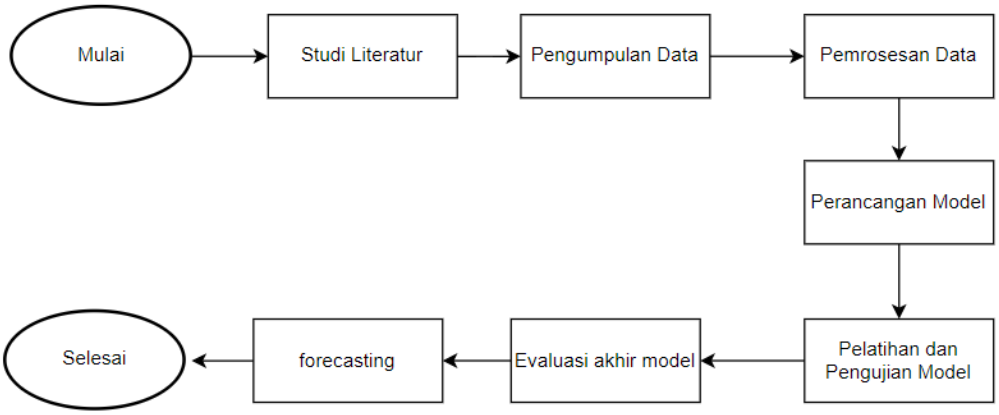
BAB III

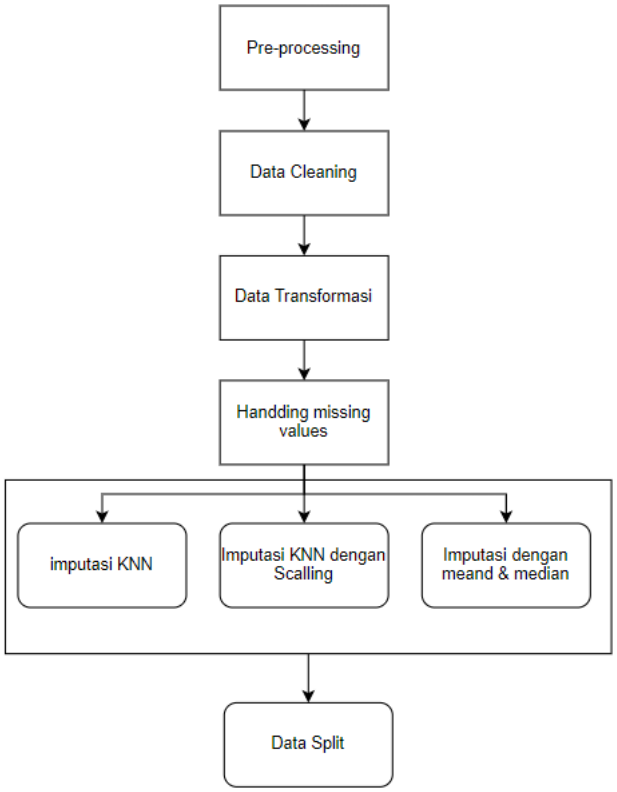
METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini ada beberapa tahap dalam meneliti. Peneliti melakukan studi literatur terhadap penelitian terdahulu yang berhubungan dengan prediksi lalu akan melakukan prediksi *multivariate* dan *univariate* data *time series* dengan LSTM dan GRU. Parameter untuk *multivariate* ada 6 dan *univariate* ada 1 yaitu PM2.5 dipilih karena merupakan partikel diudara yang paling kecil yang juga berbahaya bagi kesehatan bila sampai terhirup oleh manusia. Tahap selanjutnya, peneliti mengumpulkan data dari website resmi. Setelah itu, melakukan data *cleaning* terhadap data yang bertujuan untuk membersihkan data, seperti mengatasi *missing values* dll. Setelah itu, data *transformation* yang bertujuan untuk mengubah data *time series* menjadi data *supervised learning*. Kemudian, untuk data *split* akan dilakukan dengan dua rasio berbeda yaitu 7:3 dan 8:2. Lalu, membangun model LSTM dan GRU dan melakukan *training* model dan pengujian terhadap model. Setelah mendapat hasil dari uji coba pada masing-masing model peneliti akan melakukan evaluasi akhir menggunakan RMSE dan R2. Setelah itu akan dilakukan *forecasting* untuk beberapa waktu kedepan menggunakan model dengan performa terbaik.



Gambar 3. 1 Alur langkah penelitian



Gambar 3. 2 Alur Pemrosesan Data: Pre-processing data

3.2 Studi Literatur

Dalam melakukan studi literatur, penulis mempelajari dari penelitian terdahulu tentang sumber dan penyebab polusi udara, ISPU, *deep learning*, prediksi terhadap data *time series*. Selain itu, penulis juga menggunakan Kaggle dan github sebagai acuan tabahan bagi peneliti terkait topik penelitian. Kaggle merupakan komunitas *online* bagi para peneliti dan pengembang data, yang juga sebagai platform untuk berbagi projek *machine learning opensource* dataset. Sedangkan, Github ialah sebuah platform kolaborasi untuk developer dari semua level dan projek. Selain itu penulis juga melakukan diskusi seputar topik penelitian dengan Dosen Pembimbing Teknik Komputer penulis

3.3 Pengumpulan Data

Dataset yang digunakan bersumber dari website redahemisi.jakarta.go.id untuk data ISPU. Data yang digunakan memiliki kurun waktu 17 Januari 2022 sampai 30 September 2023.

	No	PM10	PM2.5	SO2	CO	O3	NO2	ISPU Status
0	1.0	-	-	-	-	-	-	-
1	2.0	-	-	-	-	-	-	-
2	3.0	-	-	-	-	-	-	-
3	4.0	-	-	-	-	-	-	-
4	5.0	-	-	-	-	-	-	-
...	...	...	...	...	...	...	...	...
14395	NaN	84	125	29	21	28	19	Tidak Sehat
14396	NaN	83	125	29	21	28	20	Tidak Sehat
14397	NaN	83	125	29	21	28	20	Tidak Sehat
14398	NaN	83	126	29	22	29	21	Tidak Sehat
14399	NaN	84	126	29	22	29	21	Tidak Sehat

14400 rows × 8 columns

Gambar 3. 3 Dataset Original

Pada penelitian ini, dataset yang digunakan adalah dataset berupa *time series* setiap jam dari 17 Januari 2022 sampau 30 September 2023. Jumlah keseluruhan data original sebanyak 14400 dengan 13 atribut yaitu, tanggal, jam, pm10, pm2.5, so2, co, o3, dan no2. yang ditunjukkan pada Gambar

3.4. Pada penelitian ini, dataset yang digunakan untuk *train* dan *test* model adalah data yang sama yang akan di *split*. Sedangkan, menggunakan dataset yang berbeda untuk membandingkan hasil *forecast* dari sumber yang sama namun *range* waktu yang berbeda yaitu 01 Oktober 2023 – 31 Oktober 2023 untuk melihat akurasi dari hasil prediksi dengan data sebenarnya yang belum pernah 'dilihat' model.

3.4 Pemrosesan Data

3.4.1 Data Cleaning

3.4.1.1 Multivariate

Dalam penelitian ini, untuk *multivariate* data *cleaning* yang dilakukan yaitu, menghapus kolom yang tidak dipakai, kolom yang di hapus adalah ISPU status. Setelah itu, mengubah data yang tidak terdefinisi *value* yang direpresentasikan dengan "-", maka akan data tersebut akan di *replace* menjadi *Not a Number* (Nan). Sehingga memudahkan dalam menghitung *missing values*

3.4.1.2 Univariate

Dalam penelitian ini, untuk *univariate*, data *cleaning* yang dilakukan yaitu, menghapus kolom yang tidak dipakai, kolom yang di hapus adalah PM10, SO2, CO, O3, NO2, dan ISPU Status. Setelah itu, mengubah data yang tidak terdefinisi *value* yang direpresentasikan dengan "-", maka akan data tersebut akan di *replace* menjadi *Not a Number* (Nan). Sehingga memudahkan dalam menghitung *missing values.*

3.4.2 Data Transformasi

Dalam penelitian ini, data transformasi untuk *multivariate* dan *univariate* kedunaya digunakan langkah yang sama yang dilakukan adalah menggabungkan kolom tanggal dan waktu lalu di jadikan kolom baru dengan nama dan format *Datetime*. Kemudian, kolom *Datetime* di jadikan index dari dataset. Lalu, karena tipe data untuk kolom lainnya masih berupa *object* maka tipe data untuk dataset lainnya diubah ke dalam tipe data *float*. Alasan data di ubah ke *float* karena algoritma dalam *deep learning* dalam pemrosesannya menggunakan operasi matematika sehingga perlu representasi ke tipe data numerik.

3.4.3 Data Pre-Processing

Dalam peneltian data yang digunakan memiliki *missing values* sekitar 38%, kemungkinan kurangnya data karena ada hari-hari tertentu alat atau sensor yang digunakan untuk mengukur tidak dinyalakan sehingga alat tidak mencatat dkualitas udara. Oleh karena itu, peneliti memakai tiga cara atau perlakuan berbeda dalam mengatasi missing values atau imputasi *value* yaitu:

3.4.3.1 Imputasi dengan K-nearest Neighbour

Imputasi dengan *K-nearest neighbour* (KNN), KNN adalah salah satu algoritma generalisasi KNN bertipe *supervised machine learning*. Dengan menemukan pola data baru dengan cara menghubungkan pola data yang ada dengan data baru [23]. Pada dasarnya alur algoritma KNN adalah pertama setalah ditentukan parameter terdekat dari '*neighboard*' yaitu K, selanjutnya menentukan pembobotan dengan teknik TF-IDF. Lalu melakukan perhitungan persamaan data dengan *cosine similarity*, kemudian masuk ke tahap pengurutan kemiripan data dari besar ke kecil. Setelah itu, diambil nilai K paling tinggi '*similarity' dengan* data

lalu tentukan kelasnya[24]. Pada penelitian ini nilai K yang digunakan adalah 5. Alasan peneliti menggunakan nilai K 5 karena pada uji coba pada penelitian [24], dengan menggunakan data time series berupa data cuaca dengan 9 parameter, nilai K = 5 memiliki hasil yang paling baik yaitu RMSE 0.902. Juga peneliti sendiri melakukan percobaan 3 kali dengan K 5, 6, dan 7 diperoleh nilai K 5 memiliki hasil yang lebih baik.

3.4.3.2 Imputasi dengan K-nearest Neighbour menggunakan Scalling

Imputasi dengan KNN dengan *scalling*, yang membedakan dengan dengan KNN sebelumnya adalah data sebelum masuk ke tahap KNN, dilakukan *scalling* untuk mengatasi *outlier* dengan *Robustscaler*. Nilai K yang digunakan adalah 5. Alasan menggunakan *Robustscaler* karena Teknik ini mempertahankan proporsi jarak *outlier*, juga mempertahankan informasi tentang *outlier* pada data serta memperkecil data, sehingga sangat cocok untuk data yang memiliki banyak *outlier.*

3.4.3.3 Imputasi dengan nilai mean dan median

Imputasi dengan nilai mean dan median dari masing-masing atribut. Sebelum itu, hitung nilai mean & median kemudian di isi nilai yang kosong pada setiap atribut. Gunakan nilai mean untuk atribut yang terdistribusi normal atau mendekati normal. Gunakan nilai median untuk atrribut yang memiliki *outlier* atau tidak terdistribusi normal.

Setelah itu, dataset akan dibagi atau diubah agar memiliki pola *input* dan pola *output* dari *inputan*. Agar dataset *timeseries* menjadi data *supervised learning*. Dataset diubah data *supervised learning* dengan nilai *past times* adalah 15 sedangkan nilai *future times* 1.

3.4.4 Data Split

Dalam penelitian ini, data akan dibagi dalam dua rasio berbeda yaitu 7:3 dan 8:2. Alasan peneliti membagi pembagian dataset menjadi dua rasio karena peneliti mengambil ide dari penelitian yang dilakukan oleh [9] , dimana dalam jurnal tersebut membagi data kedalam tiga rasio yaitu 7:3, 8:2, dan 9:1. Dan hasilnya dataset yang dibagi 7:3 memiliki performa baik. Oleh karena itu, menggunakan rasio 7:3 dan agar ada pembandingnya peneliti menambahkan rasio 8:2. Alasan pembagian rasio 7:3 dan 8:2 adalah karena banyak penelitian sebelumnya dan terbukti memberikan hasil yang baik, dan juga digunakan tergantung masalah yang ingin diselesaikan dan jumlah data. Salah satunya, pada penelitian [9] data yang digunakan juga *data time series* dan tujuannya juga sama yaitu melakukan prediksi untuk beberapa parameter sehingga mengambil ide untuk membagi dataset kedalam dua rasio berbeda dan juga melihat seberapa besar pengaruh dari pembagian data train dan test.

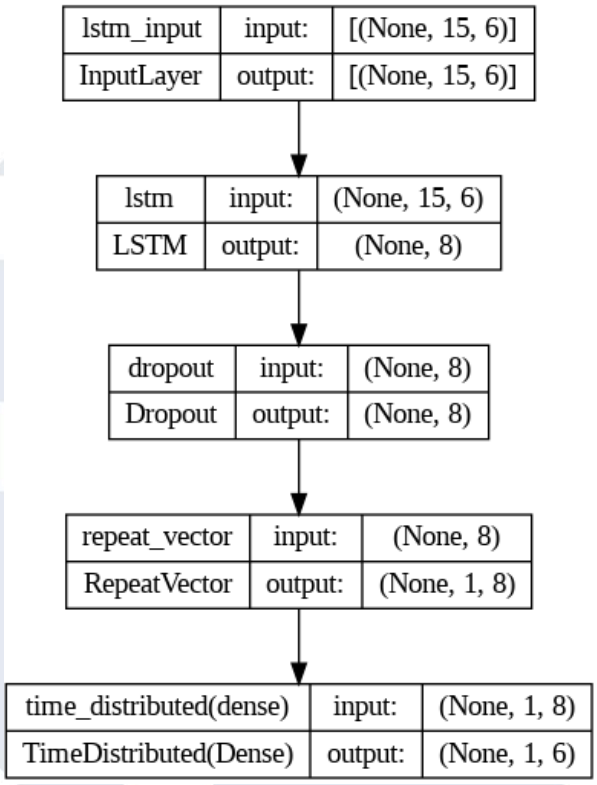
3.5 Perancangan Model

3.5.1 Multivariate

3.5.1.1 Long Short Tern Memory

Perancangan model LSTM terdiri dari sebuah *hidden layer* dengan 8-unit memori karena data yang digunakan dalam hal ini termasuk sedikit sehingga menggunakan unit memori yang kecil juga, aktivasi yang digunakan adalah *rectified linear unit* (relu) digunakan karena aktivasi ini memberikan elemen non-linear sehingga model mempelajari pola yang lebih kompleks dalam data *time series*, lebih sederhana dan cepat dalam komputasi, juga membantu mengatasi masalah gradien yang menghilang. Kemudian, *batch size* 32 karena disesuaikan dengan jumlah parameter yang digunakan yaitu 6 dan

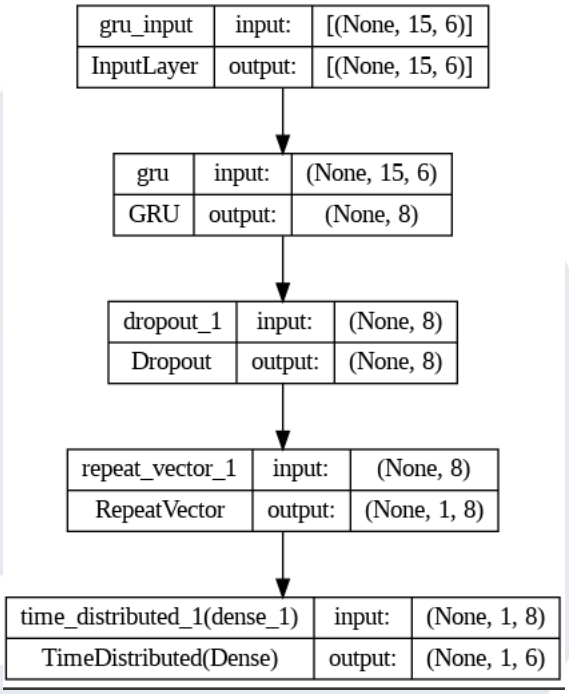
jumlah dataset yang digunakan jadi *batch size* yang digunakan tidak terlalu besar dan tidak terlalu kecil, juga dari beberapa kali percobaan yang dilakukan oleh peneliti sendiri dengan *batch size* 16, 32, dan 64 diperoleh bahwa *batch size* 32 yang menghasilkan model hasil yang memiliki hasil yang paling baik. *Optimizer* adam, *epoch* 150 karena disesuaikan juga dengan *batch size* peneliti sendiri melakukan dua kali uji coba dengan *epoch time* 100 dan 150 dan hasil yang diperoleh *epoch* 150 memperoleh hasil yang lebih baik dan fungsi *loss* MSE. Fungsi *loss* berperan untuk menghitung seberapa besar perbedaan antara data *train* dan test Selain itu, dilakukan *scalling* terhadap data menggunakan *MinMaxScaler*. *Scalling* berperan dalam mengatasi nilai data yang cenderung berbeda antar tiap atribut, juga untuk sehinga di normalisasikan dengan mengubah skala data dengan rentan antara 0 dan 1, sehingga mencegah terjadinya dominasi dari atribut lainnya dalam proses *training*. Selain itu, untuk mencegah terjadinya *overfitting* maka dilakukan *regiularization* L2, *cross validation* berperan dalam mendapatkan hasil validasi lebih akurat karena,*cross validation* akan membagi dataset menggunakan K-fold, dalam penelitian ini K-fold adalah 3 jadi akan dilakukan 3 literasi dalam *cross validation* terhadap data *train test*. dan *dropout* sebesar 0.1 digunakan untuk mengurangi *overfitting* nilai 0.1 digunakan karena model yang digunakan sederhana, ukuran dataset juga tidak terlalu besar, juga dari hasil uji coba dari nilai *dropout* 0.001 justru memberikan *overfitting.*



Gambar 3. 4 Arsitektur Model LSTM *multivariate*

3.5.1.2 Gated Reccurent Unit

Perancangan model GRU untuk arsitekturnya sebenarnya sama dengan model LSTM yaitu terdiri dari sebuah *hidden layer* dengan 8-unit memori, aktivasi yang digunakan adalah relu, *batch size* 16, *optimizer* adam, *epoch* 100 dan fungsi *loss* MSE. Selain itu, dilakukan *scalling* terhadap data menggunakan *MinMaxScaler*. Selain itu, untuk mencegah terjadinya *overfitting* maka dilakukan *regiularization* L2, *cross validation* dan *dropout* sebesar 0.1.



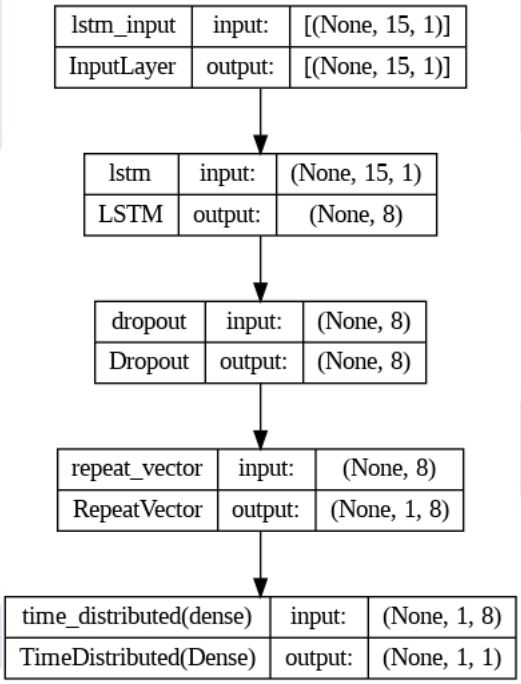
Gambar 3. 5 Arsitektur Model GRU *multivariate*

3.5.2 Univariate

3.5.2.1 Long ShortTerm Memory

Perancangan model LSTM untuk *univariate* terdiri dari sebuah *hidden layer* dengan 8 unit memori karena data yang digunakan dalam hal ini termasuk sedikit sehingga menggunakan unit memori yang kecil juga, aktivasi yang digunakan adalah *rectified linear unit* (relu) digunakan karena aktivasi ini memberikan elemen non-linear sehingga model mempelajari pola yang lebih kompleks dalam data *time series,* lebih sederhana dan cepat dalam komputasi, juga membantu mengatasi masalah gradien yang menghilang. Kemudian, *batch size* 32 dan jumlah dataset yang digunakan jadi *batch size* yang digunakan tidak terlalu besar dan tidak terlalu kecil, juga dari beberapa kali percobaan yang dilakukan oleh peneliti sendiri dengan *batch size* 16, 32, dan 64 diperoleh bahwa *batch size* 32 yang menghasilkan model hasil yang

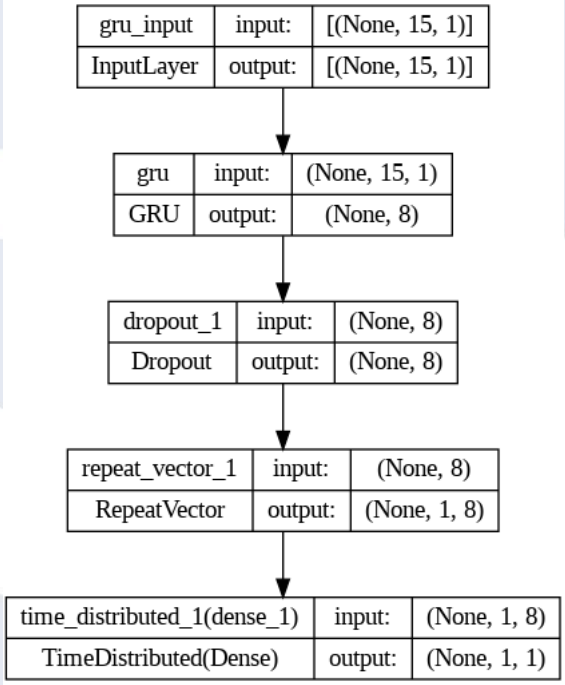
memiliki hasil yang paling baik. *Optimizer* adam, *epoch* 100 memiliki hasil yang mengalami sedikit *overfitting.* Serta memperoleh hasil yang lebih baik dan fungsi *loss* MSE. Fungsi *loss* berperan untuk menghitung seberapa besar perbedaan antara data *train* dan test Selain itu, untuk mencegah terjadinya *overfitting* maka dilakukan *regiularization* L2, *cross validation* berperan dalam mendapatkan hasil validasi lebih akurat karena,*cross validation* akan membagi dataset menggunakan K-fold, dalam penelitian ini K-fold adalah 3 jadi akan dilakukan 3 literasi dalam *cross validation* terhadap data *train test*. dan *dropout* sebesar 0.5 digunakan untuk mengurangi *overfitting.*



Gambar 3. 6 Arsitektur Model LSTM *univariate*

3.5.2.2 Gated Reccurent Unit

Perancangan model GRU untuk arsitekturnya sebenarnya sama dengan model LSTM yaitu terdiri dari sebuah *hidden layer* dengan 8 unit memori, aktivasi yang digunakan adalah relu, *batch size* 32, *optimizer* adam, *epoch* 400 dan fungsi *loss* MSE. Selain itu, Selain itu, untuk mencegah terjadinya *overfitting* maka dilakukan *regiularization* L2 0.0001, *cross validation* dan *dropout* sebesar 0.5



Gambar 3. 7 Arsitektur Model GRU *univariate*

3.6 Pelatihan dan Pengujian Model

Dalam penelitian ini, setelah model dirancang, model akan di *training* dengan data *train*. Setelah itu, hasil *training* di uji dengan data *test*. Proses *testing* model bertujuan agar mengetahui performa model yang dibangun.

3.7 Evaluasi Akhir Model

Evaluasi akhir model digunakan untuk memahami seberapa baik model dan sejauh mana model melakukan prediksi yang relevan serta akurat terhadap data yang belum pernah 'dipelajari' sebelumnya [25]. Dalam penelitian ini, evaluasi model yang digunakan adalah RMSE. Alasan penggunaan RMSE karena RMSE diukur dalam satuan yang sama dengan variabel respons, sehingga lebih mudah untuk diinterpresentasikan dan diukur dalam satuan yang sama dengan variabel respons. Nilai RMSE memberikan informasi tentang seberapa jauh prediksi dari nilai actual. Semakin kecil nilai RMSE, maka performa model semakin baik.

Selain menggunakan RMSE, digunakan juga metrik evaluasi R2, untuk mengukur seberapa baik model cocok dengan data aktual. R2 memberikan informasi seberapa banyak variasi dalam data aktual yang bisa dijelaskan oleh model. Semakin tinggi nilai R2, maka semakin baik performa model.

3.8 Forecasting

Dalam penelitian ini, setelah model dirancang, di latih dan di uji, di evaluasi. Selanjutnya dilakukan *forecasting* atau prediksi untuk 31 hari kedepan dari tanggal terakhir dalam dataset menggunakan model dengan performa terbaik. Kemudian hasil *forecasting* akan di bandingkan tingkat akurasinya dengan membandingkan dengan data *real* menggunakan metrix evaluasi model. *Forecasting* dilakukan untuk 31 hari kedepan karena jika dilihat beberapa aplikasi yang dijelaskan dibab sebelumnya, belum ada yang memprediksi untuk 31 hari kedepan. Oleh karena itu, peneliti ingin melihat performa model yang dibangun apakah mampu melakukan prediksi untuk 31 hari kedepan.