunakan data dari Maret 2020 hingga Desember 2020. Model LSTM dengan RMSE terendah digunakan untuk melakukan *forecast*. Untuk periode waktu akan dilakukan hingga 30 hari kedepan sehingga dapat diperbandingkan dengan data aktual. Hasil *forecasting* masa depan dapat dilihat pada LAMPIRAN C.

Perbandingan hasil *forecast* dengan data aktual dapat dilihat pada Gambar 6.17.



Gambar 6.17 Perbandingan Hasil Forecast dan Data Aktual Dengan Model LSTM

Grafik *forecast* terdiri atas hasil *training*, *testing* dan *forecast* masa depan.

Hasil perbandingan forecast dan data aktual dalam bentuk MAPE dan RMSE adalah nilai MAPE dengan 37.2746% dan RMSE 1993.9573. Dari grafik ataupun nilai MAPE dan RMSE, hasil *forecast* termasuk akurasi model peramalan tidak akurat. Hal tersebut terjadi karena data yang digunakan dari Maret 2020 hingga Desemeber 2020 masih belum ada perubahan besar yang terjadi. Pada akhir tahun 2020 dan tahun baru 2021 terjadi gelombang satu virus corona di Indonesia terjadi [19] dimana kasus virus corona mencapai 100,000 kasus per hari. Dan pada Juni 2021 Indonesia bermasuk gelombang dua virus corona. Karena adanya gelombang virus corona yang tidak terekspektasi sehingga model peramalan tidak dapat mengikuti pola gelombangnya yang menyebabkan ketidak akuratan model.

6.6 Hasil Uji Skenario Dengan Data 2021

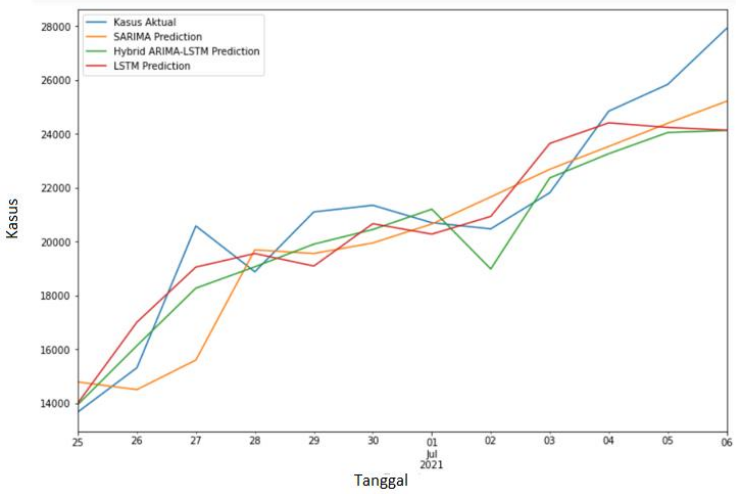
Pada bagian ini akan membahas hasil percobaan untuk menggunakan data terbaru (03 Maret 2020 hingga 06 Juli 2021).

Hasil perbandingan skenario dengan parameter dari setiap model yang terbaik dapat dilihat pada Tabel 6.41.

Tabel 6.41 Hasil Perbandingan Model Terbaik Data Terbaru

Model	MAPE	RMSE
SARIMA(0,1,1)(2,0,2)[13]	6.1266%	1828.3446
LSTM	5.8985%	1617.2049
Hybrid ARIMA-LSTM	5.7135%	1212.3503

Grafik perbandingan ketiga model dengan data aktual dapat dilihat pada Gambar 6.18.



Gambar 6.18 Grafik Perbandingan Model Peramalan

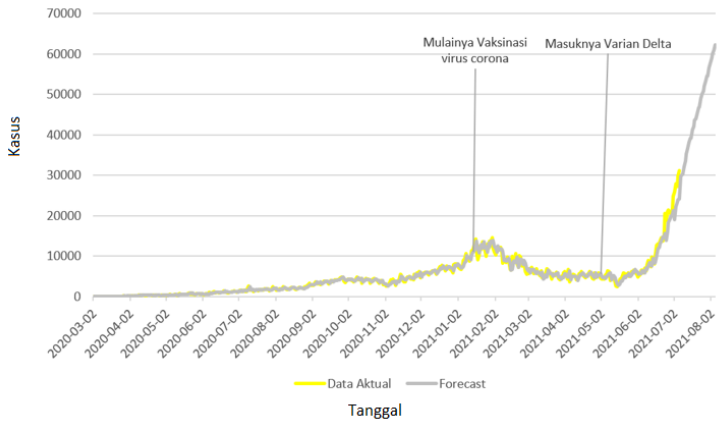
Dari hasil perbandingan model terbaik pada data terbaru dapat diketahui model terbaik adalah model hybrid ARIMA-LSTM dengan MAPE 5.7135% dan RMSE 1212.3503. Hasil dari uji skenario data terbaru menghasilkan hasil yang lebih bagus dibandingkan dengan data dari Maret 2020 hingga Desember 2020. Dari hasil pada diketahui bahwa tidak hanya model Hybrid ARIMA-LSTM yang dapat menghasilkan hasil yang lebih baik melainkan model ARIMA dan LSTM juga dapat menghasilkan hasily ang lebih baik disbanding data Maret 2020

hingga Desember 2020. Dari uji skenario data terbaru dapat diketahui bawha ketika data yang uji lebih banyak dapat menghasilkan hasil yang lebih bagus. Model Hybrid ARIMA-LSTM dapat dikatakan memiliki performa lebih baik ketika data tersedia banyak.

6.7 Hasil *Forecasting* Data Masa Depan Dengan Data 2020-2021

Pada bagian ini akan dijelaskan mengenai hasil *forecasting* masa depan dengan menggunakan data dari Maret 2020 hingga Juli 2020. Model Hybrid ARIMA-LSTM dengan RMSE terendah digunakan untuk melakukan *forecast*. Untuk periode waktu akan dilakukan hingga 30 hari ke depan. Hasil *forecasting* masa depan dapat dilihat LAMPIRAN D.

Hasil *forecasting* masa depan dalam bentuk grafik dapat dilihat pada Gambar 6.19.



Gambar 6.19 Hasil Forecast Masa Depan

Grafik *forecast* terdiri atas hasil *training*, *testing* dan *forecast* masa depan.

6.7.1 Saran Penanganan Virus Corona

Pada bagian ini akan dijelaskan saran mengenai penanganan virus corona apabila prediksi akan terus naik. Berikut adalah beberapa aksi yang dapat diterapakan untuk menangani gelombang kedua virus corona di Indonesia:

1. Disiplin menerapkan 5M.

   Protokol kesehatan untuk mencegah virus corona adalah 5M yang terdiri atas

   - Rutin mencuci tangan
   - Menggunakan masker
   - Menjaga jarak
   - Menjauhi kerumunan
   - Mengurangi mobilitas

   Dengan menjaga disiplin 5M, dapat mengurangi risiko yang dapat terjadi pada gelombang kedua virus corona di Indonesia.

2. Vaksinasi

   Untuk mencegah gelombang kedua virus corona di Indonesia, pemberian vaksin yang ampuh lawan virus corona varian delta dapat membantu. Menurut penelitian yang dilakukan oleh peneliti dari Universitas Oxford, vaksin AstraZeneca dan Pfizer-BioNTech ampuh melawan virus corona varian delta [20]. Sehingga pemberian vaksin AstraZeneca dan Pfizer-BioNTech dapat membantu menangani gelombang kedua virus corona yang terjadi karena varian delta.

3. Penambahan Rumah Sakit Virus Corona

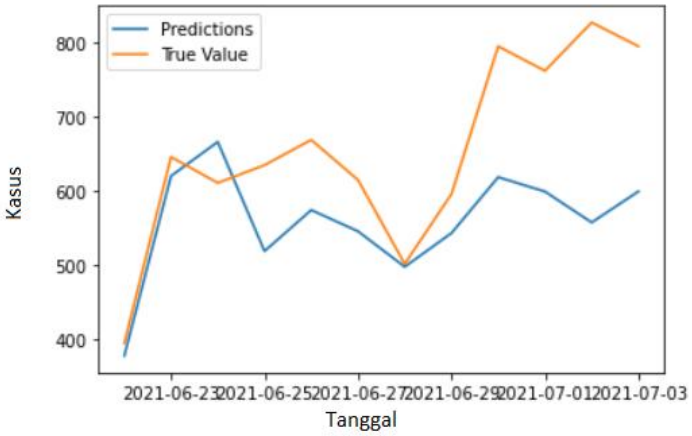
   Cara lain untuk menangani persebaran virus corona pada gelombang kedua adalah penambahan rumah sakit virus corona. Dengan bertambahnya rumah sakit virus corona,

persebaran yang terjadi karena kurangnya rumah sakit dapat ditangani.

6.8 Hasil Uji Skenario Dengan Data Korea Selatan

Pada bagian ini akan dijelaskan mengenai hasil percobaan peramalan jumlah kasus virus corona di Korea Selatan dengan menggunakan model Hybrid ARIMA-LSTM dengan tuning parameter dari subbab 6.3.3.

Hasil dari permalan dengan menggunakan data Korea Selatan dapat dilihat pada Gambar 6.20.



Gambar 6.20 Percobaan Model Hybrid ARIMA-LSTM Pada Data Korea Selatan

Dari hasil percobaan model Hybrid ARIMA-LSTM pada data Korea Selatan dapat diketahui model Hybrid ARIMA-LSTN menghasilkan nilai MAPE 14.2785% dan RMSE 129.7555. Menurut standar penilaian akurasi MAPE yang dapat dilihat pada Tabel 2.2, performa model Hybrid ARIMA-LSTM pada data Korea Selatan masih termasuk peramalan yang baik. Walaupun hasil peramalan termasuk peramalan yang baik, ketika dibandingkan dengan peramalan hybrid ARIMA-LSTM untuk data Indonesia perbedaan akurasi peramalan signifikan

berbeda sehingga pada penelitian selanjutnya dapat mencoba *tuning parameter* yang *optimal*.

6.9 Rangkuman Hasil dan Pembahasan

Hasil dan pembahasan dari penelitian peramalan jumlah kasus virus corona di Indonesia dengan metode ARIMA, LSTM dan hybrid ARIMA-LSTM dapat disimpulkan dalam beberapa point berikut:

1. Pada uji skenario data Maret 2020 hingga Desember 2020, model LSTM memiliki performa lebih baik daripada hybrid ARIMA-LSTM dan ARIMA dengan nilai MAPE sebesar 6.0197% dan nilai RMSE sebesar 473.5747. *Tuning Parameter* untuk model LSTM tersebut berada pada timestep 9 dengan *tuning parameter LSTM unit* 100, *Optimizer* RMSprop, fungsi aktivasi ReLU dan *batch size* 32. (Ditunjukkan pada Tabel 6.40).
2. Pada hasil *forecasting* masa depan, hasil dari model LSTM dengan menggunakan data Maret 2020 hingga Desember 2020 menunjukkan perbedaan signifikan dibandingkan dengan data aktual karena kondisi tidak terduga yaitu peningkatan kasus pada gelombang 1 dan gelombang 2 yang terjadi pada masa akhir liburan tahun dan pada Juni 2021 dengan masuknya varian virus corona baru (delta) (Ditunjukkan pada Gambar 6.17).
3. Pada uji skenario data Maret 2020 hingga Juli 2021 menunjukkan bahwa hybrid ARIMA-LSTM memiliki performa lebih baik dibandingkan dengan model ARIMA dan LSTM dengan menghasilkan nilai MAPE dan RMSE yang terkecil dengan nilai MAPE 5.7135% dan nilai RMSE 1212.3503. *Tuning Parameter* untuk model Hybrid ARIMA-LSTM tersebut adalah ARIMA(2,1,5), LSTM untuk *seasonal* pada timestep 8 dengan *tuning parameter LSTM unit* 50, *Optimizer* adam, fungsi aktivasi ReLU dan *batch size* 32 dan LSTM untuk *residual* pada timestep 7 dengan *tuning parameter LSTM*

unit 150, *Optimizer* adam, fungsi aktivasi ReLU dan *batch size* 1. (Ditunjukkan pada Tabel 6.41)

4. Model hybrid dengan uji skenario data Maret 2020 hingga Juli 2021 memiliki performa lebih baik dibandingkan dengan data Maret 2020 hingga Desember 2020. Sehingga dapat dikatakan bahwa lebih banyak data yang tersedia, model hybrid ARIMA-LSTM bisa memberikan performa lebih baik. (Ditunjukkan Pada Tabel 6.41)
5. Pada peramalan forecasting masa depan metode hybrid ARIMA-LSTM dengan data periode Maret 2020 hingga Juli 2021, terdapat tren kenaikan mengikuti gelombang kedua virus corona yaitu setelah masa libur idul fitri. (Ditunjukkan pada Gambar 6.19)
6. Kekurangan dari model LSTM yang digunakan untuk *forecasting* adalah model LSTM tersebut tidak dapat memprediksikan kapan akan terjadinya penurunan drastis atau akhirnya virus corona di Indonesia dibandingkan dengan model matematis lain (Richard's Curve model pada penelitian Nuraini [21]) yang dapat memprediksikan akhirnya virus corona.

BAB VII.
KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan kesimpulan dari keseluruhan proses dan hasil penelitian tugas akhir ini serta saran yang dapat diberikan untuk penelitian selanjutnya supaya dapat dikembangkan lebih baik ke depan.

7.1 Kesimpulan

Kesimpulan yang didapatkan dari penelitian tugas akhir ini adalah sebagai berikut:

1. Hasil peramalan model LSTM untuk kasus virus corona di Indonesia untuk data periode Maret 2020 hingga Desember 2020 menunjukkan model terbaik yang didapatkan adalah pada timestep 9 dengan jumlah unit LSTM 100, fungsi aktivasi ReLU, *optimizer* RMSprop dan *batch size* sebanyak 32. Nilai MAPE yang dihasilkan adalah 6.0197% dan nilai RMSE yang dihasilkan adalah 473.5747. *Tuning Parameter* untuk model Hybrid ARIMA-LSTM tersebut adalah ARIMA(2,1,5), LSTM untuk *seasonal* pada timestep 8 dengan *tuning parameter LSTM unit* 50, *Optimizer* adam, fungsi aktivasi ReLU dan *batch size* 32 dan LSTM untuk *residual* pada timestep 7 dengan *tuning parameter LSTM unit* 150, *Optimizer* adam, fungsi aktivasi ReLU dan *batch size* 1.
2. Hasil peramalan model hybrid ARIMA-LSTM pada uji skenario data Maret 2020 hingga Juli 2021 dapat menghasilkan nilai MAPE dan RMSE yang terkecil dengan nilai MAPE 5.7135% dan nilai RMSE 1212.3503.
3. Pada peramalan forecasting masa depan metode hybrid ARIMA-LSTM dengan data periode Maret 2020 hingga Juli 2021, terdapat tren kenaikan mengikuti gelombang kedua virus corona yaitu setelah masa libur idul fitri.

7.2 Saran

Berdasarkan penelitian tugas akhir ini, dapat memberikan saran untuk pengembangan penelitian selanjutnya sebagai berikut:

1. Pada penelitian selanjutnya dapat menggunakan hybrid ARIMA-LSTM dengan data yang lebih banyak.
2. Mempertimbangkan adanya faktor seperti gelombang virus corona yang tidak terduga dan adanya jenis varian virus corona yang baru yang dapat bersebar lebih cepat. Juga dapat mempertimbangkan faktor yang menurunkan virus corona, seperti adanya vaksin virus corona.
3. Mengeksplorasi penggunaan parameter yang lain dengan melakukan tuning parameter untuk model LSTM dan model hybrid ARIMA-LSTM.

DAFTAR PUSTAKA

[1] World Health Organization, "Coronavirus," World Health Organization, [Online]. Available: https://www.who.int/health-topics/coronavirus#tab=tab_1. [Diakses 31 October 2020].

[2] A. K. Sahai, N. Rath, V. Sood dan M. P. Singh, "ARIMA modelling & forecasting of COVID-19 in top five affected countries," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews,* vol. 14, no. 5, pp. 1419-1427, 2020.

[3] World Health Organization, "Coronavirus disease (COVID-19): Similarities and differences with influenza," World Health Organization, 17 March 2020. [Online]. Available: https://www.who.int/news-room/q-a-detail/coronavirus-disease-covid-19-similarities-and-differences-with-influenza. [Diakses 19 November 2020].

[4] V. Gitiyarko, "Upaya dan Kebijakan Pemerintah Indonesia Menangani Pandemi Covid-19," KOMPASPEDIA, 2020.

[5] H. Gusman dan I. L. A. Bhaskara, "Bagaimana COVID-19 Mengubah Ekonomi Indonesia & ASEAN?," tirto, 2020.

[6] P. D. Pakan, "PERAMALAN KASUS POSITIF COVID 19 DI INDONESIA MENGGUNAKAN LSTM," *JURNAL ILMIAH FLASH,* vol. 6, pp. 12-15, 2020.

[7] F. Fadly dan E. Sari, "An Approach to Measure the Death Impact of Covid-19 in Jakarta using Autoregressive Integrated Moving Average (ARIMA)," *Unnes Journal of Public Health,* vol. 9, pp. 108-116, 2020.

[8] S. Ketu dan P. K. Mishra, "A Hybrid Deep Learning Model for COVID-19 Prediction and Current Status of Clinical Trials Worldwide," *Tech Science Press,* vol. 66, no. 2, 2020.

[9] F. Jiang, Z. Zhao dan X. Shao, “Time series analysis of COVID-19 infection curve: A change-point perspective,” *Journal of Econometrics,* 2020.

[10] D. C. Montgomery, M. Kulahci dan C. L. Jennings, Introduction to time series analysis and forecasting, New Jersey: Wiley, 2016.

[11] R. J. Hyndman dan G. Athanasopoulos, Forecasting: principles and practice, 2nd edition, Melbourne: OTexts, 2018.

[12] T. Hill, L. Marquez, M. O'Connor dan W. Remus, “Artificial neural network models for forecasting and decision,” *International Journal of Forecasting,* vol. 10, pp. 5-15, 1994.

[13] D. Emmanuel, L. Albert, J. Marethia dan H. Novita, “Forecasting Indonesia Exports using a Hybrid Model ARIMA-LSTM,” dalam *5th International Conference on Computer Science and Computational Intelligence 2020,* 2020.

[14] Swamidass dan M. Paul, Encyclopedia of Production and Manufacturing Management, Boston: Springer, 2000.

[15] E. A.Mohammed, ChristopherNaugler dan B. H.Far, “Chapter 32 - Emerging Business Intelligence Framework for a Clinical Laboratory Through Big Data Analytics,” dalam *Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology*, Emerging Trends in Computer Science and Applied Computing, 2015, pp. 577-602.

[16] Keras, “Keras,” [Online]. Available: keras.io. [Diakses 6 July 2021].

[17] B. Woo Seok, H. Sin Woo, J. Gunhee, K. Jae Won, K. Sarang, L. Sujie, P. Hye Eun dan B. Jea-Hyun, “Is coronavirus disease (COVID-19) seasonal? A critical analysis of empirical and epidemiological studies at global and local scales,” *Environmental Research* , vol. 196, 2021.

[18] R. Nau, “Statistical forecasting: notes on regression and time series analysis,” [Online]. Available: https://people.duke.edu/~rnau/411home.htm. [Diakses 5 August 2021].

[19] H. K. N. Sumartiningtyas, “Kompas,” Kompas, 2 1 2021. [Online]. Available: https://www.kompas.com/sains/read/2021/01/02/100200723/kasus-covid-19-indonesia-bisa-melonjak-tapi-gelombang-1-belum-masuki?page=all. [Diakses 8 July 2021].

[20] M. Syahrial, “2 Vaksin Ini Disebut Ampuh Lawan Covid-19 Varian Delta dan Kappa,” Kompas.com, 2021.

[21] N. Nuning, K. Kamal dan A. Mochamad, “Modeling Simulation of COVID-19 in Indonesia based on Early Endemic Data,” *COMMUN. BIOMATH*, vol. 3, no. 1, pp. 1-8, 2020.

[22] A. S. TEMÜR, M. AKGÜN dan G. TEMÜR, “PREDICTING HOUSING SALES IN TURKEY USING ARIMA,,” *Journal of Business Economics and Management*, vol. 20, no. 5, pp. 920-938, 2019.

BIODATA PENULIS

Penulis bernama lengkap Lee Ye Chan dan biasa dipanggil Lee. Penulis merupakan anak kedua dari tiga bersaudara yang lahir di Busan, Korea Selatan, 20 Oktober 1998. Penulis telah menempuh beberapa pendidikan formal sebelumnya antara lain SD International School of Batam, SMP Sekolah Pelita Bangsa Lampung, dan SMA Sekolah Pelita Bangsa Lampung.

Pada tahun 2017 pasca kelulusan SMA, penulis melanjutkan pendidikan di Program Studi Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas — Institut Teknologi Sepuluh Nopember Surabaya melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) dan terdaftar sebagai mahasiswa dengan NRP 05211740000094.

Pada akhir masa perkuliahan di jurusan Sistem Informasi ITS, penulis memilih untuk mengerjakan tugas akhir di Laboratorium Rekayasa dan Intelegensi Bisnis (RDIB) dengan topik tugas akhir peramalan. Penulis dapat dihubungi melalui email di lyc98yc@gmail.com.

LAMPIRAN A

Data jumlah kasus virus corona yang terjadi di Indonesia yang digunakan sebagai data input model peramalan.

Tanggal	Kasus
2020-03-02	6
2020-03-03	0
2020-03-04	0
2020-03-05	0
2020-03-06	2
2020-03-07	0
2020-03-08	2
2020-03-09	0
2020-03-10	13
2020-03-11	8
2020-03-12	7
2020-03-13	35
2020-03-14	27
2020-03-15	21
2020-03-16	17
2020-03-17	38
2020-03-18	55
2020-03-19	78
2020-03-20	60
2020-03-21	81
2020-03-22	64
2020-03-23	65
2020-03-24	106
2020-03-25	105

Tanggal	Kasus
2020-03-26	103
2020-03-27	153
2020-03-28	109
2020-03-29	130
2020-03-30	129
2020-03-31	114
2020-04-01	149
2020-04-02	113
2020-04-03	196
2020-04-04	106
2020-04-05	181
2020-04-06	218
2020-04-07	247
2020-04-08	218
2020-04-09	337
2020-04-10	219
2020-04-11	330
2020-04-12	399
2020-04-13	316
2020-04-14	282
2020-04-15	297
2020-04-16	380
2020-04-17	407
2020-04-18	325

Tanggal	Kasus
2020-04-19	327
2020-04-20	185
2020-04-21	375
2020-04-22	283
2020-04-23	357
2020-04-24	436
2020-04-25	396
2020-04-26	275
2020-04-27	214
2020-04-28	415
2020-04-29	260
2020-04-30	347
2020-05-01	433
2020-05-02	292
2020-05-03	349
2020-05-04	395
2020-05-05	484
2020-05-06	367
2020-05-07	338
2020-05-08	336
2020-05-09	533
2020-05-10	387
2020-05-11	233
2020-05-12	484
2020-05-13	689
2020-05-14	568
2020-05-15	490
2020-05-16	529
2020-05-17	489
2020-05-18	496

Tanggal	Kasus
2020-05-19	486
2020-05-20	693
2020-05-21	973
2020-05-22	634
2020-05-23	949
2020-05-24	526
2020-05-25	479
2020-05-26	415
2020-05-27	686
2020-05-28	687
2020-05-29	678
2020-05-30	557
2020-05-31	700
2020-06-01	583
2020-06-02	467
2020-06-03	609
2020-06-04	684
2020-06-05	585
2020-06-06	703
2020-06-07	993
2020-06-08	672
2020-06-09	847
2020-06-10	1043
2020-06-11	1240
2020-06-12	979
2020-06-13	1111
2020-06-14	1014
2020-06-15	857
2020-06-16	1017
2020-06-17	1106

Tanggal	Kasus
2020-06-18	1031
2020-06-19	1331
2020-06-20	1041
2020-06-21	1226
2020-06-22	862
2020-06-23	954
2020-06-24	1051
2020-06-25	1113
2020-06-26	1178
2020-06-27	1240
2020-06-28	1385
2020-06-29	1198
2020-06-30	1082
2020-07-01	1293
2020-07-02	1385
2020-07-03	1624
2020-07-04	1301
2020-07-05	1447
2020-07-06	1607
2020-07-07	1209
2020-07-08	1268
2020-07-09	1853
2020-07-10	2657
2020-07-11	1611
2020-07-12	1671
2020-07-13	1681
2020-07-14	1282
2020-07-15	1591
2020-07-16	1522
2020-07-17	1574

Tanggal	Kasus
2020-07-18	1462
2020-07-19	1752
2020-07-20	1639
2020-07-21	1693
2020-07-22	1655
2020-07-23	1882
2020-07-24	1906
2020-07-25	1761
2020-07-26	1868
2020-07-27	1492
2020-07-28	1525
2020-07-29	1748
2020-07-30	2381
2020-07-31	1904
2020-08-01	2040
2020-08-02	1560
2020-08-03	1519
2020-08-04	1679
2020-08-05	1922
2020-08-06	1815
2020-08-07	1882
2020-08-08	2473
2020-08-09	2277
2020-08-10	1893
2020-08-11	1687
2020-08-12	1693
2020-08-13	1942
2020-08-14	2098
2020-08-15	2307
2020-08-16	2345

Tanggal	Kasus
2020-08-17	2081
2020-08-18	1821
2020-08-19	1673
2020-08-20	1902
2020-08-21	2266
2020-08-22	2197
2020-08-23	2090
2020-08-24	2037
2020-08-25	1877
2020-08-26	2447
2020-08-27	2306
2020-08-28	2719
2020-08-29	3003
2020-08-30	3308
2020-08-31	2858
2020-09-01	2743
2020-09-02	2775
2020-09-03	3075
2020-09-04	3622
2020-09-05	3269
2020-09-06	3128
2020-09-07	3444
2020-09-08	2880
2020-09-09	3046
2020-09-10	3307
2020-09-11	3861
2020-09-12	3737
2020-09-13	3806
2020-09-14	3636
2020-09-15	3141

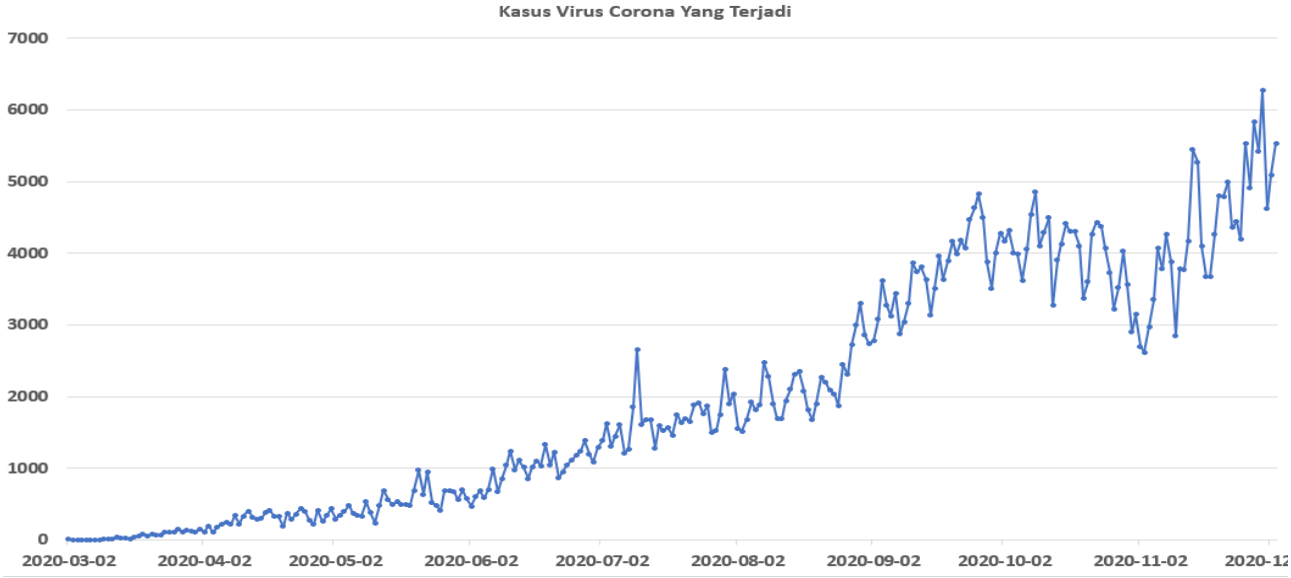
Tanggal	Kasus
2020-09-16	3507
2020-09-17	3963
2020-09-18	3635
2020-09-19	3891
2020-09-20	4168
2020-09-21	3989
2020-09-22	4176
2020-09-23	4071
2020-09-24	4465
2020-09-25	4634
2020-09-26	4823
2020-09-27	4494
2020-09-28	3874
2020-09-29	3509
2020-09-30	4002
2020-10-01	4284
2020-10-02	4174
2020-10-03	4317
2020-10-04	4007
2020-10-05	3992
2020-10-06	3622
2020-10-07	4056
2020-10-08	4538
2020-10-09	4850
2020-10-10	4094
2020-10-11	4294
2020-10-12	4497
2020-10-13	3267
2020-10-14	3906
2020-10-15	4127

Tanggal	Kasus
2020-10-16	4411
2020-10-17	4301
2020-10-18	4301
2020-10-19	4105
2020-10-20	3373
2020-10-21	3602
2020-10-22	4267
2020-10-23	4432
2020-10-24	4369
2020-10-25	4070
2020-10-26	3732
2020-10-27	3222
2020-10-28	3520
2020-10-29	4029
2020-10-30	3565
2020-10-31	2897
2020-11-01	3143
2020-11-02	2696
2020-11-03	2618
2020-11-04	2973
2020-11-05	3356
2020-11-06	4065
2020-11-07	3778
2020-11-08	4262
2020-11-09	3880

Tanggal	Kasus
2020-11-10	2853
2020-11-11	3779
2020-11-12	3770
2020-11-13	4173
2020-11-14	5444
2020-11-15	5272
2020-11-16	4106
2020-11-17	3671
2020-11-18	3671
2020-11-19	4265
2020-11-20	4798
2020-11-21	4792
2020-11-22	4998
2020-11-23	4360
2020-11-24	4442
2020-11-25	4192
2020-11-26	5534
2020-11-27	4917
2020-11-28	5828
2020-11-29	5418
2020-11-30	6267
2020-12-01	4617
2020-12-02	5092
2020-12-03	5533

LAMPIRAN B

Grafik kasus virus corona yang terjadi di Indonesia.



LAMPIRAN C

Hasil *forecasting* hingga Januari 2021 dengan model terbaik (LSTM).

Tanggal	Forecast
2020-12-04	6033.022
2020-12-05	5954.47
2020-12-06	5440.107
2020-12-07	4857.217
2020-12-08	4627.827
2020-12-09	4923.866
2020-12-10	5617.483
2020-12-11	6107.866
2020-12-12	5875.101
2020-12-13	5138.401
2020-12-14	4573.551
2020-12-15	4553.854
2020-12-16	5108.396
2020-12-17	5857.432
2020-12-18	6146.642
2020-12-19	5710.619

Tanggal	Forecast
2020-12-20	4907.76
2020-12-21	4433.06
2020-12-22	4644.016
2020-12-23	5430.322
2020-12-24	6151.503
2020-12-25	6197.07
2020-12-26	5516.499
2020-12-27	4688.714
2020-12-28	4401.931
2020-12-29	4868.048
2020-12-30	5817.341
2020-12-31	6408.57
2021-01-01	6131.969
2021-01-02	5249.919
2021-01-03	4497.689

LAMPIRAN D

Hasil *forecasting* hingga Augustus 2021 dengan model terbaik (Hybrid ARIMA-LSTM).

Tanggal	Forecast
2021-07-07	29730.96
2021-07-08	29913.53
2021-07-09	30258.39
2021-07-10	32235.18
2021-07-11	33628.06
2021-07-12	35333.29
2021-07-13	36562.99
2021-07-14	38144.17
2021-07-15	39118.94
2021-07-16	39150.67
2021-07-17	41322.8
2021-07-18	42162.58
2021-07-19	43580.87
2021-07-20	43950.11
2021-07-21	45366.7
2021-07-22	46728.69

Tanggal	Forecast
2021-07-23	46941.79
2021-07-24	48950.88
2021-07-25	50097.73
2021-07-26	50853.06
2021-07-27	52043.56
2021-07-28	53174.53
2021-07-29	54506.22
2021-07-30	54608.22
2021-07-31	56578.4
2021-08-01	57891.24
2021-08-02	59061.31
2021-08-03	59991.57
2021-08-04	60847.63
2021-08-05	62255.45
2021-07-07	29730.96

LAMPIRAN E

Hasil percobaan ARIMA untuk komponen trend dengan data Maret 2020 hingga Desember 2020.

	MAPE	RMSE
ARIMA(2,1,5)	10.0217	625.5787
ARIMA(5,1,4)	10.0345	633.5224
ARIMA(5,1,3)	10.3736	653.1211
ARIMA(5,1,2)	10.4394	656.3190
ARIMA(4,1,5)	10.4957	658.2360
ARIMA(0,1,4)	10.5535	661.5275
ARIMA(0,1,0)	10.5770	662.2197
ARIMA(3,1,5)	10.6162	663.0915
ARIMA(0,1,5)	10.6203	666.2168
ARIMA(5,1,1)	10.7162	667.0361
ARIMA(3,1,2)	10.7210	666.0249
ARIMA(5,1,0)	10.7228	667.3036
ARIMA(0,1,1)	10.7396	660.4955
ARIMA(3,1,3)	10.9472	679.8079
ARIMA(2,1,4)	10.9635	677.0921
ARIMA(4,1,4)	11.0035	680.1945
ARIMA(4,1,2)	11.0434	684.4347
ARIMA(3,1,1)	11.1080	687.9432
ARIMA(4,1,1)	11.1734	690.6174
ARIMA(4,1,3)	11.1752	690.7630
ARIMA(3,1,0)	11.1854	692.7278
ARIMA(2,1,2)	11.1937	691.5018
ARIMA(3,1,4)	11.2010	686.0710
ARIMA(0,1,2)	11.2853	684.0789
ARIMA(5,1,5)	11.3037	685.5568
ARIMA(1,1,5)	11.3074	689.7123

ARIMA(4,1,0)	11.3234	698.7144
ARIMA(1,1,2)	11.3438	698.4903
ARIMA(1,1,3)	11.3584	702.7907
ARIMA(2,1,3)	11.3799	683.4735
ARIMA(2,1,1)	11.3935	701.0187
ARIMA(0,1,3)	11.4393	695.0632
ARIMA(1,1,1)	11.4469	703.7304
ARIMA(1,1,4)	11.5036	701.4470
ARIMA(2,1,0)	11.8197	720.0103
ARIMA(1,1,0)	12.0539	725.6434