

**PENERAPAN METODE *LONG SHORT TERM MEMORY* DALAM
MEMPREDIKSI JUMLAH KASUS COVID-19**

SKRIPSI

**Oleh :
RIZKI TRISNA RULLY ABADI
NIM. 17650082**



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021**

**PENERAPAN METODE *LONG SHORT TERM MEMORY* DALAM
MEMPREDIKSI JUMLAH KASUS COVID-19**

SKRIPSI

**Diajukan kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :
RIZKI TRISNA RULLY ABADI
NIM. 17650082**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021**

LEMBAR PERSETUJUAN

PENERAPAN METODE LONG SHORT TERM MEMORY DALAM MEMPREDIKSI JUMLAH KASUS COVID-19

SKRIPSI

Oleh:
RIZKI TRISNA RULLY ABADI
NIM. 17650082

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal : 14 Desember 2021

Dosen Pembimbing I



Ajib Hanani, M.T.
NIDT. 19840731 20160801 1 076

Dosen Pembimbing II



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

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



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

LEMBAR PENGESAHAN

PENERAPAN METODE *LONG SHORT TERM MEMORY* DALAM MEMPREDIKSI JUMLAH KASUS COVID-19

SKRIPSI

Oleh :

RIZKI TRISNA RULLY ABADI
NIM. 17650082

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 14 Desember 2021

Susunan Dewan Penguji :

Penguji Utama	:	<u>Dr. Cahyo Crysdian</u> NIP. 19740424 200901 1 008
Ketua Penguji	:	<u>Fatchurrohman, M.Kom</u> NIP. 19700731 200501 1 002
Sekretaris Penguji	:	<u>Ajib Hanani, M.T.</u> NIDT. 19840731 20160801 1 076
Anggota Penguji	:	<u>Irwan Budi Santoso, M.Kom</u> NIP. 19770103 201101 1 004

()


()

()

()

Mengetahui dan Mengesahkan,
Ketua Jurusan Teknik Informatika




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

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Rizki Trisna Rully Abadi

NIM : 17650082

Fakultas / Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Skripsi : Penerapan Metode *Long Short Term Memory* Dalam Memprediksi Jumlah Kasus Covid-19.

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

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

Malang, 14 Desember 2021

Yang membuat pernyataan,



Rizki Trisna Rully Abadi

NIM.17650082

HALAMAN MOTTO

Siapapun yang diberi hak untuk hidup
memiliki kewajiban untuk tetap hidup.

(Kutipan dialog dalam Film Inseparable Bros (2019))

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Karya ilmiah ini penulis persembahkan kepada kedua orang tua dan kakak saya. Tidaklah penulis dapat berjumpa dengan kehidupan yang sekarang jika tidak dengan diiringi kasih sayang mereka. Semoga Allah berikan yang terbaik untuk mereka,

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Puji syukur penulis limpahkan ke hadirat Allah SWT atas limpahan rahmt dan karunia-Nya sehingga penulis dapat menyelesaikan studi dan skripsi. Shalawat serta salam kepada Nabi Muhammad SAW, yang telah memimpin umat manusia dari jalan yang gelap menuju jalan yang terang benderang.

Penulis mendapatkan banyak sekali dukungan dari banyak pihak selama proses pengerjaan studi dan lebih khususnya pada proses penyusunan karya ilmiah skripsi ini. Penulis haturkan banyak terimakasih kepada:

1. Kedua orang tua penulis dan kakak yang telah memberikan dukungan baik secara moral hingga material sehingga penulis dapat menyelesaikan skripsi ini.
2. Prof. DR. H. M. Zainuddin, MA selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta jajarannya.
3. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta jajarannya.
4. Dr. Fachrul Kurniawan ST., M.MT ., IPM, selaku Kepala Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta jajarannya.
5. Ajib Hanani, M.T, selaku Dosen Pembimbing I yang telah memberikan ilmu dalam proses pengarahan hingga penyelesaian skripsi ini.
6. Irwan Budi Santoso, M.Kom, selaku Dosen Pembimbing II yang telah memberikan ilmu dalam proses pengarahan hingga penyelesaian skripsi ini.

7. Teman-teman di Jurusan Teknik Informatika angkatan 2017 “Unocore” yang telah membantu untuk bertukar pikiran dan telah memberikan dukungannya kepada penulis dalam proses penyelesaian skripsi ini.
8. Teman-teman di komunitas Fun Java dan Ontaki yang telah memberikan ilmu serta ruang diskusi yang nyaman bagi penulis.
9. Seluruh keluarga besar Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
10. Seluruh teman-teman yang telah memberikan support dan semangat kepada penulis.

Penulis menyadari bahwa dalam penyusunan karya ilmiah skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini dapat memberikan manfaat termasuk penulis sendiri.

Malang, 14 Desember 2021

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
ABSTRAK	xiii
ABSTRACT.....	xiv
الملخص	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah.....	5
1.3 Tujuan Penelitian.....	5
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian.....	5
BAB II STUDI PUSTAKA.....	7
2.1 Kasus Covid-19	7
2.2 Long Short Term Memory (LSTM)	9
2.3 Mean Squared Error (MSE).....	15
2.4 Mean Absolute Error (MAE)	15
BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM.....	17
3.1 Pengumpulan Data	17
3.2 Desain Sistem	20
3.3 <i>Preprocessing</i>	20
3.4 Pembuatan Model LSTM	21
3.4.1 Arsitektur Jaringan LSTM	22
3.4.1.1 Normalisasi Data	22

3.4.1.2	Long Short Term Memory (LSTM)	23
3.4.1.3	Sigmoid	26
3.4.2	<i>Training</i> Jaringan LSTM	27
3.5	Implementasi Sistem	31
3.5.1	Spesifikasi Hardware	31
BAB IV	UJI COBA DAN PEMBAHASAN	33
4.1	Skenario Uji.....	33
4.2	Hasil Uji Coba	38
4.3	Pembahasan	48
BAB V	KESIMPULAN DAN SARAN.....	53
5.1	Kesimpulan.....	53
5.2	Saran	54
DAFTAR PUSTAKA		

DAFTAR GAMBAR

Gambar 2.1 Struktur sel pada LSTM	10
Gambar 2.2 Struktur sel pada LSTM	10
Gambar 2.3 <i>Input gate</i> pada LSTM	11
Gambar 2.4 Operasi pada hasil input gate dan perkalian C_t dan i_t pada sel LSTM	13
Gambar 2.5 Output gate pada sel LSTM	13
Gambar 3.1 Desain sistem	20
Gambar 3.2 Arsitektur jaringan.....	22
Gambar 3.3 <i>Flowchart</i> Pembuatan Model Prediksi Menggunakan LSTM	30
Gambar 4.1 Pengukuran MSE di setiap model training.....	42
Gambar 4.2 Pengukuran MAE di setiap model <i>training</i>	42
Gambar 4.3 Grafik hasil perbandingan hasil prediksi dan data asli kasus harian covid-19	43
Gambar 4.4 Grafik MSE pada proses prediksi	49

DAFTAR TABEL

Tabel 3.1 Sampel dataset kawalcovid19	18
Tabel 3.2 Tabel Atribut yang Digunakan dalam Data Kasus Covid-19 dari Kawalcovid19.id.....	21
Tabel 4.1 Sampel <i>groundtruth</i>	33
Tabel 4.2 Pembagian data dalam pengujian.....	34
Tabel 4.3 Skenario perubahan parameter <i>epoch</i> dan <i>batch size</i> pada proses <i>training</i>	36
Tabel 4.4 Perbandingan grafik pengukuran MSE dan MAE tiap-tiap model.....	39
Tabel 4.5 Tabel parameter yang digunakan dalam tahap uji coba menggunakan data uji	43
Tabel 4.6 Hasil Prediksi ditampilkan dalam tampilan tabel	44
Tabel 4.7 Penghitungan MSE dan MAE dengan data normalisasi	45
Tabel 4.8 Hasil uji coba prediksi pada data uji normalisasi.....	46
Tabel 4.9 Penghitungan MSE dan MAE dengan data denormalisasi	47
Tabel 4.10 Hasil uji coba prediksi pada data denormalisasi	48

ABSTRAK

Trisna, Rizki. 2021. Penerapan **Metode Long Short Term Memory Dalam Memprediksi Jumlah Kasus Covid-19**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Ajib Hanani, M.T, (II) Irwan Budi Santoso, M.Kom

Kata Kunci: *LSTM, Prediksi, Kasus Harian COVID-19.*

Tingkat kasus positif harian COVID-19 di Indonesia mengalami kenaikan dan penurunan yang sangat bervariasi. Hal tersebut mempengaruhi penyedia layanan kesehatan dan pembuat regulasi dalam membuat kebijakan karena kondisi kasus harian yang bervariasi. Oleh karena itu pada penelitian kali ini akan melakukan prediksi terhadap jumlah kasus harian COVID-19 di Indonesia. Long Short Term Memory dipilih sebagai metode penelitian karena Long Short Term Memory dapat mempelajari data time series dan melakukan *forecasting* dengan cukup baik. Penelitian ini menggunakan data sekunder yang diambil dari [kawalcovid19.id](http://kawalCovid19.id) yang tersedia di portal <http://sinta.ristekbrin.go.id/covid/>. Data tersebut dibagi menjadi data latih dan data uji masing-masing sebesar 300 baris data dan 60 baris data dengan satu feature “kasus harian”. Hasil pengujian dari model yang telah dibangun pada proses training menunjukkan nilai MSE dan MAE masing-masing sebesar 0.03 dan 0.15 pada data normalisasi dan 2303320.93 dan 1268.98 pada data denormalisasi.

ABSTRACT

Trisna, Rizki. 2021. **Implementing Long Short Term Memory Method in Predicting the Number of Covid-19 Cases**. Undergraduate Thesis. Department of Informatics Engineering Faculty of Science and Technology Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Ajib Hanani, M.T, (II) Irwan Budi Santoso, M.Kom

Keywords: *LSTM, Prediction, Daily Case of COVID-19*.

The daily rate of positive cases of COVID-19 in Indonesia has varied widely. This affects health care providers and regulators in making policies because of the varying daily case conditions. Therefore, in this study, we will predict the number of daily cases of COVID-19 in Indonesia. Long Short Term Memory was chosen as the research method because Long Short Term Memory can study time series data and perform forecasting quite well. This study uses secondary data taken from kawalcovid19.id which is available on the <http://sinta.ristekbrin.go.id/covid/> portal. The data is divided into training data and test data, each with 300 data lines and 60 data lines with one “daily case” feature. The test results of the model that has been built in the training process show the MSE and MAE values of 0.03 and 0.15 respectively on the normalized data and 2303320.93 and 1268.98 on the denormalized data.

الملخص

عبادي ، رزقي تريسنا رولي . 2021. تنفيذ طريقة الذاكرة طويلة المدى في توقع عدد حالات كوفيد -19 أطروحة جامعية .مقال .قسم هندسة المعلوماتية بكلية العلوم والتكنولوجيا مولانا مالك إبراهيم الدولة الإسلامية جامعة مالانج مالانج. مستشار: (١) عجيب حنائي، ماجستير (٢) اروان بودي سانتوسو

الكلمات الرئيسية : LSTM, التنبؤ ، الحالة اليومية لـ COVID-19

اختلف المعدل اليومي للحالات الإيجابية لـ COVID-19 في إندونيسيا بشكل كبير . يؤثر هذا على مقدمي الرعاية الصحية والمنظمين في وضع السياسات بسبب اختلاف ظروف الحالة اليومية. لذلك ، في هذه الدراسة ، سوف نتوقع عدد الحالات اليومية لـ COVID-19 في إندونيسيا. تم اختيار الذاكرة طويلة المدى كطريقة بحث لأن الذاكرة طويلة المدى يمكنها دراسة بيانات السلاسل الزمنية وأداء التنبؤ بشكل جيد. تستخدم هذه الدراسة بيانات ثانوية مأخوذة من kawalcovid19.id والمتوفرة على <http://sinta.ristekbrin.go.id/covid/portal>. تنقسم البيانات إلى بيانات تدريب وبيانات اختبار ، لكل منها 300 خط بيانات و 60 خط بيانات مع ميزة "حالة يومية" واحدة. تُظهر نتائج اختبار النموذج الذي تم إنشاؤه في عملية التدريب قيم MSE و MAE البالغة 0.03 و 0.15 على التوالي على البيانات المعيارية و 2303320.93 و 1268.98 على البيانات غير الطبيعية.

BAB I

PENDAHULUAN

1.1 Latar Belakang

COVID-19 merupakan penyakit yang baru muncul sekitar akhir tahun 2019 dan kini telah ditetapkan oleh WHO menjadi pandemi di seluruh dunia. COVID-19 atau Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-Cov-2) telah muncul pada akhir tahun 2019 yang diyakini oleh banyak pihak berasal dari Kota Wuhan yang terletak di Provinsi Hubei, Republik Rakyat Tiongkok. Virus COVID-19 ini menyebar dengan cepat pada manusia dan pada awalnya diduga ditularkan oleh kelelawar dan ditularkan ke manusia melalui inang perantara. Penyakit ini memiliki gejala seperti demam, batuk, sesak napas dan beberapa gejala flu lainnya. Menurut Worldometers, pandemi COVID-19 ini menyerang 210 negara dan wilayah di seluruh dunia dengan kasus dikonfirmasi hingga enam juta kasus yang tercatat hingga 30 Mei 2020 (Muhammad et al., 2020).

Perkembangan kasus COVID-19 dunia saat ini telah menjadi pandemi yang memiliki angka kematian yang terbilang tinggi. Seperti yang ditunjukkan dalam paper yang dibuat oleh Chaurasia dan Pal (2020), angka kematian yang diakibatkan kasus COVID-19 yang pada awalnya berjumlah ratusan, lalu berubah menjadi ribuan hingga jutaan kasus. Hal ini tidak bisa dianggap sebagai masalah yang ringan karena cepat atau lambat, jika jumlah kasus COVID-19 ini tidak menunjukkan penurunan dalam kurun waktu yang lebih lama akan menimbulkan masalah yang lebih serius untuk umat manusia.

Indonesia merupakan salah satu negara yang tercatat terdampak pandemi COVID-19 ini dan perkembangan jumlah kasus COVID-19 di Indonesia yang tercatat oleh salah satu sumber media yang dirilis pada bulan Agustus 2020 menunjukkan terjadinya peningkatan jumlah kasus. Dikutip dari media online detik.com, tercatat pada tanggal 19 Agustus 2020 kasus positif COVID-19 di Indonesia bertambah 1902 menjadi 144.945, pasien sembuh bertambah 2352 menjadi 98.657, dan pasien meninggal bertambah 69 orang menjadi 6.346. Jumlah ini terhitung besar karena dari seluruh kasus di dunia Indonesia menempati urutan ke-23 sebagai negara terbanyak dalam perhitungan jumlah kasus COVID-19.

Tidak terkontrolnya penyebaran virus COVID-19 di beberapa bulan terakhir, khususnya di Indonesia, membuat efek yang belum pernah ada pada pandemi sebelumnya. Hal ini menuntut adanya campur tangan dari teknologi untuk membantu melawan situasi dan mengembalikan keadaan dunia sebaik mungkin. Kecerdasan buatan merespon pada situasi seperti ini (Arulprakash dan Aruldoss, 2020). Dari mulai Januari 2020 hingga sekarang, banyak peneliti, ilmuwan, hingga dokter dari seluruh dunia telah dan sedang berusaha untuk menyelesaikan masalah pandemi yang kompleks ini sehingga hasil yang didapatkan dapat membantu pihak yang berkepentingan baik dalam pemberian regulasi suatu negara maupun pihak yang berkecimpung dalam rumah sakit sehingga angka kematian akibat pandemi COVID-19 ini dapat berkurang (Chaurasia dan Pal, 2020)

Seperti yang telah dijelaskan sebelumnya, kecerdasan buatan merupakan salah satu bidang non-klinis yang turut terlibat dalam membantu melawan situasi pandemi COVID-19 ini. Terdapat beberapa penelitian yang telah dilakukan salah

satu diantaranya adalah penelitian yang dilakukan oleh Kırbaş et al. (2020). Dalam penelitiannya, Kirbas membangun model untuk melakukan perbandingan terhadap model prediksi kasus kumulatif terkonfirmasi virus COVID-19 menggunakan dataset COVID-19 yang didapat dari *European Center for Disease Prevention and Control* (ECDC). Ia menggunakan tiga metode diantaranya *Auto-Regressive Integrated Moving Average* (ARIMA), *Nonlinear Autoregression Neural Network* (NARNN) dan *Long-Short Term Memory* (LSTM). Kirbas membandingkan ketiga metode tersebut menggunakan enam matriks pengukuran performa seperti MSE, PSNR, RMSE, NRMSE, MAPE, dan SMAPE untuk mendapatkan model prediksi yang paling akurat diantara tiga metode yang digunakan. Hasil yang ditunjukkan dalam penelitiannya adalah metode LSTM merupakan metode yang menghasilkan tingkat akurasi paling akurat dalam melakukan prediksi kasus Covid-19 di beberapa negara di eropa dibanding metode lainnya dengan nilai RMSE paling rendah.

Long Short Term Memory yang digunakan pada penelitian Kırbaş et al. (2020) merupakan metode yang dibuat untuk mengatasi masalah *gradient descent* yang terjadi pada metode Recurrent Neural Network dengan menerapkan tiga gerbang atau *gates* yang mengontrol aliran informasi yang masuk dengan masing-masing gerbang terdiri dari *input gate*, *forget gate*, dan *output gate*. Terdapat penelitian lain yang menggunakan metode LSTM tersebut, salah satu penelitian tersebut adalah yang dilakukan oleh Steven Sen et al. (2020). Pada penelitiannya, Ia melakukan perbandingan terhadap metode *Multi Layer Perceptron* (MLP) dan *Long Short Term Memory* (LSTM) dalam melakukan prediksi terhadap harga beras di PT. Food Station. Dalam penelitiannya, Steven menggunakan data aktual dari

PT. Food Station pada periode 2016 hingga 2019 sejumlah 1307 dengan distribusi sebesar 1123 data latih dan 184 data testing. Penelitiannya menghasilkan kesimpulan bahwa metode LSTM mengungguli metode MLP dengan nilai RMSE data training sebesar 0.49 dan RMSE data testing sebesar 0.27.

Disamping melihat pandemi ini dari perspektif sains yang telah dan sedang dijalankan, islam memberikan perspektif lain untuk dilihat yaitu melalui perspektif spiritualitas islam. Perspektif spiritualitas islam bekerja menemukan makna-makna yang tidak dapat dijangkau melalui perspektif sains. Dalam salah satu ayat Al Baqarah[2] ayat 19 di dalam Al Quran yang penulis kutip dari Shihab (2020) yang berbunyi :

وَأَنْفِقُوا فِي سَبِيلِ اللَّهِ وَلَا تُلْقُوا بِأَيْدِيكُمْ إِلَى التَّهْلُكَةِ. وَأَحْسِنُوا. إِنَّ اللَّهَ يُحِبُّ الْمُحْسِنِينَ

“Dan belanjakanlah (harta bendamu) di jalan Allah, dan janganlah kamu menjatuhkan tangan (diri) kamu ke dalam kebinasaan dan berbuat baiklah. Sesungguhnya Allah menyukai orang-orang yang muhsin (orang yang selalu berbuat yang lebih baik) ” (Q.S. Al-Baqarah [2]: 19).

Dan dalam potongan surat Al Maidah ayat 2 Al-Quran yang berbunyi :

... وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ ۚ وَاتَّقُوا اللَّهَ ۚ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

“... Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya Allah amat berat siksa-Nya.” (Q.S. Al-Maidah [5]: 2).

Perspektif tersebut sangatlah relevan dengan kondisi pandemi saat ini dimana saling membantu dan berbuat baik menggunakan kesempatan dan potensi yang masing-masing individu miliki, termasuk penulis, merupakan sebuah keharusan. Oleh karena itu, penulis memutuskan untuk menjalankan penelitian yang penulis

beri judul “Penerapan Metode *Long Short Term Memory* dalam Memprediksi Jumlah Kasus COVID-19”.

1.2 Pernyataan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka pernyataan masalah pada penelitian kali ini adalah seberapa tinggi nilai *Mean Squared Error* dan *Mean Absolute Error* yang dihasilkan metode *Long Short Term Memory* dalam melakukan prediksi terhadap jumlah kasus COVID-19?

1.3 Tujuan Penelitian

Sesuai dengan latar belakang dan pernyataan masalah diatas, maka tujuan dari penelitian ini adalah untuk mengukur nilai *Mean Squared Error* dan *Mean Absolute Error* yang dihasilkan metode *Long Short Term Memory* dalam melakukan prediksi terhadap jumlah kasus COVID-19.

1.4 Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah yang diperlukan agar penelitian ini sesuai dengan tujuan penelitian. Batasan masalah tersebut adalah data yang digunakan pada penelitian ini merupakan data sekunder yang penulis ambil dari portal kawalcovid19.id.

1.5 Manfaat Penelitian

Dalam melakukan penelitian ini, selain penulis mengharapkan tujuan penelitian ini sendiri dapat dicapai, peneliti juga mengharapkan adanya manfaat

yang dapat diberikan dari hasil penelitian ini. Adapun manfaat yang penulis harapkan dari penelitian ini diantaranya:

1. Manfaat bagi penyedia layanan kesehatan, mereka dapat melakukan perkiraan biaya yang dibutuhkan untuk melakukan pengadaan fasilitas kesehatan yang diperlukan untuk menangani kasus COVID-19 dengan mengacu pada sistem di penelitian ini.
2. Manfaat bagi pemerintah dan/atau pembuat regulasi, mereka dapat memberikan aturan atau regulasi yang menyesuaikan dengan data hasil prediksi yang dihasilkan dari sistem yang dibangun di penelitian ini untuk menangani kasus di daerah terdampak masing-masing.

BAB II

STUDI PUSTAKA

2.1 Kasus Covid-19

Kasus Covid-19 telah menjadi pusat perhatian dunia setelah penyakit ini diumumkan telah menjadi pandemi dan menimbulkan ancaman yang belum pernah terjadi sebelumnya di seluruh dunia. Pandemi ini telah menyebabkan beberapa perubahan tatanan di seluruh dunia dimulai dari sosial, ekonomi, politik, budaya, dan aspek-aspek yang menyangkut mengenai lingkungan. Penyakit Covid-19 ini sendiri merupakan penyakit yang disebabkan oleh virus baru yang diberi nama *severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2)*. Berdasarkan data yang diterbitkan oleh WHO, kasus pertama Covid-19 dilaporkan pada bulan Desember 2019. Seperti data yang ditunjukkan oleh Center for Systems Science and Engineering (CSSE) di Universitas Johns Hopkins, pandemi Covid-19 telah mengakibatkan 346.700 kematian dengan total kasus sebesar 5.519.876 kasus terkonfirmasi hingga 26 Mei 2020 (Rahman et al., 2020).

Selama terjadinya pandemi Covid-19, beberapa negara mengambil tindakan yang dapat mengurangi dan mengatasi penyebaran virus Covid-19 ini. Banyak penelitian di bidang medis yang telah dilakukan oleh peneliti yang bertujuan untuk meneliti dan mengobati penyakit ini. Terdapat penelitian lainnya yang juga dilakukan untuk mempelajari efek sosial, psikologi, dan ekonomi yang timbul akibat virus Covid-19 ini. Model epidemiologi, statistik, dan matematis juga telah diperkenalkan untuk melakukan prediksi terhadap distribusi untuk mengamati perubahan yang bergantung pada kondisi cuaca atau meteorologi dan juga untuk

melakukan pemeriksaan terhadap struktur epidemi yang mempengaruhi banyak negara di dunia (Yeşilkanat, 2020).

Dalam penelitian machine learning mengenai Covid-19, dibutuhkan data yang digunakan sebagai penunjang pembangunan model yang akan dibangun, baik digunakan sebagai data latih maupun data uji. Seperti yang telah disebutkan dalam penelitian Farooq dan Bazaz (2020) bahwa untuk berbagai teknik prediksi seperti neural network atau teknik estimasi parameter model lainnya, data pelatihan diperlukan untuk melakukan pelatihan terhadap model sebelum menerapkannya pada skenario di masa pengujian mendatang. Namun, kasus pandemi seperti Covid-19 ini, data pelatihan yang tersedia terus berkembang seiring dengan berjalannya waktu dan model yang akan dibangun juga perlu dilatih dan dijalankan pada saat yang sama karena parameter model dapat berubah seiring dengan adanya interupsi dari faktor eksternal yang berbeda-beda seperti kebijakan pemerintah, pemberlakuan *social distancing*, dan lainnya.

Di Indonesia, saat ini, telah menyediakan portal kumpulan data dari beberapa kontributor dalam portal tersebut yang terkait virus Covid-19 yaitu portal yang disediakan oleh Kementrian Riset dan Teknologi – Badan Riset dan Inovasi Nasional Republik Indonesia dengan alamat website <http://sinta.ristekbrin.go.id/covid/>. Portal ini juga menerima sumbangan karya yang ditujukan untuk dapat dikembangkan lebih lanjut oleh sesama peneliti mengenai isu Covid-19 yang sedang terjadi. Salah satu kontributor data pada portal tersebut adalah Kawalcovid19 yang merupakan organisasi yang dibentuk oleh relawan yang beranggotakan warga negara Indonesia yang menjadi relawan yang tinggal di dalam

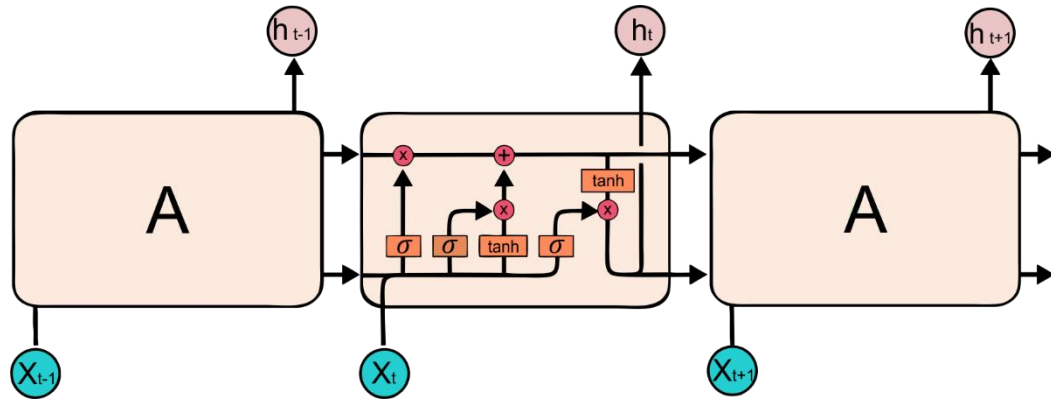
negeri maupun relawan yang tinggal di luar negeri yang juga sudah lebih dulu berada di dalam komunitas yang dinyatakan telah positif terinfeksi virus Covid-19. Data yang mereka dapatkan berasal dari kurasi dan analisis atas informasi yang beredar (baik *news explainer* ataupun *newsplainer*), dengan berdasarkan hanya pada poin-poin yang telah terbukti dan berdasarkan fakta (KawalCovid19, 2020). Oleh karena itu data tersebut dapat digunakan sebagai data latih atau data uji pada algoritma machine learning.

2.2 Long Short Term Memory (LSTM)

Long Short Term Memory merupakan salah satu pendekatan menggunakan arsitektur *Recurrent neural network* (RNN) yang pertama kali dikenalkan oleh Sepp Hochreiter and Jürgen Schmidhuber pada tahun 1997. LSTM dikenalkan untuk menyelesaikan masalah *vanishing gradient* yang terjadi dalam arsitektur *vanilla* RNN pada saat menyimpan memori dalam jangka panjang. Hal ini karena LSTM memiliki unit yang unik yang dapat mempelajari kapan waktu untuk membuka dan menutup gerbang dalam jaringan untuk mengatur pergerakan error yang konstan dalam jaringan tersebut (Sagheer & Kotb, 2019).

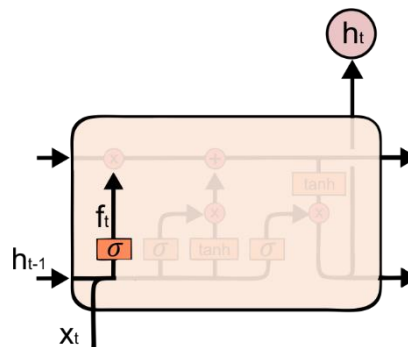
LSTM memiliki tiga *gates* atau gerbang yang masing-masing memiliki peran untuk melindungi dan mengontrol *cell state*. *Cell state* merupakan garis horizontal (jika dilihat dalam penggambaran pada gambar 2.1) yang melewati bagian atas diagram sel LSTM yang memiliki kemampuan untuk menghapus atau menambahkan informasi baru yang masuk dalam waktu t dengan memanfaatkan struktur cermat yang disebut gerbang. *Gates* atau gerbang sendiri adalah sebuah cara yang digunakan oleh LSTM untuk melakukan seleksi terhadap informasi yang

masuk ke dalam sel. Gerbang tersebut tersusun atas layer yang menggunakan operasi sigmoid dan perkalian *pointwise* (Olah, 2015).



Gambar 2.1 Struktur sel pada LSTM

Forget gate merupakan gerbang pertama yang dioperasikan dalam sel LSTM. *Forget gate* ini menentukan informasi mana yang harus dipertahankan dan yang harus dibuang dari *cell state*. Gerbang ini menerima dua input, masing-masing dari h_{t-1} dan x_t yang dimana h_{t-1} merupakan keluaran dari proses LSTM pada *time step t-1* dan x_t merupakan input pada *time step t*. Output dari gerbang ini adalah angka dengan rentang 0 hingga 1. Diagram kerja *forget gate* ditunjukkan pada gambar 2.2.



Gambar 2.2 Struktur sel pada LSTM

Rumus yang digunakan pada gerbang tersebut adalah:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

Keterangan:

f_t = *forget gate*

σ = *sigmoid function*

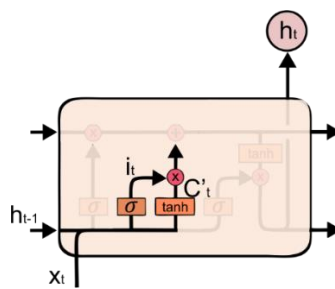
W_f = bobot pada *forget gate*

h_{t-1} = hasil output pada *time step t-1*

x_t = input pada *time step t*

b_f = bias pada *forget gate*

Setelah didapatkan nilai dengan rentang 0 hingga 1 pada *forget gate*, maka operasi selanjutnya adalah pada *input gate*. *Input gate* terdiri dari dua bagian, bagian pertama menggunakan fungsi sigmoid yang menentukan informasi mana yang ingin di-*update*. Sedangkan bagian kedua menggunakan fungsi tanh yang berfungsi untuk menentukan vektor yang akan ditambahkan pada nilai *cell state* (C'_t). Kedua bagian tersebut berfungsi untuk menentukan informasi baru apa saja yang ingin disimpan pada *cell state*. Layer *input gate* ini dapat dilihat pada gambar 2.3.



Gambar 2.3 *Input gate* pada LSTM

Rumus yang digunakan pada gerbang tersebut adalah:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$C'_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.3)$$

Keterangan:

i_t = input gate

W_i = bobot pada gerbang input

b_i = bias pada gerbang input

C'_t = kandidat nilai cell state baru yang akan ditambahkan ke C_{t-1}

\tanh = fungsi tanh

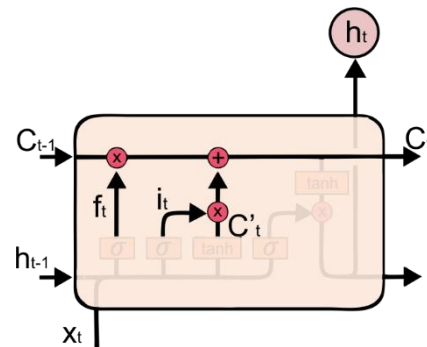
W_c = bobot pada operasi cell state baru

h_{t-1} = hasil output pada time step t-1

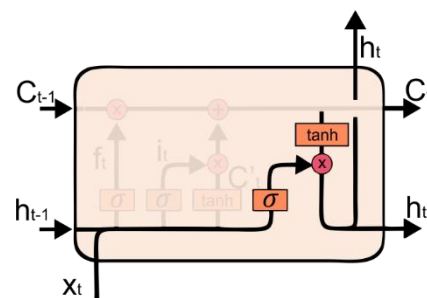
x_t = input pada time step t

b_c = bias pada operasi cell state baru

Hasil yang didapatkan dari operasi pada *forget gate* dan *input gate* selanjutnya dioperasikan agar hasil dari operasi tersebut dapat digunakan untuk meng-update *cell state* C_{t-1} . Pada operasi ini dilakukan perkalian pada state sel C_{t-1} dengan hasil dari layer *forget gate* (f_t). Lalu tambahkan hasil perkalian tersebut dengan $i_t * C_t$ seperti pada gambar 2.4. Setelah dua operasi tersebut (operasi perkalian C_{t-1} dan f_t lalu penambahan dengan $i_t * C'_t$), langkah selanjutnya adalah dengan mengoperasikan *output gate*. Pada *output gate*, layer sigmoid akan dioperasikan terlebih dahulu untuk menentukan bagian dari *cell state* apa saja yang akan digunakan sebagai hasil keluaran. Setelah operasi sigmoid dilakukan, selanjutnya operasi tanh dijalankan pada *cell state* lalu dikalikan dengan keluaran dari layer sigmoid yang telah dihitung sebelumnya (Olah, 2015). Operasi ini dapat dilihat pada gambar 2.5.



Gambar 2.4 Operasi pada hasil input gate dan perkalian C'_t dan i_t pada sel LSTM



Gambar 2.5 Output gate pada sel LSTM

Rumus yang digunakan pada gerbang tersebut adalah:

$$C_t = f_t * C_{t-1} + i_t * C'_t \quad (2.4)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.5)$$

$$h_t = o_t * \tanh(C_t) \quad (2.6)$$

Keterangan:

C_t = Cell state pada time step t

f_t = forget gate

C_{t-1} = Cell state pada time step $t-1$

C'_t = Nilai cell state baru pada hasil perhitungan gerbang input

o_t = fungsi gerbang output

σ = sigmoid function

W_o = bobot pada gerbang output

h_{t-1} = hasil output pada time step $t-1$

x_t = *input* pada *time step* t

b_o = bias pada gerbang *output*

h_t = *output* pada *time step* ke t

\tanh = fungsi \tanh

Berdasarkan penelitian yang dilakukan oleh Arfan (2019) dalam prediksi harga saham di Indonesia menggunakan algoritma LSTM, Arfan menunjukkan bahwa LSTM dapat mengolah data dengan ketergantungan jangka panjang dengan menghasilkan nilai kesalahan yang lebih kecil dari metode SVR. Arfan melakukan penelitian dengan melakukan pengumpulan data, preprocessing, pembagian, perancangan metode, pelatihan, dan pada pengujian pada proses paling akhir. Pengujian yang dilakukan menggunakan data *time series* tahun 2014 hingga 2016 sebagai data latih dan data *time series* tahun 2017 hingga 2019 sebagai data uji. Hasil terbaik yang didapatkan dalam penelitiannya, ia mendapatkan nilai error menggunakan MSE sebesar 0.0015 dalam waktu latih selama 18.2 detik.

Penelitian lain yang dilakukan oleh Rasjid et al. (2021) juga menunjukkan bahwa prediksi menggunakan metode LSTM-NN menunjukkan hasil yang lebih baik daripada metode Savitzky Golay Smoothing. Dalam penelitiannya, Rasjid dan rekan-rekannya melakukan prediksi terhadap perbandingan prediksi dari dua metode yang berbeda, masing-masing metode tersebut adalah LSTM-NN dan Savitzky Golay Smoothing. Dalam penelitiannya, Rasjid dan rekan-rekannya menggunakan dataset dari Humanitarian Data Exchange (HDX) yang berisi informasi data *time series* dari kasus kematian dan terinfeksi karena virus COVID-19 di Indonesia dalam rentang waktu 2 Maret 2020 hingga 26 Juli 2020 dengan

totoal 147 *records*. Berdasarkan hasil penelitian yang dilakukan, metode LSTM-NN dinilai lebih baik ketimbang metode Savitzky Golay Smoothing.

2.3 Mean Squared Error (MSE)

Mean Squared Error (MSE) merupakan *metrics* pengukuran error yang cukup sederhana dan sering digunakan untuk melakukan evaluasi terhadap hasil prediksi. Dimana pada MSE pada dasarnya melakukan pengukuran kesalahan yang dikuadratkan dan diambil rata-ratanya dari hasil prediksi yang dilakukan. Nilai MSE memiliki jaminan dengan tidak akan menghasilkan nilai negatif karena dalam prosesnya melakukan kuadrat dari hasil pengurangan data asli dengan data hasil prediksi (Yashwanth, 2021). Dimana rumus yang digunakan pada pengukuran MSE tersebut adalah sebagai berikut.

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - Y'_i)^2 \quad (2.7)$$

Keterangan:

MSE : Error yang dihasilkan dari perhitungan MSE

N : Banyak sample data yang dihitung error-nya

Y_i : Data aktual atau groundtruth

Y'_i : Hasil prediksi

2.4 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) merupakan *error metric* yang didalamnya menggunakan penghitungan rata-rata perbedaan nilai selisih groundtruth dan hasil prediksi sistem yang telah diabsolutkan. *Metric* ini melakukan penghukuman *error* yang tidak separah yang diberikan oleh *Mean Squared Error*, dengan begitu nilai

yang dihasilkan pada *metric* ini tidak terganggu apabila terdapat nilai *outlier* yang cukup besar. *Metric* ini menggunakan rumus sebagai berikut.

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - Y'_i| \quad (2.8)$$

Keterangan:

MAE : Error yang dihasilkan dari perhitungan MAE

N : Banyak sample data yang dihitung *error*-nya

Y_i : Data aktual atau *groundtruth*

Y'_i : Hasil prediksi

BAB III

PERANCANGAN DAN IMPLEMENTASI SISTEM

3.1 Pengumpulan Data

Dalam sub-bab ini, peneliti akan menjelaskan data yang digunakan untuk objek penelitian. Jenis data yang peneliti gunakan adalah data sekunder yang merupakan dataset disediakan oleh para ahli di bidangnya. Data sekunder tersebut akan dibagi menjadi dua kelompok data yaitu kelompok data latih atau training dan data uji atau data testing. Isi dari dua kelompok sub-dataset tersebut akan dipilih secara acak dari dataset utama. Kedua sub-dataset tersebut digunakan untuk membangun model dari sistem yang akan dibangun dimana data uji akan digunakan untuk membangun model pelatihan sementara data testing akan digunakan untuk membangun model pengujian sistem.

Data yang peneliti gunakan merupakan data sekunder yang berupa statistik perkembangan jumlah kasus Covid-19 yang dirangkum secara historis oleh website kawalcovid19.id. Data tersebut memiliki 361 baris record dan 40 atribut (total 46 atribut dengan atribut kosong) atau kolom yang disusun secara historis. Data tersebut diunggah oleh praktisi kesehatan, akademisi, dan profesional yang dikumpulkan secara sukarela mulai tanggal 02 Maret 2020 hingga 25 Februari 2021 dalam format spreadsheet google document yang dapat diunduh dalam format Microsoft Excel (.xlsx), Format OpenDocument (.ods), Dokumen PDF (.pdf), Halaman web (.html, .zip), Nilai yang dipisahkan koma (.csv), atau Nilai yang dipisahkan tab (.tsv). Untuk mempermudah pengolahan data yang peneliti lakukan, peneliti akan menggunakan format file Microsoft Excel (.xlsx) atau nilai yang

dipisahkan koma (.csv). Berikut ini merupakan sample dataset yang peneliti ambil dari dataset kawalcovid19:

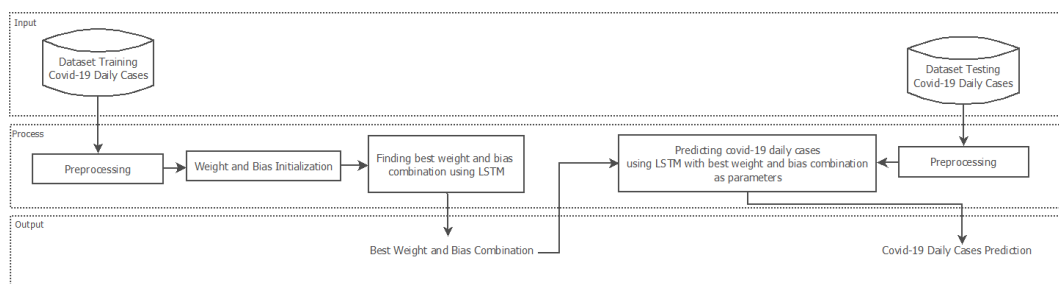
Tabel 3.1 Sampel dataset kawalcovid19

No.	Nama Atribut/Kolom	Contoh data	Record baris ke
1	Tanggal	10-Sep	192
2	Kasus harian	3,861	192
3	Total kasus	207,203	192
4	Kasus aktif	51,237	192
5	% kasus aktif	24.73%	192
6	Sembuh (baru)	2,310	192
7	Sembuh	147,510	192
8	Tingkat kesembuhan (seluruh kasus)	71.19%	192
9	Tingkat kesembuhan (closed cases)	94.58%	192
10	Meninggal (baru)	120	192
11	Meninggal Dunia	8,456	192
12	Tingkat kematian (seluruh kasus)	4.08%	192
13	Tingkat kematian (closed cases)	5.42%	192
14	Suspek aktif	95,501	192
15	Jumlah spesimen diperiksa (dari 1 April)	2,549,579	192
16	Jumlah orang diperiksa	1,469,943	192
17	Negatif	1,262,740	192
18	Spesimen	34,909	192
19	Orang yang dites	20,314	192

20	Jumlah test/juta penduduk	5,444	192
21	Test/Positif	12.30	192
22	Jumlah org yg dites / positif	7.09	192
23	Jumlah Test/Positif harian	5.26	192
24	Positif / Jumlah Tes	14.10%	192
25	Positive rate harian	19.01%	192
26	Case Growth Rate	1.90%	192
27	Tingkat positivitas mingguan	19.66%	192
28	Jumlah spesimen diperiksa (rata-rata 7 hari)	30,102	192
29	Jumlah orang diperiksa (rata-rata 7 hari)	16,665	192
30	Rasio spesimen/orang diperiksa (rata-rata 7 hari)	1.81	192
31	Test PCR Orang Nasional Rata-rata 7 hari	16665	192
32	Pos Rate Nasional - rata-rata 7 hari	19.66%	192
33	Test Orang DKI	6383	192
34	Positif DKI	1450	192
35	Pos Rate DKI	22.72%	192
36	Pos Rate DKI - 7 harian	15.93%	192
37	NON DKI	13,931	192
38	Positif Non DKI	2,411	192
39	Pos Rate Non DKI	17.31%	192
40	Pos Rate Non DKI - 7 harian	22.19%	192

3.2 Desain Sistem

Subbab ini menjelaskan tahapan untuk membuat rancangan dari alur sistem yang akan dibuat. Dimana alur sistem tersebut tercantum pada Gambar 3.1 berikut.



Gambar 3.1 Desain sistem

Gambar 3.1 merupakan desain sistem yang dibangun pada implementasi sistem menggunakan metode LSTM untuk melakukan prediksi kasus covid-19. Dalam desain sistem tersebut terdapat alur dari proses *preprocessing*, proses *learning*, dan proses prediksi dari metode LSTM.

3.3 Preprocessing

Data yang peneliti dapatkan dari website kawalcovid19.id tidak dapat langsung diproses menggunakan algoritma LSTM. Oleh karena itu dibutuhkan *preprocessing* data agar data dapat digunakan dalam proses sistem yang akan dibangun. Langkah pertama yang dilakukan adalah dengan *cleaning* dataset yaitu memilih atribut apa saja yang akan digunakan dan relevan dengan penelitian. Data awal yang peneliti dapatkan yang berupa data sekunder yang berjumlah yang berjumlah 40 atribut dan 361 baris data akan dilakukan pemilihan atribut yang sesuai dengan tujuan dari sistem yang akan dibangun. Atribut yang dipilih merupakan atribut yang akan digunakan untuk prediksi kasus baru Covid-19 baru dan atribut lain yang tidak digunakan akan dihapus. Dengan demikian didapatkan

atribut yang akan digunakan dalam penelitian ini seperti yang ditunjukkan pada tabel 3.2 berikut.

Tabel 3.2 Tabel Atribut yang Digunakan dalam Data Kasus Covid-19 dari Kawalcovid19.id

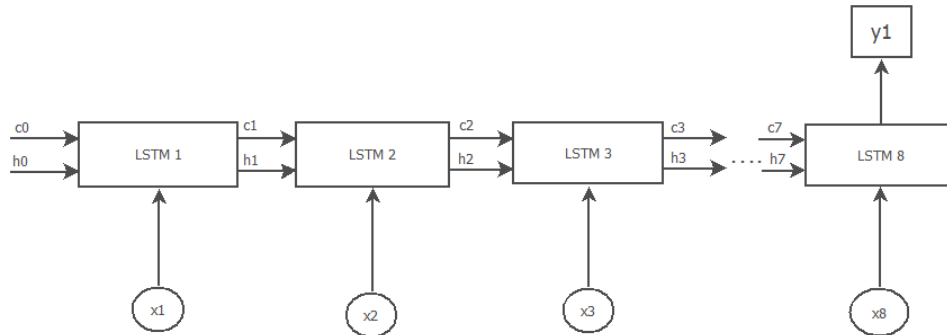
No.	Nama atribut	Keterangan
1	Kasus harian	Banyak kasus positif covid-19 baru yang ditemukan dalam satu hari

Setelah proses pemilihan atribut, langkah selanjutnya adalah dengan mengatasi nilai yang hilang atau *handling missing values*. Dalam mengatasi missing value salah satu cara dan yang paling sederhana adalah dengan menghapus baris atau sample data yang mengandung missing values . Data awal yang berjumlah 361 baris setelah dilakukan penghapusan baris yang memiliki missing values menjadi berjumlah 360 baris data.

3.4 Pembuatan Model LSTM

Pembuatan model LSTM memiliki beberapa tahap yang harus dilalui sebelum pada akhirnya didapatkan hasil berupa model yang digunakan pada prediksi kasus covid-19 menggunakan data uji. Tahapan-tahapan tersebut diantaranya adalah inisialisasi parameter dan training jaringan LSTM (Aldi et al., 2018). Tahap inisialisasi parameter bertujuan untuk menentukan nilai bobot awal, *learning rate*, jumlah *batch*, dan jumlah *epoch* sedangkan tahap training jaringan bertujuan untuk membuat model dengan parameter yang telah ditentukan sebelumnya.

3.4.1 Arsitektur Jaringan LSTM



Gambar 3.2 Arsitektur jaringan

Arsitektur yang peneliti gunakan pada penelitian ini terdiri dari 3 layer, yaitu layer input, layer LSTM dan layer output seperti yang ditunjukkan pada gambar 3.3.

3.4.1.1 Normalisasi Data

Pada layer input terdapat proses normalisasi data yang masuk dengan menggunakan *min max normalization*. Data yang telah di seleksi dan dibersihkan selanjutnya akan dilakukan transformasi data. Transformasi data dilakukan untuk mempermudah sistem menentukan hasil akhir prediksi. Transformasi yang dilakukan adalah dengan melakukan normalisasi *MinMax Scaler* terhadap data numerik kontinu. Normalisasi dilakukan untuk mengubah data kontinu ke dalam interval yang telah ditentukan sebelumnya. Hal ini dilakukan dengan tujuan meningkatkan akurasi dan mempercepat kecepatan dari sistem (Ahmad & Brown, 2009). Perhitungan normalisasi dilakukan dengan menggunakan rumus pada persamaan 3.1 berikut ini.

$$normalisasi(x) = \frac{x_i - x_{i\min}}{x_{i\max} - x_{i\min}} \quad (3.1)$$

Keterangan:

x_i : nilai data yang belum dinormalisasi

$normalisasi(x)$: nilai yang telah dinormalisasi

$x_{i\min}$: nilai paling kecil dari atribut data x_i

$x_{i\max}$: nilai tertinggi dari atribut data x_i (Loukas, 2020).

3.4.1.2 Long Short Term Memory (LSTM)

Long Short Term Memory merupakan metode yang digunakan pada penelitian kali ini. Dimana LSTM digunakan pada layer kedua dari arsitektur jaringan yang dibangun. Data input yang telah dinormalisasi diolah menggunakan fungsi-fungsi yang terdapat di dalam unit LSTM. Lalu keluaran dari unit di layer LSTM akan diolah di layer output. Dimana pada layer output terdapat fungsi aktivasi sigmoid yang dihubungkan oleh variabel bobot dan bias. Adapun tahapan proses pada layer LSTM tersebut adalah sebagai berikut:

1. Setelah input telah melalui proses normalisasi di layer input, untuk menjalankan fungsi-fungsi yang ada di layer kedua perlu diinisialisasikan bobot dan bias terlebih dahulu. Nilai awal dari bobot dan bias adalah berkisar antara -1 hingga 1.
2. Langkah selanjutnya setelah inisialisasi bobot dan bias awal adalah proses perhitungan nilai yang dihasilkan pada forget gate (ft). Persamaan yang digunakan untuk melakukan perhitungan pada forget gate (ft) adalah sebagai berikut

$$ft = \sigma(Wft \cdot [x, h_{t-1}] + bft) \quad (3.2)$$

Keterangan :

f_t = Matrix keluaran dari proses perhitungan nilai forget gate

W_{ft} = Bobot untuk forget gate

B_{ft} = Bias untuk forget gate

X = input yang telah dinormalisasi

H_{t-1} = Hidden state pada time step sebelumnya

σ = fungsi aktivasi sigmoid

3. Setelah nilai pada forget gate ditemukan, langkah selanjutnya adalah melakukan untuk nilai input gate dengan data masukkan yang telah dinormalisasi dengan bobot dan bias yang telah diinisialisasi di awal. Dimana untuk perhitungan nilai input gate tersebut menggunakan rumus perhitungan berikut:

$$it = \sigma(W_{it} \cdot [x, h_{t-1}] + b_{it}) \quad (3.3)$$

Keterangan :

It = Matriks keluaran perhitungan *input gate*

W_{it} = Bobot pada *input gate*

B_{it} = Bias untuk input gate

H_{t-1} = Hidden state yang dihasilkan timestep sebelumnya

X = input yang telah dinormalisasi

σ = Fungsi aktivasi sigmoid

4. Langkah selanjutnya adalah melakukan perhitungan untuk Cell gate. Dimana hasil dari perhitungan cell gate ($c't$). Untuk melakukan perhitungan cell gate menggunakan rumus perhitungan sebagai berikut:

$$c't = \tanh(W_{ct} \cdot [x, h_{t-1}] + b_{ct}) \quad (3.4)$$

Keterangan :

C'_t = Hasil perhitungan cell gate

W_{ct} = Bobot pada cell gate

B_{ct} = Bias pada cell gate

X = input yang telah dinormalisasi

H_{t-1} = Hidden state pada timestep sebelumnya

\tanh = Fungsi aktivasi tanh

Dimana hasil pada cell gate tersebut digunakan untuk melakukan perhitungan pada cell state bersama dengan hasil dari input gate. Dimana perhitungan untuk mendapatkan nilai cell state tersebut adalah menggunakan rumus sebagai berikut:

$$C_t = f_t * C_{t-1} + i_t * c'_t \quad (3.5)$$

Keterangan:

C_t = Hasil perhitungan cell state

F_t = Hasil perhitungan pada forget gate

C_{t-1} = Cell state pada timestep sebelumnya

i_t = Hasil perhitungan pada input gate

c'_t = Hasil perhitungan pada cell gate

5. Selanjutnya adalah melakukan penghitungan untuk mendapatkan nilai output gate (o_t). Dimana untuk mendapatkan nilai dari output gate ini dilakukan perhitungan dengan rumus sebagai berikut:

$$o_t = \sigma(W_{ot} \cdot [x, h_{t-1}] + b_{ot}) \quad (3.6)$$

Keterangan:

O_t = Matriks hasil perhitungan pada output gate

W_{ot} = Bobot pada output gate

X = Data input yang telah dinormalisasi

H_{t-1} = Hidden state timestep sebelumnya

b_{ot} – Bias pada output gate

σ = Fungsi aktivasi sigmoid

6. Setelah mendapatkan nilai di masing-masing gate, selanjutnya adalah melakukan penghitungan pada hidden state (h_t), dimana perhitungannya adalah sebagai berikut:

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

Keterangan:

H_t = hasil perhitungan pada hidden state

C_t = Hasil perhitungan cell state

O_t = Hasil perhitungan pada output gate

\tanh = fungsi aktivasi tanh

3.4.1.3 Sigmoid

Layer ketiga, yaitu layer output (Y'_t), memiliki fungsi aktivasi sigmoid yang digunakan untuk melakukan perhitungan terhadap hasil yang dihasilkan oleh layer LSTM untuk mendapatkan hasil prediksi dari sistem, dimana pada layer ini dapat dilakukan perhitungan dengan rumus sebagai berikut:

$$Y'_t = \frac{1}{1 + e^{-h_t}} \quad (3.8)$$

Keterangan :

$Y't$ = hasil prediksi dari perhitungan sigmoid

H_t = Hasil pada hidden state dikalikan dengan bobot pada output layer

3.4.2 *Training Jaringan LSTM*

Setelah tahap preprocessing pada dataset telah dilakukan dan data telah terbagi menjadi dua kelompok (data latih dan data uji), data latih selanjutnya akan digunakan training jaringan LSTM yang akan menghasilkan model LSTM. Sebelum melakukan training tersebut, terdapat beberapa parameter yang harus ditentukan terlebih dahulu, parameter tersebut adalah:

- a. Jumlah *batch size*
- b. Jumlah *sequence length*
- c. Nilai *learning rate*
- d. Jumlah *input unit*
- e. Jumlah *hidden unit*
- f. Jumlah *output unit*
- g. Jumlah *epoch*
- h. Bobot awal

Dimana *learning rate* merupakan *hyperparameter* yang mengontrol berapa banyak perubahan yang dilakukan sebagai respon terhadap estimasi error setiap kali bobot model diperbaharui (Brownlee, 2020). Sedangkan *batch* merupakan *hyperparameter* yang menentukan jumlah sample yang harus diolah sebelum memperbarui parameter internal dan keluaran dari satu *batch* digunakan untuk melakukan prediksi dan akan dihitung tingkat *error*-nya dan setiap satu batch

memiliki *batch size*. *Sequence length* adalah jumlah timestep yang harus dijalankan dalam satu kali *forward propagation*. *Input unit* dan *output unit* masing-masing menunjukkan banyaknya n-data dalam satu batch dalam satu sequence yang harus diolah dan dihasilkan dalam satu step *forward propagation* sedangkan *hidden unit* merupakan banyaknya unit yang mengisi *hidden state* di dalam sel LSTM. Yang terakhir, *epoch*, merupakan *hyperparameter* yang menentukan berapa kali algoritma *learning* akan dilakukan di seluruh set data latih. Satu *epoch* terdiri dari satu atau lebih *batch* data (Brownlee, 2018).

Setelah parameter selesai didefinisikan pada proses sebelumnya, tahapan selanjutnya adalah melakukan training terhadap jaringan LSTM. Pada penelitian kali ini, untuk melakukan *training* jaringan LSTM, peneliti menggunakan metode *Backpropagation Through Time (BPPT)* untuk melakukan *training* pada jaringan LSTM. BPPT sendiri merupakan salah satu jenis dari metode yang umum digunakan dalam pengoptimalan model yang populer dalam *Machine Learning*. Dimana BPPT sering digunakan dalam proses training yang menggunakan data input *time series*. Secara garis besar, algoritma dari metode ini dapat disimpulkan:

1. Siapkan urutan langkah waktu dari pasangan input dan output.
2. Buka gulungan jaringan lalu hitung *error* dan *error* kumulatif di setiap *sequence*.
3. Lakukan *update* bobot dari hasil penghitungan *error* kumulatif tersebut
4. Ulangi proses jika diperlukan. (Brownlee, 2017)

Proses *update* bobot dapat dilakukan dengan menggunakan rumus:

$$w_{baru} = w_{lama} - \alpha \cdot \frac{dE}{dw_t} \quad (3.2)$$

Keterangan:

W_{baru} = bobot baru pada iterasi selanjutnya

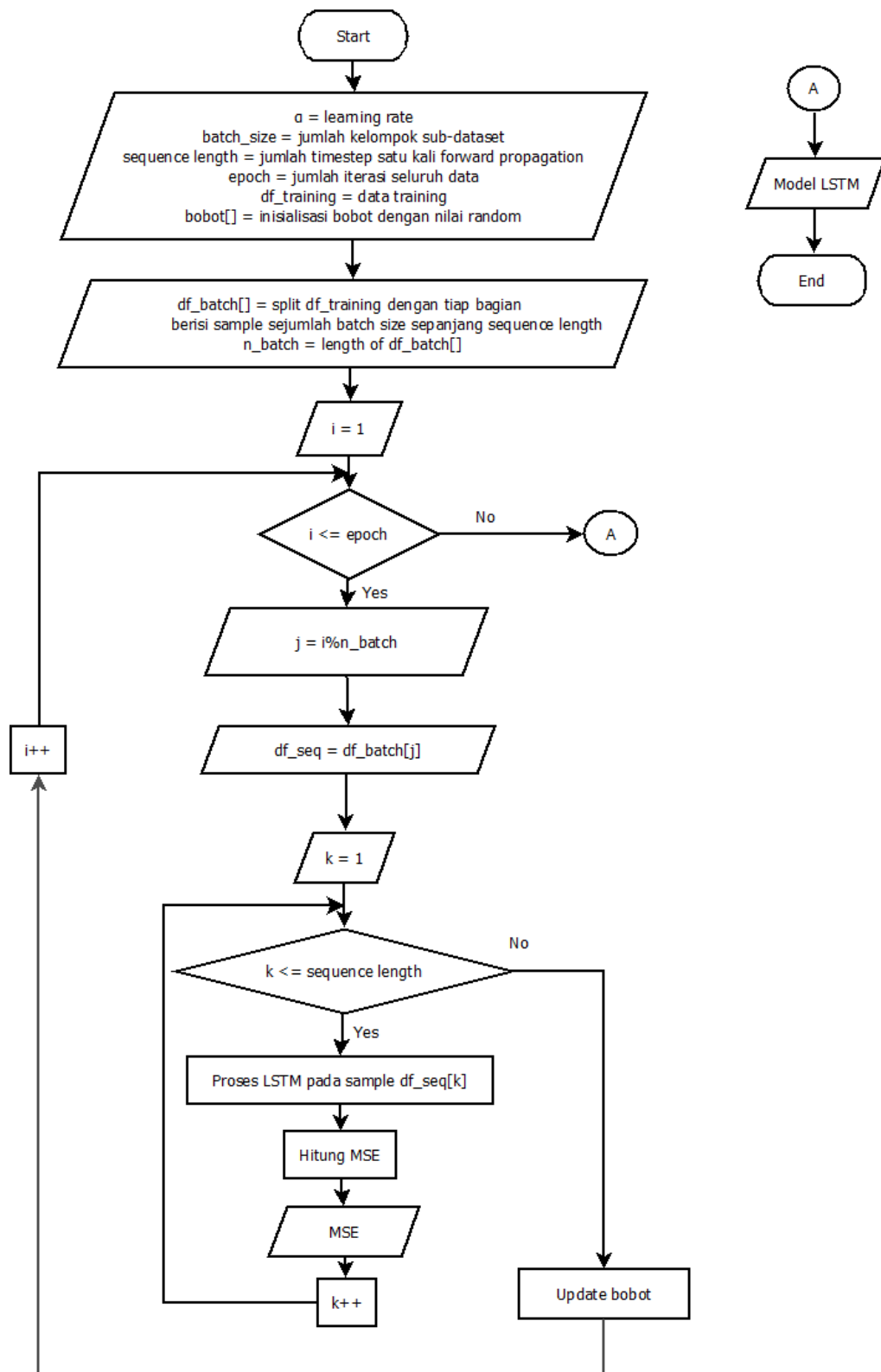
W_{lama} = bobot lama

α = *learning rate*

$\frac{dE}{dw_t}$ = penghitungan gradien/error pada *Error* terhadap bobot w_t

1. Inisialisasi nilai jumlah *batch size*, jumlah *sequence length*, nilai *learning rate*, jumlah *input unit*, jumlah *hidden unit*, jumlah *output unit (opsional)* jumlah *epoch*, dan bobot awal.
2. Inisialisasi nilai bobot dengan nilai *random*
3. Bagi data training menjadi sub-data *batch sequence* dengan isi sampel pada hasil *batching* data menggunakan parameter *batch size*, *sequence length*, dan *input unit*.
4. Hitung secara *forward* sesuai dengan alur kerja LSTM yaitu mulai dari *forget gate*, *input gate*, hingga *output gate* dan menghasilkan keluaran h_t , c_t , dan output prediksi sepanjang *time step batch sequence* data yang telah ditentukan dalam proses sebelumnya.
5. Pada setiap akhir *batch sequence* lakukan update bobot untuk menentukan bobot pada iterasi *batch sequence* selanjutnya.
6. Jika seluruh *batch sequence* data telah selesai diiterasi seperti yang ditunjukkan pada gambar 3.3, lalu ulangi proses 4 sampai 5 hingga jumlah *epoch* yang telah ditentukan.

Dengan demikian, keseluruhan langkah yang digunakan untuk melakukan training menggunakan BPPT pada penelitian kali ini adalah:



Gambar 3.3 *Flowchart* Pembuatan Model Prediksi Menggunakan LSTM

Model LSTM yang dihasilkan pada pemodelan sebelumnya berupa variasi nilai bobot, jumlah *epoch*, dan jumlah *batch size* yang menghasilkan *error* MSE paling kecil sehingga dapat digunakan untuk proses selanjutnya yaitu pada proses prediksi menggunakan data uji.

3.5 Implementasi Sistem

Implementasi sistem pada bab ini adalah tahap untuk melakukan penerapan dan dilanjutkan dengan pengujian terhadap skenario yang telah didefinisikan pada bab sebelumnya. Dalam melakukan implementasi sistem pada penelitian kali ini, peneliti menggunakan beberapa perangkat *hardware* maupun *software* sebagai media implementasinya. Perangkat tersebut diantaranya adalah laptop pribadi, *jupyter notebook*, dan *google colaboratory*.

3.5.1 Spesifikasi Hardware

Perangkat yang peneliti gunakan dalam implementasi sistem kali ini memiliki spesifikasi sebagai berikut:

- 1) Laptop pribadi :
 - a. *CPU and Memory* : Intel Core i7 CPU @ 2.60 GHz, 8192MB RAM
 - b. *Operating System* : Windows 10 Home
 - c. *Disk Size* : 32 GB Free Space
- 2) Google *colaboratory* :
 - a. *Colab Plan* : Free
 - b. *CPU and Memory* : Intel(R) Xeon(R) CPU @ 2.20GHz, 136222MB RAM
 - c. *Operating System* : Not specified

d. *Disk Size : 62 GB Free Space*

BAB IV

UJI COBA DAN PEMBAHASAN

4.1 Skenario Uji

Skenario uji merupakan evaluasi yang dilakukan untuk mengetahui tingkat *error* dari model yang telah dibangun. Terdapat beberapa tahap yang perlu dilakukan sebelum akhirnya mendapat nilai *error* dari model yang diuji.

Tahapan pertama adalah dengan mencari *groundtruth* dimana *groundtruth* sendiri merupakan data atau informasi yang digunakan untuk memberikan pembandingan terhadap hasil prediksi dari model yang telah dibangun. *Groundtruth* pada penelitian ini menggunakan data yang peneliti ambil dari portal kawalcovid19.id yang berisi data runtut waktu atau *time series* yang berjumlah 391 baris dengan atribut terpilih “Tanggal” dan “Kasus harian” yang masing-masing berisi tipe data *datetime* dan *integer*. Data *groundtruth* yang dipakai memiliki rentang waktu yang lebih panjang daripada rentang waktu yang digunakan untuk pembuatan dan pengujian model yaitu dari tanggal 2 Maret 2020 hingga 28 Maret 2021. *Groundtruth* didapatkan berdasar pada data atribut “Kasus harian” yang berisi data *integer* dengan rentang waktu yang dapat diukur atau dilihat pada atribut “Tanggal” yang berisi data *datetime*. Berikut merupakan beberapa sample yang digunakan sebagai *groundtruth*:

Tabel 4.1 Sampel *groundtruth*

Tanggal	Kasus harian
02-03-20	2
03-03-20	0

04-03-20	0
05-03-20	0
06-03-20	2
...	...
24-03-21	5,227
25-03-21	6,107
26-03-21	4,982
27-03-21	4,461
28-03-21	4,083

Dalam rangka membangun model LSTM pada proses training dan juga pengujian model tersebut di penelitian ini, peneliti tidak menggunakan keseluruhan data yang didapat dan membagi data tersebut menjadi dua bagian. Hal ini dilakukan untuk menyesuaikan relevansi terhadap sistem yang dibangun. Data yang terbagi menjadi dua dapat dilihat pada tabel 4.2 berikut.

Tabel 4.2 Pembagian data dalam pengujian

Sub-dataset	Data Latih	Data Uji
Banyak sampel	300 data (83%)	60 data (17%)
Rentang waktu	2 Maret 2020 – 27 Desember 2020	28 Desember 2020 – 25 Februari 2021

Setelah *groundtruth* telah didapatkan, pengujian dilakukan dengan mengambil data uji kemudian dibandingkan dengan data prediksi yang dihasilkan oleh model yang diuji pada rentang waktu atau *time step* yang telah ditentukan.

Pada proses *training*, proses perbandingan tersebut dilakukan bersamaan dengan pengukuran tingkat *error* yang dimana pada penelitian kali ini peneliti pengukuran tingkat *error* menggunakan *Mean Squared Error* dan *Mean Absolute Error*. Dimana MSE dapat dihitung dengan menggunakan rumus berikut:

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - Y'_i)^2 \quad (4.1)$$

Keterangan:

MSE : Error yang dihasilkan dari perhitungan MSE

N : Banyak sample data yang dihitung error-nya

Y_i : Data aktual atribut “Kasus baru” pada time step i dari groundtruth

Y'_i : Hasil prediksi atribut “Kasus baru” pada time step i dari keluaran pengujian model

Dan MAE menggunakan rumus:

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - Y'_i| \quad (4.2)$$

Keterangan:

MAE : Error yang dihasilkan dari perhitungan MAE

N : Banyak sample data yang dihitung *error*-nya

Y_i : Data aktual atribut “Kasus baru” pada *time step i* dari *groundtruth*

Y'_i : Hasil prediksi atribut “Kasus baru” pada *time step i* dari keluaran pengujian model

Proses pengukuran *error* dilakukan pada proses *training* dan juga proses pengujian menggunakan data uji sebelum dilakukan proses denormalisasi pada rumus 4.3. Pada proses *training*, pengukuran dilakukan dengan cara melakukan perubahan parameter pada saat pembentukan model lalu mengukur tingkat *error*

selama proses pembentukan model berlangsung untuk melakukan *monitoring* pergerakan tingkat *error*-nya. Parameter yang dimaksud adalah jumlah *epoch* dan jumlah *batch size*. Hal ini dilakukan untuk mencari perbandingan sekaligus mencari model dengan variasi jumlah *epoch* dan *batch size* yang paling baik sehingga menghasilkan besaran bobot dan bias yang paling sesuai untuk melakukan prediksi pada sesi uji coba menggunakan data uji dan hal ini ditunjukkan dengan tingkat tingkat *error* yang rendah. Berikut ini skenario perubahan parameter *epoch* dan *batch size* pada proses *training* yang akan dilakukan:

1. Perubahan parameter *epoch* dengan nilai *epoch*: 200, 400, 600, 800
2. Perubahan parameter *batch size* dengan nilai *batch size*: 5, 10, 15, 20, 30.

Berdasarkan skenario perubahan parameter tersebut, maka model *training* yang akan terbentuk adalah seperti pada tabel 4.3 berikut.

Tabel 4.3 Skenario perubahan parameter *epoch* dan *batch size* pada proses *training*

Model	<i>Epoch</i>	<i>Batch size</i>
Model 1	200	5
Model 2	200	10
Model 3	200	15
Model 4	200	20
Model 5	200	30
Model 6	400	5
Model 7	400	10
Model 8	400	15
Model 9	400	20

Model 10	400	30
Model 11	600	5
Model 12	600	10
Model 13	600	15
Model 14	600	20
Model 15	600	30
Model 16	800	5
Model 17	800	10
Model 18	800	15
Model 19	800	20
Model 20	800	30

$$X_t = y (X_{max} - X_{min}) + X_{min} \quad (4.3)$$

Keterangan:

X_t : Nilai yang sebenarnya

y : Nilai hasil prediksi

X_{max} : Nilai rentang maksimum pada parameter normalisasi

X_{min} : Nilai rentang minimum pada parameter normalisasi (Aulia, 2020)

Setelah hasil model yang berupa bobot dan bias telah terbentuk, langkah selanjutnya adalah dengan melakukan pemilihan model dengan hasil *error* terendah dan model tersebut akan dilakukan proses pengujian menggunakan data uji. Skenario yang digunakan adalah dengan membandingkan hasil prediksi

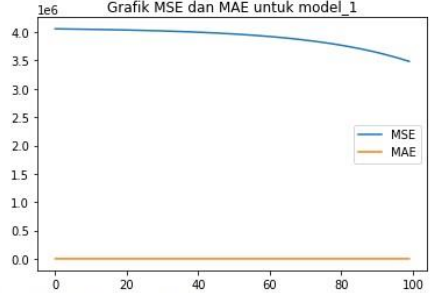
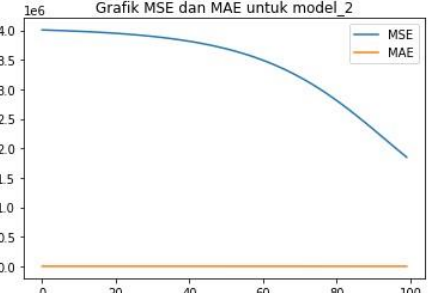
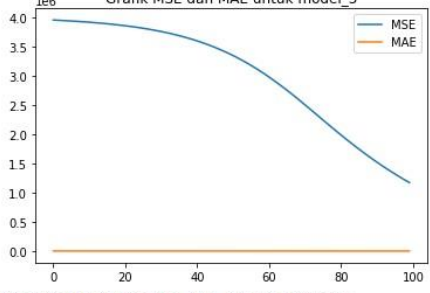
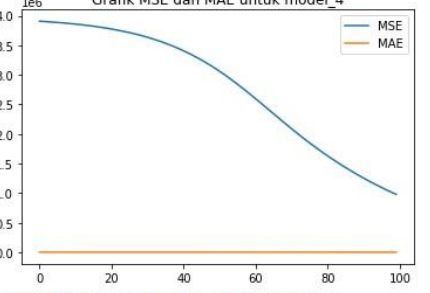
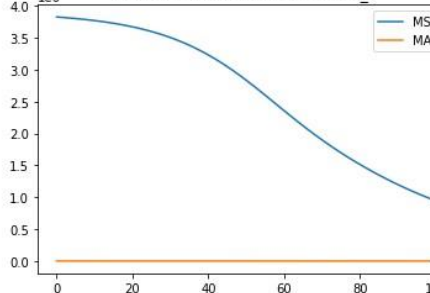
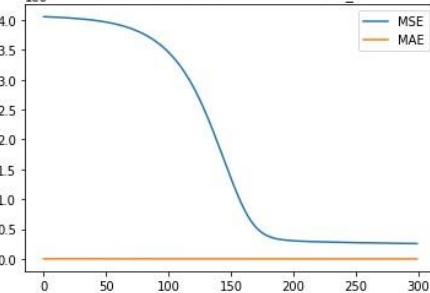
menggunakan bobot dan bias yang dihasilkan oleh model ketika dijalankan pada data uji.

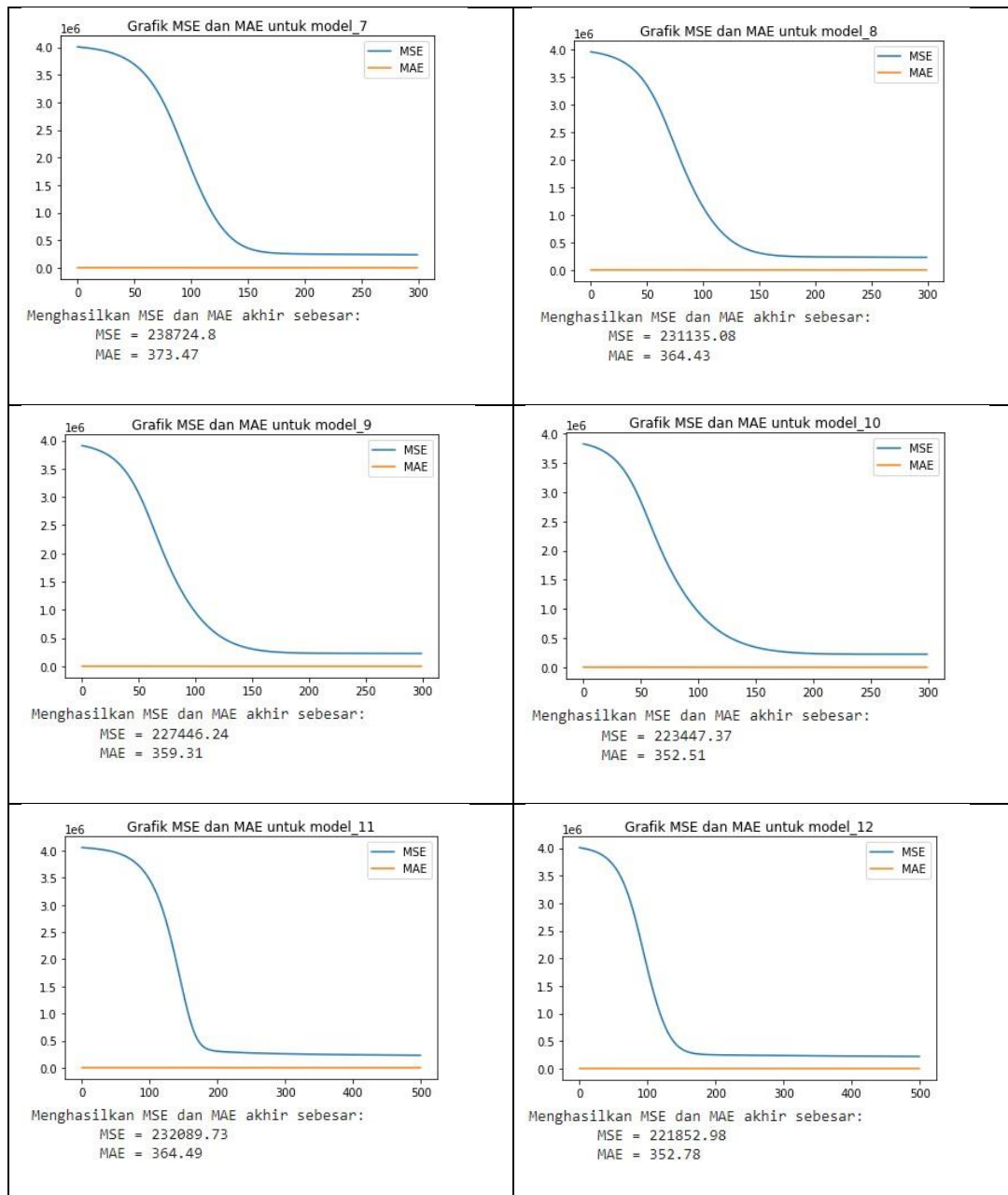
Langkah terakhir yang dilakukan dalam skenario uji adalah mengetahui tingkat *error* dari metode LSTM yang telah dibangun dan diuji menggunakan pengukuran *Mean Squared Error* dan *Mean Average Error* pada rumus 4.1 dan 4.2.

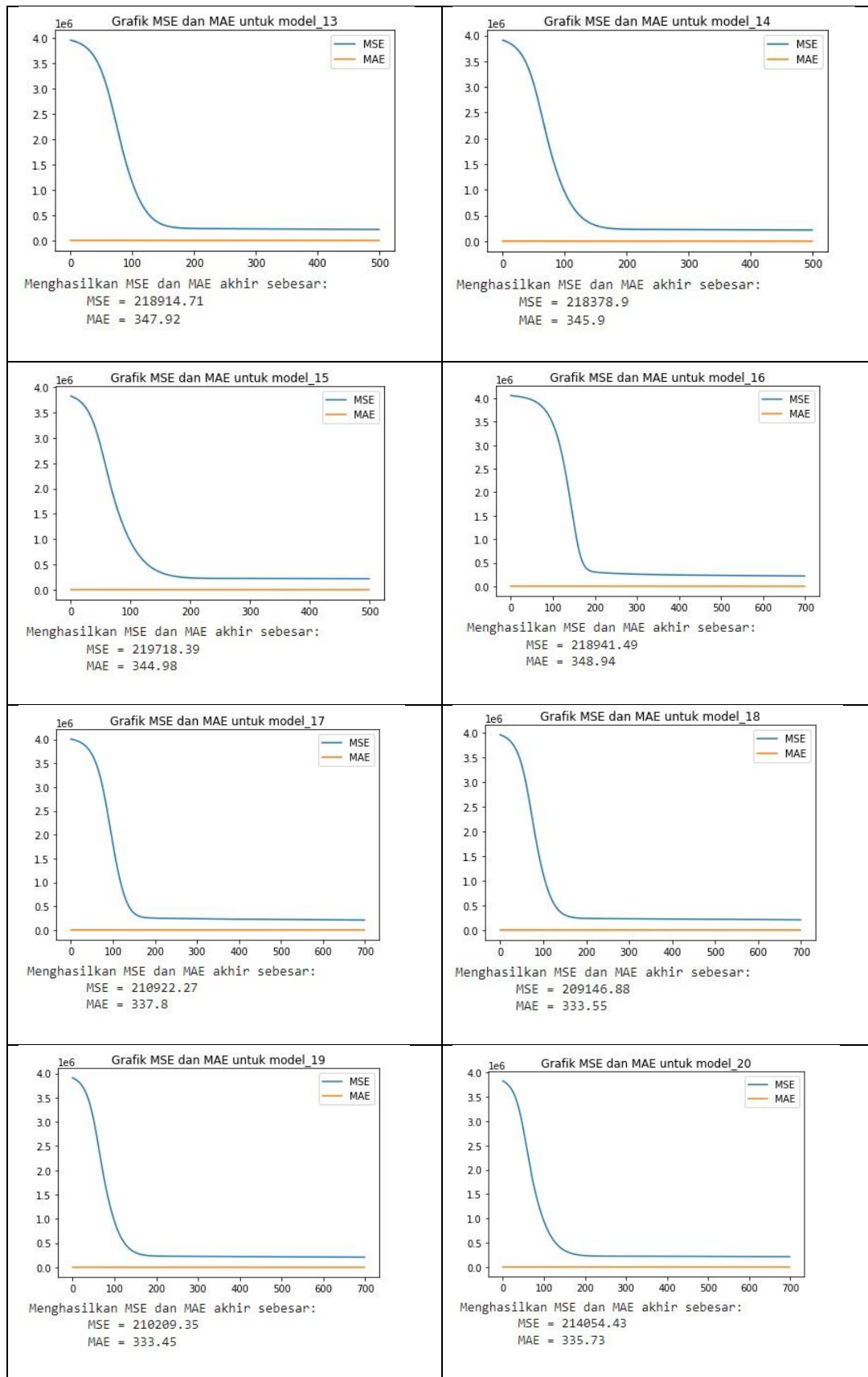
4.2 Hasil Uji Coba

Pada sub-bab hasil uji coba ini, peneliti akan memaparkan hasil uji coba yang telah dilakukan dengan berbasis pada skenario uji coba yang telah dibuat sebelumnya. Di dalam skenario uji coba terdapat model-model yang telah dibentuk pada proses training sebelumnya. Pada proses training, model-model dibentuk dengan melakukan pengkombinasian antara jumlah *epoch* dan jumlah *batch size* lalu dilakukan proses training sehingga masing-masing model menghasilkan kombinasi bobot dan bias dan hasil pengukuran *error* terhadap prediksi kasus harian covid-19. Hasil pengukuran MSE dan MAE di masing-masing model dapat dilihat pada tabel 4.1 berikut.

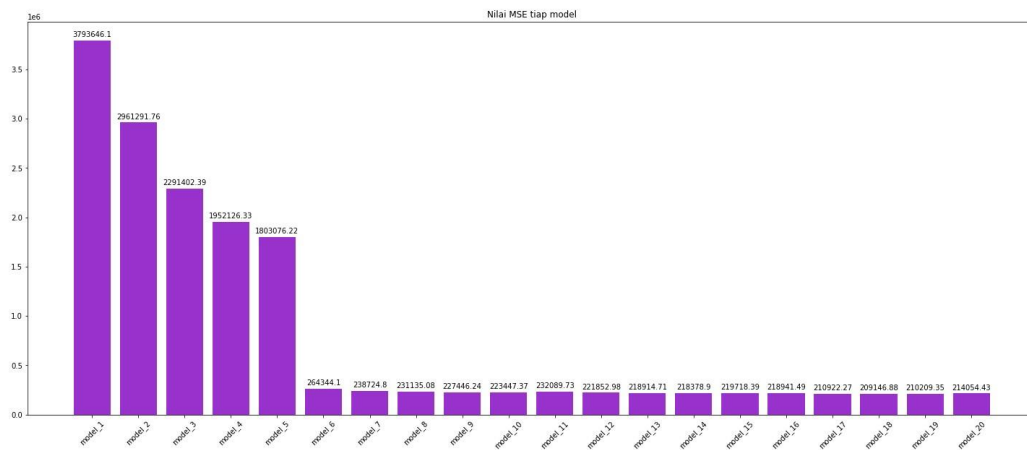
Tabel 4.4 Perbandingan grafik pengukuran MSE dan MAE tiap-tiap model

Grafik Pengukuran MSE dan MAE pada model training	
<p>Grafik MSE dan MAE untuk model_1</p>  <p>Menghasilkan MSE dan MAE akhir sebesar: MSE = 3793646.1 MAE = 1680.35</p>	<p>Grafik MSE dan MAE untuk model_2</p>  <p>Menghasilkan MSE dan MAE akhir sebesar: MSE = 2961291.76 MAE = 1486.16</p>
<p>Grafik MSE dan MAE untuk model_3</p>  <p>Menghasilkan MSE dan MAE akhir sebesar: MSE = 2291402.39 MAE = 1296.65</p>	<p>Grafik MSE dan MAE untuk model_4</p>  <p>Menghasilkan MSE dan MAE akhir sebesar: MSE = 1952126.33 MAE = 1190.0</p>
<p>Grafik MSE dan MAE untuk model_5</p>  <p>Menghasilkan MSE dan MAE akhir sebesar: MSE = 1803076.22 MAE = 1139.08</p>	<p>Grafik MSE dan MAE untuk model_6</p>  <p>Menghasilkan MSE dan MAE akhir sebesar: MSE = 264344.1 MAE = 397.04</p>

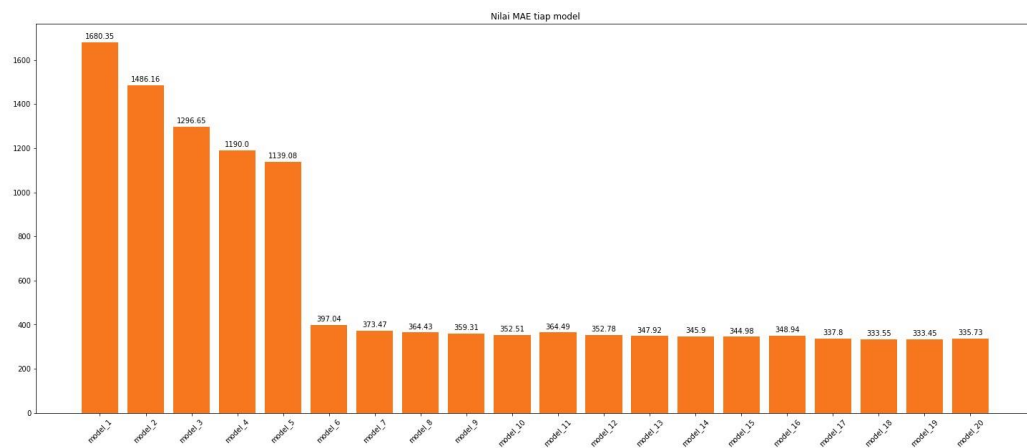




Dari uji coba pemilihan model terbaik pada tabel 4.1 dan 4.2 dapat diambil kesimpulan yang ditunjukkan pada gambar berikut.



Gambar 4.1 Pengukuran MSE di setiap model training



Gambar 4.2 Pengukuran MAE di setiap model *training*

Berdasarkan grafik perbandingan hasil pengukuran *error* terhadap model *training* dengan variasi *batch size* dan *epoch* pada gambar diatas, model 18 memiliki nilai *error* MSE paling rendah yaitu 209146.88 jika dibandingkan dengan lainnya. Sehingga, berdasarkan hal tersebut maka kombinasi parameter yang digunakan dalam penelitian kali ini adalah dengan kombinasi yang diberikan oleh model 18 yaitu:

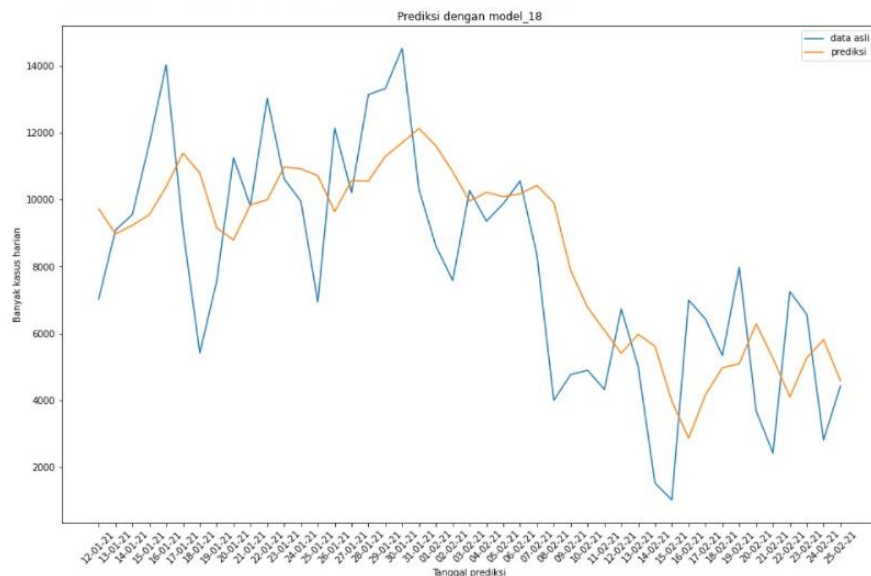
Tabel 4.5 Tabel parameter yang digunakan dalam tahap uji coba menggunakan data uji

Nama Model	Epoch	Batch size
Model 18	800	15

Data yang digunakan dalam proses uji coba merupakan data uji dari data sekunder yang memiliki sumber data yang sama dengan data training. Masing-masing model diharapkan agar dapat melakukan prediksi selama 35 hari setelah tanggal terakhir yang diinputkan ke dalam model dalam proses uji coba tersebut dengan tingkat *error* yang se-minimal mungkin.

Berikut ini akan ditampilkan hasil uji coba prediksi menggunakan kombinasi bobot dan bias pada model 18 menggunakan data uji:

```
Prediksi dengan spesifikasi model:
Nama model: model_18
Epoch: 800
Batch size: 15
Data uji dimulai dari tanggal 28-12-20 hingga 11-01-21
Expected output:
Hasil prediksi jumlah kasus harian dari tanggal 12-01-21 hingga 25-02-21
```



Gambar 4.3 Grafik hasil perbandingan hasil prediksi dan data asli kasus harian covid-19

Tabel 4.6 Hasil Prediksi ditampilkan dalam tampilan tabel

Hasil prediksi dengan normalisasi			Hasil prediksi setelah denormalisasi		
Tanggal	Hasil Prediksi	Data aktual	Tanggal	Hasil Prediksi	Data aktual
12-01-21	0.669696	0.483957	12-01-21	9723	7026
13-01-21	0.618172	0.626039	13-01-21	8975	9089
14-01-21	0.635849	0.658241	14-01-21	9231	9556
15-01-21	0.657498	0.803786	15-01-21	9546	11669
16-01-21	0.715105	0.966066	16-01-21	10382	14025
17-01-21	0.784230	0.627078	17-01-21	11385	9104
18-01-21	0.743315	0.373038	18-01-21	10791	5416
19-01-21	0.630616	0.520660	19-01-21	9155	7559
20-01-21	0.605403	0.774931	20-01-21	8789	11250
21-01-21	0.678016	0.675092	21-01-21	9843	9801
22-01-21	0.688346	0.897738	22-01-21	9993	13033
23-01-21	0.755913	0.731417	23-01-21	10974	10619
24-01-21	0.752081	0.684903	24-01-21	10919	9943
25-01-21	0.737896	0.477839	25-01-21	10713	6937
26-01-21	0.664069	0.835642	26-01-21	9641	12132
27-01-21	0.727901	0.703370	27-01-21	10568	10212
28-01-21	0.727302	0.905009	28-01-21	10559	13139
29-01-21	0.777501	0.917359	29-01-21	11288	13318
30-01-21	0.805898	1.000000	30-01-21	11700	14518
31-01-21	0.835148	0.709488	31-01-21	12125	10300
01-02-21	0.799641	0.593259	01-02-21	11609	8613
02-02-21	0.746312	0.522276	02-02-21	10835	7582
03-02-21	0.685623	0.707525	03-02-21	9954	10272
04-02-21	0.703609	0.644044	04-02-21	10215	9350
05-02-21	0.694836	0.680402	05-02-21	10088	9878
06-02-21	0.700710	0.727378	06-02-21	10173	10560
07-02-21	0.718010	0.573984	07-02-21	10424	8333
08-02-21	0.682002	0.275623	08-02-21	9901	4001
09-02-21	0.542603	0.328486	09-02-21	7878	4769
10-02-21	0.466574	0.337258	10-02-21	6774	4896
11-02-21	0.419756	0.297899	11-02-21	6094	4325
12-02-21	0.372163	0.463412	12-02-21	5403	6728
13-02-21	0.411437	0.345106	13-02-21	5973	5010
14-02-21	0.386749	0.105148	14-02-21	5615	1527
15-02-21	0.273415	0.070175	15-02-21	3969	1019
16-02-21	0.197470	0.481879	16-02-21	2867	6996
17-02-21	0.287372	0.442405	17-02-21	4172	6423
18-02-21	0.342678	0.367613	18-02-21	4975	5337
19-02-21	0.350651	0.549400	19-02-21	5091	7976
20-02-21	0.433298	0.253924	20-02-21	6291	3686
21-02-21	0.361806	0.166898	21-02-21	5253	2423
22-02-21	0.282336	0.499307	22-02-21	4099	7249
23-02-21	0.362656	0.452562	23-02-21	5265	6570
24-02-21	0.400079	0.193790	24-02-21	5808	2813
25-02-21	0.316050	0.304594	25-02-21	4588	4422

Dari hasil uji yang ditampilkan pada gambar 4.3, hasil prediksi tersebut dapat dilakukan penghitungan nilai MSE dan MAE pada tabel 4.7 berikut.

Tabel 4.7 Penghitungan MSE dan MAE dengan data normalisasi

Nomor	Tanggal	Label (Y)	Prediksi (Y')	Penghitungan MSE		Penghitungan MAE	
				Label-prediksi	square(labels-pred)	Label-prediksi	absolute(label-prediksi)
1	12-01-21	0.48396	0.6696964	-0.18573984	0.034499287	-0.18573984	0.18573984
2	13-01-21	0.62604	0.6181724	0.00786637	6.18797E-05	0.00786637	0.00786637
3	14-01-21	0.65824	0.635849	0.02239201	0.000501402	0.02239201	0.02239201
4	15-01-21	0.80379	0.6574983	0.14628753	0.021400042	0.14628753	0.14628753
5	16-01-21	0.96607	0.7151047	0.2509618	0.062981826	0.2509618	0.2509618
6	17-01-21	0.62708	0.7842302	-0.15715261	0.024696942	-0.15715261	0.15715261
7	18-01-21	0.37304	0.7433151	-0.37027722	0.137105221	-0.37027722	0.37027722
8	19-01-21	0.52066	0.6306162	-0.10995603	0.012090328	-0.10995603	0.10995603
9	20-01-21	0.77493	0.6054026	0.16952818	0.028739803	0.16952818	0.16952818
10	21-01-21	0.67509	0.678016	-0.00292367	8.54785E-06	-0.00292367	0.00292367
11	22-01-21	0.89774	0.6883464	0.20939138	0.043844752	0.20939138	0.20939138
12	23-01-21	0.73142	0.7559127	-0.02449536	0.000600023	-0.02449536	0.02449536
13	24-01-21	0.6849	0.7520815	-0.06717841	0.004512939	-0.06717841	0.06717841
14	25-01-21	0.47784	0.7378964	-0.26005705	0.067629671	-0.26005705	0.26005705
15	26-01-21	0.83564	0.6640687	0.17157307	0.029437318	0.17157307	0.17157307
16	27-01-21	0.70337	0.7279015	-0.02453119	0.000601779	-0.02453119	0.02453119
17	28-01-21	0.90501	0.7273021	0.17770719	0.031579844	0.17770719	0.17770719
18	29-01-21	0.91736	0.7775012	0.13985796	0.019560249	0.13985796	0.13985796
19	30-01-21	1	0.8058983	0.19410167	0.03767546	0.19410167	0.19410167
20	31-01-21	0.70949	0.835148	-0.12566042	0.01579054	-0.12566042	0.12566042
21	01-02-21	0.59326	0.7996408	-0.20638135	0.042593263	-0.20638135	0.20638135
22	02-02-21	0.52228	0.7463118	-0.22403569	0.050191992	-0.22403569	0.22403569
23	03-02-21	0.70753	0.685623	0.02190244	0.000479717	0.02190244	0.02190244
24	04-02-21	0.64404	0.7036085	-0.05956421	0.003547895	-0.05956421	0.05956421
25	05-02-21	0.6804	0.6948365	-0.01443484	0.000208365	-0.01443484	0.01443484
26	06-02-21	0.72738	0.70071	0.02666769	0.000711166	0.02666769	0.02666769
27	07-02-21	0.57398	0.71801	-0.14402567	0.020743394	-0.14402567	0.14402567
28	08-02-21	0.27562	0.6820019	-0.40637864	0.165143601	-0.40637864	0.40637864
29	09-02-21	0.32849	0.5426031	-0.21411744	0.045846279	-0.21411744	0.21411744
30	10-02-21	0.33726	0.466574	-0.12931635	0.01672272	-0.12931635	0.12931635
31	11-02-21	0.2979	0.4197559	-0.12185651	0.014849009	-0.12185651	0.12185651
32	12-02-21	0.46341	0.3721634	0.09124838	0.008326268	0.09124838	0.09124838
33	13-02-21	0.34511	0.4114371	-0.06633086	0.004399783	-0.06633086	0.06633086
34	14-02-21	0.10515	0.3867493	-0.28160151	0.079299413	-0.28160151	0.28160151
35	15-02-21	0.07018	0.2734146	-0.2032392	0.041306173	-0.2032392	0.2032392
36	16-02-21	0.48188	0.1974703	0.28440878	0.080888357	0.28440878	0.28440878
37	17-02-21	0.44241	0.287372	0.15503332	0.024035331	0.15503332	0.15503332
38	18-02-21	0.36761	0.3426777	0.02493538	0.000621773	0.02493538	0.02493538
39	19-02-21	0.5494	0.350651	0.19874881	0.03950109	0.19874881	0.19874881
40	20-02-21	0.25392	0.4332984	-0.17937413	0.032175078	-0.17937413	0.17937413
41	21-02-21	0.1669	0.3618059	-0.19490836	0.037989267	-0.19490836	0.19490836
42	22-02-21	0.49931	0.2823358	0.21697169	0.047076712	0.21697169	0.21697169
43	23-02-21	0.45256	0.3626557	0.08990665	0.008083206	0.08990665	0.08990665
44	24-02-21	0.19379	0.4000791	-0.20628867	0.042555015	-0.20628867	0.20628867
45	25-02-21	0.30459	0.3160504	-0.01145672	0.000131256	-0.01145672	0.01145672
Total					1.38074397	Total	6.59077227
Total/banyak data					0.0306832	Total/banyak data	0.14646161

Dimana, nilai MSE dan MAE tersebut didapatkan dengan melakukan penghitungan menggunakan rumus 4.1 dan 4.2 dengan nilai-nilai yang didefinisikan sebagai berikut:

N : Banyak sample data yang dihitung error-nya

Y_i : Data aktual hasil normalisasi atribut “Kasus baru” pada time step i dari groundtruth

Y'_i : Hasil prediksi hasil normalisasi atribut “Kasus baru” pada time step i dari keluaran

Berdasarkan pendefinisian dan penghitungan pada tabel 4.7 tersebut didapatkan nilai MSE dan MAE seperti yang ditunjukkan pada tabel berikut ini:

Tabel 4.8 Hasil uji coba prediksi pada data uji normalisasi

<i>MSE</i>	0.031
<i>MAE</i>	0.147

Tabel 4.9 Penghitungan MSE dan MAE dengan data denormalisasi

Nomor	Tanggal	Label (Y)	Prediksi (Y')	Penghitungan MSE		Penghitungan MAE	
				Label-prediksi	square(labels-pred)	Label-prediksi	absolute(label-prediksi)
1	12-01-21	10047	11656	-1609	2588881	-1609	1609
2	13-01-21	11278	11210	68	4624	68	68
3	14-01-21	11557	11363	194	37636	194	194
4	15-01-21	12818	11551	1267	1605289	1267	1267
5	16-01-21	14224	12050	2174	4726276	2174	2174
6	17-01-21	11287	12649	-1362	1855044	-1362	1362
7	18-01-21	9086	12294	-3208	10291264	-3208	3208
8	19-01-21	10365	11318	-953	908209	-953	953
9	20-01-21	12568	11099	1469	2157961	1469	1469
10	21-01-21	11703	11728	-25	625	-25	25
11	22-01-21	13632	11818	1814	3290596	1814	1814
12	23-01-21	12191	12403	-212	44944	-212	212
13	24-01-21	11788	12370	-582	338724	-582	582
14	25-01-21	9994	12247	-2253	5076009	-2253	2253
15	26-01-21	13094	11607	1487	2211169	1487	1487
16	27-01-21	11948	12161	-213	45369	-213	213
17	28-01-21	13695	12155	1540	2371600	1540	1540
18	29-01-21	13802	12590	1212	1468944	1212	1212
19	30-01-21	14518	12836	1682	2829124	1682	1682
20	31-01-21	12001	13090	-1089	1185921	-1089	1089
21	01-02-21	10994	12782	-1788	3196944	-1788	1788
22	02-02-21	10379	12320	-1941	3767481	-1941	1941
23	03-02-21	11984	11794	190	36100	190	190
24	04-02-21	11434	11950	-516	266256	-516	516
25	05-02-21	11749	11874	-125	15625	-125	125
26	06-02-21	12156	11925	231	53361	231	231
27	07-02-21	10827	12075	-1248	1557504	-1248	1248
28	08-02-21	8242	11763	-3521	12397441	-3521	3521
29	09-02-21	8700	10555	-1855	3441025	-1855	1855
30	10-02-21	8776	9896	-1120	1254400	-1120	1120
31	11-02-21	8435	9491	-1056	1115136	-1056	1056
32	12-02-21	9869	9078	791	625681	791	791
33	13-02-21	8844	9419	-575	330625	-575	575
34	14-02-21	6765	9205	-2440	5953600	-2440	2440
35	15-02-21	6462	8223	-1761	3101121	-1761	1761
36	16-02-21	10029	7565	2464	6071296	2464	2464
37	17-02-21	9687	8344	1343	1803649	1343	1343
38	18-02-21	9039	8823	216	46656	216	216
39	19-02-21	10614	8892	1722	2965284	1722	1722
40	20-02-21	8054	9608	-1554	2414916	-1554	1554
41	21-02-21	7300	8989	-1689	2852721	-1689	1689
42	22-02-21	10180	8300	1880	3534400	1880	1880
43	23-02-21	9775	8996	779	606841	779	779
44	24-02-21	7533	9320	-1787	3193369	-1787	1787
45	25-02-21	8493	8592	-99	9801	-99	99
Total					103649442	Total	57104
Total/banyak data					2303320.93	Total/banyak data	1268.977

Dimana, nilai MSE dan MAE tersebut didapatkan dengan melakukan penghitungan menggunakan rumus 4.1 dan 4.2 dengan nilai-nilai yang didefinisikan sebagai berikut:

N : Banyak sample data yang dihitung error-nya

Y_i : Data aktual hasil denormalisasi atribut “Kasus baru” pada time step i dari

groundtruth

Y_i : Hasil prediksi hasil denormalisasi atribut “Kasus baru” pada time step i dari keluaran

Berdasarkan pendefinisian dan penghitungan pada tabel 4.8 tersebut didapatkan nilai MSE dan MAE seperti yang ditunjukkan pada tabel berikut ini:

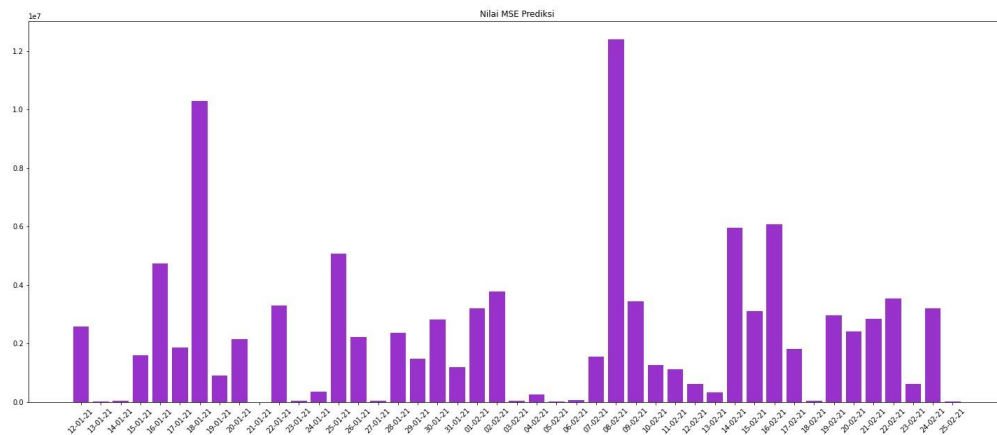
Tabel 4.10 Hasil uji coba prediksi pada data denormalisasi

<i>MSE</i>	2303320.93
<i>MAE</i>	1268.98

4.3 Pembahasan

Setelah proses pengujian selesai, maka akan dapat diketahui tingkat *error* dari prediksi yang dilakukan oleh model terpilih selama 35 hari selanjutnya. Dalam penghitungan *Mean Squared Error* dan *Mean Absolute Error* testing didapatkan kesimpulan bahwa pengujian algoritma LSTM menggunakan parameter dari model 18 dan menggunakan data uji menunjukkan hasil pengujian dengan tingkat *error* MSE dan MAE pada data normalisasi masing-masing sebesar 0.031 dan 0.147 dan pada data denormalisasi masing-masing sebesar 2303320.93 dan 1268.98.

Untuk mempermudah menganalisa nilai *error* pada hasil prediksi, maka disajikan dalam bentuk diagram batang sebagai berikut.



Gambar 4.4 Grafik MSE pada proses prediksi

Diagram batang pada gambar 4.4 menunjukkan bahwa prediksi yang dilakukan pada kasus harian covid-19 menggunakan data uji memiliki tingkat error MSE paling rendah pada tanggal 21 Januari 2021 seperti yang ditunjukkan pada tabel 4.11. Nilai MAE pada tanggal tersebut sebesar 25 dan MSE sebesar 625. Walaupun demikian, nilai MSE menunjukkan angka yang besar padahal jika dihitung secara persentase rata-rata absolut *error* terhadap nilai aktualnya menunjukkan angka yang tidak terlalu besar. Hal ini dikarenakan perhitungan *Mean Squared Error* sangat terikat dengan konsep dasar variansi. Sama seperti yang terdapat di variansi atau rata-rata yang digunakan dalam variansi rentan terhadap data dengan selisih yang besar dari rata-rata data yang digunakan. Angka MSE dan MAE yang besar tersebut disebabkan karena data yang digunakan dalam proses pengujian merupakan data dengan nilai belasan ribu kasus harian sehingga menyebabkan penghitungan selisih hasil prediksi dan *groundtruth* yang merupakan dasar dari penghitungan MSE dan MAE menjadi lebih besar dibanding data dengan rentang nilai yang lebih kecil. Dan ketika melakukan perhitungan menemukan data dengan selisih yang cukup besar, maka ketika selisih tersebut masuk ke dalam

rumus *squared* yang merupakan kuadrat dari selisih akan menghasilkan angka *error* yang tinggi (Agrawal, 2021). Hal tersebut menunjukkan nilai MSE yang besar tidak berarti bahwa prediksi yang dilakukan memiliki persentase rata-rata absolut yang tinggi karena pengukuran MSE bergantung kepada jenis data yang digunakan pada sistem yang dibangun dan pada penelitian kali ini penulis menggunakan data yang menyebabkan angka tersebut menjadi tinggi daripada nilai persentase rata-rata absolutnya.

Dari grafik pada gambar 4.3 tersebut menunjukkan bahwa kasus penyebaran covid-19 memiliki kemungkinan untuk dilakukan prediksi namun dengan toleransi *error* seperti yang telah disebutkan. Hal tersebut disebabkan karena tingkat *error* yang tidak dapat lebih kecil lagi karena dapat disebabkan oleh perubahan jumlah kasus harian yang tidak terduga seperti disebabkan oleh perubahan kebijakan penanganan pandemi yang berlaku pada saat pandemi berlangsung sehingga mempengaruhi grafik kasus harian covid-19.

Nilai MSE dan MAE dari penelitian yang dilakukan menunjukkan angka yang tinggi karena tidak mudahnya untuk memprediksi data kasus harian covid-19. Hal ini menunjukkan perlunya perbaikan untuk penelitian yang akan datang pada sistem yang dibangun agar prediksi yang dilakukan memiliki tingkat akurasi yang lebih tinggi dan *error* yang lebih rendah lagi.

Penghitungan tingkat *error* dari prediksi atau perkiraan memiliki banyak manfaat salah satu diantaranya adalah untuk menambahkan pertimbangan dalam penentuan metode yang digunakan dalam prediksi di berbagai bidang. Seperti kegiatan pengukuran tingkat *error* prediksi yang peneliti lakukan dalam penelitian

kali ini, yaitu pengukuran *error* pada prediksi terhadap jumlah kasus harian covid-19, yang memiliki tujuan untuk mengetahui tingkat *error* yang diberikan oleh algoritma LSTM dengan melakukan prediksi terhadap kemungkinan jumlah kasus harian covid-19 yang akan terjadi di masa mendatang dan ingin memberikan manfaat kepada pengambil keputusan agar dapat melakukan efisiensi dalam pemanfaatan sumber daya dengan baik dan tepat.

Dalam ajaran islam, melakukan kegiatan yang bertujuan untuk memberikan manfaat bagi sesama manusia adalah hal yang sangat dianjurkan seperti yang disampaikan pada salah satu ayat Al Baqarah[2] ayat 19 di dalam Al Quran yang berbunyi :

وَأَنْفِقُوا فِي سَبِيلِ اللَّهِ وَلَا تُلْقُوا بِأَيْدِيكُمْ إِلَى التَّهْلُكَةِ وَأَحْسِنُوا إِنَّ اللَّهَ يُحِبُّ الْمُحْسِنِينَ

“Dan belanjakanlah (harta bendamu) di jalan Allah, dan janganlah kamu menjatuhkan tangan (diri) kamu ke dalam kebinasaan dan berbuat baiklah. Sesungguhnya Allah menyukai orang-orang yang muhsin (orang yang selalu berbuat yang lebih baik) ” (Q.S. Al-Baqarah [2]: 19).

Dan dalam potongan surat Al Maidah ayat 2 Al-Quran yang berbunyi :

... وَتَعَاوَنُوا عَلَى الْبِرِّ وَالتَّقْوَىٰ وَلَا تَعَاوَنُوا عَلَى الْإِثْمِ وَالْعُدْوَانِ ۚ وَاتَّقُوا اللَّهَ إِنَّ اللَّهَ شَدِيدُ الْعِقَابِ

“... Dan tolong-menolonglah kamu dalam (mengerjakan) kebajikan dan takwa, dan jangan tolong-menolong dalam berbuat dosa dan pelanggaran. Dan bertakwalah kamu kepada Allah, sesungguhnya Allah amat berat siksa-Nya.” (Q.S. Al-Maidah [5]: 2).

Maka dengan adanya penelitian kali ini, merupakan salah satu cara yang dilakukan untuk memberikan manfaat dengan cara melakukan prediksi terhadap jumlah kasus harian covid-19 menggunakan algoritma LSTM yang peneliti harapkan agar mengetahui tingkat *error* dari algoritma tersebut dalam

memperkirakan jumlah kasus harian covid-19. Dengan begitu dapat mempermudah dan memberikan pertimbangan terhadap para pengambil keputusan yang sedang bekerja maupun yang melakukan penelitian dalam hal prediksi baik prediksi kasus covid-19 ataupun kegiatan prediksi dalam bidang lain.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian yang dilakukan merupakan penelitian terhadap prediksi jumlah kasus harian covid-19 menggunakan algoritma *Long Short Term Memory*. Dalam penelitian ini, peneliti menggunakan data sekunder yang didapatkan dari kawalcovid19.id yang merupakan dataset yang dikumpulkan secara independen oleh banyak peneliti maupun relawan dan berisi data catatan perkembangan kasus covid-19 yang terjadi di indonesia yang disajikan dalam urutan time series. Data tersebut peneliti bagi menjadi dua yaitu data *training* dan data uji. Pembentukan model yang digunakan untuk pengujian merupakan hasil dari proses *training* yang menghasilkan variasi *epoch*, *batch size*, serta parameter bobot dan bias dengan tingkat *error* prediksi yang paling rendah. Model terpilih tersebut selanjutnya dilakukan pengukuran *error* melalui prediksi untuk kasus harian dengan durasi 30 setelah hari terakhir yang dimasukkan sebagai input dalam proses pengujian menggunakan data uji. Dimana proses pengujian tersebut menghasilkan kesimpulan bahwa algoritma LSTM dengan menggunakan parameter bobot dan bias hasil training serta epoch sejumlah 800 dan batch size 30 menghasilkan tingkat *error* dengan nilai tingkat error MSE dan MAE pada data normalisasi masing-masing sebesar 0.03 dan 0.15 dan pada data denormalisasi masing-masing sebesar 2303320.93 dan 1268.98.

5.2 Saran

Peneliti sadar bahwa dalam penelitian ini masih memiliki kekurangan, sehingga untuk penelitian selanjutnya peneliti harapkan akan adanya pengembangan dan perbaikan agar hasil yang didapatkan semakin baik. Berikut adalah saran dari peneliti yang diharapkan dapat membantu bagi penelitian selanjutnya:

1. Fitur yang digunakan dalam penelitian dapat ditambah menyesuaikan dengan dataset yang didapat.
2. Penelitian selanjutnya dapat menggunakan metode optimasi pada proses update bobot.
3. Penelitian selanjutnya dapat menggunakan *error metrics* MAPE.

DAFTAR PUSTAKA

- Agrawal, R. (2021, May 29). *Evaluation Metrics for Your Regression Model—Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/05/know-the-best-evaluation-metrics-for-your-regression-model/>
- Ahmad, A., & Brown, G. (2009). *Data transformation for decision tree ensembles*.
- Aldi, M. W. P., Jondri, J., & Aditsania, A. (2018). Analisis dan Implementasi Long Short Term Memory Neural Network untuk Prediksi Harga Bitcoin. *EProceedings of Engineering*, 5(2).
- Arfan, A. (2019). Prediksi Harga Saham Di Indonesia Menggunakan Algoritma Long Short-Term Memory. *Prosiding SeNTIK*, 3(1).
- Arulprakash, E., & Aruldoss, M. (2020). A Study on Fight Against COVID-19 from Latest Technological Intervention. *SN Computer Science*, 1(5), 277. <https://doi.org/10.1007/s42979-020-00301-0>
- Aulia, N. (2020). *Prediksi Harga Ethereum Berdasarkan Informasi Blockchain Menggunakan Metode Long Short Term Memory*.
- Brownlee, J. (2017, June 22). A Gentle Introduction to Backpropagation Through Time. *Machine Learning Mastery*. <https://machinelearningmastery.com/gentle-introduction-backpropagation-time/>
- Brownlee, J. (2018, July 20). *Difference Between a Batch and an Epoch in a Neural Network*. <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/#:~:text=The%20batch%20size%20is%20a%20number%20of%20samples%20processed%20before,samples%20in%20the%20training%20dataset.>
- Brownlee, J. (2020, September 12). *Understand the Impact of Learning Rate on Neural Network Performance*. <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>
- Chaurasia, V., & Pal, S. (2020). COVID-19 Pandemic: ARIMA and Regression Model-Based Worldwide Death Cases Predictions. *SN Computer Science*, 1(5), 288. <https://doi.org/10.1007/s42979-020-00298-6>
- Farooq, J., & Bazaz, M. A. (2020). A deep learning algorithm for modeling and forecasting of COVID-19 in five worst affected states of India. *Alexandria Engineering Journal*. <https://doi.org/10.1016/j.aej.2020.09.037>

- KawalCovid19. (2020, November 17). *KawalCOVID19 untuk mengurai derasnya informasi seputar COVID-19*. <https://kawalcovid19.id/tentang-kami>
- Kırbaç, İ., Sözen, A., Tuncer, A. D., & Kazancıoğlu, F. Ş. (2020). Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches. *Chaos, Solitons & Fractals*, 138, 110015. <https://doi.org/10.1016/j.chaos.2020.110015>
- Loukas, S. (2020, May 28). *Everything you need to know about Min-Max normalization: A Python tutorial*. <https://towardsdatascience.com/everything-you-need-to-know-about-min-max-normalization-in-python-b79592732b79>
- Muhammad, L. J., Islam, Md. M., Usman, S. S., & Ayon, S. I. (2020). Predictive Data Mining Models for Novel Coronavirus (COVID-19) Infected Patients' Recovery. *SN Computer Science*, 1(4), 206. <https://doi.org/10.1007/s42979-020-00216-w>
- Olah, C. (2015, August 27). Understanding LSTM Networks. *Understanding LSTM Networks*. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Rahman, M., Islam, M., Shimanto, M. H., Ferdous, J., Rahman, A. A.-N. S., Sagor, P. S., & Chowdhury, T. (2020). A global analysis on the effect of temperature, socio-economic and environmental factors on the spread and mortality rate of the COVID-19 pandemic. *Environment, Development and Sustainability*. <https://doi.org/10.1007/s10668-020-01028-x>
- Rasjid, Z. E., Setiawan, R., & Effendi, A. (2021). A Comparison: Prediction of Death and Infected COVID-19 Cases in Indonesia Using Time Series Smoothing and LSTM Neural Network. *5th International Conference on Computer Science and Computational Intelligence 2020*, 179, 982–988. <https://doi.org/10.1016/j.procs.2021.01.102>
- Sagheer, A., & Kotb, M. (2019). Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*, 323, 203–213.
- Shihab, M. Q. (2020). *Al-Quran dan Maknanya*. Lentera Hati.
- Steven Sen, Dedy Sugiarto, & Abdul Rochman. (2020). Rice Price Prediction Using Multilayer Perceptron (MLP) and Long Short Term Memory (LSTM). *Ultimatics : Jurnal Teknik Informatika*, 12(1). <https://doi.org/10.31937/ti.v12i1.1572>
- Yashwanth, N. V. S. (2021, January 1). *Evaluation metrics & Model Selection in Linear Regression*. Medium. <https://towardsdatascience.com/evaluation-metrics-model-selection-in-linear-regression-73c7573208be>

Yeşilkanat, C. M. (2020). Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm. *Chaos, Solitons & Fractals*, 140, 110210.
<https://doi.org/10.1016/j.chaos.2020.110210>