



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

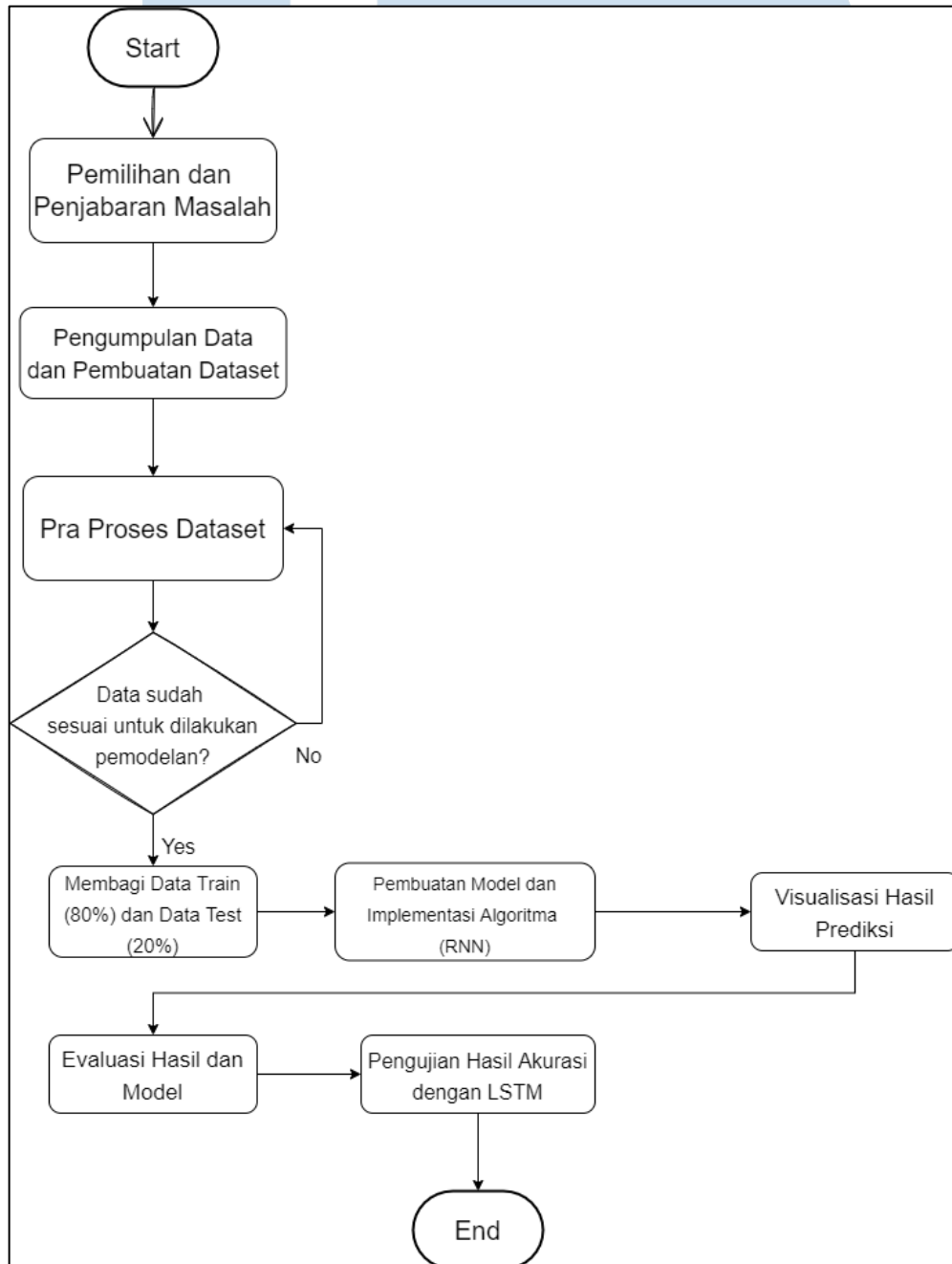
This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB 3

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Pada penelitian yang akan dilakukan, akan digunakan beberapa metode yang dijabarkan pada *flowchart* sebagai berikut.



Gambar 3.1. Flowchart metodologi penelitian

3.1.1 Pemilihan dan Penjabaran Masalah

Permasalahan yang dipilih adalah prediksi jumlah penderita Covid-19 di Kabupaten Tangerang. Permasalahan akan diselesaikan dengan menggunakan metode Recurrent Neural Network, kemudian akan dibuat sebuah model untuk prediksi nilai serta akan dilakukan evaluasi model tersebut dengan pengujian akurasi dan *error rate*.

3.1.2 Pengumpulan Data dan Pembuatan Dataset

Pengumpulan data akan dilakukan hanya di Kabupaten Tangerang. Data diambil per hari dari web <https://covid19.tangerangkab.go.id/sebaran-data> akan memuat variabel pengukur yaitu jumlah kasus konfirmasi. Data yang telah didapatkan akan dikumpulkan dalam bentuk *xlsx* lalu akan dilakukan pra proses dengan menggunakan *Python*.

3.1.3 Pra Proses Dataset

Pada pra proses, data yang telah didapatkan sebelumnya akan dilakukan normalisasi dengan menggunakan *min-max scaler*. Kemudian pada langkah ini harus dipastikan bahwa baris dan kolom yang ada pada dataset tidak boleh kosong, setelah itu akan dibuat sebuah variabel baru untuk menampung variabel yang akan digunakan dalam pembuatan model. Variabel baru tersebut akan berbentuk dalam matriks, setelah semua terpenuhi maka akan dilanjutkan ke pembuatan model.

3.1.4 Pembuatan Model dan Implementasi Algoritma

Kemudian penelitian akan menggunakan algoritma Recurrent Neural Network dalam pembuatan model. Data hasil dari pra proses kemudian akan dibagi menjadi data untuk *training* sebesar 80% dan data untuk *test* sebesar 20%. Pada pengaplikasiannya, akan digunakan layer LSTM untuk membantu RNN dalam pembuatan model prediksi jumlah penderita Covid-19.

3.1.5 Evaluasi Model

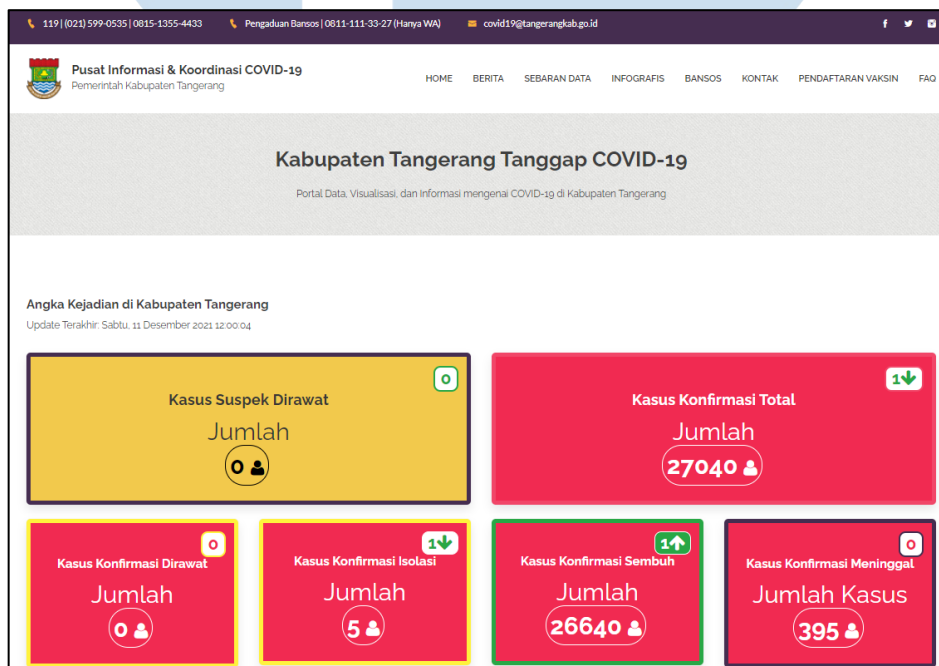
Pada tahap ini model yang sudah jadi akan dilakukan evaluasi dengan *Mean Squared Error* (MSE) dan LSTM. MSE akan digunakan untuk menghitung *error rate* pada model sedangkan LSTM akan digunakan untuk akurasi yang didapat dari model yang dibuat.

3.2 Perancangan Sistem

Pada proses perancangan sistem akan dilakukan pengumpulan serta pembuatan *dataset* ke dalam aplikasi Microsoft Excel untuk dijadikan *file xlsx*. Setelah *dataset* telah terkumpul maka tahap selanjutnya yaitu membuat *flowchart* untuk pra proses dan *flowchart* pemodelan. Kemudian di bawah ini merupakan detail untuk pengumpulan serta pembuatan *dataset* dan pembuatan *flowchart* yang dimaksud.

3.2.1 Pembuatan Dataset

Dataset yang dibuat diambil dari *website* resmi pemerintah Kabupaten Tangerang yang khusus menangani kasus Covid-19 yaitu <https://covid19.tangerangkab.go.id/sebaran-data>. Tampilan *website* dapat dilihat pada Gambar 3.2. Data akan diambil per hari serta akan ditampung di dalam *file xlsx*.



Gambar 3.2. Website covid-19 kabupaten tangerang

Dataset yang telah didapatkan akan ditampung dalam *file xlsx* yang akan berisi variabel yang diambil dari *website* Covid-19 Kabupaten Tangerang. Variabel yang diambil antara lain yaitu konfirmasi kasus yang terkena Covid-19 dan tanggal dari konfirmasi kasus tersebut. Kemudian perancangan *dataset* dapat dilihat pada tabel 3.1.

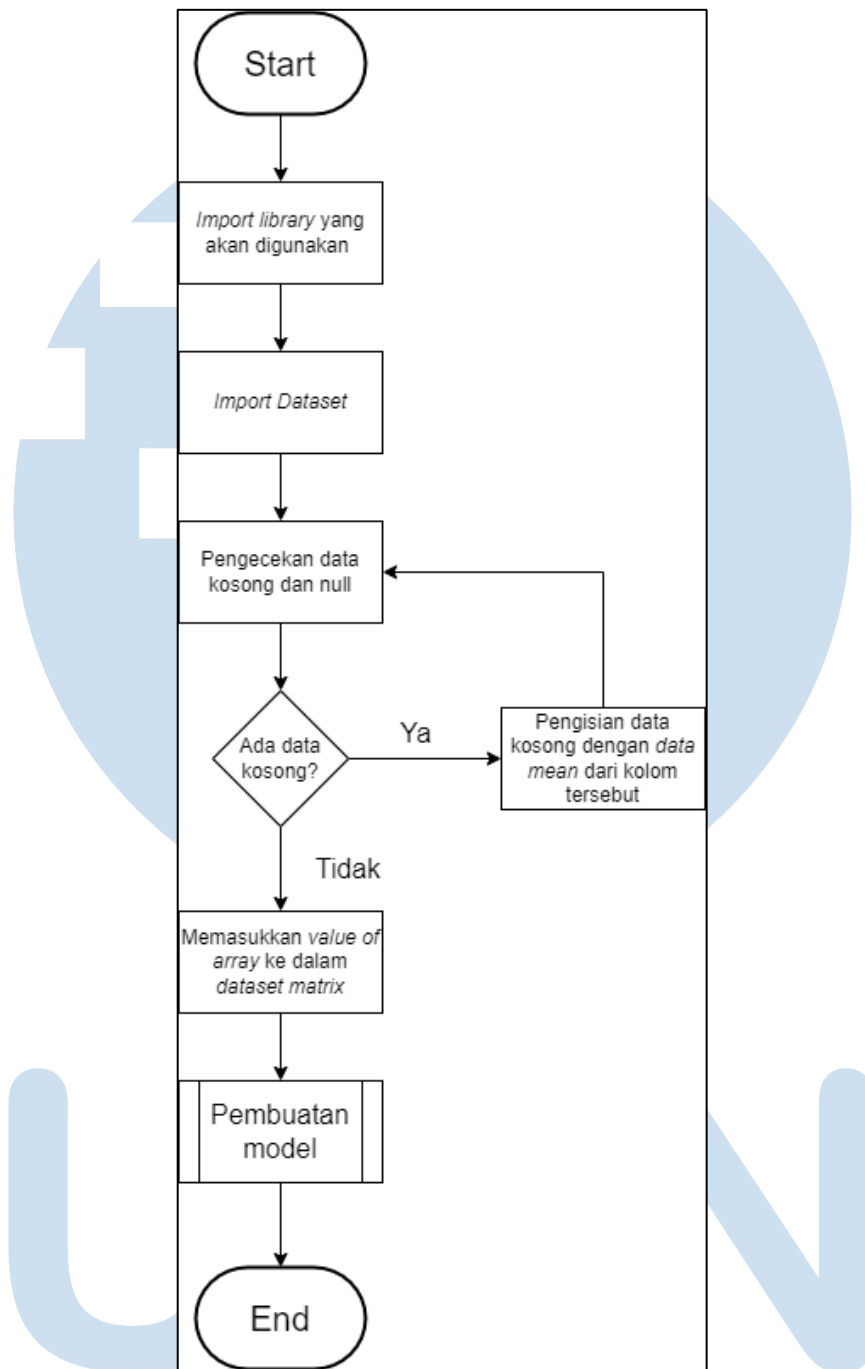
Tabel 3.1. Rancangan dataset covid-19 kabupaten tangerang

Dates	Confirmation Dates
2021-06-24	157
2021-06-25	163
.....
.....
2021-12-12	0
2021-12-13	0

3.2.2 Flowchart Preprocessing

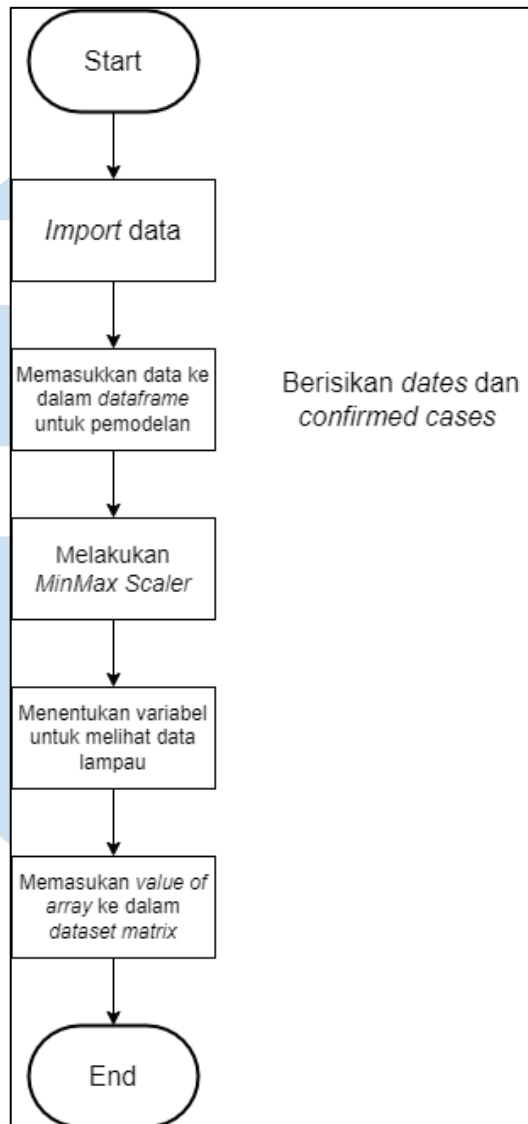
Preprocessing merupakan fase pemrosesan awal untuk *dataset* serta persiapan data-data yang diperlukan untuk pembetulan model. Pemrosesan awal *dataset* dapat dilihat pada Gambar 3.3. Di dalam fase ini akan dilakukan *import* untuk setiap *library* yang akan dipakai, baik dalam pra proses, pembuatan model, dan evaluasi. Apabila terdapat data kosong di dalam *dataset* maka akan dilakukan pengisian pada data kosong tersebut dengan menggantinya dengan data *mean* dari kolom yang bersangkutan.





Gambar 3.3. Flowchart pengecekan data kosong pada *dataset*

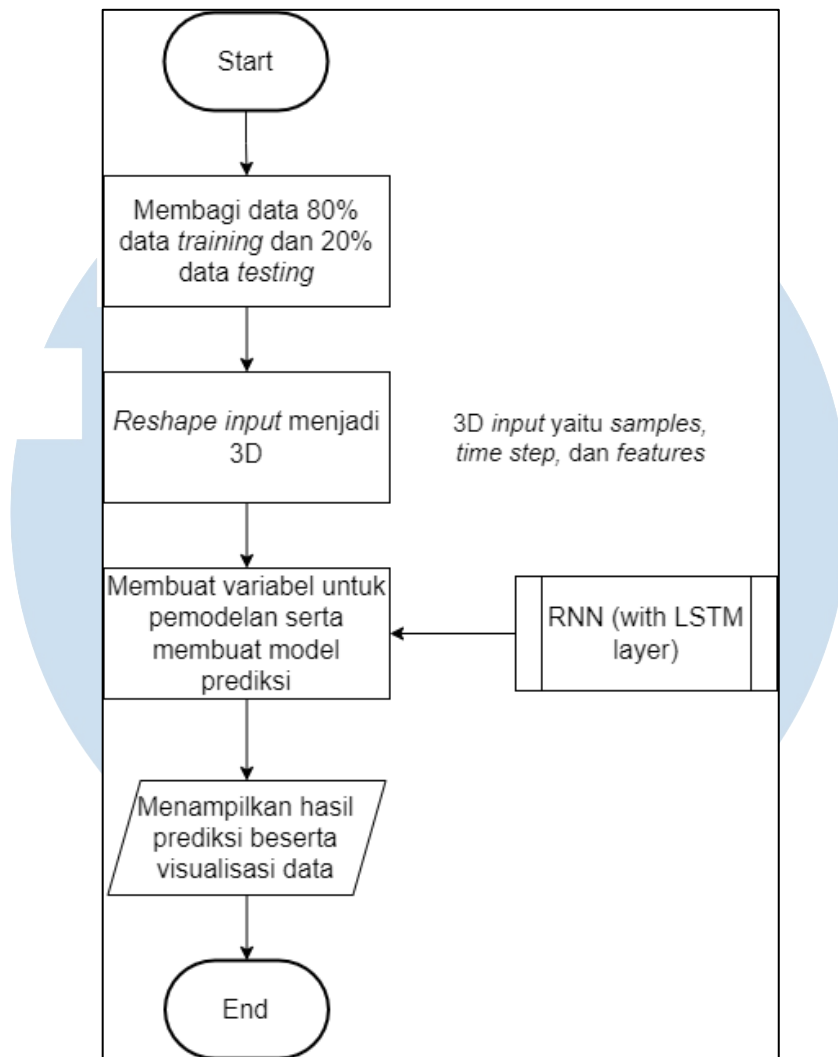
Setelah memastikan bahwa tidak ada nilai kosong pada *dataset*, maka akan dilanjutkan ke tahap pembuatan model seperti pada Gambar 3.4. Di dalam tahap tersebut, akan disiapkan variabel-variabel untuk pemodelan, lalu akan dilakukan *scaling data* menggunakan *MinMax scaler*, serta variabel untuk melihat kembali data lampau sebanyak 30 hari.



Gambar 3.4. Transformasi data kedalam *dataset matrix* untuk pemodelan

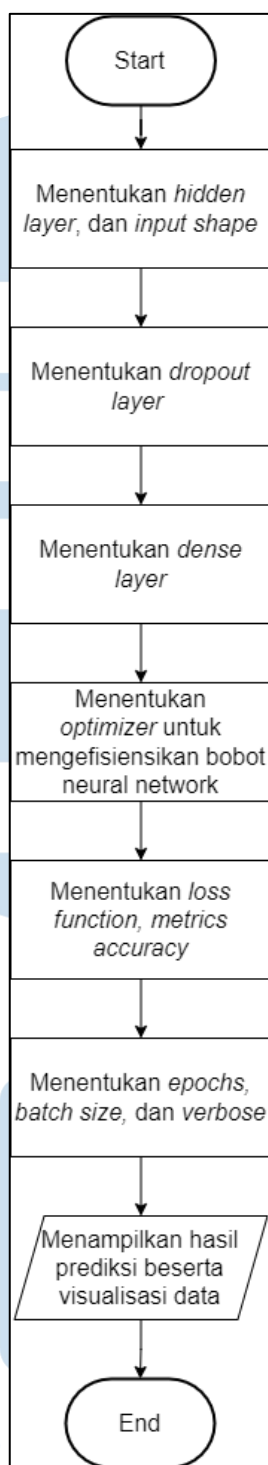
3.2.3 Flowchart Pemodelan

Pada proses pemodelan akan digunakan Recurrent Neural Network dengan *layer* Long-Short Term Memory sebagai metode prediksi, kemudian di dalam pemodelan data akan dibagi menjadi 80% sebagai data *training* dan 20% sebagai data *testing*. Alasan dibuatnya data *training* lebih besar karena metode RNN membutuhkan data *training* yang cukup banyak untuk meningkatkan akurasi dari model. Kemudian di bawah ini merupakan *flowchart* alur proses pemodelan.



Gambar 3.5. Flowchart alur proses pemodelan

Setelah data telah dibagi menjadi data *train* dan data *test* sebagai mana yang telah dijabarkan pada Gambar 3.5. Maka data tersebut akan dilakukan *reshapping input* menjadi bentuk *3D input* yaitu *samples*, *time step*, dan *features* kemudian sebelum dilakukan pemodelan akan dibuat variabel *look_back* yang dapat melihat data lampau, serta variabel *model* untuk menampung beberapa *layer* seperti *LSTM layer*, *dropout layer*, dan *dense layer* yang akan dijelaskan pada Gambar 3.6.



Gambar 3.6. Flowchart pemodelan dengan recurrent neural network LSTM

Setelah menentukan *hidden layer* dan *input shape* pada *layer LSTM* maka selanjutnya akan menambahkan *dropout layer* sebanyak 20% untuk mencegah terjadinya *overfitting* pada proses pemodelan, lalu menambahkan *dense layer* dimana *layer* ini akan mengklasifikasi *output* yang akan ditampilkan dengan

activation function yang berfungsi untuk membantu *neural network* memahami pola kompleks yang terdapat di dalam data [29]. Kemudian *optimizer* yang digunakan pada penelitian ini adalah *Adam optimizer* yang merupakan algoritma optimisasi tambahan yang dapat digunakan sebagai pengganti *classical stochastic gradient descent* untuk memperbarui bobot jaringan secara iteratif berdasarkan *training data* [30].

Sebelum model akan dijalankan maka ditentukan *loss function* dan *metrics accuracy*, didalam penelitian ini akan digunakan MSE (Mean Squared Error) sebagai *loss function* dan *metrics accuracy* untuk mengalkulasi seberapa sering prediksi sama dengan label yang ada. Kemudian untuk menjalankan model membutuhkan *tuning* untuk meminimalisir kompleksitas dengan menentukan *epochs*, *batch size*, dan *verbose*. *Epochs* digunakan untuk menunjukkan jumlah lintasan dari seluruh *training dataset* yang telah selesai, *batch size* merupakan *hyperparameter* yang digunakan untuk mendefinisikan jumlah *sample* yang harus diselesaikan sebelum memperbarui parameter yang ada di dalam model internal, sedangkan *verbose* untuk menampilkan hasil dari jumlah *epochs* yang dihasilkan.

UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA