

BAB II

LANDASAN TEORI

2.1 Teori yang digunakan

2.1.1 Covid-19

Covid-19 atau dapat dikenal sebagai virus corona ini merupakan penyakit menular yang disebabkan oleh sindrom pernafasan akut coronavirus2 (*Severe Acute Respiratory Syndrome Corona Virus 2* atau SARS-coV-2) [14]. Corona Virus ini menyebabkan penyakit infeksi saluran pernafasan seperti flu, MERS (*Middle East Respiratory Syndrome*) dan SARS (*Severe Acute Respiratory Syndrome*). China merupakan negara pertama yang terdeteksi adanya virus tersebut di kota Wuhan, ibukota provinsi Hubei, China pada bulan desember 2019. Virus covid-19 ini sendiri mulai menyebar luas ke berbagai negara didunia dan menyebar secara cepat termasuk ke Indonesia, dengan adanya laporan 2 orang yang terkonfirmasi positif pada tanggal 2 Maret 2020. *World health organizaion (WHO)* sendiri yang merupakan organisasi Kesehatan dunia dibawah naungan Perserikatan Bangsa-Bangsa (PBB) menetapkan wabah penyakit yang disebabkan oleh covid-19 ini sebagai pandemi global[15].

2.1.2 Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM)

Pemberlakuan pembatasan kegiatan masyarakat (PPKM) awalnya dinamakan sebagai pembatasan sosial berskala besar (PSBB). Pemerintah Indonesia sendiri khususnya Jakarta mulai menetapkan atau menerapkan PSBB di Jakarta pada tanggal 10 April - 23 April 2020. Pembatasan sosial berskala besar ini dilakukan dengan cara sebagai berikut yaitu membuat sekolah dan tempat kerja libur, adanya pembatasan dalam kegiatan keagamaan, pembatasan dalam tempat umum, kegiatan sosial budaya, dan moda transportasi. PSBB yang awalnya hanya ditetapkan selama 14 hari maka diperpanjang terus hingga 11 januari 2021, ini dilakukan karena dalam proses pemberlakuan PSBB sendiri tidak membuahkan hasil yang cukup baik yaitu terjadinya penurunan dalam penambahan kasus positif covid-19. PSBB di Jakarta sendiri pada periode 11-25 Januari diatur dalam Keputusan Gubernur DKI Nomor 19 Tahun 2021

tentang Pemberlakuan, Jangka Waktu dan Pembatasan Aktivitas Luar Rumah PSBB, dikarenakan periode PSBB Jakarta ini berjalan seiringan dengan pemberlakuan pembatasan kegiatan masyarakat (PPKM) Jawa-Bali, yaitu pada tanggal 11-25 Januari 2021, maka PSBB yang dilakukan oleh Pemprov DKI Jakarta akan mengikuti kebijakan PPKM pusat. Dalam proses perjalanan PPKM sendiri terdapat beberapa istilah yaitu dimulai dari PPKM Jawa-Bali, PPKM Mikro, PPKM Darurat, dan PPKM Level 1-4[16].

2.1.3 Citra VIIRS

Visible Infrared Imaging Radiometer Suites (VIIRS) DNB free cloud composites adalah merupakan salah satu dari data penginderaan jauh yang bisa memberikan informasi seperti polusi cahaya dan jumlah tutupan awan[17]. *National Oceanography Atmospheric Administration (NOAA)* dan *United Nation Aeronautics and Space Administration (NASA)* bekerja sama dalam melakukan peluncuran *Suomi National Polar Satellite* yang merupakan satelit yang membawa program berupa VIIRS. Data citra VIIRS ini sendiri memiliki cakupan wilayah perekaman sebesar 3000km dengan memiliki spektrum panjang gelombang yang digunakan yaitu berada diantara a 0.5 – 0.9 μm , dan juga ia memiliki resolusi radiometrik mencapai 14 bit dan juga dapat mendeteksi limit batas cahaya mencapai $-2\text{E Watt/cm}^2 \text{ *sr}$. VIIRS sendiri memiliki kapabilitas dalam mendeteksi api dan cahaya di malam hari menggunakan saluran VNIR.

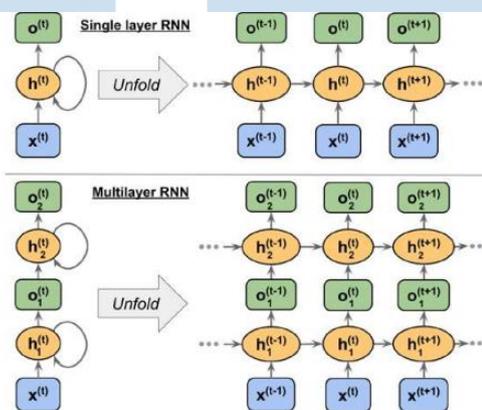
2.1.4 Google Earth Imagery

Google earth imagery adalah sebuah teknologi yang dikembangkan oleh perusahaan Google yang berupa sebuah *virtual globe*, peta dan program informasi sebuah geografis[18]. Google earth sendiri menampilkan berbagai macam gambar satelit permukaan bumi yang memiliki resolusi yang sangat bervariasi sehingga dapat mempermudah pengguna untuk memperoleh berbagai informasi secara visual, seperti kota, rumah, jalan, sungai, *night-light* dan lain sebagainya. Data yang didapatkan pada Google Earth ini sendiri berasal dari berbagai macam variasi satelit perusahaan yang mengorbit di dekat planet bumi.

2.2 Framework yang digunakan

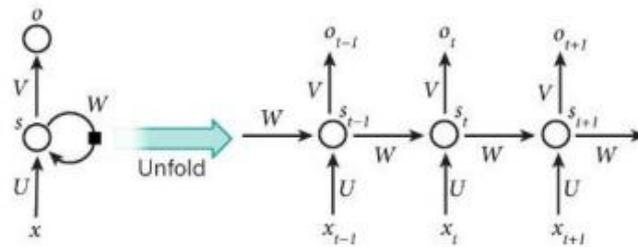
2.2.1 Recurrent Neural Network

Recurrent Neural Network (RNN) adalah salah satu jenis arsitektur jaringan saraf berulang yang mempunyai desain khusus agar dapat mengatasi permasalahan dengan data berurutan (*sequence data*)[19]. RNN sendiri pertama kali dikembangkan oleh Jeff Elman pada tahun 1990, yang merupakan juga variasi dari *Artificial Neural Network*.



Gambar 2. 1 Arsitektur RNN
Sumber: [20]

Perbedaan yang paling utama yang terdapat dalam model ini adalah sinyal dapat mengalir secara *forward* dan *backward* secara berulang seperti pada gambar 2.1. Selain melewati *input* antar layer, *output* yang dihasilkan dari setiap *layer* juga akan menuju ke *context layer* agar dapat digunakan untuk sebagai inputan pada *timestep* berikutnya. RNN sendiri menyimpan informasi di *context layer*, yang itu membuatnya dapat untuk mempelajari urutan data dan juga menghasilkan *output* atau urutan lain. RNN sendiri dapat dikatakan bahwa ia memiliki memori yang berisikan hasil rekaman informasi yang dihasilkan sebelumnya[21].



Gambar 2. 2 Proses RNN saat Menghitung proses didepannya
 Sumber: [21]

Gambar 2.1 merupakan RNN pada posisi yang tidak dibuka ke *full network*. Dengan membuka gulungan RNN maka artinya kita menuliskan seluruh jaringan dengan urutan (*sequence*) secara lengkap, berikut merupakan keterangan dari pada gambar 2.1[21]:

- a. X_t adalah input pada setiap time step.
- b. S_t adalah *hidden state* pada setiap *time step* t .
- c. O_t adalah output untuk setiap step t .

Dikarenakan RNN merupakan salah satu dari jenis *neural network*, dalam neural network sendiri terdapat beberapa jenis yaitu *artificial neural network* (ANN) dan juga *convolutional neural network* (CNN). ANN sendiri adalah sebuah model algoritma yang termotivasi berdasarkan struktur atau fungsi dari jaringan neuron pada manusia[22]. Sedangkan CNN sendiri adalah operasi konvolusi yang menggabungkan beberapa lapisan pemrosesan, menggunakan beberapa elemen yang beroperasi secara parallel dan juga ini terinspirasi dari jaringan saraf manusia[23], dan pada tabel 2.1 dibawah ini merupakan perbedaan utama antara algoritma RNN, ANN dan juga CNN[24]:

Tabel 2. 1 Perbedaan Algoritma ANN, RNN & CNN

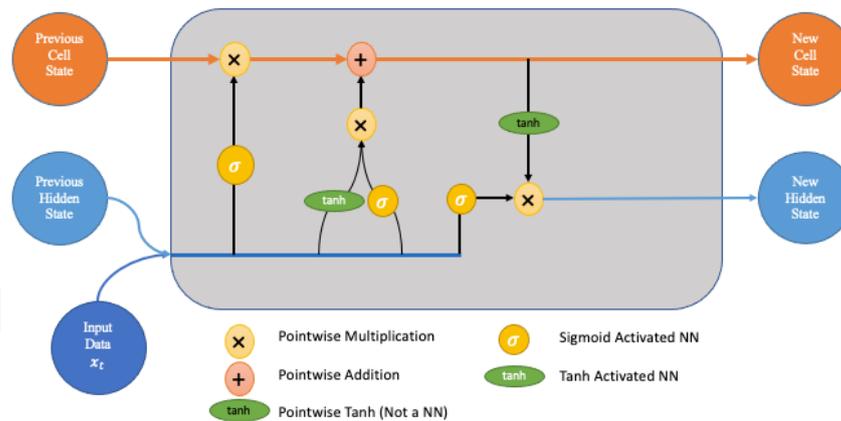
	ANN	RNN	CNN
Architecture	<i>Feedforward</i>	<i>Recurrent</i>	<i>Feedforward with shared weight</i>
Input	<i>Tabular Data, Text Data</i>	<i>Sequence Data</i>	Gambar
Keuntungan Utama	Mempunyai toleransi kesalahan, dapa	Dapat mengingat setiap	Mempunyai akurasi yang tinggi

	bekerja pada <i>incomplete knowledge</i>	informasi,, prediksi <i>time series data</i>	pada <i>image recognition</i>
Parameters	<i>Fixed Number</i>	<i>Variable Length</i>	<i>Fixed Number</i>
Kompleksitas	Relatif Simple	Sedang	Relatif Kompleks
Kelemahan	Tergantung Hardware yang digunakan, perilaku jaringan tidak dapat dijelaskan	Dapat terjadi penyapuan Gradien & Peledakan Gradien	Membutuhkan data train yang besar, tidak <i>encode</i> posisi dan orientasi dari objeknya

2.2.2 Long-Short Term Memory

Long-Short Term Memory (LSTM) adalah turunan dari algoritma Recurrent Neural Network (RNN) yang terbukti stabil dan juga *powerful* untuk memprediksi atau *modelling* data *long-range dependencies* untuk *sequence modeling*[25]. LSTM sendiri pertama kali diperkenalkan oleh Hochreiter & Schmidhuber pada tahun 1997[9]. LSTM sendiri memiliki desain arsitektur yang mirip dengan algoritma *Recurrent Neural Network*, akan tetapi jika RNN memiliki kekurangan yaitu tidak bisa untuk belajar dalam menghubungkan informasi yang ada karena memori lama yang tersimpan akan tidak berguna dengan seiringnya waktu yang berjalan akibat menimbunnya memori yang baru terhadap memori yang lama. Kekurangan yang dimiliki RNN itu sendiri tidak berlaku di LSTM karena LSTM dapat mengatur memori pada setiap *input* dengan menggunakan *memory cells & gate units* yang ada. LSTM sendiri memiliki *memory cell* dan *gate units* pada setiap *neurons* yang ada dan berfungsi untuk mengatur memori dalam setiap *neurons*nya[26]. Gambar 2.3 dibawah ini merupakan alur kerja dalam setiap *memory cell* pada setiap *neurons* LSTM yang ada.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 2. 3 Alur Kerja Dalam Memory Cell LSTM
 Sumber: [27]

Terdapat 4 proses dalam fungsi aktivasi pada setiap *input* pada *neurons* yang disebut sebagai *gates units*. *Gates units* sendiri yaitu *forget gates*, *input gates*, *cell gates*, dan *output gates*. *Forget gates*, terdapat informasi pada setiap data yang di *input* akan diolah dan dipilih data mana saja yang akan disimpan atau dibuang pada *memory cells*. Fungsi aktivasi yang digunakan dalam *forget gates* ini adalah fungsi aktivasi sigmoid. Dimana hasil *output* dari fungsi tersebut adalah antara 0 dan 1. Jika hasil *output* adalah 1 maka semua data akan disimpan dan begitu juga sebaliknya, jika hasil *output*-nya adalah 0 maka semua data akan dibuang. Berikut rumus dari *forget gates*[26]:

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

Dalam *input gates* sendiri terdapat dua *gates* yang akan dilaksanakan yaitu pertama akan diputuskan manakah nilai yang akan diperbarui menggunakan fungsi aktivasi sigmoid, kemudian selanjutnya fungsi tanh akan membuat vector nilai yang baru yang akan disimpan pada *memory cell*. Berikut rumus dari *input gates*[26]:

$$f_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\check{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

Dalam *cell gates* sendiri akan mengganti nilai pada *memory cells* yang sebelumnya dengan nilai yang baru. Dimana nilai baru yang didapatkan dengan

cara menggabungkan nilai yang terdapat pada *forget gate* dan *input gate*. Berikut merupakan rumus dari *cell gates*[26]:

$$c_t = f_t * c_{t-1} + i_t * \check{c}_t \quad (4)$$

Dalam *output gates* juga terdapat dua *gates* yang akan dilaksanakan, pertama akan diputuskan mana nilai pada bagian *memory cell* mana yang akan dikeluarkan dengan menggunakan fungsi aktivasi sigmoid. Setelah diputuskan maka akan ditempatkan nilai pada *memory cell* dengan menggunakan fungsi aktivasi tanh. Kedua *gates* tersebut dikalikan sehingga menghasilkan nilai baru yang akan dikeluarkan. Adapun rumus dari *output gates*, sebagai berikut[26]:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t \tanh(c_t) \quad (6)$$

Berikut adalah informasi tentang notasi rumus diatas:

- $f_t = \text{Forget gate}$
- $i_t = \text{Input gate}$
- $\check{c}_t = \text{Candidate}$
- $c_t = \text{Cell gate}$
- $o_t = \text{Output gate}$
- $b = \text{Bias}$
- $h = \text{Hidden State sebelumnya}$
- $\sigma = \text{Sigmoid}$
- $W = \text{Weight matrix}$

2.2.3 Time Series Analysis

Menurut Robinson, *Time series* adalah urutan titik data untuk suatu variable yang biasanya diukur pada waktu yang secara berturut-turut pada interval waktu yang seragam[28]. *Time series analysis* ini dilakukan biasanya untuk memperoleh pola dari data tersebut dengan menggunakan *historical* data yang fungsinya yaitu untuk dapat digunakan untuk memprediksi suatu nilai dimasa yang akan datang[29]. Untuk melakukan analisis *time series* ini sendiri perlu diperiksa terlebih dahulu apakah data atau nilai yang diinput sudah stasioner atau tidak stasioner.

Dalam hal mengetahui data tersebut stasioner atau tidak stasioner maka dapat diperiksa dengan analisis *autocorrelation function (ACF)* dan *Partial Autocorrelation Function (PACF)*. Untuk mencari koefisien korelasi sederhana y_t dengan y_{t-1} dapat dicari dengan rumus berikut[29]:

$$r_k = \frac{\sum_t^n (Y_t - \bar{y})(Y_{t-1} - \bar{y})}{\sum_t^n (Y_{t-1} - \bar{y})^2} \quad (7)$$

Dimana:

- r_k adalah koefisien autokorelasi *lag* ke k
- Dimana $k = 0, 1, 2, \dots, k$
- n adalah jumlah data yang digunakan
- Y_t adalah nilai data dari *time series* yang sudah stasioner
- \bar{y} adalah rata-rata peubah acak Y

Sedangkan untuk mencari autokorelasi parsial adalah sebagai berikut untuk rumus autokorelasi partials[29]:

$$\rho_k = \phi_1 \rho_{k-1} + \phi_2 \rho_{k-2} + \dots + \phi_p \rho_{k-p} \quad (8)$$

Dimana:

- ρ = nilai dari fungsi ACF
- ϕ = nilai dari fungsi PCAF
- k = *time lag*, dengan $k = 1, 2, \dots, \rho$.

2.2.4 RMSE (*Root Mean Square Error*)

Root Mean Square Error atau biasa disebut dengan RMSE adalah sebuah cara pengukuran yang cukup sering digunakan dari perbedaan antara nilai-nilai yang diprediksi sebuah model atau estimator dan nilai tersebut benar-benar diamati. Untuk menilai apakah sebuah model memiliki hasil prediksi atau prakiraan yang baik maka nilai RMSE semakin rendah atau mendekati angka 0, sebaliknya jika angka pada nilai RMSE semakin tinggi maka nilai prediksi atau prakiraan semakin tidak baik. Perbedaan-perbedaan individual dapat disebut juga residual jika perhitungan yang dilakukan untuk sampel data yang

digunakan untuk estimasi, dan disebut juga kesalahan dama melakukan prediksi jika dihitung *out of sample*. Persamaan dari *root mean square error* itu sendiri adalah[30]:

$$\text{RMSE} = \sqrt{\frac{\sum(y_t - \hat{y}_t)^2}{n}} \quad (9)$$

Dimana:

RMSE = *Root Mean Square Error*

N = Jumlah Sampel

y_t = Nilai Aktual Indeks

\hat{y}_t = Nilai Prediksi Indeks

2.3 Tools yang digunakan

2.3.1 Python

Python merupakan bahasa pemrograman yang masuk dalam bahasa pemrograman tingkat tinggi dan juga bahasa ini sangat populer saat ini. Python sendiri saat ini bisa dibilang sangat populer dikarenakan terdapat kenaikan kebutuhan di bidang *data science*, *machine learning*, *artificial intelligent*, dan semua kebutuhan di bidang *big data*[31]. Python sendiri dikembangkan oleh Guido Van Rossum pada tahun 1989. versi pertama dari python sendiri yaitu versi 0.9.0 di tahun 1991[32]. Pada awal rilis, bahasa pemrograman python sendiri sudah memiliki fitur seperti beberapa tipe data dan fungsi untuk menangani kesalahan, kemudian berkembang kembali pada tahun 1994 dengan versi Python 1.0 dan hingga sekarang python sudah berkembang hingga versi python 3.11 yang rilis di tahun 2022[25-26].

Python sendiri memiliki library nya sendiri, yang dimaksud library yaitu kumpulan modul-modul yang berisikan kode yang dapat digunakan secara berulang kali dalam program yang berbeda. Macam-macam library python yaitu[34]:

- TensorFlow
- Numpy

- Scipy
- Pandal
- Matplotlib
- Keras
- Scikit-Learn

2.3.2 Anaconda

Anaconda merupakan *software* atau sebuah platform *data science* yang biasanya digunakan melakukan pemrograman *data science* menggunakan python ataupun r. Anaconda sendiri merupakan software yang di bangun oleh perusahaan bernama Anaconda Inc. yang didirikan pada tahun 2012. Pada tahun 2012-2014, Anaconda Inc. membuat Anaconda dan Conda atau “Anaconda Server” yang pertama. Perkembangan perusahaan Anaconda sendiri sudah dapat dibidang cukup sukses dikarenakan sekarang software anaconda sendiri sudah mempunyai pengguna sebanyak 25 juta pengguna di seluruh dunia[35].

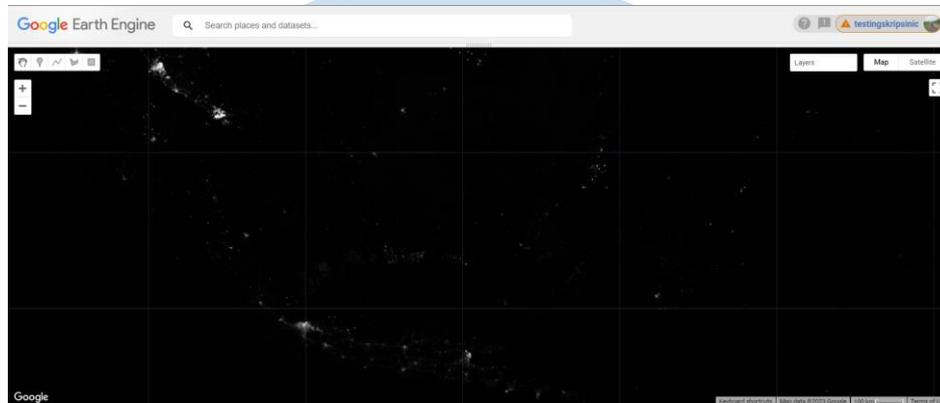
2.3.3 Tableau



Gambar 2. 4 Logo Tableau
Sumber: Google Images (2023)

Tableau adalah salah satu dari banyaknya *tools* dalam visualisasi data. Tableau sendiri mampu mengolah data dalam jumlah yang sangat besar. Tableau pertama kali diluncurkan pada tahun 2003, ia dapat menghasilkan visualisasi yang canggih dengan hanya *drag and drop* data yang diperlukan untuk visualisasi. Tableau sendiri dilengkapi dengan berbagai jenis visualisasi mulai dari *bar chart*, *plots*, *heatmap*, *maps* dan masih banyak lainnya[36].

2.3.4 VIIRS Stray Light Corrected Nighttime Day/Night Band Composites Version 1



Gambar 2. 5 Data Gambar VIIRS melalui Google Earth Engine
Sumber: Google Earth Engine

Gambar 2.5 diatas merupakan citra VIIRS (*Visible Infrared Imaging Radiometer*) yang di ambil dari platform Google Earth Engine. VIIRS sendiri adalah salah satu instrument yang dibawa oleh satelit NPP (*National Polar-Orbiting Partnership*) yang diluncurkan pada tanggal 28 Oktober 2011, akan tetapi satelit bumi telah melakukan pengamatan pada malam hari sudah lebih dari 50 tahun, tetapi untuk arsip *Night-Time Light* didirikan pada tahun 1992 oleh *National Oceanic and Atmospheric Administration (NOAA)*[37]. *Nighttime-Light* sendiri adalah salah satu produk data satelit yang paling dikenal oleh banyak orang dan memberikan gambaran visual kehadiran manusia di seluruh permukaan bumi[38]. Gambar malam hari yang didapat melalui sensor *The Defense Meteorological Satellite Program (DMSP) – Operational Linescan System (OLS)* dari *band Visible and Near-infrared (VNIR)* akan membantu para ilmuwan mengamati dan mendeteksi jika adanya sumber polusi cahaya, misalnya lampu kota, gas dan juga kebakaran. Setelah DMSP-OLS, muncullah sensor yang terbaru yaitu sensor *Visible Infrared Imaging Radiometer Suite (VIIRS)* yang ini menjadi sumber data yang baru yang menghasilkan hasil yang lebih baik dalam menghasilkan data NTL[39]. Dalam beberapa tahun terakhir, sensor VIIRS yang dilengkapi dengan *Day/Night Band (DNB)*, telah lebih baik dari DMSP-OLS dalam banyak hal, seperti rentang dinamis yang lebih besar, resolusi spasial yang lebih baik, dan batas deteksi

yang rendah. Data yang digunakan dalam penelitian ini merupakan data yang diambil melalui Google Earth Engine dengan menggunakan dataset VIIRS yaitu VIIRS Stray Light Corrected Nighttime Day/Night Band Composites Version 1. Dataset ini tersedia mulai dari bulan januari 2014 hingga bulan januari 2023. Gambar 2.6 dibawah ini merupakan contoh code yang dapat digunakan di Google Earth Engine code editor untuk melihat preview data secara langsung.

```
var dataset = ee.ImageCollection('NOAA/VIIRS/DNB/MONTHLY_V1/VCMSLCFG')
    .filter(ee.Filter.date('2017-05-01', '2017-05-31'));
var nighttime = dataset.select('avg_rad');
var nighttimeVis = {min: 0.0, max: 60.0};
Map.setCenter(-77.1056, 38.8904, 8);
Map.addLayer(nighttime, nighttimeVis, 'Nighttime');
```

Gambar 2. 6 Code default Google Earth Engine untuk VIIRS Stray Light Corrected Nighttime Day/Night Band Composites Version 1
Sumber: Google Earth Engine

2.3.5 Google Earth Engine

Google Earth Engine adalah sebuah platform yang dimiliki oleh Google yang dimana platform ini merupakan *computing platform* yang dapat memungkinkan pengguna untuk melakukan *geospatial* analisis dalam infrastruktur yang dimiliki oleh Google. Terdapat beberapa cara untuk menggunakan platform tersebut, yaitu terdapat *code editor* secara langsung untuk berbasis website untuk menulis dan menjalankan *script*. Pengguna dapat juga menggunakan menu *Explorer* untuk melihat katalog data yang dimiliki dan menjalankan analisis yang simpel. Kemudian, terdapat juga *library* yang disediakan oleh Google yang dapat dijalankan di bahasa pemrograman seperti python dan JavaScript *wrappers* dengan menggunakan web API (Application Programming Interface) yang sudah disediakan oleh Google sendiri[40].

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.4 Penelitian terdahulu

Berikut ini merupakan beberapa penelitian terdahulu yang berkaitan dengan topik Analisis Deforestasi menggunakan Algoritma Machine Learning yang dirangkum dalam tabel:

Tabel 2. 2 Penelitian Terdahulu

No.	Nama Peneliti	Jurnal, Volume, Tahun	Judul Artikel	Hasil Penelitian
1	Samuel Ady Sanjaya, Melissa Indah Fianty	Jurnal Teknologi Terapan, Volume 7 No. 2, Tahun 2023	<i>Time-based Geospatial Analysis of Night-Time Light Data and Citizen Movement Restriction During Covid-19 Period</i> [12]	Terdapat penurunan jumlah intensitas cahaya ketika masa pandemi covid-19 masuk ke dalam Indonesia sebesar 1%-16%. Penurunan ini juga didukung dengan adanya penurunan ekstrim mobilitas masyarakat di Indonesia dengan data yang didapatkan dari Google mencapai -70% dari kondisi awal sebelum pandemi. Akan tetapi, level pembatasan kegiatan masyarakat (PPKM) tidak berpengaruh

No.	Nama Peneliti	Jurnal, Volume, Tahun	Judul Artikel	Hasil Penelitian
				<p>begitu besar terhadap perubahan intensitas cahaya, kemudian, terdapat juga beberapa hal yang dipertimbangkan yaitu beberapa faktor dari alam juga mempengaruhi intensitas cahaya di malam hari seperti awan, hujan, kabut hingga banjir. Ini yang menyebabkan tangkapan intensitas cahaya oleh kamera satelit menjadi terdistorsi.</p>
2	Efrike Sofyani Putri, Mujiono Sadikin	Jurnal Ilmiah Teknik Informatik FORMAT, Volume 10 No. 2, Tahun 2021	Prediksi Penjualan Produk Untuk Mengestimasi Kebutuhan Bahan Baku Menggunakan Perbandingan Algoritma LSTM dan ARIMA[13]	<p>Penelitian ini menggunakan LSTM dengan optimizer SGD dan ARIMA. Penggunaan data <i>time series</i> jauh lebih baik menggunakan</p>

No.	Nama Peneliti	Jurnal, Volume, Tahun	Judul Artikel	Hasil Penelitian
				<p>LSTM dibandingkan dengan menggunakan ARIMA karena LSTM mendapatkan nilai akurasi yang cukup tinggi. Hasil menggunakan LSTM yaitu 0,22 untuk nilai RMSE dalam data testing sedangkan ARIMA di 60,21 dan juga Nilai MAPE di metode LSTM yaitu 29,57% dan metode ARIMA di 73%.</p> <p>Dari hasil yang ada penggunaan metode LSTM jauh lebih baik daripada menggunakan metode ARIMA untuk memprediksi penjualan makanan frozen food.</p>

No.	Nama Peneliti	Jurnal, Volume, Tahun	Judul Artikel	Hasil Penelitian
3	Bibhuti Bhusan Sahoo, Anshuman Singh, Ramakar Jha, Deepak Kumar	Acta Geophysica, Juli 2019	<i>Long Short-Term Memory (LSTM) Recurrent Neural Network for Low-Flow Hydrological Time Series Forecasting</i> [41].	<p>Penelitian ini bertujuan untuk mengetahui apakah LSTM-RNN cocok dengan <i>low-flow time series forecasting</i>. Data yang digunakan yaitu data yang didapatkan dari Basantapurgauging station yang terletak di Mahanadi River Basin, India.</p> <p>Penelitian ini sendiri menggunakan 3 metode yaitu LSTM-RNN, RNN dan juga naïve. LSTM sendiri mendapatkan nilai RMSE & MAE yang paling kecil yaitu 0,487 dan 0,361, Sedangkan RNN yaitu 0,561 dan 0,391 untuk nilai RMSE dan</p>

No.	Nama Peneliti	Jurnal, Volume, Tahun	Judul Artikel	Hasil Penelitian
				MAE, dan untuk Naïve yaitu 0,793 dan 0,617 untuk nilai RMSE dan MAEnya. Dari nilai yang ada, maka LSTM-RNN merupakan metode paling cocok dan efektif untuk membuat model secara bulanan untuk <i>low-flow discharge time series</i> dan dapat memberikan hasil yang signifikan untuk performa prediksinya dibandingkan dengan RNN dan <i>benchmark naïve</i> untuk data time series.
4	Muhammad Wildan Putra Aldi, Jondri,	e-Proceeding of	Analisis dan Implementasi Long Short-Term Memory Neural	Model LSTM yang digunakan untuk memprediksi harga bitcoin mendapatkan

No.	Nama Peneliti	Jurnal, Volume, Tahun	Judul Artikel	Hasil Penelitian
	Annisa Aditsania	Engineering, Volume 5, Agustus 2018	Network untuk prediksi Harga Bitcoin[26].	nilai akurasi yang cukup tinggi dengan adanya pengujian beberapa parameter komposisi data, jumlah pola <i>time series</i> , jumlah hidden neuron dan max epoch. Pada penelitian ini didapatkan hasil terbaik dengan komposisi data yaitu data latih 70% dan 30% data uji, parameter 1 pola time series, jumlah <i>hidden neuron</i> yaitu 25 dan <i>max epoch</i> sebesar 100 dengan akurasi rata-rata pada data latih yaitu 93.36% dan data testing 93.5%.
5	K.E. ArunKumar, Dinesh V. Kalaga, Ch.	Chaos, Solitons and Fractals, Volume	<i>Forecasting of Covid-19 using deep layer Recurrent Neural</i>	Penelitian ini menghasilkan report untuk 60 hari prediksi kedepan

No.	Nama Peneliti	Jurnal, Volume, Tahun	Judul Artikel	Hasil Penelitian
	Mohan Sai Kumar, Masahiro Kawaji, Timothy M Brenza	146, May 2021	<i>Networks (RNN) with Gated Recurrent Units (GNU) and Long Short-Term Memory (LSTM) cells</i> [42].	dari covid-19 menggunakan RNN-LSTM dan RNN-GRU model. Penelitian ini menggunakan nilai MSE dan RMSE untuk melihat model mana yang paling bagus untuk melakukan prediksi. Hasilnya yaitu untuk prediksi konfirmasi kasus, LSTM model lebih baik untuk beberapa negara yaitu USA, Brazil, South Africa, Peru, Chile dan Iran sedangkan GRU model lebih baik untuk India, Russia, Mexico, dan UK. Sedangkan untuk <i>recovered case</i> , LSTM model baik untuk negara India, South Africa, Chile,

No.	Nama Peneliti	Jurnal, Volume, Tahun	Judul Artikel	Hasil Penelitian
				UK dan Iran, sedangkan GRU menghasilkan model yang baik untuk USA, Brazil, Russia, Mexico dan Peru. Sedangkan untuk memprediksi <i>fatalities data</i> , LSTM mempunyai model yang baik pada Brazil, India, Russia, South Afriza, Mexico dan Iran. Sedangkan untuk negara USA, Peru, Chile, dan UK model GRU mempunyai hasil yang lebih baik dibandingkan LSTM.

Berdasarkan dari kelima penelitian terdahulu yang sudah diterangkan diatas, maka dapat disimpulkan bahwa LSTM dapat memprediksi data yang berupa time series. Nilai RMSE menjadi nilai yang dilihat untuk melihat apakah model atau algoritma yang digunakan untuk melihat hasil prediksi yang dilakukan oleh beberapa model itu baik atau tidak. Semakin nilai dari RMSE kecil maka hasil

akurasi yang dihasilkan itu baik dan sebaliknya, jika hasil nilai RMSE besar maka nilai hasil akurasi yang didapatkan itu kurang baik. Pada penelitian terdahulu, mereka menggunakan data *time series* yang berarti penggunaan LSTM pada *time series* data merupakan hal yang cocok. Pembeda dengan penelitian sebelumnya yaitu objek penelitiannya, yaitu *Nighttime Light Data*. Dari beberapa penelitian terdahulu yang sudah di masukkan maka, penelitian Nomor 1, 2 dan 4 merupakan penelitian yang diadopsi dalam penelitian ini, dikarenakan pada penelitian nomor 1 itu merupakan penelitian yang selaras dengan penelitian kali ini akan tetapi dalam kebaruannya penelitian ini menggunakan algoritma untuk membuat prediksi kedepannya dan tidak hanya sebatas melakukan *monitoring* saja, kemudian untuk penelitian nomor 2 sendiri diadopsi dikarenakan pada penelitian nomor 2, ia menggunakan SGD sebagai *optimizer* yang digunakan dalam model LSTM-nya sedangkan penelitian nomor 4 sendiri diadopsi karena penelitian ini menggunakan perbandingan untuk penggunaan *data training & testing* sebesar 70% dan 30% sesuai dengan penggunaan dalam penelitian nomor 4.

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA