

**PERAMALAN JUMLAH KASUS COVID-19
MENGUNAKAN *JOINT LEARNING***

SKRIPSI



**UIN SUNAN AMPEL
S U R A B A Y A**

Disusun Oleh:

**MUHAMMAD RIZQI NUR
NIM: H76217063**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA
2022**

PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini,

Nama : Muhammad Rizqi Nur

NIM : H76217063

Program Studi : Sistem Informasi

Angkatan : 2017

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul: “PERAMALAN JUMLAH KASUS COVID-19 MENGGUNAKAN JOINT LEARNING”. Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 12 Januari 2022

Yang menyatakan,



Muhammad Rizqi Nur

NIM H76217063

LEMBAR PERSETUJUAN PEMBIMBING

JUDUL : PERAMALAN JUMLAH KASUS COVID-19
MENGGUNAKAN JOINT LEARNING
NAMA : MUHAMMAD RIZQI NUR
NIM : H76217063

Mahasiswa tersebut telah melakukan proses bimbingan dan dinyatakan layak
untuk mengikuti Sidang Skripsi.

Surabaya, 13 Desember 2021

Menyetujui,

Dosen Pembimbing 1



(Faris Mushlihul Amin, M. Kom)

NIP. 198808132014031001

Dosen Pembimbing 2



(Ahmad Yusuf, M. Kom)

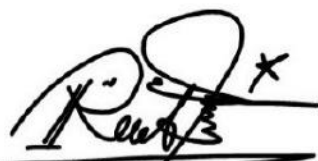
NIP. 199001202014031003

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi Muhammad Rizqi Nur ini telah dipertahankan
di depan tim penguji skripsi
Surabaya, 23 Desember 2021

Mengesahkan,
Dewan Penguji

Penguji I



(Mujib Ridwan, S.Kom., M.T.)
NIP 198604272014031004

Penguji II



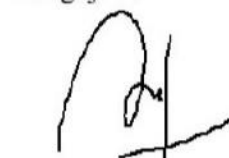
(Khalid, M.Kom)
NIP 197906092014031002

Penguji III



(Faris Mushlihul Amin, M.Kom)
NIP 198808132014031001

Penguji IV



(Ahmad Yusuf, M.Kom)
NIP 199001202014031003

Mengetahui,
Dekan Fakultas Sains dan Teknologi
UIN Sunan Ampel Surabaya



Prof. Dr. Ir. Hj. Fatimatur Rusydiyah, M.Ag.
NIP 197312272005012003



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN

Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : Muhammad Rizqi Nur
NIM : H76217063
Fakultas/Jurusan : Sains dan Teknologi / Sistem Informasi
E-mail address : rizqinur2010@gmail.com

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :

☒ Skripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (.....)

yang berjudul :

PERAMALAN JUMLAH KASUS COVID-19 MENGGUNAKAN JOINT LEARNING

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Eksklusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara **fulltext** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 12 Januari 2022

Penulis

(Muhammad Rizqi Nur)

ABSTRACT

COVID-19 CASE COUNT FORECASTING USING JOINT LEARNING

COVID-19 is a dangerous illness because it spreads quickly and easily. Vaccines are already available but the pandemic isn't likely to end quick. Forecasting is hoped to help handle the pandemic. Deep learning, specially LSTM, has been used to forecast COVID-19 case count in some regions. However, deep learning models need a lot of data while COVID-19 daily data are scarce. Still, COVID-19 doesn't happen in just one region but also in neighboring regions. This research aims to use joint learning with data from other regions to improve model performance with fewer data and to use the model to forecast until 9 months since the date of last data taken. Joint learning is done by making models share some modules and training the models together. To overcome the different data scale and pandemic age in the regions, the data was first transformed into SIRD variables and was evaluated using RMSSE. Joint learning failed to improve the model performance in this research. The proposed model performance is better than ARIMA-SIRD and SIRD model but isn't better than normal encoder-decoder LSTM and only reaches RMSSE below one occasionally. Pandemic in Kab. Sampang, Kab. Tuban, Kota Madiun, and Kota Pasuruan which are in the same cluster were forecasted to end quickly in May 2021, while the active case count in Kab. Bojonegoro was forecasted to increase drastically in June 2021 to its peak in late July after infecting 78.48% of the population. All of those forecasts are way off because the models were only trained to forecast 14 days ahead and there was the new delta variant which weren't in the training data. Solely for Kab. Bojonegoro, the model captured the shape pretty well, although the scale is way off.

Keywords: joint learning, SIRD, COVID-19, forecasting, deep learning, RMSSE

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	ii
LEMBAR PERSETUJUAN PEMBIMBING	iii
PENGESAHAN TIM PENGUJI SKRIPSI.....	iv
LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	v
KATA PENGANTAR	vi
DAFTAR ISI.....	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN.....	xxiii
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah.....	4
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	5
1.6. Sistematika Penulisan Skripsi	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Tinjauan Penelitian Terdahulu	6
2.2 Teori Dasar yang Digunakan	10
2.2.1 COVID-19	10
2.2.2 Model SIRD.....	12
2.2.3 Peramalan	12
2.2.4 Machine Learning	15
2.2.5 Clustering.....	16
2.2.6 Deep Learning.....	19
2.2.7 Recurrent Neural Network.....	21
2.2.8 One-Dimensional Convolution	23

DAFTAR TABEL

Tabel 2.1 Nilai q_{α} untuk Bonferroni-Dunn test (Demšar, 2006).....	29
Tabel 3.1 <i>Hyperparameter</i>	47
Tabel 4.1. Hasil observasi tanggal terhadap pola jumlah kasus positif aktif (I), perubahannya (Delta I), dan beta.....	54
Tabel 4.2 Keunggulan dan kekurangan transformasi variabel SIRD diskrit dalam membuat <i>time series</i> menjadi stasioner	62
Tabel 4.3 Nilai <i>silhouette</i> dan jumlah <i>single cluster</i> untuk tiap jumlah cluster (n) terpilih untuk tiap kelompok	63
Tabel 4.4 Hasil <i>clustering</i> untuk kelompok pembatasan data 90 hari.....	63
Tabel 4.5 Hasil <i>clustering</i> untuk kelompok 180 hari	64
Tabel 4.6 Hasil <i>clustering</i> untuk kelompok 366 hari	64
Tabel 4.7 Hasil <i>clustering</i> untuk kelompok sebelum vaksinasi	65
Tabel 4.8 <i>Hyperparameter</i> terbaik yang ditemukan Optuna untuk model utama dengan kolom K=konvergen	70
Tabel 4.9 Perbandingan <i>tuning</i> model utama dengan model <i>baseline</i>	72
Tabel 4.10 <i>Epoch</i> tiap model untuk tiap kabupaten/kota untuk kelompok 90 hari dalam bentuk <i>epoch</i> terbaik/total <i>epoch</i>	74
Tabel 4.11 Hasil uji statistik rata-rata <i>rank</i> untuk kelompok 90 hari.....	80
Tabel 4.12 <i>Epoch</i> tiap model untuk tiap kabupaten/kota untuk kelompok 180 hari dalam bentuk <i>epoch</i> terbaik/total <i>epoch</i>	82
Tabel 4.13 Hasil uji statistik rata-rata <i>rank</i> untuk kelompok 180 hari	86
Tabel 4.14 <i>Epoch</i> tiap model untuk tiap kabupaten/kota untuk kelompok 366 hari dalam bentuk <i>epoch</i> terbaik/total <i>epoch</i>	88
Tabel 4.15 Hasil uji statistik rata-rata <i>rank</i> untuk kelompok 366 hari	93
Tabel 4.16 <i>Epoch</i> tiap model untuk tiap kabupaten/kota untuk kelompok sebelum vaksinasi dalam bentuk <i>epoch</i> terbaik/total <i>epoch</i>	95
Tabel 4.17 Hasil uji statistik rata-rata <i>rank</i> untuk kelompok sebelum vaksinasi	100
Tabel 4.18 Persentase pencapaian RMSSE di bawah 1 untuk tiap kelompok	103
Tabel 4.19 Persentase pencapaian RMSSE di bawah 1 untuk tiap label	103

DAFTAR LAMPIRAN

Lampiran 1 Tanggal PSBB di Jawa Timur	143
Lampiran 2 Tanggal PPKM	144
Lampiran 3 Tanggal PPKM Mikro.....	150
Lampiran 4 Tanggal PPKM Darurat dan Level 1-4	151
Lampiran 5 Tanggal pilkada Jawa Timur 2020.....	152
Lampiran 6 Tanggal libur panjang dan tanggal lainnya	153
Lampiran 7 Rekap jumlah tanggal berpengaruh dari observasi	155
Lampiran 8 Nilai <i>silhouette</i> dan jumlah <i>single cluster</i> untuk tiap jumlah cluster (n) yang dicoba untuk kelompok 90 hari diurutkan dari nilai <i>silhouette</i> terbesar	156
Lampiran 9 Nilai <i>silhouette</i> dan jumlah <i>single cluster</i> untuk tiap jumlah cluster (n) yang dicoba untuk kelompok 180 hari diurutkan dari nilai <i>silhouette</i> terbesar ..	156
Lampiran 10 Nilai <i>silhouette</i> dan jumlah <i>single cluster</i> untuk tiap jumlah cluster (n) yang dicoba untuk kelompok 366 hari diurutkan dari nilai <i>silhouette</i> terbesar	157
Lampiran 11 Nilai <i>silhouette</i> dan jumlah <i>single cluster</i> untuk tiap jumlah cluster (n) yang dicoba untuk kelompok sebelum vaksinasi diurutkan dari nilai <i>silhouette</i> terbesar	157
Lampiran 12 Sebaran percobaan <i>hyperparameter</i> oleh Optuna untuk model utama	158
Lampiran 13 RMSSE evaluasi untuk jumlah kasus positif aktif (i) kelompok 90 hari	161
Lampiran 14 RMSSE evaluasi untuk jumlah kasus sembuh (r) kelompok 90 hari	161
Lampiran 15 RMSSE evaluasi untuk jumlah kasus meninggal (d) kelompok 90 hari	162
Lampiran 16 RMSSE evaluasi untuk jumlah kasus positif aktif (i) kelompok 180 hari	162
Lampiran 17 RMSSE evaluasi untuk jumlah kasus sembuh (r) kelompok 180 hari	163

PENDAHULUAN

1.1. Latar Belakang

COVID-19 adalah penyakit yang disebabkan oleh virus SARS-CoV-2. Sebagian besar pasien hanya menunjukkan gejala ringan mirip flu biasa, tapi sebagian menunjukkan gejala parah yang dapat berujung pada kematian. COVID-19 berbahaya karena ia dapat menyebar dari manusia ke manusia lain dalam masa inkubasi yang tidak menunjukkan atau hanya menunjukkan sedikit gejala selama 2-14 hari (rata-rata 5.2 hari), sehingga penyebarannya cepat, luas, dan sulit dilacak. (Harapan et al., 2020; Park et al., 2020; Singhal, 2020; Sohrabi et al., 2020; Wu et al., 2020)

Sejak 17 Juni 2020, Indonesia telah menjadi negara dengan jumlah kasus kumulatif COVID total, aktif, harian, dan meninggal tertinggi di ASEAN (Mukaromah, 2020). Hingga 16 April 2021, tercatat 1.594.722 total kasus COVID-19 di Indonesia dengan 107.297 (6.7%) kasus aktif, 1.444.229 (90.6%) kasus sembuh, dan 43.196 (2.7%) kasus meninggal (Komite Penanganan COVID-19 dan Pemulihan Ekonomi Nasional, 2021). Pada 16 April 2021, Jawa Timur adalah provinsi dengan jumlah total kasus tertinggi keempat dengan 143.900 (9.0%) total kasus positif (Komite Penanganan COVID-19 dan Pemulihan Ekonomi Nasional, 2021; Pemerintah Provinsi Jawa Timur, 2021). Jumlah total kasus tersebut terdiri dari 1.941 (1,35%) kasus aktif, 131.606 (91,46%) kasus sembuh, dan 10.353 (7,19%) kasus meninggal (Pemerintah Provinsi Jawa Timur, 2021).

Meskipun vaksin sudah mulai tersedia, pandemi COVID-19 tidak akan segera berakhir. Prancis kembali melakukan *lockdown* setelah adanya gelombang infeksi COVID-19 baru (VOA News, 2021). Jumlah kasus baru di Polandia meningkat tajam melampaui jumlah tertinggi sebelumnya pada November 2021 (VOA News, 2021). Filipina kembali melakukan *lockdown* karena lambatnya vaksinasi (Batino & Calonzo, 2021). Virus COVID-19 telah bermutasi menjadi beberapa *strain* baru yang lebih mudah menular dan dapat menginfeksi kembali

buruk. Lonjakan kematian di Italia terjadi karena layanan kesehatan gagal mengantisipasi lonjakan pasien dan pemerintah gagal mengontrol penyebaran COVID-19 dengan baik (Rudan, 2020). Peramalan jumlah kasus COVID-19 dapat membantu mengantisipasi lonjakan kasus sehingga persiapan bisa lebih matang dan meminimalkan dampaknya (Devaraj et al., 2021; Petropoulos & Makridakis, 2020).

Model *deep learning* mampu menangkap pola yang rumit dengan banyaknya parameter di dalamnya. Namun, banyaknya parameter dalam model *deep learning* mengharuskannya belajar dari data dalam jumlah besar untuk dapat memberikan kinerja yang baik (Ajiboye et al., 2015; Pan & Yang, 2010), sedangkan data jumlah kasus harian COVID-19 tidak banyak. *Transfer learning* berusaha mengatasi ini dengan membuat model *machine learning* menggunakan pengetahuan dari *dataset* atau *model* lain (Goodfellow et al., 2016; Pan & Yang, 2010). Akan tetapi, *transfer learning* hanya dapat dilakukan dengan satu sumber dan satu target saja. *Joint learning* dapat digunakan untuk melatih model dengan data dari lebih dari dua domain. *Joint learning* adalah bentuk *transfer learning* dimana beberapa model dilatih bersamaan dan berbagi sebagian *layer* untuk mempelajari pola umum pada *layer* tersebut (Zhang et al., 2019). Dengan *joint learning*, diharapkan dapat ditemukan sebuah pola umum pandemi COVID-19 dari banyak daerah yang memiliki dinamika pandemi yang mirip.

dengan menggunakan data lain dari beberapa daerah yang mirip untuk menemukan pola umum pandemi. Model yang telah dikembangkan akan digunakan untuk meramalkan jumlah kasus hingga 9 bulan setelah tanggal data yang terakhir diambil. Penelitian ini menggunakan data perkembangan harian jumlah kumulatif kasus positif aktif, sembuh, dan meninggal dari kota dan kabupaten di Jawa Timur hingga 20 Maret 2021 yang diperoleh dari *website* COVID-19 Jawa Timur (Pemerintah Provinsi Jawa Timur, 2021). Model dibuat untuk meramal jumlah kasus sejauh 14 hari ke depan sebagai waktu yang cukup untuk menentukan PSBB dan penambahan kapasitas rumah sakit (Goic et al., 2021). Model dievaluasi dengan mengukur nilai *Root Mean Square Scaled Error* (RMSSE) dan membandingkannya dengan enam model *baseline* untuk mengukur kegunaan dari tiap bagian model serta model ARIMA dan SIRD (*susceptible-infected-recovered-dead*) sebagai model pembanding.

Berdasarkan latar belakang, masalah dirumuskan sebagai:

1.3. Batasan Masalah

1.4. Tujuan Penelitian

BAB II

2.1 Tinjauan Penelitian Terdahulu

Lai et. al (Lai et al., 2017) menggunakan 1D *convolution* sebelum GRU pada LSTNet untuk menangkap pola temporal. LSTNet menggunakan *autoregression* (AR) untuk mengatasi skala data sehingga model *deep learning* hanya perlu mempelajari sisa yang tidak dapat dijelaskan dengan model AR. LSTNet memperhitungkan *seasonality* dengan menggunakan *skip connection* pada *recurrent neural network* (RNN). LSTNet menggunakan *rectified linear unit* (ReLU) sebagai ganti *hyperbolic tan* (tanh) pada *Gated Recurrent Unit* (GRU). LSTNet menunjukkan kinerja yang lebih baik untuk sebagian besar dataset daripada model-model lain yang dibandingkan oleh penulis. Penelitian ini menggunakan penelitian Lai et al. sebagai dasar dalam menggunakan 1D *convolution* untuk *representation layer* yang membangun representasi untuk RNN.

Epstein et al. (Epstein et al., 2018) mengemukakan *framework transfer learning* fleksibel yang disebut *joint-autoencoder* (JAE). JAE membangun *autoencoder* yang memiliki dua cabang untuk tiap *task*, yaitu *private branch* dan *shared branch*. Tiap model berbagi bagian tengah *shared branch* yang sama sehingga *shared branch* mempelajari representasi yang dapat digunakan semua *task*. JAE dilatih terlebih dahulu sebagai *autoencoder* hingga mencapai kinerja yang cukup baik lalu dilatih untuk masing-masing *task*. JAE dapat digunakan untuk lebih dari dua *task*. JAE meninggalkan model pembandingnya cukup jauh pada *transfer learning* dari dataset SVHN ke MNIST dan kompetitif untuk dataset lainnya. Penelitian ini menggunakan penelitian Epstein et al. sebagai dasar untuk membagi model menjadi *private branch* dan *shared branch*.

Calafiore et al. (Calafiore et al., 2020) memodifikasi model SIRD (*susceptible-infected-recovered-dead*) dengan memodelkan ketiga parameternya sebagai fungsi waktu. Calafiore et al. menggunakan model SIRD diskrit untuk melihat nilai parameter sebenarnya sebelum membuat fungsi waktu untuk tiap

model SEIR (*susceptible-exposed-infected-recovered*). SEIR sebagai *layer* kedua dimodelkan sebagai *recurrent neural network* yang menerima *latent variable* h dari *layer* pertama untuk menghasilkan βSI , yaitu arus perpindahan dari kompartemen S (*susceptible*) ke E (*exposed*). Variabel lainnya, seperti periode inkubasi dan kecepatan pemulihan, juga bisa berasal dari *layer* pertama, namun dalam penelitian ini digunakan nilai rata-rata yang dilaporkan. Dua variasi SIRNET diajukan dengan perbedaan hanya pada *layer* pertama. Variasi pertama menggunakan *multilayer perceptrons* (MLP) untuk menghasilkan *latent variable* h , sedangkan variasi kedua menggunakan LSTM. Penulis tidak membandingkan model ini dengan model lain, karena tujuan utama penelitian ini adalah menunjukkan bagaimana *social distancing* sangat mempengaruhi perkembangan pandemi berdasarkan data mobilitas. Penelitian ini menggunakan penelitian Soures et al. sebagai dasar untuk menggunakan *deep learning*, khususnya LSTM, untuk meramalkan variabel model berbasis SIR.

Li et al. (Li et al., 2021) mengembangkan model ALERT untuk meramal jumlah total kasus positif COVID-19. ALERT menggunakan *transfer learning* dan mekanisme *attention* serta menggunakan *lockdown* sebagai salah satu prediktor.

Data jumlah kasus positif dalam sebuah *window* dimasukkan ke dalam LSTM dengan mekanisme *attention*. Data *lockdown* yang berupa biner dan berada dalam *window* yang sama dimasukkan ke dalam MLP. Hasil LSTM kasus positif dan MLP *lockdown* ini dijumlahkan untuk menghasilkan *output* ramalan. *Transfer learning* dilakukan dengan terlebih dahulu melatih model dengan data dari negara-negara sumber pilihan. Penulis membandingkan model ALeRT dengan 4 model *baseline*, yang merupakan model ALeRT dengan sebagian model dihilangkan. Hasil evaluasi model dengan *Mean Absolute percentage Error* (MAPE) menunjukkan kinerja yang meningkat dari model *baseline* 1 hingga ALeRT. Penulis menyebutkan dua keterbatasan penelitian ini, yaitu (1) *lockdown* dimodelkan sebagai biner sedangkan tingkat *lockdown* tiap negara berbeda-beda dan (2) model mungkin bisa menggunakan lebih banyak prediktor seperti jumlah pasien sembuh atau kapasitas rumah sakit. Penelitian ini menggunakan penelitian Li et al. sebagai dasar untuk argumen bahwa *transfer learning* dapat dilakukan untuk peramalan jumlah kasus COVID-19, bahwa model perlu memperhatikan *lockdown*, bahwa prediktor lain patut dicoba, dan sebagai referensi dalam menentukan model *baseline* untuk menilai kegunaan tiap bagian model.

2.2 Teori Dasar yang Digunakan

2.2.1 COVID-19

COVID-19 adalah penyakit yang disebabkan oleh virus SARS-CoV-2. Penyakit ini pertama kali ditemukan di Wuhan, China, dan dihubungkan dengan pasar grosir *seafood* Huanan, Wuhan. Penderita COVID-19 menunjukkan gejala batuk kering, sesak nafas, demam, penyakit pencernaan, nyeri otot, sakit kepala, serta menunjukkan adanya infiltrat paru-paru jika diperiksa. Sebagian besar pasien hanya menunjukkan gejala ringan mirip flu biasa, tapi sebagian menunjukkan gejala parah yang dapat berujung pada kematian dengan angka kematian di rumah sakit awalnya 2% hingga 15%. Gejala COVID-19 yang parah lebih sering ditemukan pada orang tua dan orang-orang dengan imun lemah karena penyakit lain. COVID-19 berbahaya karena ia dapat menyebar dari manusia ke manusia lain dalam masa inkubasi yang tidak menunjukkan atau hanya menunjukkan sedikit gejala selama 2-14 hari (rata-rata 5.2 hari) sehingga penyebarannya cepat, luas, dan sulit dilacak. Pada 30 Januari 2020, WHO mendeklarasikan situasi darurat global terkait COVID-19. (Harapan et al., 2020; Park et al., 2020; Singhal, 2020; Sohrabi et al., 2020; World Health Organization, 2020; Wu et al., 2020)

Penyebaran COVID-19 dapat diminimalkan dengan deteksi awal, isolasi, penanganan yang tepat, serta pelacakan kontak. Banyak negara yang menutup akses luar negeri sementara untuk menangani COVID-19. Orang-orang yang baru datang dari tempat yang terdampak COVID-19 diisolasi selama 14 hari untuk berjaga-jaga. Selain dari sisi kesehatan, berbagai negara juga menutup sekolah demi mencegah penyebaran COVID-19. (Singhal, 2020; Sohrabi et al., 2020; Viner et al., 2020; Wu et al., 2020)

Meskipun COVID-19 berawal di China, negara pertama yang mengalami penyebaran COVID-19 yang tidak terkontrol bukanlah China tapi Italia. Ini terjadi karena China dengan cepat mengisolasi Wuhan dan beberapa kota sekitarnya; isolasi ini sering disebut *lockdown*. *Lockdown* tidak hanya mengisolasi wilayah dari wilayah sekitarnya, namun aktivitas dan kontak penduduk dalam wilayah tersebut juga sangat dibatasi. *Lockdown* terbukti efektif di China dengan penyebaran COVID-19 yang lebih lambat terutama di wilayah di sekitar tempat

Meskipun *lockdown* terbukti efektif, *lockdown* sangatlah “mahal” dan memiliki banyak efek samping negatif bagi ekonomi maupun kesehatan. Karena itu, *local lockdown* menjadi alternatif. *Local lockdown* adalah versi *lockdown* yang kurang ketat. Masyarakat tetap diperbolehkan bekerja jika pekerjaan itu cukup vital bagi masyarakat. Masyarakat dianjurkan tetap di rumah sesering mungkin dan menjaga jarak setidaknya 2 meter. *Local lockdown* memang tidak seefektif *lockdown* sesungguhnya, namun tetap terbukti memperlambat penyebaran COVID-19 di Inggris. (Mahase, 2020; Melnick & Ioannidis, 2020)

Indonesia mulai serius menangani COVID-19 dengan Instruksi Presiden Nomor 4 Tahun 2020. Pada akhir Maret 2020, Presiden menyatakan Indonesia dalam keadaan darurat COVID-19 dan membuat kebijakan Pembatasan Sosial Berskala Besar (PSBB) bagi beberapa kota dengan jumlah kasus tinggi. Meskipun PSBB adalah *local lockdown*, dampaknya secara ekonomi tetaplah besar bagi rakyat Indonesia yang sebagian besar tidak bisa bekerja dari rumah sehingga menyebabkan pandangan negatif terhadap PSBB dan mengurangi efektivitasnya. (Djalante et al., 2020; Setyawan & Lestari, 2020)

2.2.2 Model SIRD

Model SIR (*susceptible-infected-recovered*) adalah model kompartemental yang umum digunakan dalam pemodelan epidemi atau penyakit menular. Model SIR diperkenalkan oleh W. O. Kermack dan A. G. McKendrick pada tahun 1927. Model ini adalah model sederhana dimana populasi secara perlahan berubah dari rentan (*susceptible*) ke terinfeksi (*infected*) lalu sembuh (*recovered*). Model ini dibangun atas asumsi bahwa populasi relatif tetap sehingga populasi dianggap konstan. (Teguh et al., 2020; Yulida & Karim, 2020)

Model SIR sering dikembangkan dengan menambah kompartemen baru. Dua di antaranya adalah penambahan kompartemen E (*exposed*) untuk memodelkan periode inkubasi (Soures et al., 2020) dan penambahan kompartemen D (*dead*) untuk memodelkan sebagian populasi yang meninggal (Chatterjee et al., 2020).

$$\begin{aligned}\frac{dS}{dt} &= -\beta \frac{IS}{N} \\ \frac{dI}{dt} &= \beta \frac{IS}{N} - \gamma I - \delta I \\ \frac{dR}{dt} &= \gamma I \\ \frac{dD}{dt} &= \delta I\end{aligned}\tag{2.1}$$

N adalah jumlah total populasi; S (*susceptible*) adalah jumlah sebagian populasi yang masih rentan terhadap virus; I (*infected*) adalah jumlah sebagian populasi yang sudah terinfeksi dan bisa menginfeksi S; R (*recovered*) adalah jumlah sebagian populasi yang sudah sembuh dan kebal; dan D (*dead*) adalah jumlah sebagian populasi yang meninggal. Parameter β adalah kecepatan penyebaran virus, parameter γ adalah kecepatan penyembuhan dari virus; dan parameter δ adalah *death rate*. (Chatterjee et al., 2020; Nadler et al., 2020)

2.2.3 Peramalan

Peramalan *time series* adalah masalah bisnis penting dan area aplikasi *machine learning* yang sukses. Peramalan digunakan dalam banyak aspek bisnis modern, seperti kontrol pergudangan dan manajemen pelanggan dan juga perencanaan dari produksi hingga distribusi lalu keuangan dan pemasaran.

Dengan demikian, peramalan memiliki dampak finansial yang besar hingga jutaan dolar untuk setiap titik akurasi yang diperoleh. Meskipun demikian, terkadang *machine learning* dan *deep learning* masih kalah dari peramalan dengan pendekatan statistik klasik. (Oreshkin et al., 2020)

Model *AutoRegressive Integrated Moving Average* (ARIMA) adalah salah satu pendekatan analisis *time series* yang paling sering digunakan. Model ARIMA menggunakan data historis untuk menghasilkan ramalan akurat. Model ARIMA merupakan gabungan dari model *AutoRegressive* (AR) dan *Moving Average* (MA). Bagian AR dari ARIMA melakukan regresi dengan nilai-nilai sebelumnya. Bagian MA dari ARIMA melakukan *moving average* pada *error term*. I pada ARIMA berarti *integrated*; model ARIMA melakukan ARMA pada turunan *time series* yang didapat dari *differencing* untuk membuat *time series* menjadi stasioner lalu dikembalikan dengan menjumlahkannya secara kumulatif. Stasioneritas penting agar analisis regresi klasik valid. *Differencing* dilakukan dengan mengurangi setiap nilai pada *series* dengan nilai sebelumnya. (Box et al., 2016; Peter & Silvia, 2012)

Sebuah model dapat memiliki dua jenis variabel, yaitu endogen dan eksogen. Variabel endogen adalah variabel yang berusaha dijelaskan dengan model. Variabel eksogen adalah variabel independen yang mempengaruhi variabel endogen. Variabel eksogen digunakan oleh model sebagai input untuk menjelaskan variabel endogen. (Mankiw, 2009)

Ada banyak metode untuk mengukur kinerja ramalan. Hyndman & Koehler (Hyndman & Koehler, 2006) menyebutkan kekurangan metode-metode pengukuran. *Scale-dependent measures* (MSE, RMSE, MAE) dipengaruhi skala dan sensitif terhadap *outlier*. *Measures based on percentage errors* (MAPE, sMAPE) bernilai tak hingga atau tak terdefinisi ketika $Y_t = 0$. *Relative measures* tidak dapat menilai ramalan *out-of-sample* dengan banyak *input time series*.

Hyndman & Koehler (Hyndman & Koehler, 2006) memperkenalkan *Mean Absolute Scaled Error* (MASE) sebagai metode pengukuran kinerja ramalan yang tidak terpengaruh skala, kokoh, dan dapat digunakan dalam segala situasi. MASE dihitung dengan membagi **setiap** *error* dengan *Mean Absolute Error* (MAE) dari

Root Mean Square Scaled Error (RMSSE) adalah alternatif dari MAE. RMSSE digunakan sebagai ganti MAE (Hyndman & Koehler, 2006). RMSSE digunakan dalam kompetisi peramalan M5. RMSSE didefinisikan dengan rumus berikut (Makridakis et al., 2020):

$$\text{RMSSE} = \sqrt{\frac{\frac{1}{h} \sum_{t=n+1}^{n+h} (y_t - \hat{y}_t)^2}{\frac{1}{n-1} \sum_{t=2}^n (y_t - y_{t-1})^2}}$$

di mana:

- h = panjang *horizon forecast* (panjang peramalan)
- n = jumlah data prediktor
- y_t = nilai sebenarnya pada waktu t
- $\hat{y}_t$ = nilai ramalan pada waktu t

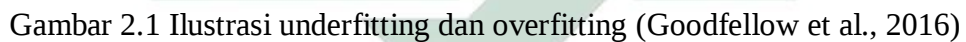
Hyndman dan Athanasopoulos (Hyndman & Athanasopoulos, 2018) menyebutkan peramalan dilakukan setidaknya dengan lima langkah dasar:

$$\text{RMSSE} = \sqrt{\frac{\frac{1}{h} \sum_{t=n+1}^{n+h} (y_t - \hat{y}_t)^2}{\frac{1}{n-1} \sum_{t=2}^n (y_t - y_{t-1})^2}} \quad (2.3)$$

- h = panjang *horizon forecast* (panjang peramalan)
- n = jumlah data prediktor
- y_t = nilai sebenarnya pada waktu t
- $\hat{y}_t$ = nilai ramalan pada waktu t

1. Penentuan masalah
2. Pengumpulan informasi, yaitu data untuk peramalan dan informasi tentang hal yang diramalkan
3. Analisis awal (eksplorasi data); mengidentifikasi pola, tren, *seasonality*, *outlier*, dan pengaruh lingkungan
4. Pemilihan dan pembuatan model
5. Evaluasi dan penggunaan model

Algoritma *machine learning* adalah algoritma yang dapat belajar dari data. *Machine learning* memungkinkan komputer menyelesaikan *task* yang terlalu sulit dengan program biasa. *Task* dalam *machine learning* biasanya dijelaskan sebagai bagaimana model *machine learning* harus memproses sebuah *example* (contoh). Beberapa contoh *task* yang umum adalah klasifikasi, regresi, transkripsi, dan penerjemahan. *Example* adalah kumpulan (*feature*) dari sebuah objek atau kejadian yang perlu diproses. Secara umum, *machine learning* terbagi menjadi dua jenis, yaitu *supervised* dan *unsupervised*. *Supervised machine learning* belajar dari data yang telah diberi label atau target, sedangkan *unsupervised machine learning* tidak memerlukan label. Label dari sebuah *example* adalah keluaran yang diharapkan dari model *machine learning* jika diberi *example* tersebut. Kinerja model dapat ditingkatkan dengan meningkatkan kualitas data dan dengan mengoptimasi *hyperparameter* model tersebut. (Goodfellow et al., 2016)



15

K dengan rata-rata koefisien siluet tertinggi dipilih sebagai nilai K optimal. (Kodinariya & Makwana, 2013)

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (2.4)$$

2.2.6 Deep Learning

Deep learning adalah salah satu bentuk spesifik *machine learning*. *Machine learning* klasik memiliki setidaknya tiga keterbatasan yang memotivasi digunakannya *deep learning*. Keterbatasan yang pertama adalah bahwa *machine learning* klasik kurang mampu mengatasi banyak dimensi karena banyaknya kombinasi yang mungkin dan sedikitnya sampel yang dimiliki. Biasanya, untuk mengatasi ini, dilakukan analisis untuk memilih dimensi yang dianggap penting saja, namun beberapa dimensi yang tampak tidak penting mungkin justru penting untuk beberapa kasus tertentu. Keterbatasan yang kedua adalah anggapan bahwa generalisasi yang baik adalah yang halus (Gambar 2.1 bagian tengah), sedangkan ini tidak selalu benar. Keterbatasan yang terakhir adalah asumsi bahwa tiap data saling berhubungan dan data yang tidak terhubung dengan baik hanyalah *noise* atau *outlier* yang tidak perlu dianggap penting. (Goodfellow et al., 2016)

Deep learning menyediakan *framework* yang kuat untuk *supervised learning*. Pada dasarnya, model *deep learning* berusaha menjadi fungsi yang melakukan *mapping* dari sekelompok *input* ke *output* yang sesuai. *Deep learning* dapat merepresentasikan fungsi yang lebih kompleks dengan menambah *layer* dan *unit* dalam sebuah *layer*. Model *deep learning* sering juga disebut sebagai *neural network* karena terinspirasi dari jaringan syaraf manusia dengan setiap *node* atau *unit* sebagai *neuron*. Model *deep learning* atau *neural network* yang paling umum adalah *deep feedforward network* yang juga sering disebut *feedforward neural network* atau *multilayer perceptrons* (MLP). Model ini disebut *feedforward* karena data hanya berjalan satu arah dari *input* ke *output*. Model ini disebut *network* karena terdiri dari banyak *node* berupa fungsi sederhana yang sering disebut *activation function* atau fungsi aktivasi. Panjang keseluruhan atau jumlah *layer* dari jaringan ini menjadi kedalaman (*depth*) dari model itu, sehingga model ini disebut *deep*. (Goodfellow et al., 2016)

RNN biasa memiliki masalah *vanishing gradient* dimana ia tidak dapat mengingat maupun mempelajari data terdahulu, terutama jika *output*-nya hanya satu. Ini disebut juga masalah ketergantungan jangka panjang karena RNN tidak dapat menggunakan informasi yang terpaut terlalu jauh (jangka panjang). Ini terjadi karena bobot akhir yang dimiliki data pada awal *sequence* akan mengecil secara eksponensial seiring diaplikasikannya sel RNN pada *sequence*. Sebaliknya, *exploding gradient* dimana gradien menjadi sangat besar juga mungkin terjadi meskipun jauh lebih jarang. Beberapa arsitektur RNN dengan *gate* (*gated RNN*) dibuat untuk mengatasi masalah ini; yang paling umum adalah *Long Short Term Memory* (LSTM). (Goodfellow et al., 2016; Yu et al., 2019)



22

merepresentasikan data dalam bentuk yang lebih baik untuk model. *Representation learning* tidak harus dilakukan dengan *deep learning*; justru *deep learning* yang dapat dilihat sebagai tumpukan *representation learning*. Suatu model klasifikasi *deep learning* dapat dilihat sebagai model yang terdiri dari dua bagian; yaitu model yang mengekstrak *feature* dari data mentah dan model klasifikasi yang menggunakan *feature* tersebut. *Feature* yang diekstrak ini adalah representasi lebih baik dari data mentah sehingga lebih mudah digunakan oleh model klasifikasi. *Feature* ini dibangun dari tumpukan *hidden layer* dimana tiap *layer* membangun representasi untuk *layer* berikutnya. (Goodfellow et al., 2016)

Machine learning secara umum berasumsi bahwa seluruh data yang digunakan, baik data untuk *training* maupun data baru, berasal dari ruang dan distribusi yang sama. Pada kenyataannya, data mungkin cukup langka sehingga tidak cukup untuk membangun model *machine learning* yang baik. Sebagai model yang belajar dari data, jumlah data sangat mempengaruhi kinerja model *machine learning* terutama untuk data baru. *Transfer learning* berusaha mengatasi ini dengan membuat model *machine learning* dapat memperoleh pengetahuan dari *model* lain. (Ajiboye et al., 2015; Goodfellow et al., 2016; Pan & Yang, 2010)

2.2.11 Joint Learning

Joint learning adalah salah satu turunan *transfer learning*. *Joint learning* dilakukan dengan melatih model-model yang berbagi sebagian *layer* untuk membangun representasi yang membuat dataset-dataset dari domain yang berbeda menjadi berada dalam satu domain yang sama yang dapat digunakan oleh tiap model. Karena *joint learning* dilakukan dengan banyak dataset, *joint learning* sebaiknya dilakukan setelah mengelompokkan dataset (*clustering*) dan hanya melakukan *joint learning* dengan model dari *cluster* yang sama. Pengelompokan ini dilakukan atas dasar bahwa dataset dari domain yang mirip seharusnya memiliki parameter yang mirip juga untuk model yang sama. Pengelompokan ini dapat dilakukan lebih dari satu kali dengan melakukannya setiap beberapa *layer* (untuk beberapa tingkat representasi data). (Zhang et al., 2019)

2.2.12 Machine Learning Python

Python telah menjadi alat utama untuk komputasi ilmiah dalam dua dekade terakhir. Hal ini mungkin tidak terduga karena Python tidak dibuat untuk tujuan ini. Python diperkenalkan pada akhir tahun 1980-an sebagai bahasa untuk *scripting* dan pembelajaran. Popularitas Python datang dari ekosistem *package* pihak ketiga yang besar dan aktif, terutama NumPy untuk memanipulasi data berbasis *array*, Pandas untuk manipulasi data heterogen dan berlabel, SciPy untuk keperluan komputasi ilmiah umum, Matplotlib untuk visualisasi data, IPython untuk eksekusi dan berbagi kode dengan interaktif, Scikit-learn untuk *machine learning*, dan lain-lain. Selain itu, Python juga menarik karena kesederhanaan dan keindahan kodenya. (VanderPlas, 2016b, 2016a)

Dengan meningkatnya popularitas *deep learning*, berbagai *deep learning* telah dibuat untuk Python. Kebanyakan alat yang populer seperti TensorFlow dan Theano membangun *deep learning network* dengan membuat *graph* statis yang merepresentasikan alur komputasi. Pendekatan ini membuat alur komputasi dapat terlihat sejak awal sehingga secara teori dapat digunakan untuk meningkatkan kinerja dan skalabilitas. Akan tetapi, pendekatan ini menghambat kemudahan penggunaan, kemudahan *debugging*, serta fleksibilitas model. (Paszke et al., 2019)

PyTorch adalah alat *deep learning* untuk Python yang menggunakan komputasi tensor dinamis dengan diferensiasi otomatis dan akselerasi GPU. PyTorch menjadi sangat populer dalam riset *deep learning* karena fleksibilitas dan kemudahan penggunaannya. PyTorch menggunakan sintaks yang sama dengan program Python biasa dan memberikan kontrol penuh kepada penggunanya. Tidak hanya itu, PyTorch melakukan semua ini dengan tetap menjaga kinerja yang kompetitif dengan alat *deep learning* lainnya. (Paszke et al., 2019)

2.2.13 Optuna

Model *machine learning*, khususnya *deep learning*, disempurnakan dengan eksperimen berulang kali untuk mencari *hyperparameter* dan arsitektur yang optimal. Kompleksitas model semakin meningkat dengan meningkatnya popularitas *deep learning* sehingga kebutuhan penyesuaian *hyperparameter* secara otomatis dan efisien juga semakin meningkat. Karena itu, berbagai alat optimasi *hyperparameter* telah dikembangkan dengan berbagai algoritma. Namun, ada masalah yang sering diabaikan oleh alat-alat ini, yaitu pengguna diharuskan mendefinisikan ruang *hyperparameter* dengan baik. Seiring meningkatnya kompleksitas model, definisi ruang *hyperparameter* menjadi sulit. Jika ruang *hyperparameter* tidak terdefinisi dengan baik, pencarian dapat menjadi sia-sia. (Akiba et al., 2019)

Optuna adalah alat pencarian *hyperparameter* yang dibuat untuk mengatasi masalah definisi ruang *hyperparameter*, efisiensi, dan skalabilitas yang telah dijelaskan sebelumnya. Selain itu, Optuna juga didesain dengan mengutamakan fleksibilitas dan kemudahan penggunaan. Optuna memungkinkan definisi ruang pencarian secara dinamis, memungkinkan kustomisasi dalam algoritmanya, serta menggunakan arsitektur yang mudah digunakan baik untuk proyek skala kecil maupun besar. (Akiba et al., 2019)

Optuna melakukan optimasi *hyperparameter* dengan cara *Bayesian optimization* yang didasarkan pada *bayesian reasoning*. *Bayesian optimization* menentukan parameter untuk dicoba berdasarkan *prior* yang dibentuk dari hasil percobaan atau pengamatan sebelumnya (Shahriari et al., 2016). *Prior* ini kemudian diperbarui dengan hasil percobaan baru untuk percobaan berikutnya.

Optuna secara *default* menggunakan algoritma *Tree-structured Parzen Estimator* (TPE) (Bergstra et al., 2011) untuk menentukan parameter yang akan dicoba. Algoritma TPE membangun dua distribusi dari hasil percobaan, yaitu distribusi $l(x)$ yang dibangun dari parameter baik yang menghasilkan kinerja di atas kuantil tertentu, dan distribusi parameter buruk $g(x)$ yang dibangun dari percobaan di bawah kuantil tersebut. Distribusi yang dibangun TPE adalah distribusi Parzen bertingkat. Karena menggunakan distribusi bertingkat (*hierarchical, tree*), TPE mendukung *hyperparameter* bertipe kategori dan *hyperparameter* kondisional. TPE akan mengambil sampel distribusi dari distribusi baik $l(x)$ dan menghitung peningkatan kinerja yang diharapkan dengan menghitung $l(x)/g(x)$.

Signifikansi komparasi dari model-model *machine learning* perlu dibuktikan secara statistik, terutama ketika membandingkan banyak model atau dengan banyak dataset. t-test untuk setiap kombinasi pasangan tidak dapat digunakan untuk ini karena cenderung mengalami error tipe 1 (*false positive*). Demsar (Demšar, 2006) memperkenalkan cara membandingkan banyak algoritma dengan banyak dataset. Demsar melakukan ini dengan menggunakan Friedman Test pada rata-rata *rank* untuk menguji adanya perbedaan signifikan dan jika ada akan dilakukan Nemeyi *post-hoc test* untuk mengetahui letak perbedaan.

28

Islam mengajarkan untuk melakukan persiapan atau perencanaan. Untuk melakukan persiapan, seseorang perlu tahu apa yang sekiranya akan terjadi. Namun, dalam surat Luqman ayat 34, Islam mengajarkan bahwa manusia tidak dapat mengetahui dengan pasti apa yang akan terjadi di masa depan. K.H. Mas Yusuf menjelaskan bahwa ini tidak bertentangan dengan persiapan, tapi menjelaskan bahwa Allah berkuasa penuh atas masa depan. Di dunia ini ada *sunatullah* dan *qudratullah*. *Sunatullah* adalah hukum alam yang “pasti” seperti $1+1=2$. Terkait itu, manusia dapat memperkirakan apa yang akan terjadi. Akan tetapi, Allah berkuasa untuk menetapkan hal-hal yang mungkin berbeda dari *sunatullah*; ini disebut *qudratullah*. Manusia tidak dapat mengetahui *qudratullah* dan tidak dapat mengubahnya, sehingga manusia tidak dapat mengetahui dengan pasti mengenai masa depan yang merupakan gabungan dari *sunatullah* dan *qudratullah*. Meskipun begitu, sejauh *sunatullah*, manusia dapat memperkirakan masa depan dan mempersiapkan untuk itu.

إِنَّ اللَّهَ عِنْدَهُ عِلْمُ السَّاعَةِ وَيُنَزِّلُ الْغَيْثَ وَيَعْلَمُ مَا فِي الْأَرْحَامِ ۖ وَمَا تَدْرِي نَفْسٌ مَّاذَا

Artinya: “Sesungguhnya hanya di sisi Allah ilmu tentang hari Kiamat; dan Dia yang menurunkan hujan, dan mengetahui apa yang ada dalam rahim. Dan tidak ada seorang pun yang dapat mengetahui (dengan pasti) apa yang akan dikerjakannya besok. Dan tidak ada seorang pun yang dapat mengetahui di bumi mana dia akan mati. Sungguh, Allah Maha Mengetahui, Maha Menenal.”

Dengan demikian, peramalan jumlah kasus COVID-19 dalam rangka persiapan dan kewaspadaan terhadap lonjakan jumlah COVID-19 layak untuk dilakukan dan memiliki landasan Islami yang cukup. Peramalan ini tidak mungkin dapat memberikan jumlah kasus pasti, tapi dapat memberitahu kemungkinan kenaikan kasus COVID-19 berdasarkan pola yang ditemukan dari data lampau. Peramalan ini tidak sepatutnya disalahgunakan untuk menjadi lengah dan sembrono ketika diramalkan pandemi akan mereda. Seseorang harus tetap sabar dan waspada karena pandemi belum selesai, sebagaimana firman Allah dalam Q.S. Ali Imran ayat 200. K.H. Mas Yusuf menambahkan bahwa penyalahgunaan sesuatu yang diberikan bukan tanggung jawab dari pemberinya jika niatnya baik.

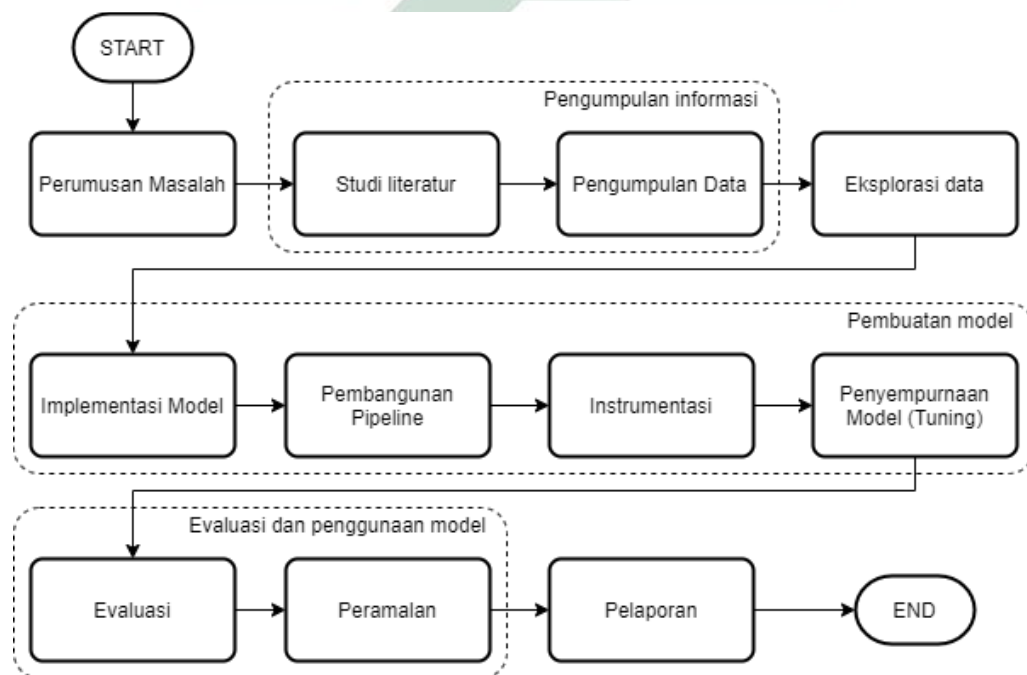
Q.S. al-Isra' ayat 27 berbunyi sebagai berikut:

Artinya: “Sesungguhnya orang-orang yang pemboros itu adalah saudara setan dan setan itu sangat ingkar kepada Tuhannya.”

BAB III

3.1 Desain Penelitian

Dalam sub-bab ini dijelaskan alur penelitian yang digunakan dalam penelitian ini. Berdasarkan lima langkah dasar peramalan (Hyndman & Athanasopoulos, 2018) dan metodologi praktis *deep learning* (Goodfellow et al., 2016), alur penelitian dirumuskan pada Gambar 3.1.



Gambar 3.1 Alur penelitian

3.1.1 Perumusan Masalah

Penelitian ini dilatarbelakangi oleh keterbatasan data dalam peramalan jumlah kasus COVID-19. Dari latar belakang tersebut, dibuat dua rumusan masalah, yaitu:

1. Bagaimana metode *joint learning* untuk meningkatkan kinerja model LSTM dalam peramalan jumlah kasus COVID-19 suatu daerah?
2. Bagaimana perkembangan jumlah kasus COVID-19 hingga 9 bulan setelah tanggal data yang terakhir diambil menurut model yang telah dikembangkan?

Model bertujuan untuk meramal sejauh 14 hari ke depan. Jangka waktu peramalan ini cukup bagi rumah sakit untuk meningkatkan kapasitas ICU dan alih fungsi ruangan dan peralatan (Goic et al., 2021). Jangka waktu peramalan ini juga cukup panjang bagi pemerintah untuk menentukan periode PSBB yang biasanya diberlakukan selama 14 hari. *Long Short Term Memory* (LSTM) dilaporkan mampu menghasilkan ramalan akurat hingga 30 hari (Wang et al., 2020).

Studi literatur dilakukan dengan mencari dan membaca literatur terkait *deep learning*, *transfer learning*, *joint learning*, model SIR (*susceptible-infected-recovered*), peramalan *time series*, dan peramalan COVID-19. Literatur dicari menggunakan Google dan Google Scholar.

Data utama yang digunakan dalam penelitian ini adalah data perkembangan harian jumlah kumulatif kasus positif aktif, sembuh, dan meninggal, COVID-19 di kota atau kabupaten di Jawa Timur. Selain jumlah kasus sebagai variabel endogen, digunakan empat variabel eksogen:

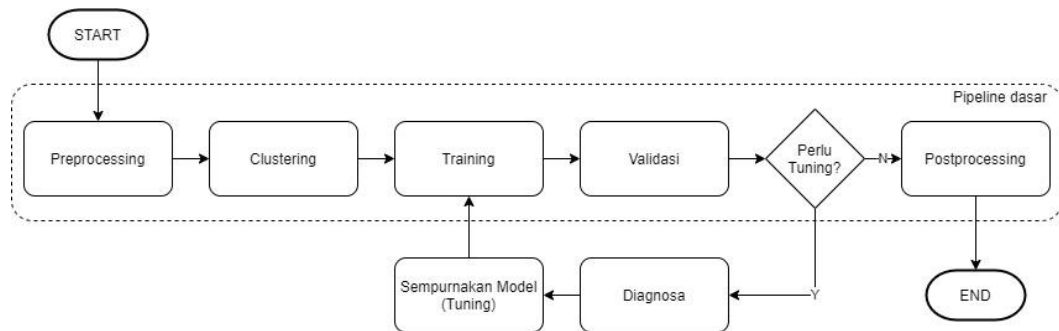
Semua data yang digunakan adalah data sekunder. Data jumlah kasus kumulatif harian COVID-19 tiap kabupaten/kota Jawa Timur diperoleh dari *website* JATIM TANGGAP COVID-19 milik Pemerintah Provinsi Jawa Timur (Pemerintah Provinsi Jawa Timur, 2021). Data jumlah kasus kumulatif harian COVID-19 diperoleh dengan mengambilnya dalam bentuk JSON dari API

3.1.5 Implementasi Model

Dalam tahap ini, model diimplementasikan untuk memungkinkan *joint learning*, namun *training* belum dilakukan sehingga yang dibuat hanya kerangkanya saja (*class*). *Joint learning* dipilih sebagai bentuk *transfer learning* yang dapat menggunakan lebih dari satu sumber untuk memaksimalkan penggunaan data daerah-daerah terjangkit pandemi. Model dibuat untuk meramalkan sejauh 14 hari ke depan. Model akan dibuat dengan bahasa Python menggunakan *framework* PyTorch (Paszke et al., 2019). Model dibuat dengan melakukan *joint learning* (Zhang et al., 2019) pada *shared branch* (Epstein et al., 2017).

Long Short Term Memory (LSTM) telah terbukti dapat meramalkan jumlah kasus COVID-19 dengan baik secara langsung (Devaraj et al., 2021; Elsheikh et al., 2021; Gautam, 2021; Nadler et al., 2020) atau dengan meramalkan variabel model berbasis SIR (Li et al., 2021; Soures et al., 2020). Berdasarkan itu, LSTM akan digunakan sebagai model dasar. Arsitektur *encoder-decoder* digunakan untuk LSTM karena peramalan dilakukan untuk lebih dari satu hari dan untuk memungkinkan input yang berbeda pada jarak prediktor dan jarak ramalan.

Penelitian ini mengimplementasikan *joint learning* pada model dasar dengan arsitektur keseluruhan yang digambarkan pada Gambar 3.2. Model *deep learning* dibagi menjadi dua bagian, yaitu *representation layer* dan *head layer*. *Representation layer* membangun representasi untuk *head layer* sebagai model dasar peramalan. Model dibagi menjadi *private branch* dan *shared branch* dan *joint learning* dilakukan pada *shared branch* (Epstein et al., 2018). H_p dan H_s pada gambar adalah *hidden state* dari *encoder*.



Gambar 3.4 Rencana *pipeline* keseluruhan

Pipeline dibangun untuk mendukung alur keseluruhan model mulai dari persiapan data hingga validasi. Rencana *pipeline* keseluruhan untuk model yang dikembangkan terdapat pada Gambar 3.4. *Pipeline* secara keseluruhan terdiri dari *pipeline* dasar yang dihubungkan dengan proses diagnosa (dengan instrumentasi) dan *tuning* (penyempurnaan model). Sub-bab ini hanya akan menjelaskan *pipeline* dasar karena instrumentasi dan penyempurnaan model akan dijelaskan dalam sub-bab berikutnya.

Data perlu disiapkan dulu agar dapat digunakan dengan baik; ini dilakukan dalam tahap *preprocessing* (Hapke & Nelson, 2020). Pertama-tama, data dimuat dulu untuk tiap kota/kabupaten. Lalu, data yang kosong (semuanya nol) di awal atau akhir *time series* dihilangkan. Data kosong di awal *time series* berarti pandemi belum dimulai untuk daerah itu, sedangkan data kosong di akhir *time series* dianggap hilang. Kemudian, data kosong yang berada di tengah-tengah *time series* diisi dengan interpolasi linear dengan nilai sebelum dan sesudahnya. Setelah itu, kasus positif aktif dihitung dengan mengurangi kasus positif total dengan kasus positif sembuh dan meninggal. Data jumlah tes kumulatif diubah diubah menjadi jumlah tes harian. Data periode PSBB dan tanggal penting diubah menjadi *time series* biner.

$$I_t = I_{Total,t} - (R_t + D_t) \quad (3.1)$$

Data jumlah kasus COVID-19 akan ditransformasikan menjadi variabel SIRD (*susceptible-infected-recovered-dead*) diskrit untuk menghilangkan tren, membuat data dalam jangkauan 0 hingga 1, dan untuk menghilangkan bias

$$R_t = R_{t-1} + \gamma_t I_{t-1}$$

$$D_t = D_{t-1} + \delta_t I_{t-1}$$

Kemudian, sebagian besar nilai bisa disubstitusikan dengan kasus kumulatif harian dengan cara mengatur ulang persamaan

$$\gamma_t I_{t-1} = R_t - R_{t-1} = \Delta R_t$$

$$\delta_t I_{t-1} = D_t - D_{t-1} = \Delta D_t$$

$$\frac{\beta_t S_{t-1} I_{t-1}}{N} = I_t - I_{t-1} + (\Delta R_t + \Delta D_t) = \Delta I_t + \Delta R_t + \Delta D_t = -$$

Setelah itu, β , γ , dan δ dapat dihitung dengan:

$$\gamma_t = \frac{\Delta R_t}{I_{t-1}}$$

$$\delta_t = \frac{\Delta D_t}{I_{t-1}}$$

$$\frac{\beta_t S_{t-1} I_{t-1}}{N} = I_t - I_{t-1} + (\Delta R_t + \Delta D_t) = \Delta I_t + \Delta R_t + \Delta D_t = -$$

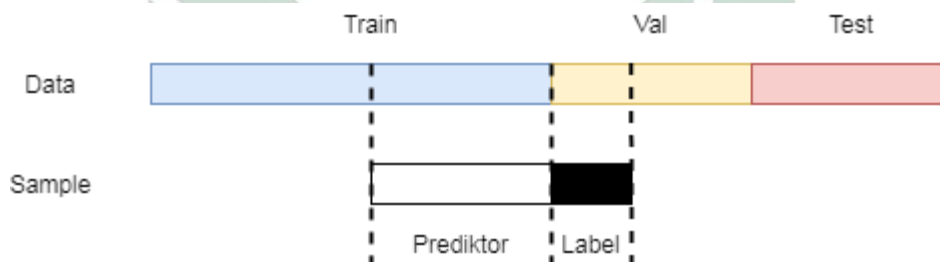
Setelah itu, β , γ , dan δ dapat dihitung dengan:

Setelah itu, β , γ , dan δ dapat dihitung dengan:

Setelah data jumlah kasus COVID-19 ditransformasikan, dibagi menjadi menjadi 4 kelompok dengan jumlah data (kasus) berbeda, yaitu (1) 3 bulan, (2) 6 bulan, (3) 1 tahun (365 hari).

Setelah data jumlah kasus COVID-19 ditransformasikan, dibagi menjadi menjadi 4 kelompok dengan jumlah data (kasus) berbeda, yaitu (1) 3 bulan, (2) 6 bulan, (3) 1 tahun (365 hari).

Tiap *time series* dari tiap kelompok jumlah dibagi lagi menjadi 50% data *training*, 25% data validasi, dan 25% data *testing* (Hastie et al., 2009). Data *training* digunakan untuk melatih model, data validasi digunakan untuk penyempurnaan model (*tuning*), dan data *evaluasi* digunakan untuk mengukur kinerja model untuk data yang belum pernah dilihat seperti ketika penggunaan sesungguhnya nanti. Data validasi dan data *testing* yang ditentukan adalah label sehingga sebagian kecil *time series* sebelumnya sebagai prediktor untuk label yang lebih awal sebagian berasal dari ujung *training* (Gambar 3.5). Ini dilakukan untuk memungkinkan variasi panjang prediktor tanpa mempengaruhi jumlah sampel validasi dan *testing*. Akibatnya, terdapat *tradeoff* antara panjang prediktor dan jumlah sampel *training*. Selain itu, model memang pernah melihat sebagian data itu ketika *training*, tapi tidak untuk konteks yang sama dengan validasi dan evaluasi. Tahap *preprocessing* diakhiri dengan menormalisasikan data ke dalam jangkauan 0 hingga 1 dengan batas atas dan bawah hanya dari data *training* dan validasi.



Transfer learning dari sumber yang buruk justru akan mempersulit *training* hingga menghasilkan model yang lebih buruk daripada model tanpa *transfer learning*; ini disebut *negative transfer* (Weiss et al., 2016). Dalam penelitian ini, semua model memiliki *task* peramalan yang sama dan hanya berbeda pada *domain*. Model yang dilatih dengan dataset dari domain yang mirip seharusnya berkonvergensi ke parameter yang mirip (Zhang et al., 2019). Karena itu,

dilakukan *clustering* untuk mengelompokkan *domain* serupa dengan mengelompokkan data *time series* daerah-daerah di Jawa Timur. *Clustering* dilakukan untuk setiap kelompok jumlah data dengan gabungan data *training* dan validasi.

Joint learning dilakukan dengan melatih model dari *cluster* yang sama bersamaan; semua model dari *cluster* yang sama berbagi *shared branch* yang sama. *Training* dilakukan sebanyak jumlah *batch* paling sedikit dari semua daerah untuk menghindari bias jumlah data, sehingga tidak semua *batch* digunakan dari dataset dengan banyak *batch* pada sebuah iterasi. Akan tetapi, sampel *batch* diambil secara acak berulang kali sehingga diharapkan seluruh data mendapat bagian dalam *training*.

$$\text{MSSE} = \frac{\frac{1}{h} \sum_{t=n+1}^{n+h} (y_t - \hat{y}_t)^2}{\frac{1}{n-1} \sum_{t=2}^n (y_t - y_{t-1})^2} \quad (3.6)$$

Validasi model ramalan dilakukan dengan menghitung nilai MSSE dari variabel SIRD diskrit. Validasi dilakukan untuk mengukur kinerja model pada data di luar *training set* dan untuk penyempurnaan model (*tuning*). Validasi

Hasil ramalan model *deep learning* tidak bisa langsung digunakan atau dievaluasi karena label juga ditransformasikan pada tahap *preprocessing*. Tahap *postprocessing* meliputi langkah-langkah yang perlu dilakukan untuk mengembalikan ramalan ke bentuk semula, yaitu jumlah kasus COVID-19. Proses ini dilakukan dengan melakukan kebalikan dari tiap langkah transformasi variabel SIRD dengan urutan yang juga berkebalikan. Variabel SIRD diskrit ditransformasikan kembali menjadi jumlah kasus COVID-19 dengan mencari nilai delta kompartemen dari tiap variabel:

Keempat nilai delta tersebut kemudian digunakan untuk membangun kembali tiap kompartemen SIRD.

3.1.7 Instrumentasi

46

aplikasi untuk visualisasi model dan eksperimen *machine learning* dalam bentuk *dashboard*.

3.1.8 Penyempurnaan Model

Penyempurnaan model dilakukan dengan penyesuaian *hyperparameter* atau penyesuaian arsitektur model (Goodfellow et al., 2016). Dalam penelitian ini, *hyperparameter* dioptimalkan menggunakan Optuna (Akiba et al., 2019) dan sebagian *hyperparameter* mempengaruhi arsitektur. Model untuk semua target berbagi *hyperparameter* yang sama. Optuna mengoptimasi *hyperparameter* dengan nilai *Mean Square Scaled Error* (MSSE) validasi. Keseluruhan *pipeline* dieksekusi untuk setiap percobaan *hyperparameter* atau desain; model dilatih ulang setiap kali eksekusi. Penyempurnaan model dilakukan selama dua minggu.

Tabel 3.1 *Hyperparameter*

Hyperparameter	Tipe	Nilai
<i>State size</i> model dasar	Integer	[3, 56]
<i>Stack</i> representasi	Integer	[1, 20]
<i>Stack</i> adapter	Integer	[0, 5]
Jumlah <i>kernel</i> CNN1D	Integer	[3, 37]
Ukuran <i>kernel</i> CNN1D	Integer	[3, 15]
<i>Mean</i> W0	Float	[0.0, 1.0]
<i>Std</i> W0	Float	[0.0, 0.5]
Combine head shared depth	Integer	[0, 5]
Combine <i>head depth</i>	Integer	[1, 20]
Tambahan panjang input	Integer	[0, 4]
Ukuran <i>batch</i>	Integer	$16 \times 2^{[0,5]}$
Learning Rate	Float	$[10^{-5}, 10^{-2}]$
Bobot <i>loss</i> sumber	Float	[0.5, 1.0]
Fungsi Aktivasi	Object	{ReLU, LeakyReLU, Sigmoid, Tanh, SELU}
Variabel Eksogen	List	{All, Beta, Delta_I, I, Corr, Nonbinary, None}

Selain *hyperparameter* model, ada enam *hyperparameter* lain. Tambahan panjang *input* menjadi salah satu *hyperparameter* sehingga pembuatan *batch* dilakukan di dalam Optuna. Ini dapat dilakukan karena data *validation set* dan *test set* dihitung untuk label. Pilihan tanggal penting dan bobot daerah sumber juga menjadi *hyperparameter*; bobot daerah sumber memiliki nilai maksimal 1 yang berarti berbobot sama dengan daerah *target*. Terakhir, *learning rate*, fungsi aktivasi, dan ukuran *batch* juga menjadi *hyperparameter*.

Evaluasi dilakukan untuk mengukur kinerja model untuk data baru (Goodfellow et al., 2016), mengetahui kebergunaan tiap bagian model (Li et al., 2021), dan membandingkan model dengan model lain. Untuk mengukur kinerja model untuk data baru, model dilatih menggunakan data *training set* hingga nilai validasi mulai naik, lalu dievaluasi menggunakan *test set* yang belum pernah dilihat oleh model (Hastie et al., 2009). Kinerja yang diukur ini menjadi estimasi kinerja model untuk peramalan sesungguhnya.

kinerja model dengan tingkat keterbatasan data yang berbeda. Model hanya diharapkan mencapai nilai RMSSE di bawah satu (lebih baik daripada model *naïve*) dan memiliki kinerja lebih baik daripada model pembanding. Signifikansi perbedaan model diuji dengan Friedman *test* dan Bonferroni-Dunn *post-hoc test* (Demšar, 2006).

1. Model dasar (LSTM encoder-decoder) tanpa joint learning
2. Model tanpa representation layer
3. Model utuh dengan *shared branch* yang hanya dilatih dengan satu daerah dengan data terbanyak dalam *cluster* yang sama
4. Model utuh dengan *shared branch* yang dilatih dengan semua *dataset*
5. Model utuh yang sepenuhnya *shared*
6. Model utuh yang sepenuhnya *private*

1. Model ARIMA pada variabel SIRD diskrit
2. Model SIRD

Sesuai rumusan masalah kedua, model digunakan untuk meramalkan jumlah kumulatif kasus positif aktif, sembuh, dan meninggal COVID-19 hingga 9 bulan setelah tanggal data yang terakhir diambil. Peramalan dilakukan untuk daerah-

BAB IV

4.1. Pengumpulan Data

Data jumlah kasus kumulatif harian COVID-19 tiap kabupaten/kota Jawa Timur diperoleh dari *website* JATIM TANGGAP COVID-19 milik Pemerintah Provinsi Jawa Timur (Pemerintah Provinsi Jawa Timur, 2021). Data jumlah kasus tersebut diperoleh dengan mengambilnya dalam bentuk JSON dari API *endpoint*: https://covid19dev.jatimprov.go.id/xweb/drax/data/tgl_awal/tgl_akhir. API *endpoint* tersebut diakses dengan GET *request* dengan *tgl_awal* dan *tgl_akhir* dalam bentuk yyyy-mm-dd. API *endpoint* tersebut akan memberikan data satu hari untuk semua kota/kabupaten yaitu pada tanggal *tgl_akhir*. Dari JSON tiap daerah, diambil nilai *confirm* (jumlah kumulatif kasus positif total), *confirm_selesai* (jumlah kumulatif kasus sembuh), dan *meninggal_rtprcr* (jumlah kumulatif kasus meninggal). Didapatkan data dari 38 kabupaten/kota dengan data terpendek dari Kabupaten Sampang dengan 313 data (mulai 12 Mei 2020), data terpanjang dari Kota Surabaya, Kota Malang, Kabupaten Magetan, dan Kabupaten Sidoarjo dengan 366 data (mulai 20 Maret 2020), dan rata-rata 350,5 data. Sebagian kabupaten/kota memiliki data yang hilang pada hari tertentu.

Data jumlah tes dan vaksinasi kumulatif harian diperoleh dari *repository* Github Our World In Data (OWID) (Hasell et al., 2020; Mathieu et al., 2021). Data dari Github OWID yang tersedia adalah data per negara atau negara bagian AS, sehingga data tes dan vaksinasi yang dapat diperoleh adalah data keseluruhan Indonesia (data tes dan vaksinasi Jawa Timur tidak tersedia). Data dari Github OWID diperoleh dengan cara mengunduh data tersebut. Data jumlah tes berawal dari sebelum 20 Maret 2020, sedangkan data jumlah vaksinasi berawal dari 12 Januari 2021. Kedua data ini juga memiliki kekosongan pada hari tertentu.

Rekap data periode PSBB terbaru tidak ditemukan, sehingga data periode PSBB akan dikumpulkan dari situs-situs berita secara manual. Ditemukan bahwa PSBB seiring waktu digantikan oleh PPKM dan PPKM mikro, sehingga keduanya

juga digunakan. Di antara 38 kabupaten/kota, 6 melakukan PSBB, 29 melakukan PPKM, dan 38 (semuanya) melakukan PPKM mikro.

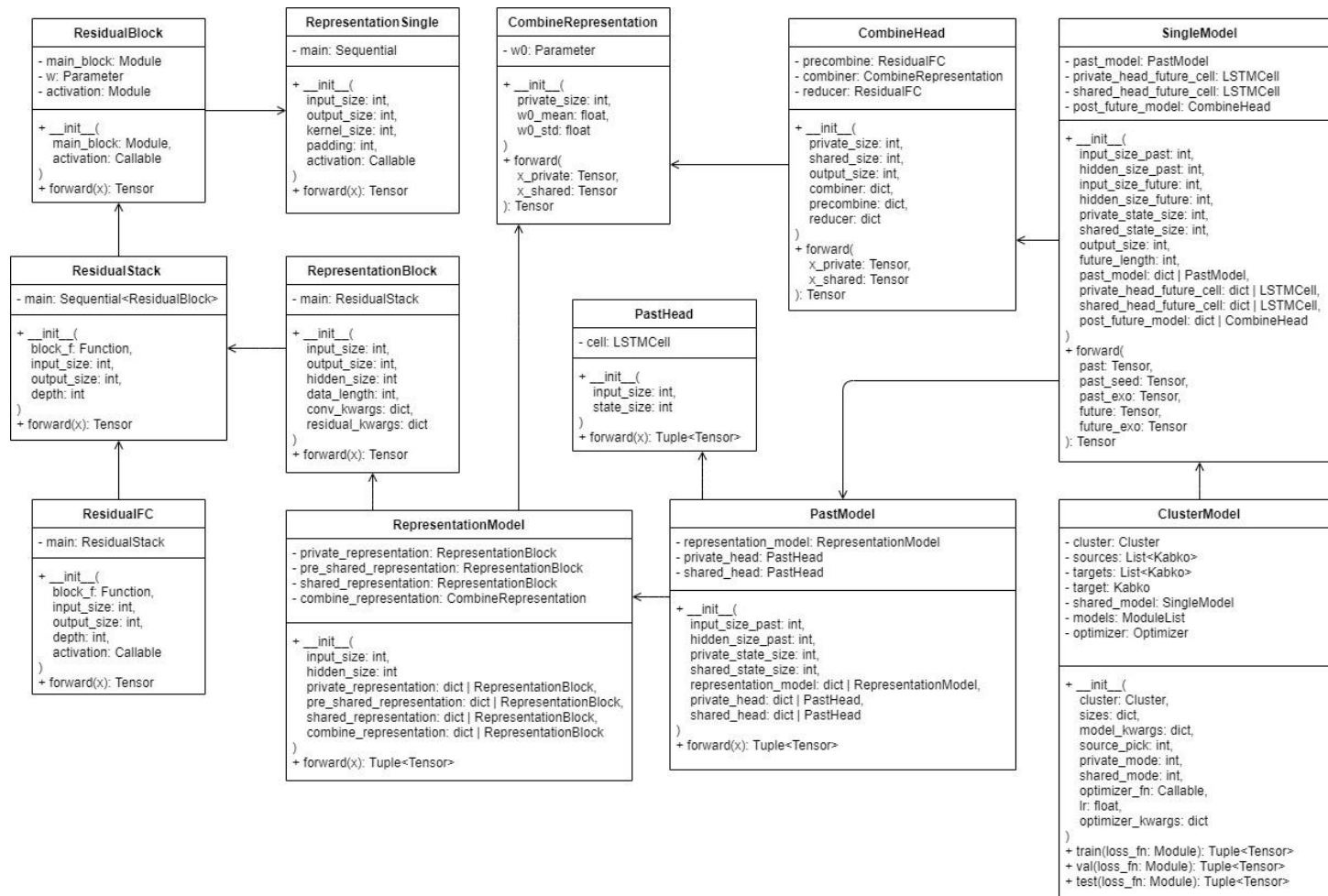
Untuk menghitung variabel SIRD, diperlukan data populasi daerah yang diperoleh dari *website* BPS Jawa Timur (Badan Pusat Statistik Provinsi Jawa Timur, 2021).

Eksplorasi data dilakukan untuk memahami data, mengidentifikasi pola, tren, *seasonality*, dan sebagainya (Hyndman & Athanasopoulos, 2018). Koefisien korelasi dihitung untuk tiap variabel eksogen, termasuk tanggal, dengan jumlah kasus, perubahannya, dan variabel SIRD (ilustrasi pada Gambar 4.1). Tanggal diubah menjadi data *time-series* biner untuk dapat dicari koefisien korelasinya. Karena tidak semua data kontinu dan normal, digunakan Kendall Tau untuk menghitung koefisien korelasi. Koefisien korelasi dihitung untuk setiap kabupaten/kota dan diambil yang terbesar. Variabel dengan koefisien korelasi di bawah 0.3 dianggap bisa diabaikan (Akoglu, 2018), sehingga sebaliknya, variabel dengan koefisien korelasi 0.3 atau lebih akan digunakan. Matriks korelasi yang dihasilkan terdapat pada Gambar 4.2. Dari matriks korelasi Gambar 4.2, didapat variabel berikut: *delta_test*, *vac_full_s*, *ppkm*, *ppkm_mikro*, *psbb*, *ramadhan*.

kumulatif kasus positif aktif, perubahannya, dan variabel beta dari semua kabupaten/kota. Observasi hanya dilakukan pada ketiga grafik tersebut karena tanggal tersebut hanya diperkirakan atau terbukti mempengaruhi mobilitas atau perilaku penduduk dan, sebagai akibatnya, jumlah kasus positif aktif. Tanggal yang terlihat berpengaruh untuk lebih dari 50% kabupaten/kota digunakan dalam penelitian ini. Hasil observasi dipaparkan pada Tabel 4.1.

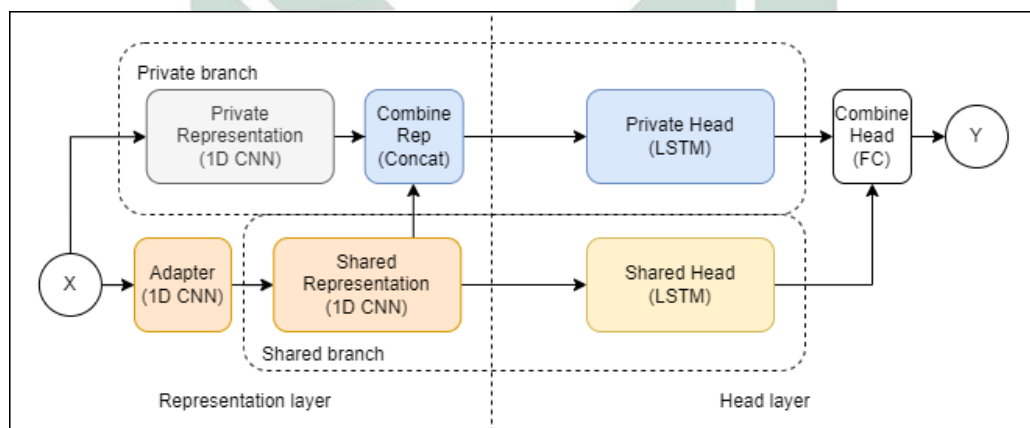
Tabel 4.1. Hasil observasi tanggal terhadap pola jumlah kasus positif aktif (I), perubahannya (ΔI), dan beta.

Grafik	Tanggal
Jumlah kasus positif aktif (I)	hut_ri, idul_adha, idul_fitri_siswa, idul_fitri_umum, libur_semester_ganjil, maulid_nabi, maulid_nabi_ext, natal, natal_ext, pilkada, ppkm, ppkm_mikro, psbb, tahun_baru_hijriyah, tahun_baru_hijriyah_ext, tahun_baru_masehi
Perubahan jumlah kasus positif aktif (Delta I)	hut_ri, idul_adha, idul_fitri_siswa, idul_fitri_umum, libur_semester_ganjil, maulid_nabi, maulid_nabi_ext, natal, natal_ext, pilkada, ppkm, ppkm_mikro, psbb, tahun_baru_hijriyah, tahun_baru_hijriyah_ext, tahun_baru_masehi
Beta	hari_buruh, idul_adha, idul_fitri_siswa, idul_fitri_umum, imlek, libur_awal_puasa, libur_semester_ganjil, maulid_nabi, maulid_nabi_ext, natal, natal_ext, pilkada, ppkm, ppkm_mikro, psbb, ramadhan, tahun_baru_hijriyah, tahun_baru_hijriyah_ext, tahun_baru_masehi
Gabungan	hari_buruh, hut_ri, idul_adha, idul_fitri_siswa, idul_fitri_umum, imlek, libur_awal_puasa, libur_semester_ganjil, maulid_nabi, maulid_nabi_ext, natal, natal_ext, pilkada, ppkm, ppkm_mikro, psbb, ramadhan, tahun_baru_hijriyah, tahun_baru_hijriyah_ext, tahun_baru_masehi



Gambar 4.4 Class diagram implementasi model

RepresentationSingle adalah *wrapper 1D convolution*. RepresentationBlock adalah ResidualStack (meskipun diimplementasikan tidak sebagai turunan tapi *wrapper*) dengan RepresentationSingle sebagai blok utama. RepresentationBlock digunakan untuk blok *Private Representation*, *Adapter (Pre-shared Representation)*, dan *Shared Representation* (Gambar 4.5). Class CombineRepresentation menyambung tensor dari alur *private* dan *shared* setelah mengalikan tensor *private* dengan tensor parameter w_0 (Gambar 4.6). Class RepresentationLayer meliputi keseluruhan *Representation Layer*.



iliki tren yang cukup jelas (Gambar 4.10) dan Adf 3, sehingga data *time series* ini non-stasioner. el SIRD diskrit *beta*, grafik sudah terlihat stasioner memberikan nilai $p=0.0021$, sehingga data *time series* ini sudah meliputi *differencing* satu kali.

Rolling Mean & Rolling Standard Deviation for i with window=30

g Mean
g Std

2020-07 2020-08 2020-09 2020-10 2020-11 2020-12 2021-01 2021-02

4.10 Grafik jumlah kasus positif aktif Kabupaten Pangasinan

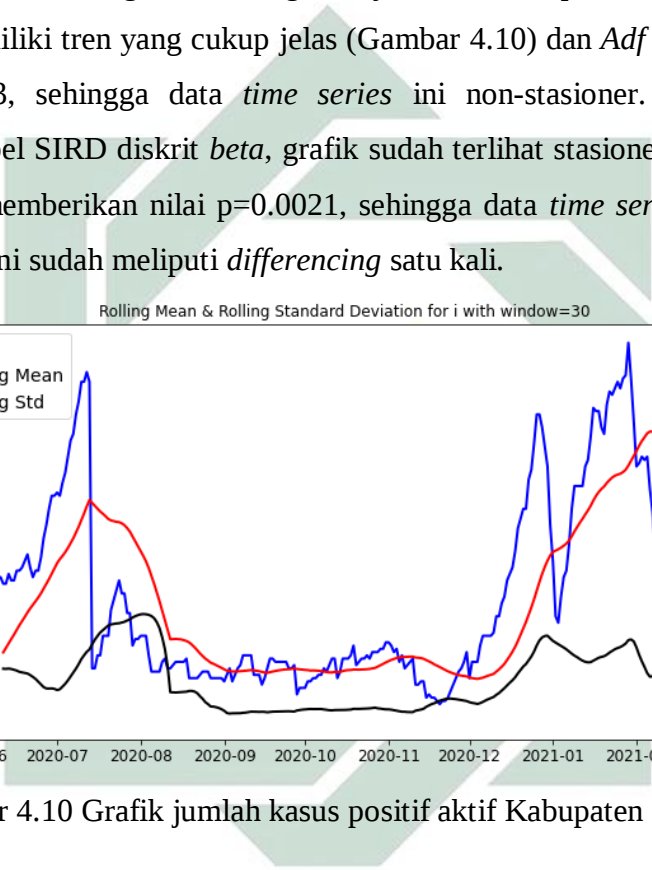
iliki tren yang cukup jelas (Gambar 4.10) dan Adf 3, sehingga data *time series* ini non-stasioner. el SIRD diskrit *beta*, grafik sudah terlihat stasioner memberikan nilai $p=0.0021$, sehingga data *time series* ini sudah meliputi *differencing* satu kali.

Rolling Mean & Rolling Standard Deviation for i with window=30

g Mean
g Std

2020-07 2020-08 2020-09 2020-10 2020-11 2020-12 2021-01 2021-02

4.10 Grafik jumlah kasus positif aktif Kabupaten Pangasinan



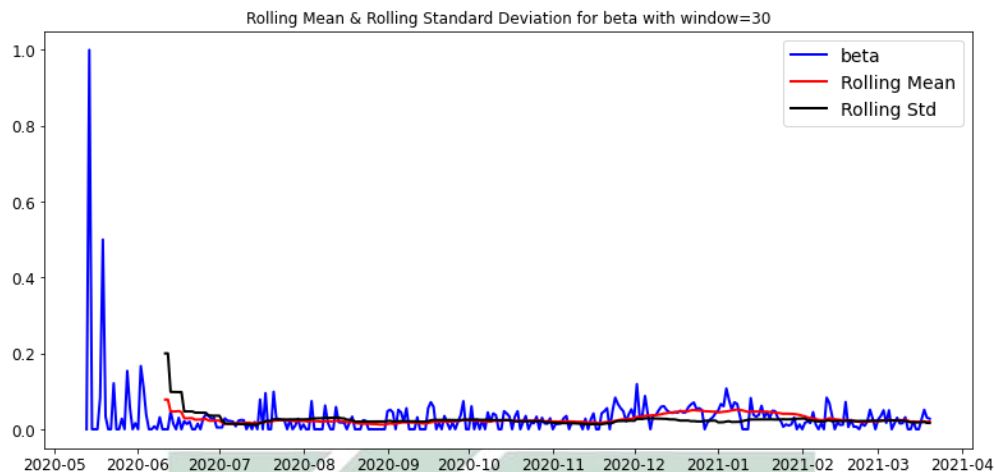
iliki tren yang cukup jelas (Gambar 4.10) dan Adf 3, sehingga data *time series* ini non-stasioner. el SIRD diskrit *beta*, grafik sudah terlihat stasioner memberikan nilai $p=0.0021$, sehingga data *time series* ini sudah meliputi *differencing* satu kali.

Rolling Mean & Rolling Standard Deviation for i with window=30

g Mean
g Std

2020-07 2020-08 2020-09 2020-10 2020-11 2020-12 2021-01 2021-02

4.10 Grafik jumlah kasus positif aktif Kabupaten Pangasinan



Gambar 4.11 Grafik *beta* Kabupaten Sampang

Transformasi ke variabel SIRD diskrit sudah meliputi *differencing* satu kali. Untuk *beta* dan jumlah kasus positif aktif, ditemukan bahwa 4 dari 38 kabupaten/kota memiliki *beta* yang masih belum stasioner sedangkan 1 dari 38 kabupaten/kota memiliki jumlah kasus positif aktif yang memerlukan *differencing* dua kali untuk menjadi stasioner. Untuk *gamma* dan jumlah kasus sembuh, ditemukan bahwa 20 dari 38 kabupaten/kota memiliki *gamma* yang belum stasioner, sedangkan 27 dari 38 kabupaten/kota memiliki jumlah kasus sembuh yang memerlukan *differencing* dua kali untuk menjadi stasioner. Untuk *delta* dan jumlah kasus meninggal, ditemukan bahwa hanya 8 dari 38 kabupaten/kota memiliki *delta* yang belum stasioner, sedangkan 26 dari 38 kabupaten/kota memiliki jumlah kasus meninggal yang memerlukan *differencing* dua kali untuk menjadi stasioner. Ini berarti transformasi ke variabel SIRD lebih baik daripada *differencing* untuk membuat *time series* menjadi stasioner kecuali untuk *beta* yang sedikit lebih buruk. Detail keunggulan dan kekurangan transformasi ini dipaparkan pada Tabel 4.2.

Tabel 4.3 Nilai *silhouette* dan jumlah *single cluster* untuk tiap jumlah cluster (n) terpilih untuk tiap kelompok

Kelompok	N	Silhouette	Non-Single	Single
90 Hari	4	0.2667	4	0
180 Hari	4	0.5383	3	1
366 Hari	4	0.4075	3	1
Sebelum Vaksinasi	2	0.4734	2	0

Tabel 4.4 Hasil *clustering* untuk kelompok pembatasan data 90 hari

Cluster	Target	Sources
0	Kab. Tuban (90, 2020-04-09)	Kab. Banyuwangi, Kab. Blitar, Kab. Jember, Kab. Situbondo
1	Kab. Bojonegoro (90, 2020-04-10)	Kab. Kediri, Kab. Pamekasan
2	Kota Madiun (90, 2020-05-07)	Kab. Bondowoso, Kab. Madiun, Kab. Magetan, Kab. Malang, Kab. Ngawi, Kab. Trenggalek, Kota Blitar, Kota Kediri, Kota Malang, Kota Probolinggo
3	Kab. Sampang (90, 2020-05-13)	Kab. Bangkalan, Kab. Gresik, Kab. Jombang, Kab. Lamongan, Kab. Lumajang, Kab. Mojokerto, Kab. Nganjuk, Kab. Pacitan, Kab. Pasuruan, Kab. Ponorogo, Kab. Probolinggo, Kab. Sidoarjo, Kab. Sumenep, Kab. Tulungagung, Kota Batu, Kota Mojokerto, Kota Pasuruan, Kota Surabaya

Tabel 4.7 Hasil *clustering* untuk kelompok sebelum vaksinasi

Cluster	Target	Sources
0	Kab. Sampang (254, 2020-05-13)	Kab. Bangkalan, Kab. Banyuwangi, Kab. Blitar, Kab. Bondowoso, Kab. Gresik, Kab. Jember, Kab. Jombang, Kab. Lamongan, Kab. Lumajang, Kab. Madiun, Kab. Magetan, Kab. Malang, Kab. Mojokerto, Kab. Nganjuk, Kab. Ngawi, Kab. Pacitan, Kab. Pasuruan, Kab. Ponorogo, Kab. Probolinggo, Kab. Sidoarjo, Kab. Situbondo, Kab. Sumenep, Kab. Trenggalek, Kab. Tuban, Kab. Tulungagung, Kota Batu, Kota Blitar, Kota Kediri, Kota Madiun, Kota Malang, Kota Mojokerto, Kota Probolinggo, Kota Surabaya
1	Kota Pasuruan (286, 2020-04-11)	Kab. Bojonegoro, Kab. Kediri, Kab. Pamekasan

4.4.3. Preprocessing (2)

Setelah *clustering*, data tiap daerah dinormalisasikan ke dalam *range* 0 hingga 1 dengan MinMaxScaler bersamaan dengan daerah lainnya dalam *cluster* yang sama. Data tiap daerah lalu dipotong menjadi beberapa sampel dengan panjang 44 atau lebih (masih dalam bentuk pandas DataFrame), yaitu 14 label sebagai jarak peramalan dan setidaknya 30 prediktor sebagai dasar peramalan. Data lalu dipisahkan label dan variabel eksogennya. Kemudian, data dijadikan *numpy array* dan diberi objek Kabko sebagai *reference* ke daerah yang memilikinya. Terakhir, data dimuat dalam PyTorch DataLoader untuk digunakan dengan model PyTorch.

4.4.4. Training

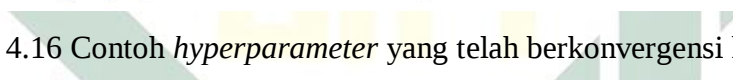
Joint learning dilakukan dengan melatih model dari *cluster* yang sama bersamaan; semua model dari *cluster* yang sama berbagi *shared branch* yang sama. *Training* dilakukan sebanyak jumlah *batch* paling sedikit dari semua daerah untuk menghindari bias jumlah data, sehingga tidak semua *batch* digunakan dari dataset dengan banyak *batch* pada sebuah iterasi. *Batch* diambil secara acak. *Loss* daerah sumber diboboti lebih kecil agar pelatihan model fokus pada target.

Training dilakukan dengan *Mean Square Scaled Error* (MSSE) sebagai *loss function*. MSSE didapat dari menghilangkan akar dari *Root Mean Square Scaled Error* (RMSSE) untuk memudahkan perhitungan gradien. MSSE adalah MSE (*mean squared error*) peramalan dibagi MSE peramalan naif satu langkah.

Validasi model ramalan dilakukan dengan menghitung nilai MSSE dari variabel SIRD diskrit. Validasi dilakukan untuk mengukur kinerja model pada data di luar *training set* dan untuk penyempurnaan model (*tuning*). Validasi dilakukan menggunakan data validasi yang telah disisihkan sebelumnya. Validasi MSSE digunakan untuk mengidentifikasi *overfitting* serta menghindarinya dengan menghentikan *training* ketika *loss* validasi mulai naik (Hastie et al., 2009) atau jika tidak ada perkembangan dalam waktu yang lama. Nilai MSSE validasi diusahakan serendah mungkin dan diharapkan di bawah satu.

Hasil ramalan model *deep learning* tidak bisa langsung digunakan atau dievaluasi karena label juga ditransformasikan pada tahap *preprocessing*. Tahap *postprocessing* meliputi langkah-langkah yang perlu dilakukan untuk mengembalikan ramalan ke bentuk semula, yaitu jumlah kasus COVID-19. Proses ini dilakukan dengan melakukan kebalikan dari tiap langkah transformasi variabel SIRD dengan urutan yang juga berkebalikan.

Instrumentasi *pipeline* diperlukan untuk diagnosa model (Goodfellow et al., 2016). Instrumentasi dilakukan menggunakan Captum (Kokhlikyan et al., 2009), Optuna (Akiba et al., 2019), dan TensorBoard (TensorFlow, 2021). Metode Saliency dari Captum digunakan untuk menghitung bobot keseluruhan dari tiap *feature* dari tiap input (Gambar 4.12) dan bobot keseluruhan dari tiap *layer* (Gambar 4.13). Optuna digunakan untuk visualisasi sebaran *hyperparameter* sejauh ini (Gambar 4.14); visualisasi ini digunakan untuk memeriksa apakah *hyperparameter* sudah berkonvergensi. TensorBoard digunakan untuk visualisasi grafik *training loss* dan *validation loss* serta grafik penghitung *early stopping*; visualisasi ini digunakan untuk *debugging* mekanisme *early stopping*.

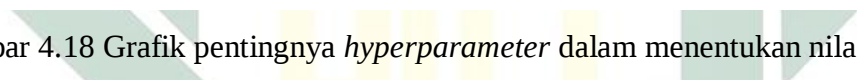


erkonvergenzi

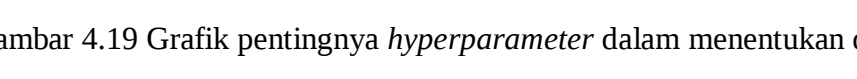


erkonvergenzi

).



Parameter	Value
representation_past_private_depth	0.19
combine_head_depth	0.17
representation_past_shared_depth	0.00
hidden_size_past	0.00



percobaan

hanya dilakukan dengan memilih *order* terbaik dari hasil eksplorasi yang dicoba untuk dataset gabungan *training* dan validasi.

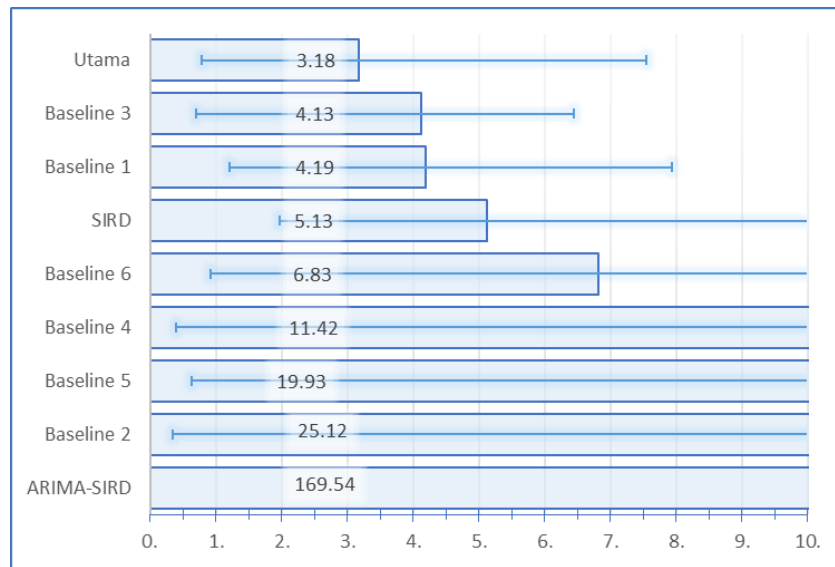
Tabel 4.9 Perbandingan *tuning* model utama dengan model *baseline*

Model	CPU (x2 core)	Percobaan Selesai	Parameter Konvergen	Nilai Objektif Terbaik
Utama	3	593/697	14/24	54.7022
Baseline 1	1	1178/1231	3/10	53.8513
Baseline 2	2	433/485	7/15	54.4529
Baseline 3	1	684/738	8/24	54.6473
Baseline 4	4	349/458	10/24	55.2519
Baseline 5	4	314/423	10/24	56.2065
Baseline 6	1	774/825	12/23	54.2258

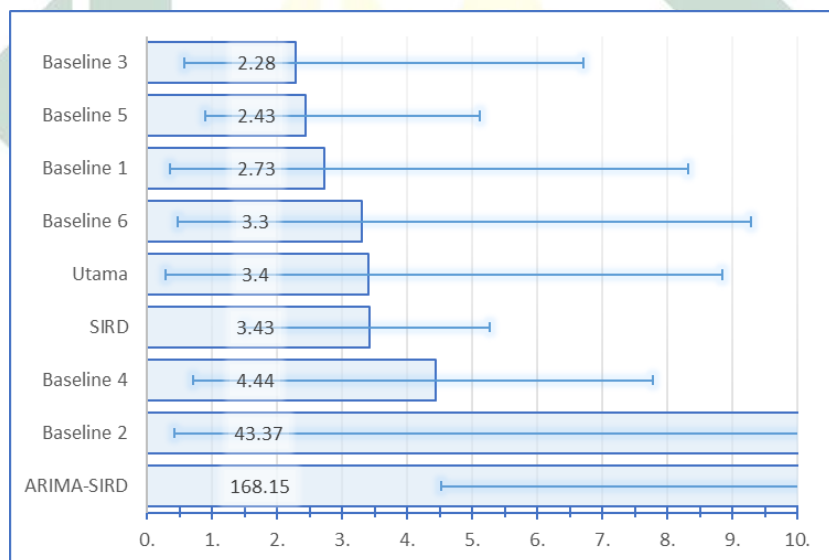
4.7. Evaluasi

Evaluasi dilakukan untuk mengukur kinerja model untuk data baru (Goodfellow et al., 2016), mengetahui kebergunaan tiap bagian model (Li et al., 2021), dan membandingkan model dengan model lain. Untuk mengukur kinerja model untuk data baru, model dilatih menggunakan *training set* hingga nilai validasi mulai naik untuk menghindari *overfitting* atau setidaknya 100 *epoch*, lalu dievaluasi menggunakan *test set* yang belum pernah dilihat oleh model (Hastie et al., 2009). Kinerja yang diukur ini menjadi estimasi kinerja model untuk peramalan sesungguhnya.

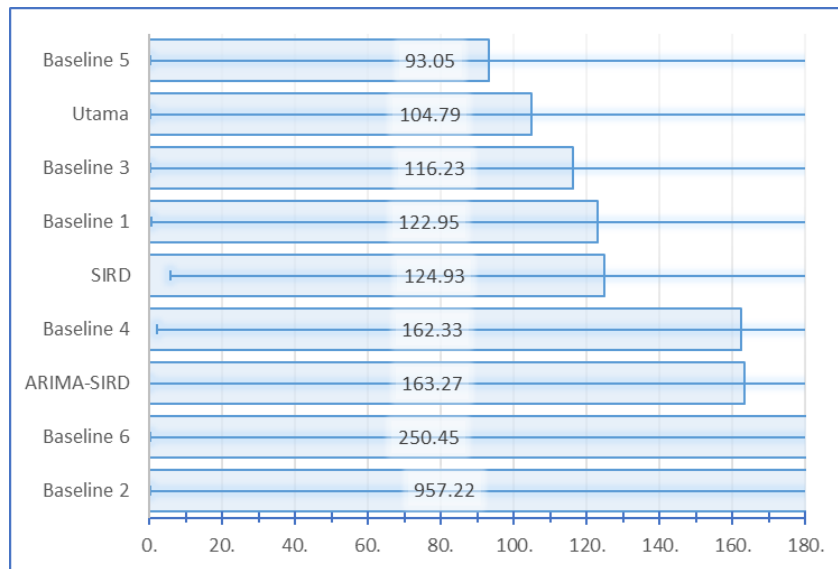
Evaluasi model ramalan dilakukan dengan menghitung nilai *Root Mean Square Scaled Error* (RMSSE) rata-rata dari ramalan jumlah total kasus positif, sembuh, dan meninggal. Model dievaluasi untuk tiap daerah target dari tiap kelompok jumlah data, yaitu dengan data 3 bulan (90 hari), 6 bulan (180 hari), 1 tahun (365 hari), serta data sebelum ada vaksinasi. Pembagian data menjadi beberapa kelompok dengan jumlah data berbeda dilakukan untuk mengukur kinerja model dengan tingkat keterbatasan data yang berbeda. Model hanya diharapkan mencapai nilai RMSSE di bawah satu (lebih baik daripada model *naïve* untuk peramalan dalam sampel) dan memiliki kinerja lebih baik daripada



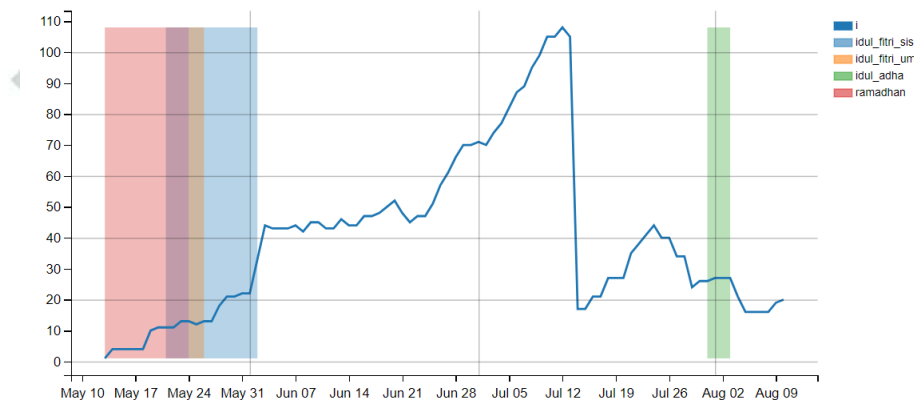
Gambar 4.20 Rata-rata RMSSE evaluasi untuk jumlah kasus positif aktif (i) kelompok 90 hari



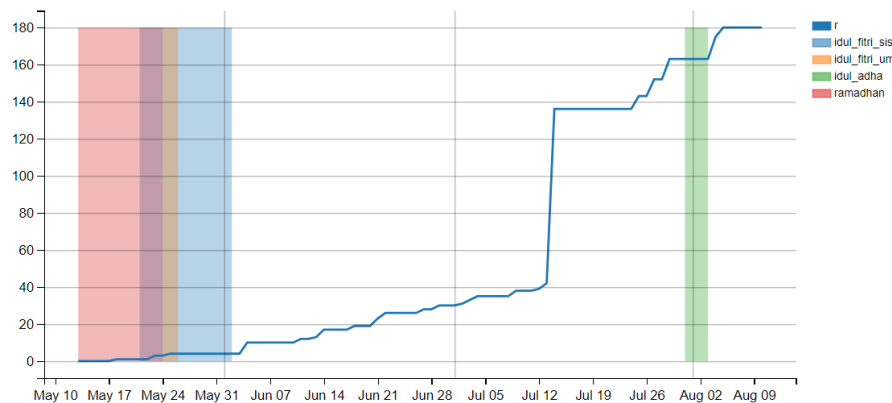
Gambar 4.21 Rata-rata RMSSE evaluasi untuk jumlah kasus sembuh (r) kelompok
90 hari



Gambar 4.22 Rata-rata RMSSE evaluasi untuk jumlah kasus meninggal (d) kelompok 90 hari



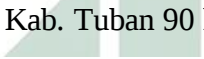
Gambar 4.23 Grafik jumlah kasus positif aktif (i) Kab. Sampang 90 hari pertama



Gambar 4.24 Grafik jumlah kasus sembuh (r) Kab. Sampang 90 hari pertama



26 Grafik jumlah kasus sembuh (r) Kab. Tuban 90

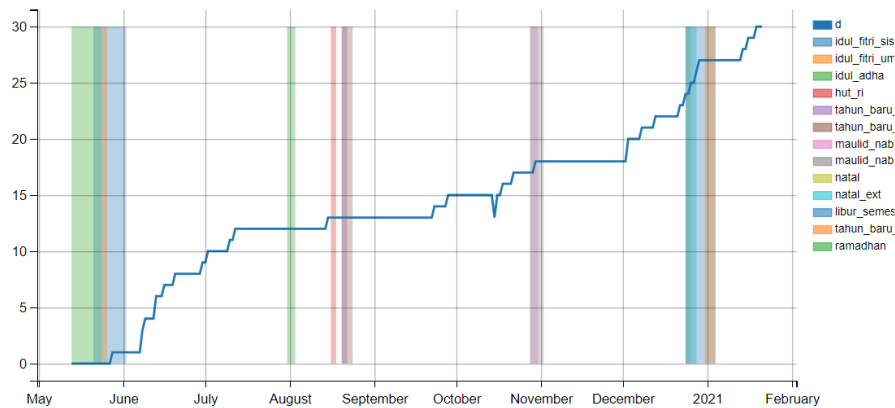


Jul 12 Jul 19 Jul 26 Aug

Kelompok ketiga ini menggunakan semua data dan memiliki dinamika yang rumit. Hampir setengah dari model-model (9 dari 21) yang dilatih untuk evaluasi kelompok ini terbentuk dari jumlah epoch (training step) yang sangat kecil (Tabel 4.14) Ini dapat berarti model tidak berkonvergensi dengan baik, yang bisa disebabkan oleh data yang terlalu sedikit, pola data yang terlalu sulit, atau model yang tidak cocok.

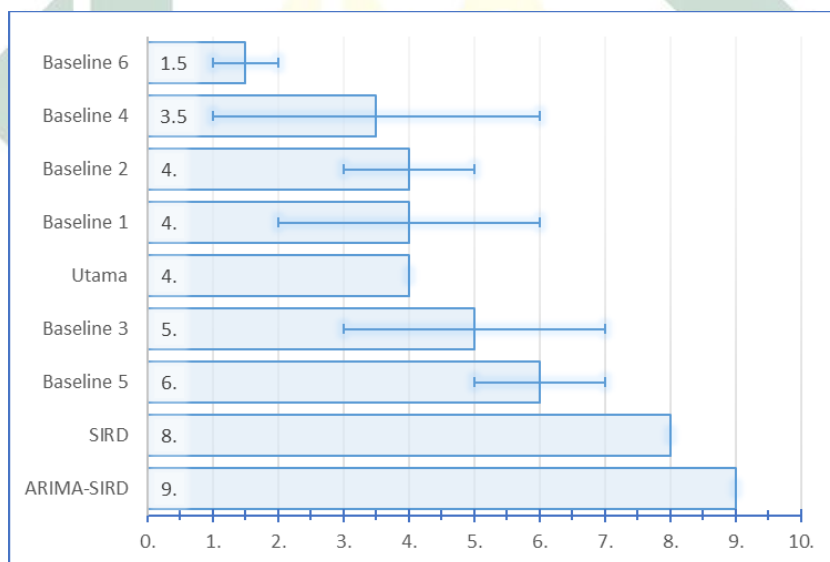
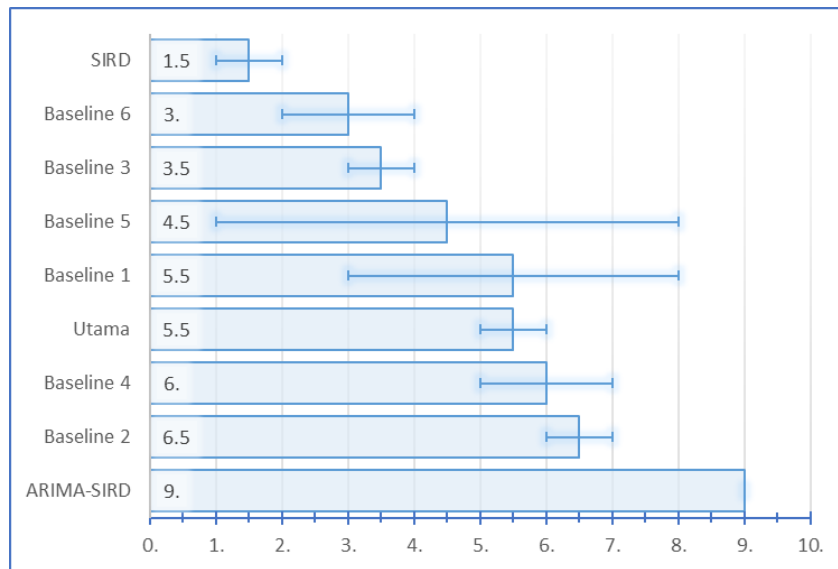
Model	Kab. Tuban	Kab. Sampang	Kota Pasuruan	Rata-rata
Utama	70/126	55/113	67/103	64.00/114.00
Baseline 1	82/111	87/111	100/108	89.67/110.00
Baseline 2	90/110	17/103	10/103	39.00/105.33
Baseline 3	63/105	36/109	32/109	43.67/107.67
Baseline 4	105/109	105/109	105/109	105.00/109.00
Baseline 5	10/109	54/103	5/112	23.00/108.00
Baseline 6	34/107	48/107	11/104	31.00/106.00
Rata-rata	64.86/111.00	57.43/107.86	47.14/106.86	56.48/108.57

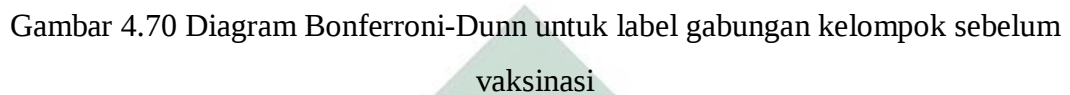
88



Gambar 4.65 Grafik jumlah kasus meninggal (d) Kab. Sampang hingga 21 Jan 2021

Evaluasi dilanjutkan dengan uji statistik untuk menguji signifikansi perbedaan tiap model menggunakan rata-rata *rank* (Gambar 4.66 hingga Gambar 4.69). Hasil uji *Adjusted Friedman* (FF) menghasilkan $p > 0.05$ untuk semua label dan hanya mendapat $p < 0.05$ ketika semua label digabungkan (Tabel 4.15). Perbedaan *rank* tiap model *baseline* dan pembandingan dengan model utama yang melebihi nilai CD Bonferroni-Dunn hanya model ARIMA-SIRD untuk label gabungan (Gambar 4.70). Model tersebut memiliki rata-rata *rank* yang lebih buruk dari model utama. Ini berarti model utama lebih baik dari model ARIMA-SIRD secara signifikan. Dengan demikian, model utama hanya lebih baik secara signifikan dari model ARIMA-SIRD untuk kelompok sebelum vaksinasi. Karena model utama tidak berbeda secara signifikan dengan model *baseline* manapun, model utama tidak lebih baik daripada LSTM biasa (*baseline* 1) dan bagian-bagian model utama selain LSTM kurang berguna.





Evaluasi masing-masing kelompok telah dibahas. Bagian ini berisi kesimpulan dari keempat kelompok. Untuk setiap jenis model, sekitar setengah dari model yang dibuat memiliki *best epoch* (jumlah *training step* dimana model mendapat *validation loss* terkecil) yang kecil. Sebagian besar dari model-model itu berasal dari kelompok data 90 hari yang memiliki jumlah data paling sedikit tapi juga pola data paling sederhana, sehingga ini mungkin disebabkan oleh data yang terlalu sedikit atau pola yang terlalu mudah. Meskipun begitu, jumlah model yang seperti itu juga tidak sedikit untuk kelompok lainnya, sehingga mungkin juga model tidak cocok atau pola data terlalu sulit.

101

Semua model memperoleh nilai yang sangat buruk untuk jumlah kasus meninggal (d) Kota Madiun pada kelompok 90 hari dan Kab. Sampang pada kelompok sebelum vaksinasi. Pada Kota Madiun kelompok 90 hari, grafik sangat rata dan ada lonjakan di ujung yang menjadi bagian dari *test set*. Model hanya dapat mempelajari data yang rata sedangkan *test set* memiliki lonjakan yang sangat mendadak, sehingga memang sangat sulit diprediksi. Selain itu, nilai RMSSE yang sangat besar juga disebabkan oleh pembagi RMSSE yang bernilai sangat kecil ketika tidak ada perubahan jumlah kasus dalam waktu lama (grafik rata/*flat*). Pada Kab. Sampang kelompok sebelum vaksinasi, tren grafik sudah terlihat secara kasat mata, tapi masih memiliki banyak bagian yang rata. Meskipun begitu, jumlah kasus meninggal (d) Kab. Sampang pada kelompok 90 hari, 180 hari, dan 366 hari tidak memiliki masalah ini. Untuk penelitian selanjutnya, mungkin sebaiknya data *training* dihaluskan terlebih dahulu dan digunakan alternatif RMSSE yang tidak terlalu terdampak oleh data yang rata.

1. Model utama tidak lebih baik daripada model LSTM *encoder-decoder* biasa
2. Model utama lebih baik daripada model ARIMA-SIRD
3. Model utama mungkin lebih baik dari model SIRD
4. *Joint learning* dengan model semua daerah sekaligus tanpa *clustering* dapat memperburuk model

Tabel 4.18 Persentase pencapaian RMSSE di bawah 1 untuk tiap kelompok

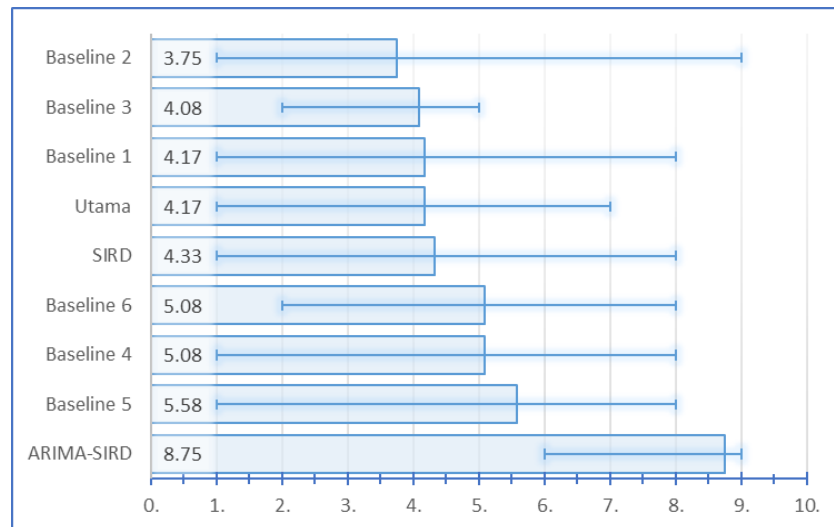
Model	Kelompok Pembatasan Data				Total
	90 Hari	180 Hari	366 Hari	Sebelum Vaksinasi	
Utama	33.33%	0.00%	11.11%	0.00%	13.89%
Baseline 1	16.67%	11.11%	11.11%	0.00%	11.11%
Baseline 2	25.00%	11.11%	11.11%	0.00%	13.89%
Baseline 3	41.67%	0.00%	0.00%	0.00%	13.89%
Baseline 4	16.67%	0.00%	11.11%	0.00%	8.33%
Baseline 5	25.00%	22.22%	11.11%	0.00%	16.67%
Baseline 6	25.00%	0.00%	0.00%	0.00%	8.33%
ARIMA-SIRD	8.33%	0.00%	0.00%	0.00%	2.78%
SIRD	0.00%	0.00%	0.00%	0.00%	0.00%
Total	21.30%	4.94%	6.17%	0.00%	9.88%

Tabel 4.19 Persentase pencapaian RMSSE di bawah 1 untuk tiap label

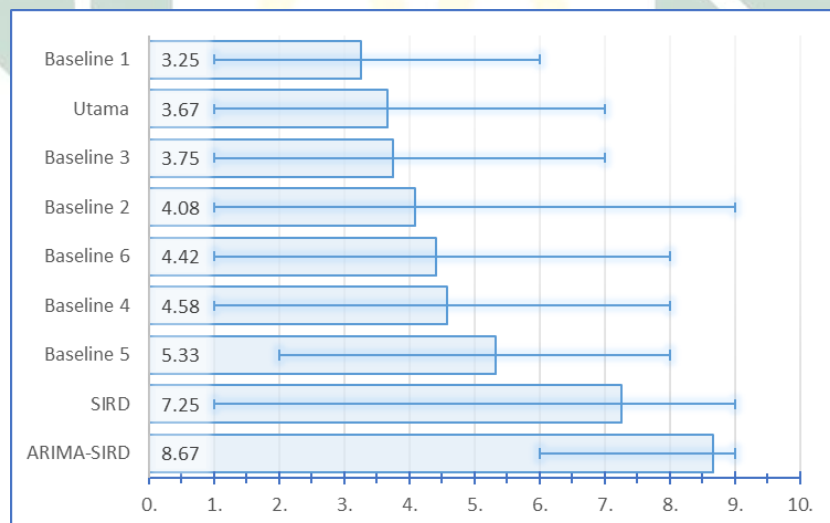
Model	I	R	D	Total
Utama	8.33%	25.00%	8.33%	13.89%
Baseline 1	0.00%	25.00%	8.33%	11.11%
Baseline 2	8.33%	25.00%	8.33%	13.89%
Baseline 3	8.33%	25.00%	8.33%	13.89%
Baseline 4	8.33%	16.67%	0.00%	8.33%
Baseline 5	8.33%	25.00%	16.67%	16.67%
Baseline 6	8.33%	8.33%	8.33%	8.33%
ARIMA-SIRD	0.00%	0.00%	8.33%	2.78%
SIRD	0.00%	0.00%	0.00%	0.00%
Total	5.56%	16.67%	7.41%	9.88%

Jika semua kelompok digabungkan dan dievaluasi, model ARIMA-SIRD dinilai lebih buruk secara signifikan untuk tiap label dan label gabungan, sedangkan model SIRD dinilai lebih buruk secara signifikan untuk label gabungan dan semua label kecuali jumlah kasus positif aktif (i). Model *baseline* 4 tidak dinilai berbeda secara signifikan dalam uji statistik kelompok gabungan. Rata-rata *rank* dipaparkan pada Gambar 4.71 hingga Gambar 4.74, sedangkan hasil uji

statistik dan diagram Bonferroni-Dunn secara berurutan dipaparkan pada Tabel 4.20 dan Gambar 4.75 hingga Gambar 4.78.



Gambar 4.71 Rata-rata *rank* tiap model untuk jumlah kasus positif aktif (i) kelompok gabungan



Gambar 4.72 Rata-rata *rank* tiap model untuk jumlah kasus sembuh (r) kelompok gabungan

4.8. Peramalan

Sesuai rumusan masalah kedua, model digunakan untuk meramalkan jumlah kumulatif kasus positif aktif, sembuh, dan meninggal COVID-19 hingga 9 bulan setelah tanggal data yang terakhir diambil. Model yang digunakan adalah model utama. Peramalan dilakukan untuk gabungan dari daerah-daerah *target* dari kelompok-kelompok sebelumnya. Peramalan dilakukan dengan model yang dilatih dengan gabungan data *training* dan validasi dan divalidasi dengan data *test* hingga nilai validasi naik untuk menghindari *overfitting*, atau setidaknya 100 *epoch*.

Berbeda dari sebelumnya, *clustering* untuk peramalan ini dilakukan menggunakan semua data. Jumlah *cluster* yang terpilih dengan nilai *silhouette* tertinggi adalah 2 dengan nilai *silhouette* 0.3757. Hasil *clustering* dipaparkan pada Tabel 4.21.

Tabel 4.21 Hasil *clustering* untuk peramalan

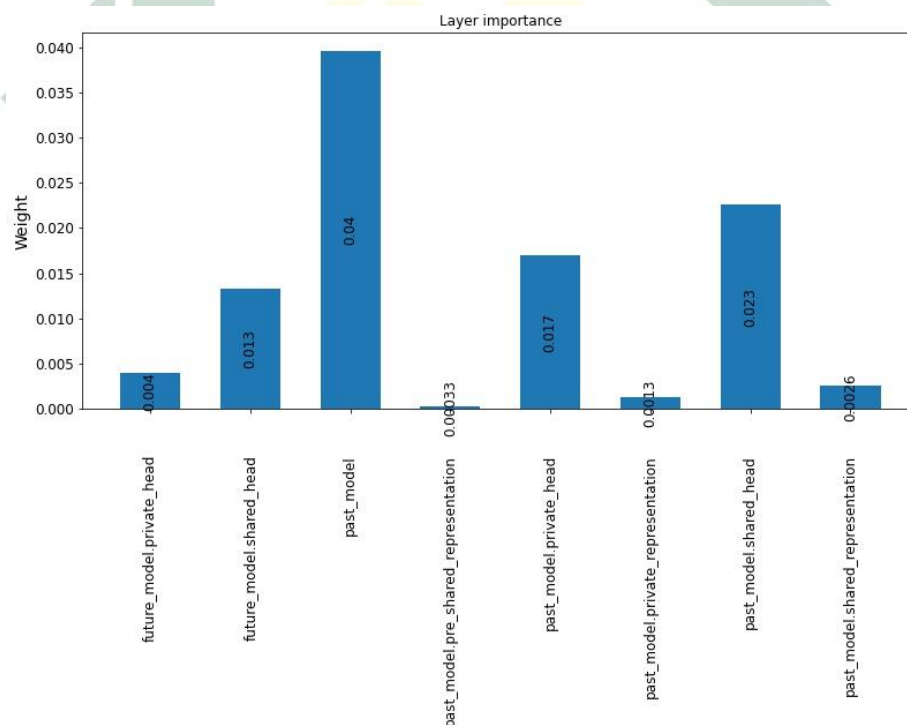
Cluster	Target	Sources
0	Kab. Sampang (312, 2020-05-13) Kab. Tuban (346, 2020-04-09) Kota Madiun (318, 2020-05-07) Kota Pasuruan (344, 2020-04-11)	KAB. BANGKALAN, KAB. BANYUWANGI, KAB. BLITAR, KAB. BONDOWOSO, KAB. GRESIK, KAB. JEMBER, KAB. JOMBANG, KAB. LAMONGAN, KAB. LUMAJANG, KAB. MADIUN, KAB. MAGETAN, KAB. MALANG, KAB. MOJOKERTO, KAB. NGANJUK, KAB. NGAWI, KAB. PACITAN, KAB. PASURUAN, KAB. PONOROGO, KAB. PROBOLINGGO, KAB. SIDOARJO, KAB. SITUBONDO, KAB. SUMENEP, KAB. TRENGGALEK, KAB. TULUNGAGUNG, KOTA BATU, KOTA BLITAR, KOTA KEDIRI, KOTA MALANG, KOTA MOJOKERTO, KOTA PROBOLINGGO, KOTA SURABAYA
1	Kab. Bojonegoro (344, 2020-04-10)	KAB. KEDIRI, KAB. PAMEKASAN

Tabel 4.22 Epoch untuk tiap kabupaten/kota untuk peramalan

Cluster	Kabupaten/Kota	Best Epoch	Total Epoch
0	Kab. Sampang	55	164
	Kab. Tuban	55	164
	Kota Madiun	55	164
	Kota Pasuruan	55	164
1	Kab. Bojonegoro	50	134

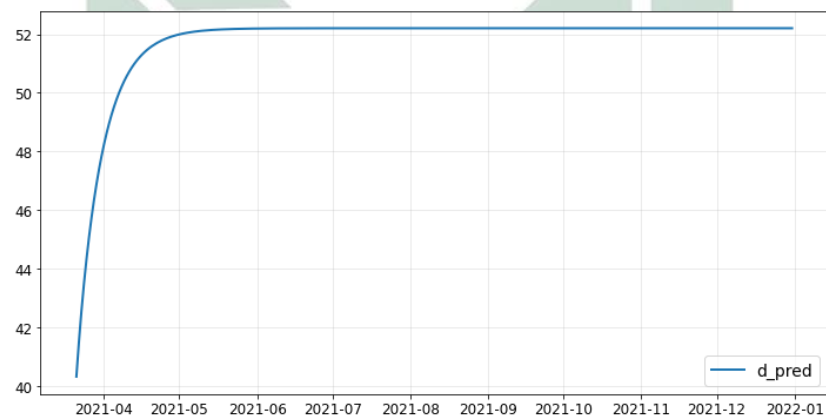
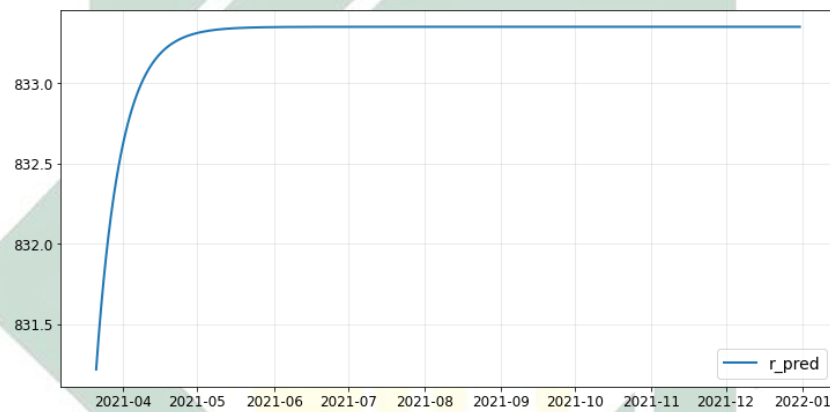
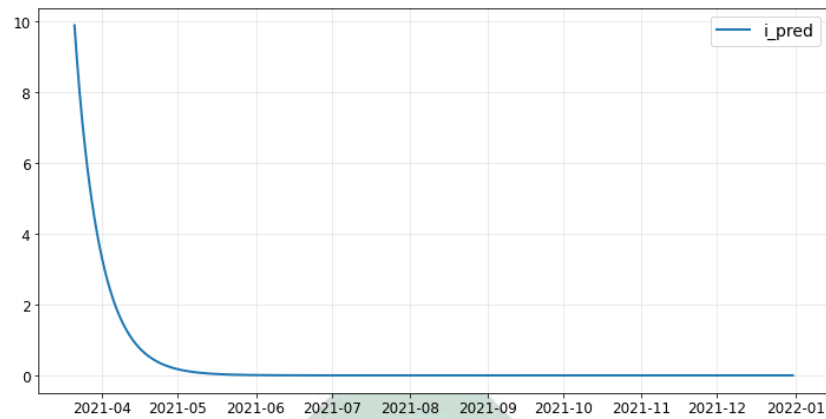
4.8.1. Kab. Sampang

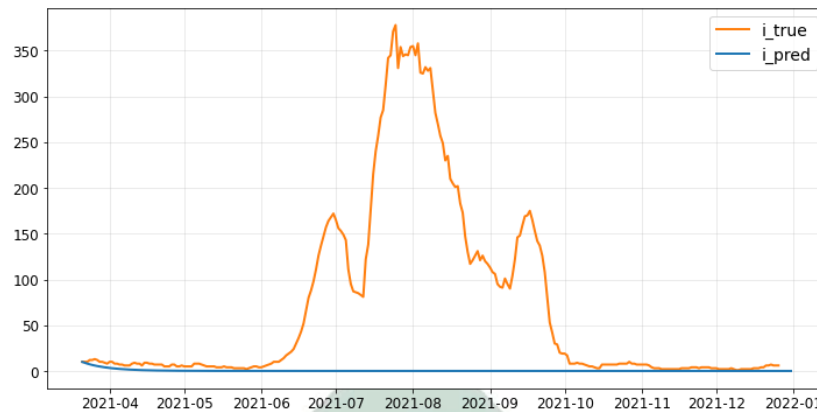
Attribution pada tiap blok/layer dengan metode Saliency (Gambar 4.79) menunjukkan bahwa bagian model yang penting untuk model adalah bagian *head*, yaitu LSTM. Meskipun begitu, bagian *shared* juga berpengaruh, yang berarti seharusnya model mempelajari sesuatu dari *joint learning*.



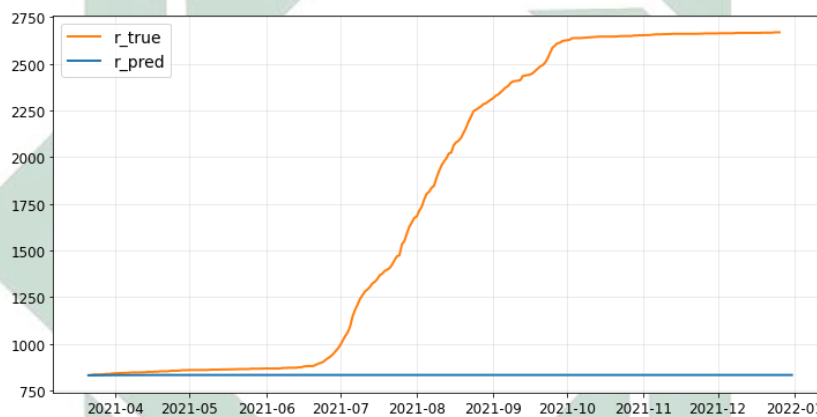
Gambar 4.79 *Saliency attribution* pada blok/layer model utama untuk peramalan
Kab. Sampang

Attribution pada *input feature* untuk dengan metode *Saliency* (Gambar 4.80) menunjukkan bahwa jumlah tes harian (*delta_test*) dan jumlah orang yang divaksinasi penuh (*vac_full_s*) tidak penting bagi model. Dari hasil *attribution*,

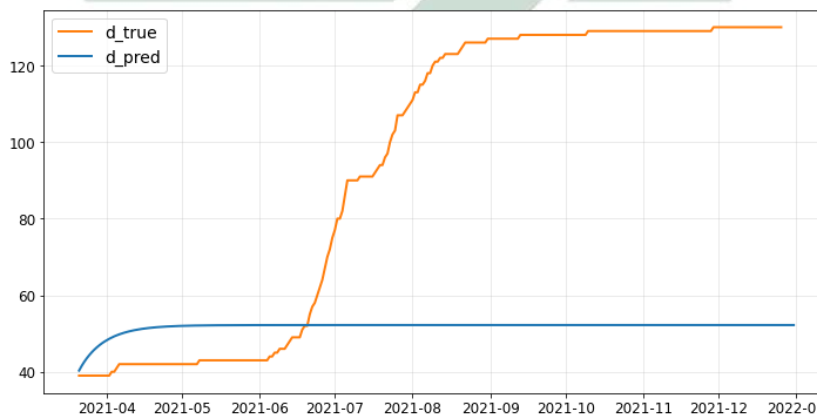




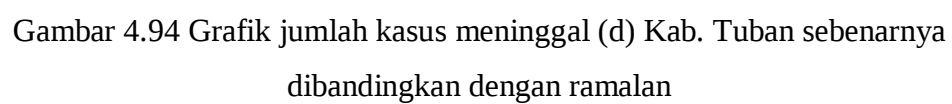
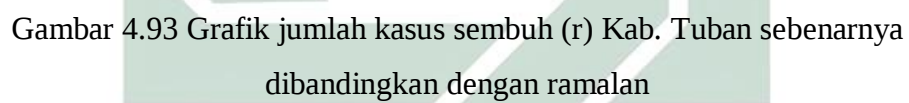
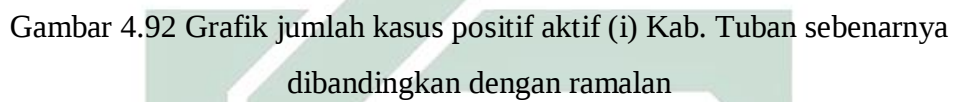
Gambar 4.84 Grafik jumlah kasus positif aktif (i) Kab. Sampang sebenarnya dibandingkan dengan ramalan



Gambar 4.85 Grafik jumlah kasus sembuh (r) Kab. Sampang sebenarnya dibandingkan dengan ramalan



Gambar 4.86 Grafik jumlah kasus meninggal (d) Kab. Sampang sebenarnya dibandingkan dengan ramalan



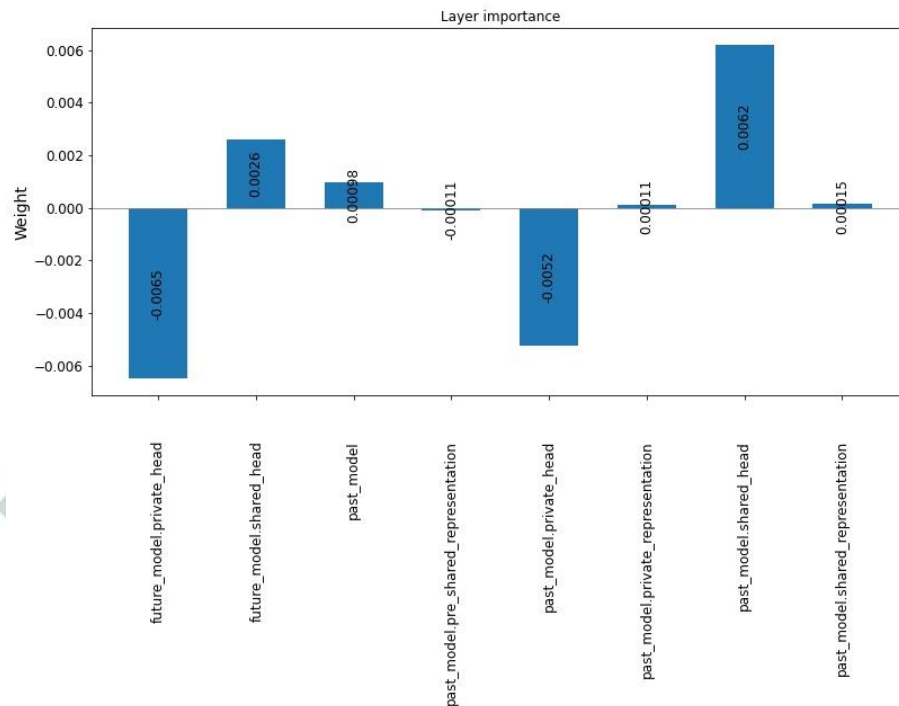
Attribution pada tiap blok/layer dengan metode Saliency (Gambar 4.95) menunjukkan bahwa bagian model yang penting untuk model adalah bagian *head*, yaitu LSTM. Meskipun begitu, bagian *shared* juga berpengaruh, yang berarti seharusnya model mempelajari sesuatu dari *joint learning*.

menunjukkan bahwa jumlah tes harian (δ) dan jumlah orang yang divaksinasi penuh (vac_full_s) tidak penting bagi model. Dari hasil analisis, dapat disimpulkan bahwa tanggal penting memiliki bobot yang tidak kalah dari parameter lain (β , γ , δ) dan beberapa tanggal justru lebih penting dari parameter lain. Tanggal penting ini adalah tanggal 14 Maret sebagai variabel endogen, yang berarti dianggap sangat penting oleh model. Tanggal penting lain yang memiliki bobot lebih besar ini adalah Maulid Nabi *extended* (diperpanjang hari libur nasional pada tanggal 14 Maret berikutnya yang dekat) dan hari buruh.



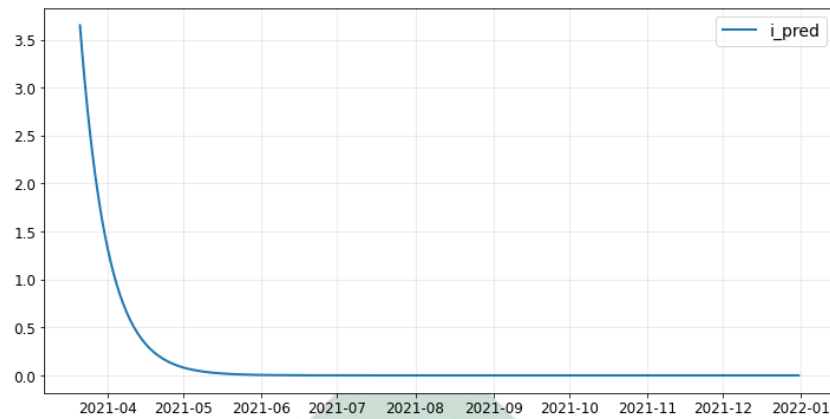
Kota Madiun

Attribution pada tiap blok/layer dengan metode Saliency (Gambar 4.103) menunjukkan bahwa bagian model yang penting untuk model adalah bagian *head*, yaitu LSTM. Meskipun begitu, bagian *shared* juga berpengaruh, yang berarti seharusnya model mempelajari sesuatu dari *joint learning*.

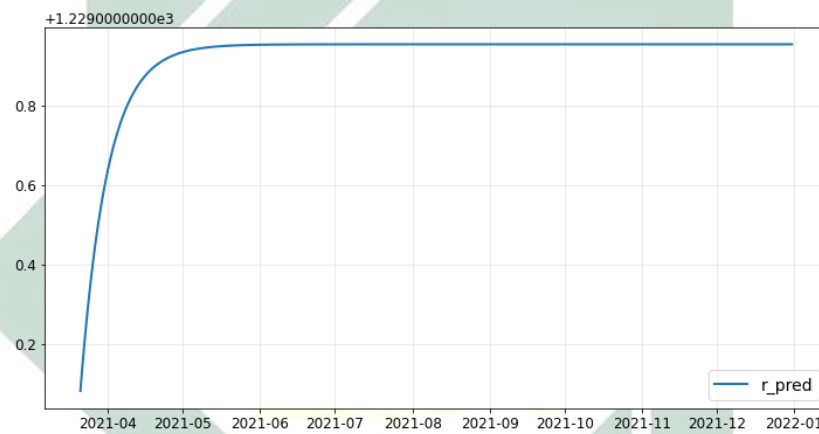


Gambar 4.103 *Saliency attribution* pada blok/layer model utama untuk peramalan Kota Pasuruan

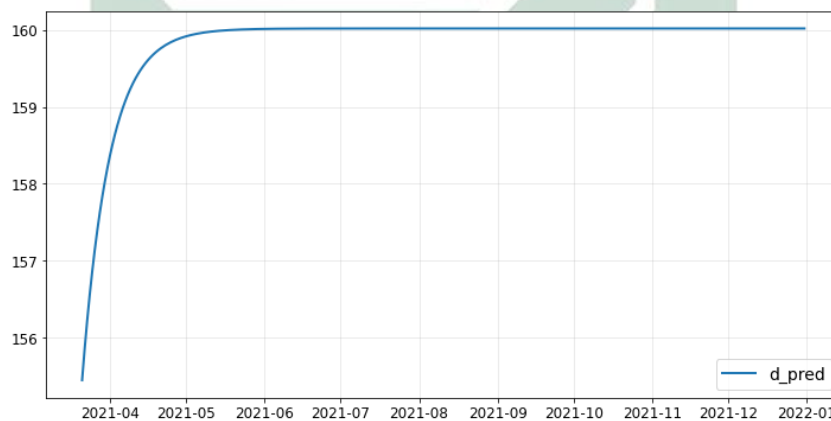
Attribution pada *input feature* untuk dengan metode *Saliency* (Gambar 4.104) menunjukkan bahwa jumlah tes harian (*delta_test*) dan jumlah orang yang divaksinasi penuh (*vac_full_s*) tidak penting bagi model. Dari hasil *attribution*, tampak bahwa tanggal penting memiliki bobot yang tidak kalah dari variabel endogen (*beta*, *gamma*, *delta*) dan beberapa tanggal justru lebih besar dari variabel endogen, yang berarti dianggap sangat penting oleh model. Tanggal yang berbobot lebih besar ini adalah Maulid Nabi *extended* (diperpanjang hingga hari libur berikutnya yang dekat) dan hari buruh.



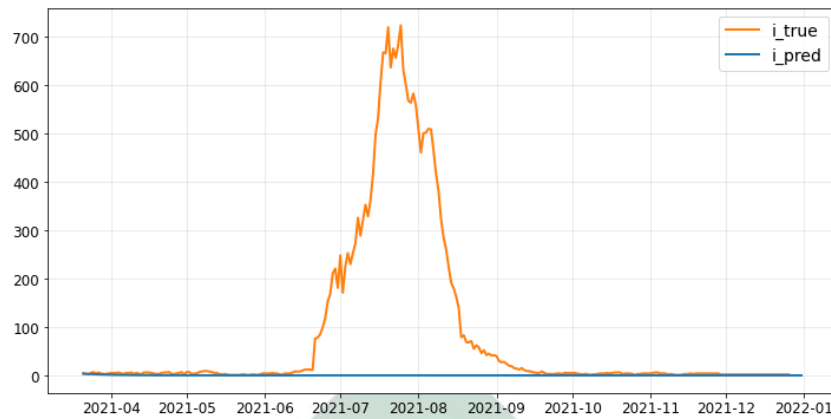
Gambar 4.105 Grafik ramalan jumlah kasus positif aktif (i) Kota Pasuruan



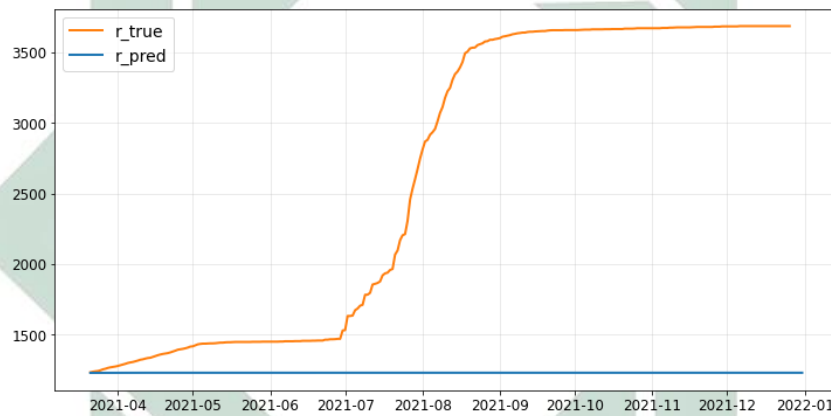
Gambar 4.106 Grafik ramalan jumlah kasus sembuh (r) Kota Pasuruan



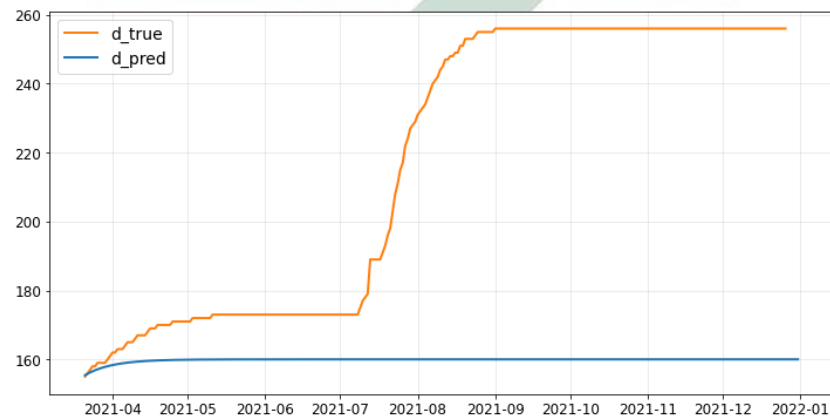
Gambar 4.107 Grafik ramalan jumlah kasus meninggal (d) Kota Pasuruan



Gambar 4.108 Grafik jumlah kasus positif aktif (i) Kota Pasuruan sebenarnya dibandingkan dengan ramalan

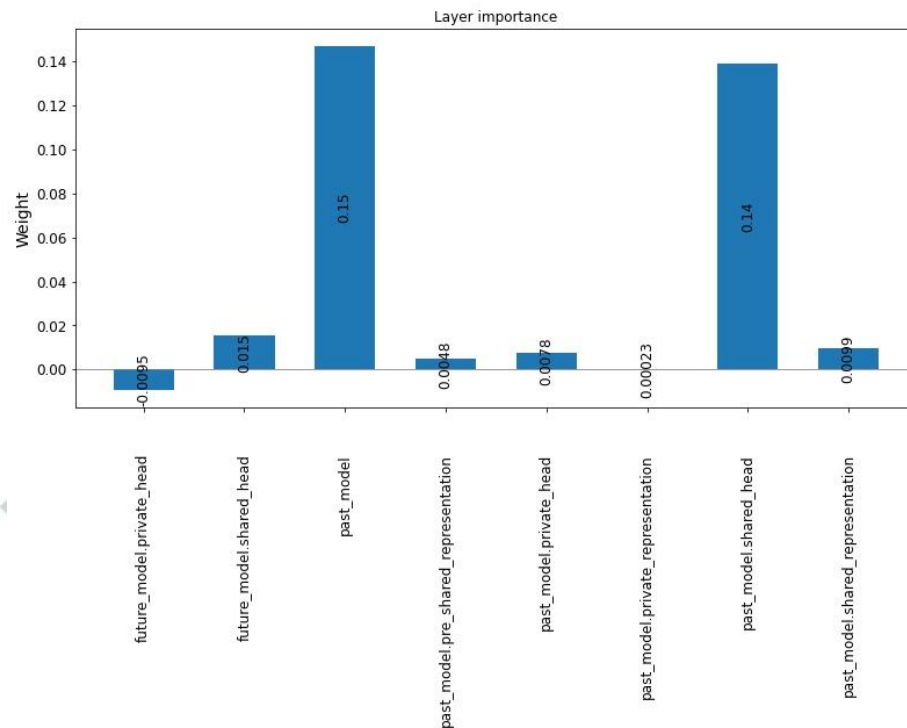


Gambar 4.109 Grafik jumlah kasus sembuh (r) Kota Pasuruan sebenarnya dibandingkan dengan ramalan



Gambar 4.110 Grafik jumlah kasus meninggal (d) Kota Pasuruan sebenarnya dibandingkan dengan ramalan

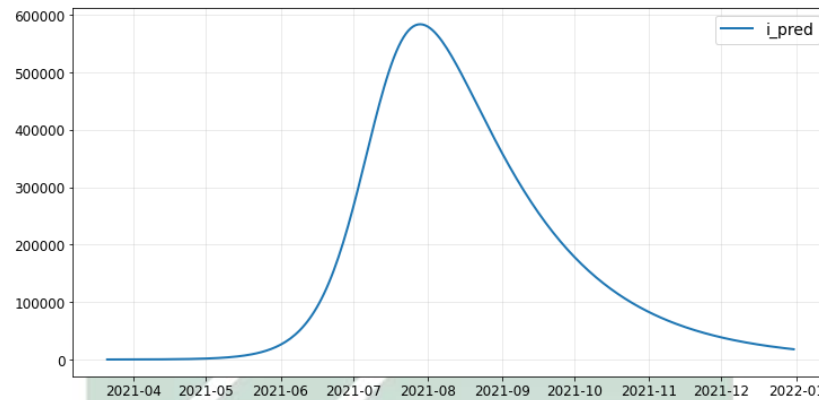
Attribution pada tiap blok/layer dengan metode Saliency (Gambar 4.111) menunjukkan bahwa bagian model yang penting untuk model adalah bagian *head*, yaitu LSTM. Meskipun begitu, bagian *shared* sangat berpengaruh, yang berarti seharusnya model mempelajari banyak dari *joint learning*.



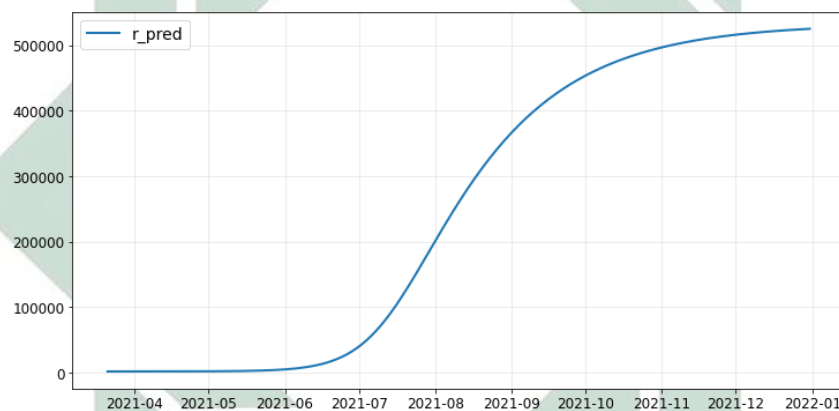
Gambar 4.111 *Saliency attribution* pada blok/layer model utama untuk peramalan
Kab. Bojonegoro

Attribution pada *input feature* untuk dengan metode *Saliency* (Gambar 4.112) menunjukkan bahwa jumlah tes harian (*delta_test*) dan jumlah orang yang divaksinasi penuh (*vac_full_s*) tidak penting bagi model. Dari hasil *attribution*, tampak bahwa tanggal penting memiliki bobot yang tidak kalah dari variabel endogen (*beta*, *gamma*, *delta*) dan beberapa tanggal justru lebih besar dari variabel endogen, yang berarti dianggap sangat penting oleh model. Tanggal yang berbobot lebih besar ini adalah bulan Ramadhan, libur idul fitri siswa dan umum, Maulid Nabi dan *extended* (diperpanjang hingga hari libur berikutnya yang dekat), HUT RI, dan hari buruh.

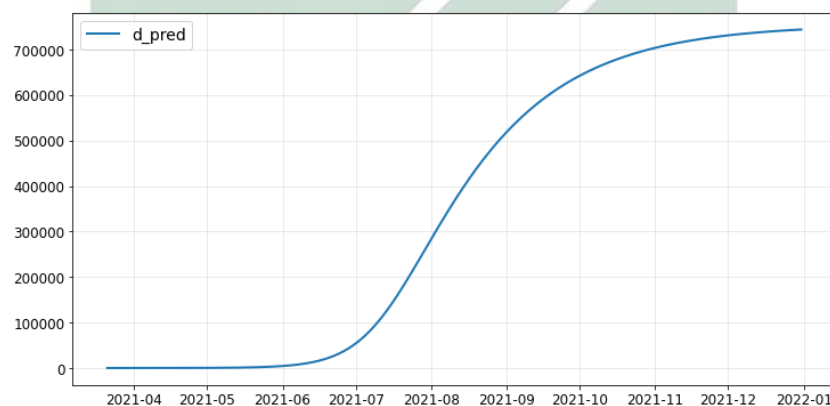
sebenarnya dengan cukup baik (Gambar 4.119, Gambar 4.120, dan Gambar 4.121), meskipun dengan skala yang terlalu besar.



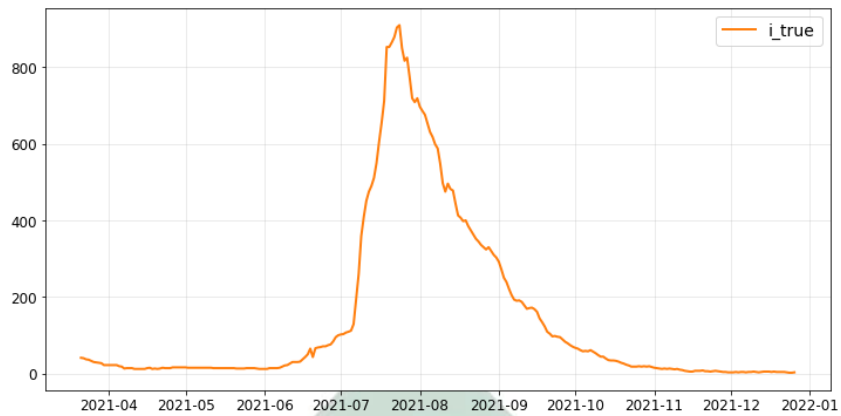
Gambar 4.113 Grafik ramalan jumlah kasus positif aktif (i) Kab. Bojonegoro



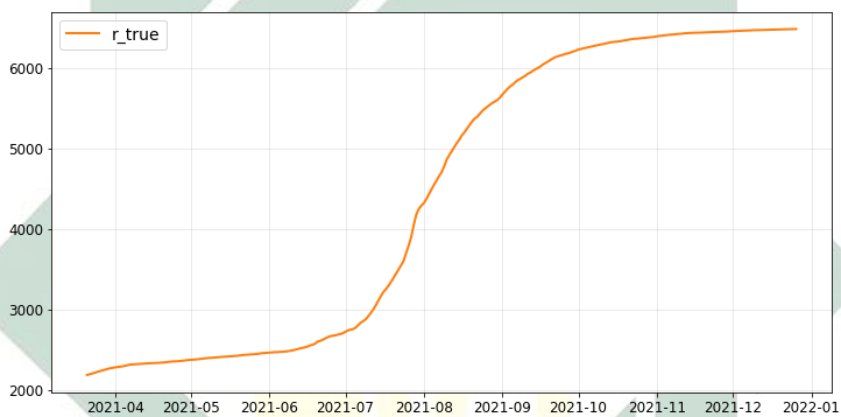
Gambar 4.114 Grafik ramalan jumlah kasus sembuh (r) Kab. Bojonegoro



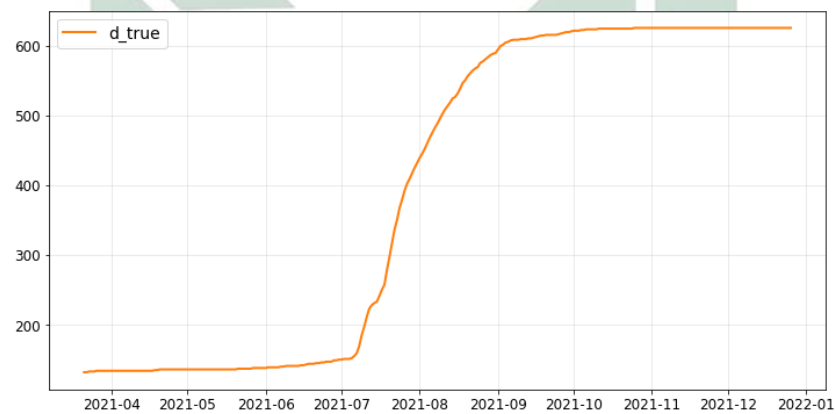
Gambar 4.115 Grafik ramalan jumlah kasus meninggal (d) Kab. Bojonegoro



Gambar 4.119 Grafik jumlah kasus positif aktif (i) Kab. Bojonegoro sebenarnya



Gambar 4.120 Grafik jumlah kasus sembuh (r) Kab. Bojonegoro sebenarnya



Gambar 4.121 Grafik jumlah kasus meninggal (d) Kab. Bojonegoro sebenarnya

Semua hasil ramalan tersebut berbeda jauh dari kenyataannya karena model hanya dilatih untuk meramal sejauh 14 hari dan adanya varian delta yang tidak ada dalam data *training*. Khusus untuk Kab. Bojonegoro, model berhasil menangkap bentuk grafik dengan cukup baik, tapi skalanya terlalu besar.

5.2. Saran

Selain transformasi variabel SIRD, terdapat pula perlakuan yang sama untuk semua model yang menggunakan *joint learning*, yaitu *clustering*. *Clustering* dilakukan menggunakan metode DTW Barycenter Averaging pada variabel SIRD (*beta*, *gamma*, dan *delta*). Sebelumnya telah ditemukan bahwa pemilihan sumber *joint learning* yang buruk, yaitu dengan menggunakan semua daerah tanpa *clustering*, dapat memperburuk model. Metode *clustering* atau pemilihan sumber yang digunakan dalam penelitian ini bisa jadi kurang tepat sehingga *joint learning* tidak berfungsi dengan baik dan model sulit atau gagal berkonvergensi. Penelitian

peramalan naif satu langkah yang menjadi pembagi RMSSE. Model ini adalah yang paling tepat dengan nilai kesalahan nol. Dengan demikian, sulit mencapai nilai RMSSE di bawah satu untuk *time series* yang stabil, seperti jumlah kasus meninggal. Ini seharusnya juga berlaku untuk model yang memiliki konsep yang sama, yaitu membagi *error* peramalan dengan *error* peramalan naif satu langkah dalam sampel (prediksi). Untuk lebih lanjut untuk mencari alternatif RMSSE dan MASE terhadap masalah ini.

peramalan naif satu langkah yang menjadi pembagi RMSSE. Model ini adalah yang paling tepat dengan nilai kesalahan nol. Dengan demikian, sulit mencapai nilai RMSSE di bawah satu untuk *time series* yang stabil, seperti jumlah kasus meninggal. Ini seharusnya juga berlaku untuk model yang memiliki konsep yang sama, yaitu membagi *error* peramalan dengan *error* peramalan naif satu langkah dalam sampel (prediksi). Untuk lebih lanjut untuk mencari alternatif RMSSE dan MASE terhadap masalah ini.

peramalan naif satu langkah yang menjadi pembagi RMSSE. Model ini adalah yang paling tepat dengan nilai kesalahan nol. Dengan demikian, sulit mencapai nilai RMSSE di bawah satu untuk *time series* yang stabil, seperti jumlah kasus meninggal. Ini seharusnya juga berlaku untuk model yang memiliki konsep yang sama, yaitu membagi *error* peramalan dengan *error* peramalan naif satu langkah dalam sampel (prediksi). Untuk lebih lanjut untuk mencari alternatif RMSSE dan MASE terhadap masalah ini.

- <https://health.detik.com/berita-detikhealth/d-5370421/kapan-vaksin-covid-19-diberikan-ke-masyarakat-ini-jadwalnya>
- Badan Pusat Statistik Provinsi Jawa Timur. (2021). *BPS Provinsi Jawa Timur*. BPS Provinsi Jawa Timur. Diambil 17 April 2020, dari <https://jatim.bps.go.id/>
- Batino, C., & Calonzo, A. (2021). *Philippine Capital Returns to Lockdown as Vaccinations Lag*. Bloomberg. Diambil 5 April 2020, dari <https://www.bloomberg.com/news/articles/2021-03-29/philippine-capital-returns-into-lockdown-as-vaccinations-lag>
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyperparameter optimization. *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, NIPS 2011*, 1–9.
- Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2016). *Time Series Analysis* (5 ed.). John Wiley & Sons, Inc. http://www.ru.ac.bd/stat/wp-content/uploads/sites/25/2019/03/504_05_Box_Time-Series-Analysis-Forecasting-and-Control-2015.pdf
- Calafiore, G. C., Novara, C., & Possieri, C. (2020). A time-varying SIRD model for the COVID-19 contagion in Italy. *Annual Reviews in Control, January*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7587010/pdf/main.pdf>
- Chatterjee, S., Sarkar, A., Chatterjee, S., Karmakar, M., & Paul, R. (2020). Studying the progress of COVID-19 outbreak in India using SIRD model. *Indian Journal of Physics*. <https://doi.org/10.1007/s12648-020-01766-8>
- covid19.go.id. (2021a). *Data Vaksinasi COVID-19 (Update per 3 April 2021)*. covid19.go.id. Diambil 5 April 2021, dari <https://covid19.go.id/p/berita/data-vaksinasi-covid-19-update-3-april-2021>
- covid19.go.id. (2021b). *Tahapan Kedatangan Vaksin COVID-19 di Indonesia*. covid19.go.id. Diambil 5 April 2020, dari <https://covid19.go.id/masyarakat-umum/tahapan-kedatangan-vaksin-covid-19-di-indonesia>
- covid19.go.id. (2021c). *Vaksin COVID-19 GRATIS, Tanpa syarat!* covid19.go.id. Diambil 5 April 2021, dari <https://covid19.go.id/edukasi/masyarakat->

- umum/vaksin-covid-19-gratis-tanpa-syarat
 covid19.go.id. (2021d). *Vaksinasi COVID-19 Tahap Kedua Dimulai*.
 covid19.go.id. Diambil 5 April 2020, dari
<https://covid19.go.id/berita/vaksinasi-covid-19-tahap-kedua-dimulai>
- Cui, Z., Chen, W., & Chen, Y. (2016). Multi-Scale Convolutional Neural Networks for Time Series Classification. *arXiv:1603.06995*.
<https://doi.org/10.21629/JSEE.2017.01.18>
- Cuturi, M., & Blondel, M. (2017). Soft-DTW: A differentiable loss function for time-series. *Proceedings of the 34th International Conference on Machine Learning*, 70, 894–903.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7, 1–30.
- Devaraj, J., Madurai Elavarasan, R., Pugazhendhi, R., Shafiullah, G. M., Ganesan, S., Jeysree, A. K., Khan, I. A., & Hossain, E. (2021). Forecasting of COVID-19 cases using deep learning models: Is it reliable and practically significant? *Results in Physics*, 21(February 2021), 103817.
<https://doi.org/10.1016/j.rinp.2021.103817>
- Dinas Pendidikan Provinsi Jawa Timur. (2020). *Kalender pendidikan*. Dinas Pendidikan Provinsi Jawa Timur.
- Djalante, R., Lassa, J., Setiamarga, D., Sudjatma, A., Indrawan, M., Haryanto, B., Mahfud, C., Sinapoy, M. S., Djalante, S., Rafliana, I., Gunawan, L. A., Surtiari, G. A. K., & Warsilah, H. (2020). Review and analysis of current responses to COVID-19 in Indonesia: Period of January to March 2020. *Progress in Disaster Science*, 6, 100091.
<https://doi.org/10.1016/j.pdisas.2020.100091>
- Elsheikh, A. H., Saba, A. I., Elaziz, M. A., Lu, S., Shanmugan, S., Muthuramalingam, T., Kumar, R., Mosleh, A. O., Essa, F. A., & Shehabeldeen, T. A. (2021). Deep learning-based forecasting model for COVID-19 outbreak in Saudi Arabia. *Process Safety and Environmental Protection*, 149, 223–233. <https://doi.org/10.1016/j.psep.2020.10.048>
- Epstein, B., Meir, R., & Michaeli, T. (2017). Joint auto-encoders: A flexible

- multi-task learning framework. *ECDM*.
- Epstein, B., Meir, R., & Michaeli, T. (2018). Joint auto-encoders: A flexible multi-task learning framework. *Iclr2018*, 1–13.
- Ernis, D. (2020). *Sengkarut Data Covid-19, Angka Kematian Ternyata Lebih dari 13 Ribu*. Tempo.co. Diambil 2 Agustus 2020, dari <https://majalah.tempo.co/read/nasional/160878/sengkarut-data-covid-19-angka-kematian-ternyata-lebih-dari-13-ribu>
- Gautam, Y. (2021). Transfer Learning for COVID-19 cases and deaths forecast using LSTM network. *ISA Transactions*, xxxx. <https://doi.org/10.1016/j.isatra.2020.12.057>
- Goic, M., Bozanic-Leal, M. S., Badal, M., & Basso, L. J. (2021). COVID-19: Short-term forecast of ICU beds in times of crisis. *PLoS ONE*, 16(1 January), 1–24. <https://doi.org/10.1371/journal.pone.0245272>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Gugus Tugas Percepatan Penanganan COVID-19. (2020). *Peta Sebaran | Gugus Tugas Percepatan Penanganan COVID-19*. Gugus Tugas Percepatan Penanganan COVID-19. Diambil 2 Agustus 2020, dari <https://covid19.go.id/peta-sebaran>
- Hapke, H., & Nelson, C. (2020). *Building Machine Learning Pipelines*. “O’Reilly Media, Inc.”
- Harapan, H., Itoh, N., Yufika, A., Winardi, W., Keam, S., Te, H., Megawati, D., Hayati, Z., Wagner, A. L., & Mudatsir, M. (2020). Coronavirus disease 2019 (COVID-19): A literature review. *Journal of Infection and Public Health*, 13(5), 667–673. <https://doi.org/10.1016/j.jiph.2020.03.019>
- Hasell, J., Mathieu, E., Beltekian, D., Macdonald, B., Giattino, C., Ortiz-Ospina, E., Roser, M., & Ritchie, H. (2020). A cross-country database of COVID-19 testing. *Scientific Data*, 7(1), 1–7. <https://doi.org/10.1038/s41597-020-00688-8>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction* (2 ed.). Springer Science & Business Media. <https://web.stanford.edu/~hastie/ElemStatLearn/>

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition Kaiming. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778. <https://doi.org/10.1002/chin.200650130>
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice* (2 ed.). OTexts. <https://otexts.com/fpp2/>
- Hyndman, R. J., & Koehler, A. B. (2006). Another Look at Measures of Forecast Accuracy Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4), 679–688. <https://doi.org/10.1016/j.ijforecast.2006.03.001>
- Jason, Roslynlia, & Gunawan, A. A. S. (2021). Forecasting Social Distancing impact on COVID-19 in Jakarta using SIRD Model. *Procedia Computer Science*, 179(2019), 662–669. <https://doi.org/10.1016/j.procs.2021.01.053>
- Jhaveri, R. (2021). The COVID-19 mRNA Vaccines and the Pandemic: Do They Represent the Beginning of the End or the End of the Beginning? *Clinical Therapeutics*, xxx(xxx), 1–8. <https://doi.org/10.1016/j.clinthera.2021.01.014>
- Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical Systems and Signal Processing*, 151, 107398. <https://doi.org/10.1016/j.ymssp.2020.107398>
- Kodinariya, T. M., & Makwana, P. R. (2013). Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1(6), 2321–7782.
- Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., & Reblitz-richardson, O. (2009). Captum: A unified and generic model interpretability library for PyTorch An Overview of the Algorithms. *arXiv*, 1–11.
- Komite Penanganan COVID-19 dan Pemulihan Ekonomi Nasional. (2021). *Peta Sebaran COVID-19 | Covid19.go.id*. Covid19.go.id. Diambil 16 April 2021, dari <https://covid19.go.id/peta-sebaran-covid19>
- Lai, G., Yang, Y., Chang, W. C., & Liu, H. (2017). Modeling long- and short-term

- temporal patterns with deep neural networks. *arXiv*, 95–104.
- Lau, H., Khosrawipour, V., Kocbach, P., Mikolajczyk, A., Schubert, J., Bania, J., & Khosrawipour, T. (2020). The positive impact of lockdown in Wuhan on containing the COVID-19 outbreak in China. *Journal of travel medicine*, 27(3), 1–7. <https://doi.org/10.1093/jtm/taaa037>
- Li, Y., Jia, W., Wang, J., Guo, J., Liu, Q., Li, X., Xie, G., & Wang, F. (2021). ALeRT-COVID: Attentive Lockdown-awaRe Transfer Learning for Predicting COVID-19 Pandemics in Different Countries. *Journal of Healthcare Informatics Research*, 98–113. <https://doi.org/10.1007/s41666-020-00088-y>
- Mahase, E. (2020). Covid-19: How does local lockdown work, and is it effective? *BMJ (Clinical research ed.)*, 370(July), m2679. <https://doi.org/10.1136/bmj.m2679>
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M5 Accuracy Competition: Results, Findings and Conclusions. *Research Gate, October*, 1–44. <https://www.researchgate.net/publication/344487258>
- Mankiw, N. G. (2009). *Macroeconomics* (3 ed.). Worth Publishers. <http://imas.staff.gunadarma.ac.id/Downloads/files/63058/Ebook+Macro+Economi+N.+Gregory+Mankiw.pdf>
- Mathieu, E., Ritchie, H., Ortiz-Ospina, E., Roser, M., Hasell, J., Appel, C., Giattino, C., & Rod s-Guirao, L. (2021). A global database of COVID-19 vaccinations. *Nature Human Behaviour*, 5(7), 947–953. <https://doi.org/10.1038/s41562-021-01122-8>
- Melnick, E. R., & Ioannidis, J. P. A. (2020). Should governments continue lockdown to slow the spread of covid-19? *BMJ (Clinical research ed.)*, 369(June), m1924. <https://doi.org/10.1136/bmj.m1924>
- Mukaromah, V. F. (2020). *Lewati Singapura, Kasus Covid-19 di Indonesia Kini Terbanyak di ASEAN*. Kompas.com. Diambil 2 Agustus 2020, dari <https://www.kompas.com/tren/read/2020/06/17/175103265/lewati-singapura-kasus-covid-19-di-indonesia-kini-terbanyak-di-asean?page=all>
- Nadler, P., Arcucci, R., & Guo, Y. (2020). A Neural SIR Model for Global

- Forecasting. *Proceedings of Machine Learning Research*, 136, 254–266.
- Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *ICLR 2020*, 1–31.
- Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359. <https://doi.org/10.1109/TKDE.2009.191>
- Paparrizos, J., & Gravano, L. (2015). K-shape: Efficient and accurate clustering of time series. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2015-May, 1855–1870. <https://doi.org/10.1145/2723372.2737793>
- Park, M., Cook, A. R., Lim, J. T., Sun, Y., & Dickens, B. L. (2020). A Systematic Review of COVID-19 Epidemiology Based on Current Evidence. *Journal of Clinical Medicine*, 9(4), 967. <https://doi.org/10.3390/jcm9040967>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *33rd Conference on Neural Information Processing Systems*.
- Pemerintah Provinsi Jawa Timur. (2021). *JATIM TANGGAP COVID-19*. JATIM TANGGAP COVID-19. Diambil 16 April 2021, dari <http://infocovid19.jatimprov.go.id/>
- Peter, D., & Silvia, P. (2012). ARIMA vs. ARIMAX – which approach is better to analyze and forecast macroeconomic time series? *International Conference Mathematical Methods in Economics*, 2, 136–140.
- Petitjean, F., Ketterlin, A., & Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3), 678–693. <https://doi.org/10.1016/j.patcog.2010.09.013>
- Petropoulos, F., & Makridakis, S. (2020). Forecasting the novel coronavirus COVID-19. *PLoS ONE*, 15(3), 1–8.

- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & De Freitas, N. (2016). Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175. <https://doi.org/10.1109/JPROC.2015.2494218>
- Singhal, T. (2020). A Review of Coronavirus Disease-2019 (COVID-19). 87(April), 281–286.
- Sirekbasan, S. (2021). The biggest challenge in the fight against the COVID-19 pandemic: Vaccine hesitancy. *Ukrainian Journal of Nephrology and Dialysis*, 1(69), 3–5. [https://doi.org/10.31450/ukrjnd.1\(69\).2021.01](https://doi.org/10.31450/ukrjnd.1(69).2021.01)
- Sohrabi, C., Alsafi, Z., O'Neill, N., Khan, M., Kerwan, A., Al-Jabir, A., Iosifidis, C., & Agha, R. (2020). World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). *International Journal of Surgery*, 76(February), 71–76. <https://doi.org/10.1016/j.ijssu.2020.02.034>
- Soures, N., Chambers, D., Carmichael, Z., Daram, A., Shah, D. P., Clark, K., Potter, L., & Kudithipudi, D. (2020). SIRNET: Understanding social distancing measures with hybrid neural network model for covid-19 infectious spread. *arXiv*.
- Tavenard, R., Vandewiele, G., Divo, F., Androz, G., Holtz, C., Payne, M., & Woods, E. (2020). Tslern, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research*, 21, 1–6.
- Teguh, R., Sigit, A., & Adji, F. F. (2020). Pemodelan penyebaran Infeksi COVID-19 di Kalimantan, 2020. *Research Gate*. https://www.researchgate.net/publication/341709835_Pemodelan_penyebaran_Infeksi_COVID-19_di_Kalimantan_2020
- TensorFlow. (2021). *TensorBoard | TensorFlow*. TensorFlow. Diambil 19 April 2021, dari <https://www.tensorflow.org/tensorboard>
- VanderPlas, J. (2016a). *A Whirlwind Tour of Python*. O'Reilly Media, Inc.
- VanderPlas, J. (2016b). *Python data science handbook: Essential tools for working with data* (1 ed.). O'Reilly Media, Inc. <https://books.google.co.id/books?id=6omNDQAAQBAJ&printsec=frontcove>

- Viner, R. M., Russell, S. J., Croker, H., Packer, J., Ward, J., Stansfield, C., Mytton, O., Bonell, C., & Booy, R. (2020). School closure and management practices during coronavirus outbreaks including COVID-19: a rapid systematic review. *The Lancet Child and Adolescent Health*, 4(5), 397–404. [https://doi.org/10.1016/S2352-4642\(20\)30095-X](https://doi.org/10.1016/S2352-4642(20)30095-X)
- VOA News. (2021). *Several European Countries Under New COVID Lockdown Restrictions*. VOA News. Diambil 5 April 2021, dari <https://www.voanews.com/covid-19-pandemic/several-european-countries-under-new-covid-lockdown-restrictions>
- Wang, P., Zheng, X., Ai, G., Liu, D., & Zhu, B. (2020). Time series prediction for the epidemic trends of COVID-19 using the improved LSTM deep learning method: Case studies in Russia, Peru and Iran. *Chaos, Solitons and Fractals*, 140.
- Weiss, K., Khoshgoftaar, T. M., & Wang, D. D. (2016). A survey of transfer learning. In *Journal of Big Data* (Vol. 3, Nomor 1). Springer International Publishing. <https://doi.org/10.1186/s40537-016-0043-6>
- World Health Organization. (2020). *Archived: WHO Timeline - COVID-19*. World Health Organization Newsroom. Diambil 2 Agustus 2020, dari <https://www.who.int/news-room/detail/27-04-2020-who-timeline---covid-19>
- Wouters, O. J., Shadlen, K. C., Salcher-Konrad, M., Pollard, A. J., Larson, H. J., Teerawattananon, Y., & Jit, M. (2021). Challenges in ensuring global access to COVID-19 vaccines: production, affordability, allocation, and deployment. *The Lancet*, 397(10278), 1023–1034. [https://doi.org/10.1016/S0140-6736\(21\)00306-8](https://doi.org/10.1016/S0140-6736(21)00306-8)
- Wu, Y. C., Chen, C. S., & Chan, Y. J. (2020). The outbreak of COVID-19: An overview. *Journal of the Chinese Medical Association*, 83(3), 217–220. <https://doi.org/10.1097/JCMA.0000000000000270>
- Yamak, P. T., Yujian, L., & Gadosey, P. K. (2019). A comparison between ARIMA, LSTM, and GRU for time series forecasting. *ACM International Conference Proceeding Series*, 49–55.

<https://doi.org/10.1145/3377713.3377722>

- Yechezkel, M., Weiss, A., Rejwan, I., Shahmoon, E., Ben-Gal, S., & Yamin, D. (2021). Human mobility and poverty as key drivers of COVID-19 transmission and control. *BMC Public Health*, 1–13. <https://doi.org/10.1101/2020.06.04.20112417>
- Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 4(January), 3320–3328.
- Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, 31, 1235–1270. <https://doi.org/10.1162/NECO>
- Yulida, Y., & Karim, M. A. (2020). Pemodelan Matematika Penyebaran COVID-19 di Provinsi Kalimantan Selatan. *Media Bina Ilmiah*, 14(10), 3257–3264. <https://ejurnal.binawakya.or.id/index.php/MBI/article/view/572/pdf>
- Zhang, Z., Huang, X., Huang, Q., Zhang, X., & Li, Y. (2019). Joint learning of neural networks via iterative reweighted least squares. *CVPR Workshops*, 1–9.