# **FINAL PROJECT**

# **PREDICTIVE MODELING & ANALYTICS**

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

Anggota:

Naufal Makarim Aschafitz (5026211074)

Daffa Rheza Prayoga (5026211175)

Wiweko Dwitya Anindito (5026211077)

# Overview Tugas

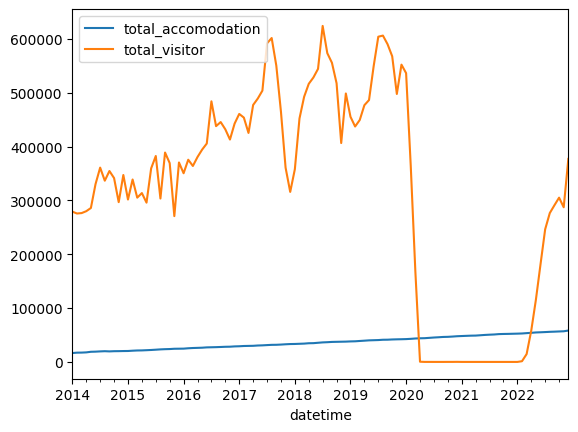
Tugas akhir mata kuliah Predictive Modelling & Analytics meliputi implementasi *supervised machine learning* berjenis peramalan. Dataset yang digunakan adalah pariwisata di Pulau Bali di tahun 2015 - 2022. Dari data tersebut akan dilakukan peramalan menggunakan empat metode yaitu RNN, LSTM, GRU, dan SVR pada tiga skenario pembagian *train-test* berbeda.

# Permasalahan

Data yang digunakan untuk peramalan adalah data pariwisata Pulau Bali di tahun 2015 - 2022. Dataset yang diberikan adalah file dengan tipe .csv. Dataset terdiri dari 78 baris data dan tiga kolom yaitu: datetime (periode bulanan), total\_visitor yaitu jumlah pengunjung suatu periode, dan total\_accomodation yaitu total akomodasi pada suatu periode. Berikut adalah cuplikan dari dataset:

| datetime | total\_visitor | total\_accomodation |
| --- | --- | --- |
| 01/01/2014 | 279257 | 16495 |
| 02/01/2014 | 275795 | 17221 |
| 03/01/2014 | 276573 | 17330 |
| 04/01/2014 | 280096 | 17737 |

Dan berikut adalah grafik dari dataset:



Pada tahun 2020 hingga 2022 terlihat angka pengunjung Bali mengalami penurunan secara drastis, hal ini diakibatkan oleh ditetapkanya kebijakan isolasi mandiri. Penurunan yang drastis ini dapat mempengaruhi hasil dari peramalan.

# Metodologi

Dilakukan peramalan untuk 12 periode kedepan (Januari 2023 - Desember 2023) menggunakan empat metode berbeda yaitu:

1. Recurrent Neural Network
2. Long Short Term Memory
3. Gated Recurrent Unit
4. dan Support Vector Regression

Untuk keempat metode tersebut dilakukan peramalan untuk tiga skenario pembagian *train-test* yaitu:

1. 80%-20%
2. 70%-30%
3. 60%-40%

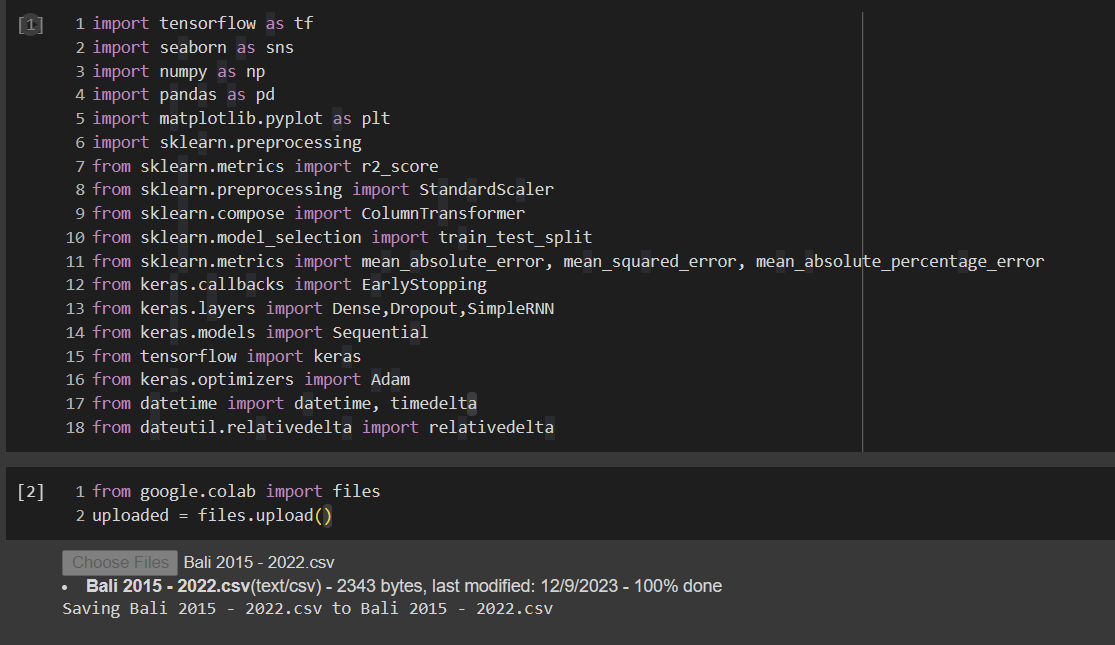
# Hasil eksperimen

## RNN

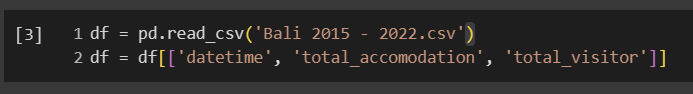
### Tahapan

#### Impor

Impor dataset dan *library* yang dibutuhkan.

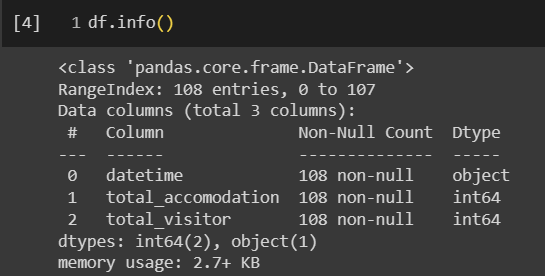


Setelah itu read dataset menggunakan pd.read\_csv.

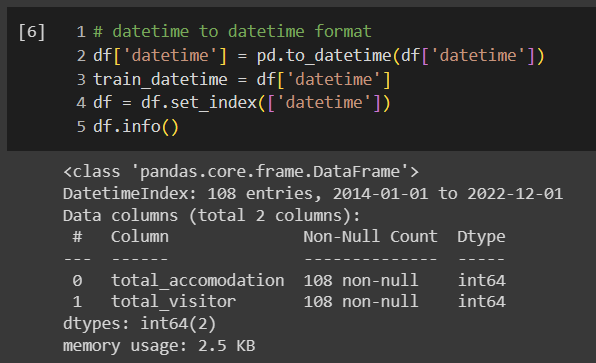


#### Pra-proses

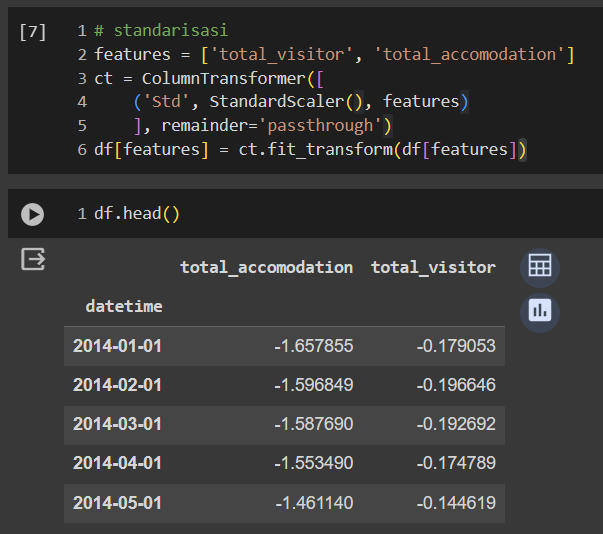
Cek jumlah data yang null menggunakan .info()



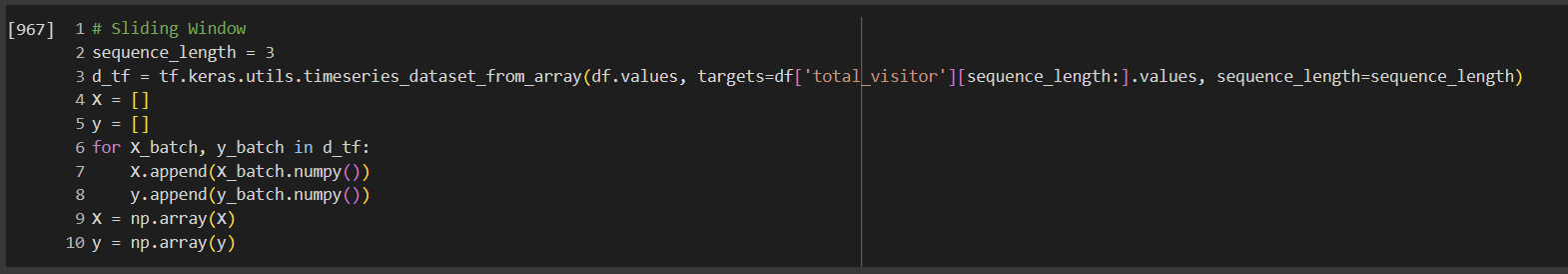
Terlihat bahwa dataset terdiri dari 108 entri dan semua entri tidak ada yang mengandung nilai null. Terlihat juga bahwa tipe objek kolom datetime adalah *object*, sehingga perlu dilakukan konversi ke tipe *datetime* dan juga dijadikan *index*.



Selanjutnya untuk menyamakan *range* data dan mengurangi variansi dilakukan *scaling* data. Untuk *scaling* digunakan StandardScaler() yang akan mengubah rata-rata data menjadi 0 dan standar deviasi menjadi 1.



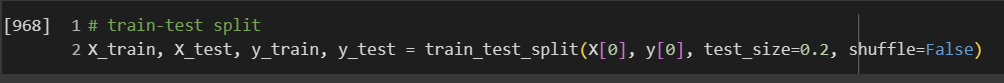
Terakhir dataset akan diubah bentuknya menjadi *sliding window* dengan *sequence length* sebesar 3. Variabel independen yang digunakan adalah kolom total\_visitor dan total\_accomodation, sedangkan variabel dependen/target yang digunakan adalah total\_visitor. Sesuai dengan persyaratan input model, bentuk *dataframe* akan diubah menjadi *array* tiga dimensi.



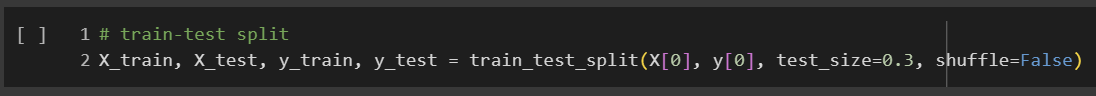
#### Splitting

Dataset yang sudah dipraproses dibagi menjadi dua bagian yaitu *training* dan *testing*. Pembagian dilakukan menggunakan train\_test\_split(). Dilakukan tiga skenario splitting yaitu:

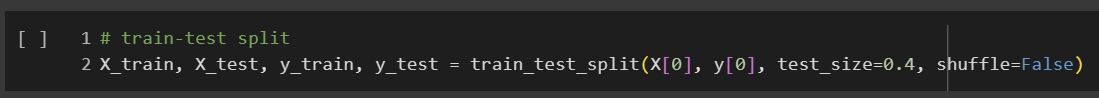
1. 80-20



1. 70-30



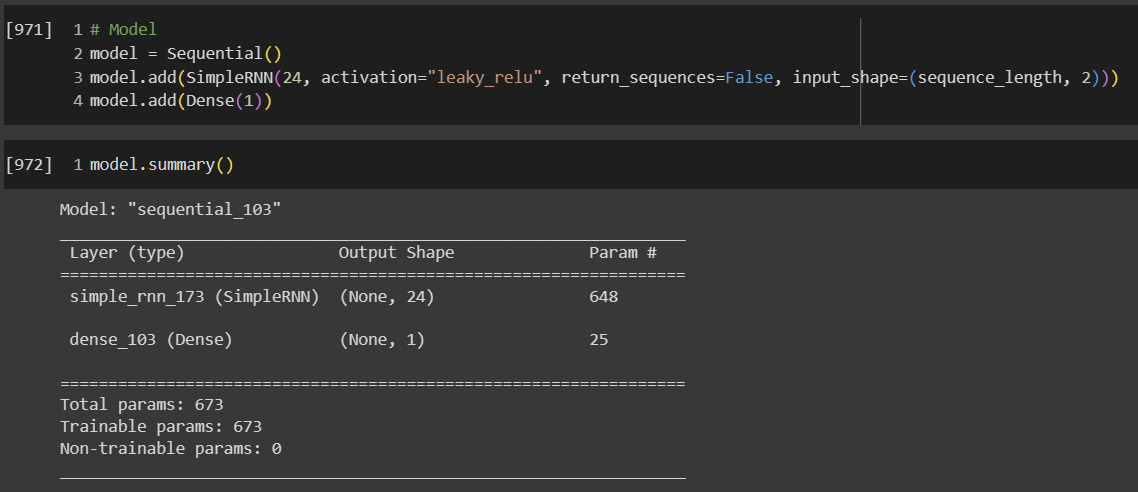
1. 60-40



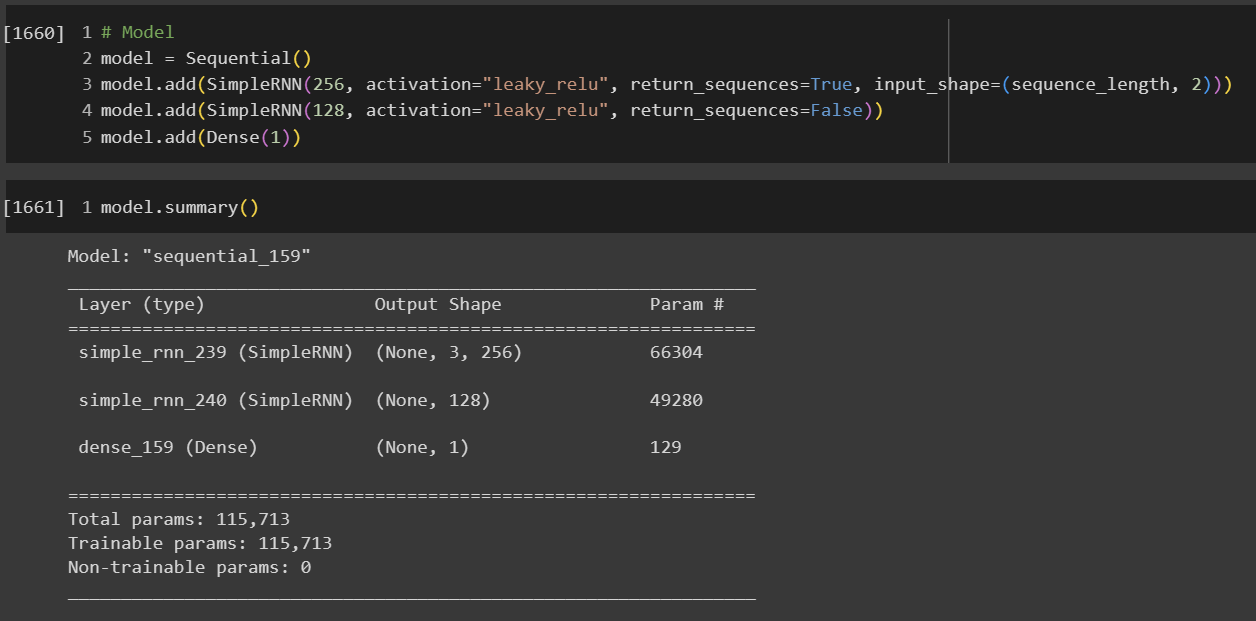
#### Model

Pembuatan model dilakukan menggunakan *library* Tensorflow-Keras. Pada tahapan ini dilakukan *hyperparameter tuning* secara manual untuk mencapai hasil R2 terbaik untuk tiap skenario. Pada model digunakan layer berupa SimpleRNN untuk mengutilisasi fungsi RNN dan juga diimplementasikan. Berikut adalah model yang digunakan untuk tiap skenario:

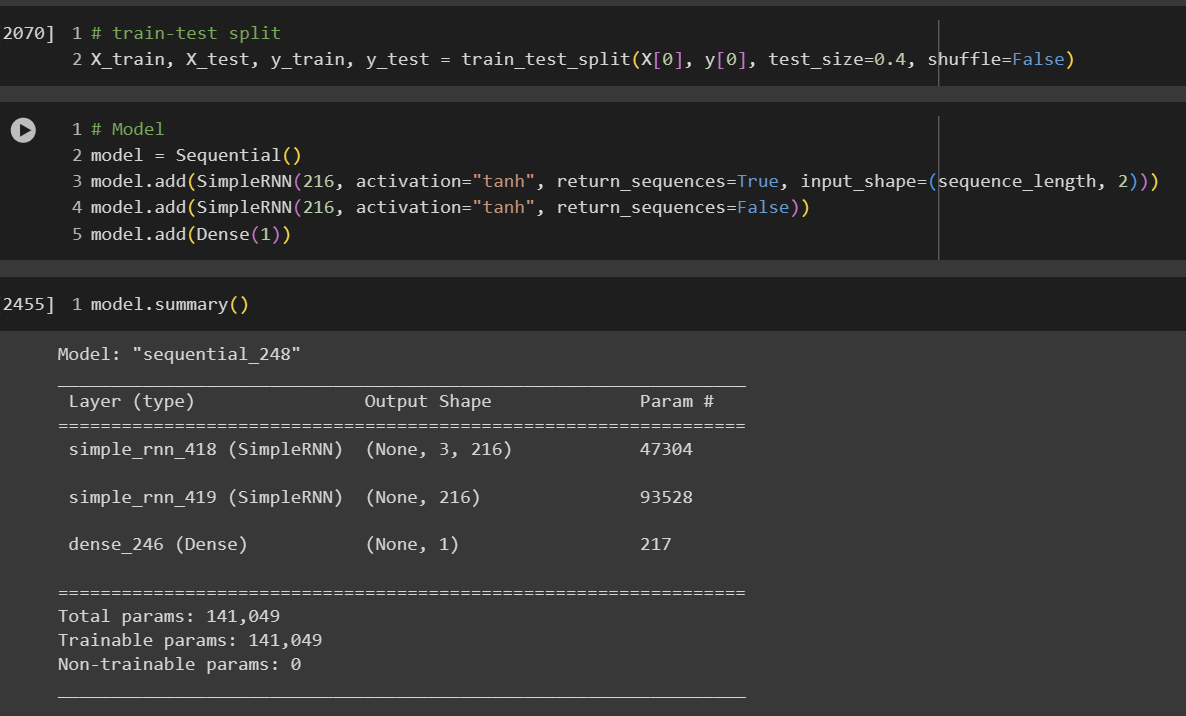
1. 80-20



1. 70-30



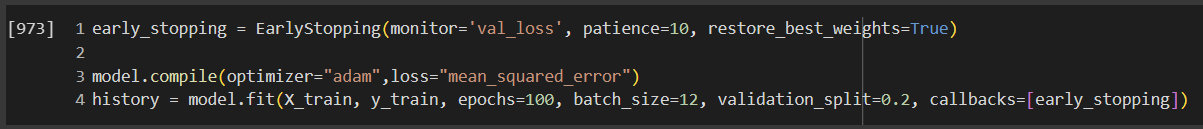
1. 60-40



#### Pelatihan

Dilakukan pelatihan untuk X\_train pada ketiga skenario dengan metode yang sama. Pelatihan dilakukan dengan epoch sejumlah 100/50 dan batch\_size sebesar 12. Digunakan juga EarlyStopping() untuk mencegah terjadinya *overfitting*.

1. 80-20 & 70-30

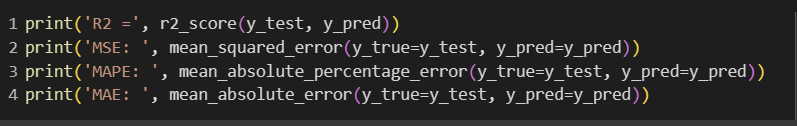


1. 60-40

#### Prediksi

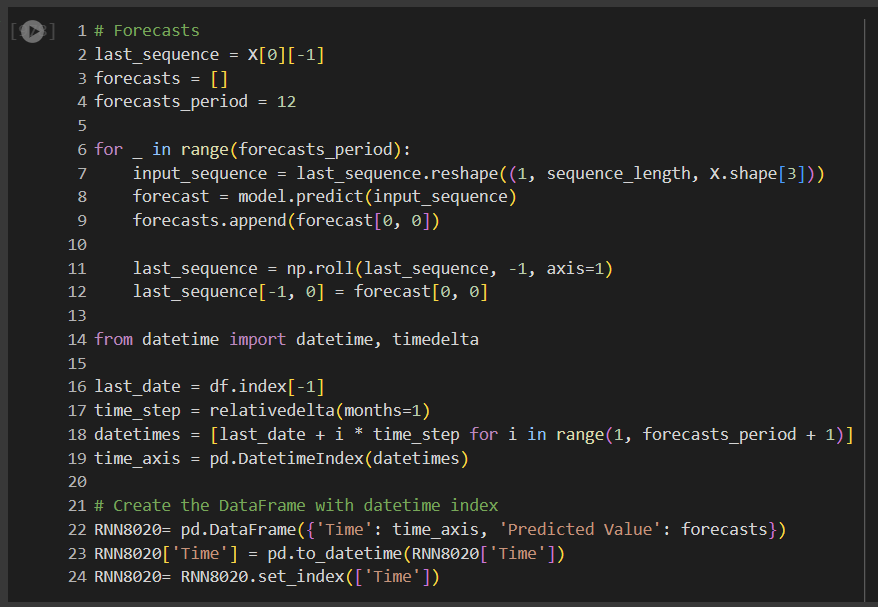
Menggunakan ketiga model yang sudah dilatih, dilakukan prediksi untuk data X\_test. Kemudian hasil prediksi dievaluasi menggunakan empat metrik yaitu: skor R2, MSE, MAPE, dan MAE. Hasil evaluasi tersebut dijadikan bahan acuan untuk menentukan kualitas dari model dan melakukan penyesuaian *hyperparameter*.





#### Peramalan

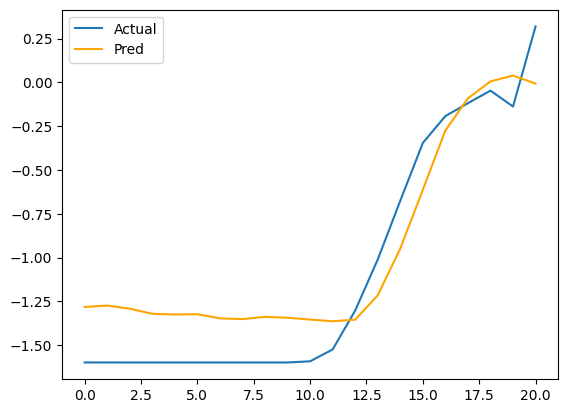
Kemudian dilakukan peramalan menggunakan ketiga model yang sudah dilatih untuk 12 periode kedepan, yaitu dari Januari 2023 hingga Desember 2023.



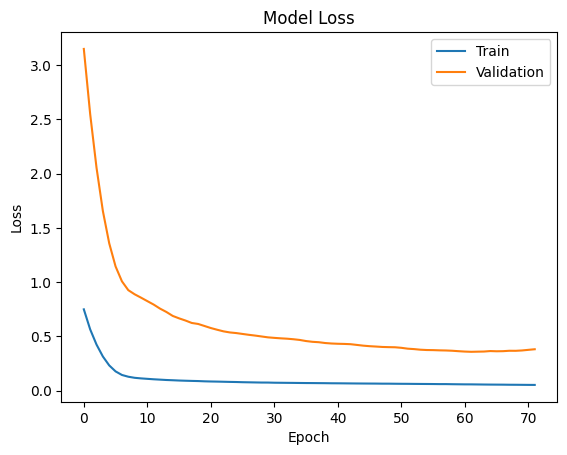
### Hasil Prediksi

Dari ketiga model RNN didapatkan hasil prediksi sebagai berikut:

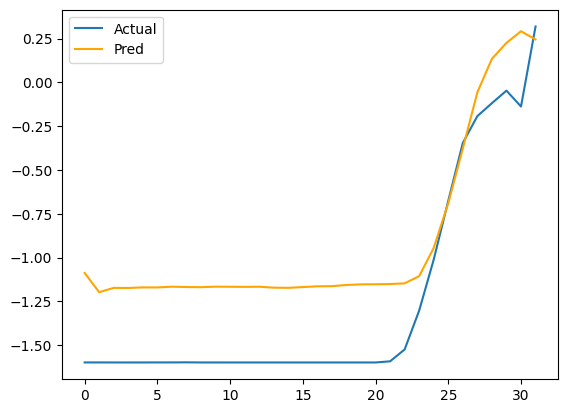
1. 80-20



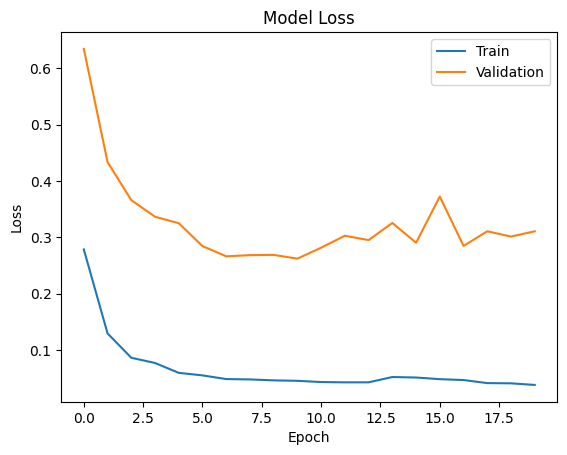
Dengan loss sebagai berikut:



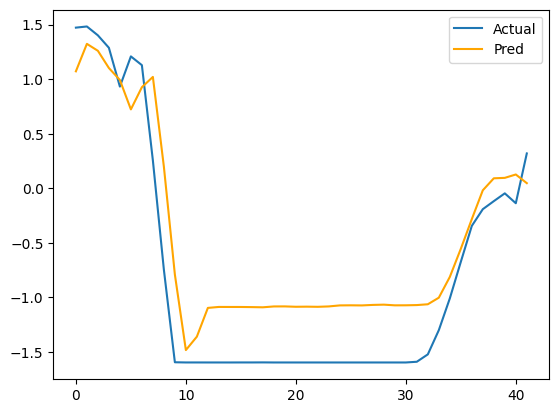
1. 70-30



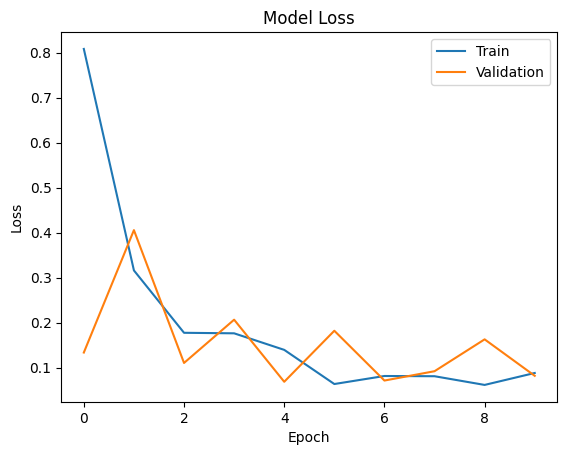
Dengan loss sebagai berikut:



1. 60-40



Dengan loss sebagai berikut:



### Evaluasi

Berikut adalah hasil evaluasi prediksi X\_test dari ketiga model:

| Evaluasi/  *Split* | R² | MSE | MAPE | MAE |
| --- | --- | --- | --- | --- |
| 80-20 | 0.873 | 0.057 | 0.357 | 0.221 |
| 70-30 | 0.590 | 0.146 | 0.579 | 0.356 |
| 60-40 | 0.828 | 0.205 | 0.530 | 0.405 |

## LSTM

### Tahapan

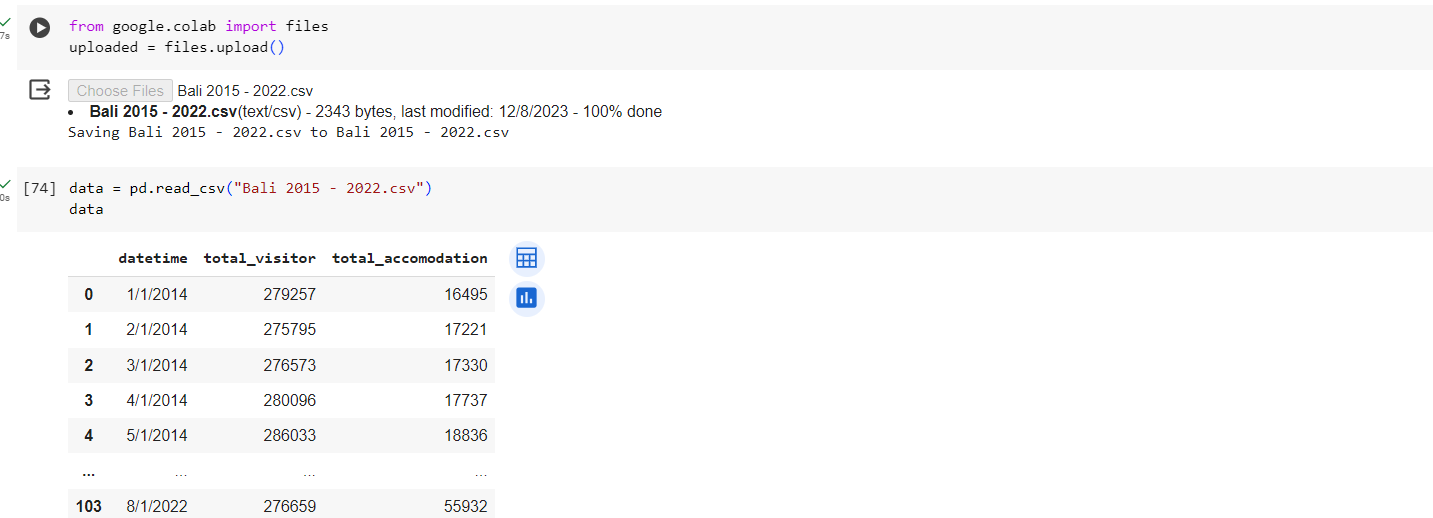
#### Import

Import library yang dibutuhkan



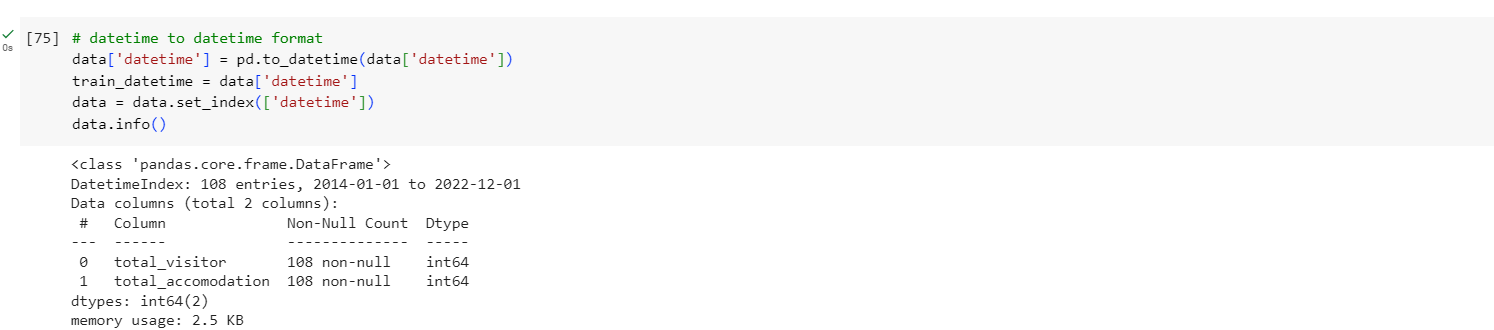
#### Load Dataset

Load dataset yang digunakan

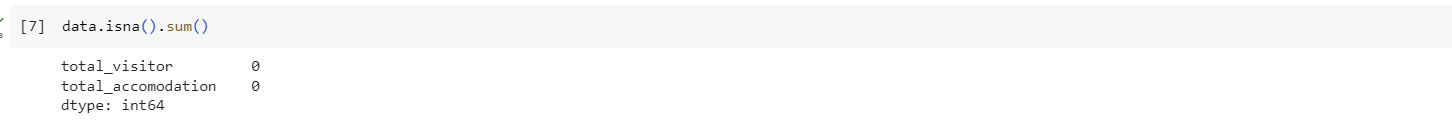


#### Pra-proses

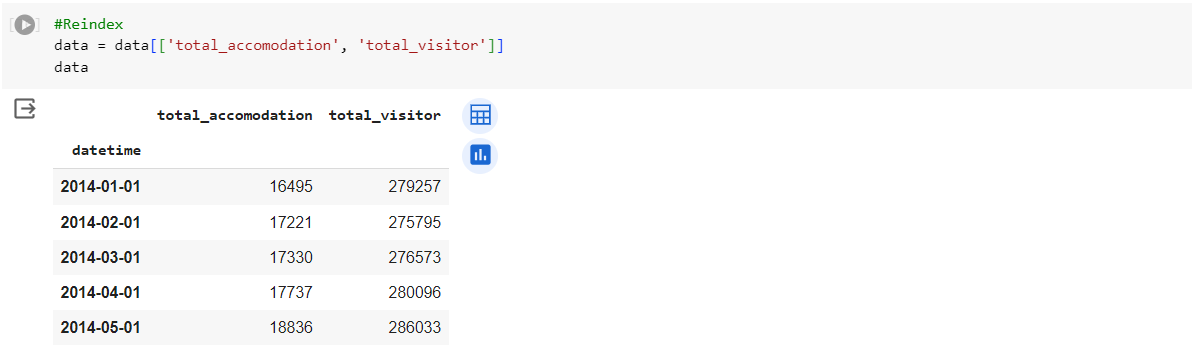
Terdapat beberapa proses yang dilakukan. Pertama mennganti format kolom “datetime” menjadi datetime format dan mengubahnya ke bentuk index. Selain itu, juga dilakukan mengecek nilai null dari data dengan menggunakan .info().



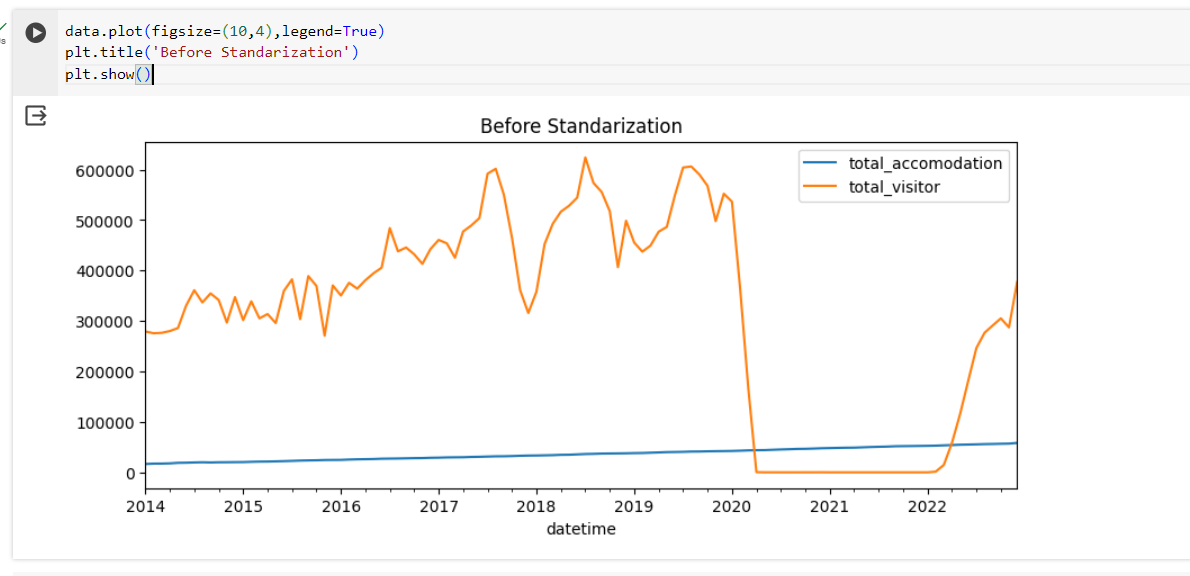
Lalu, mengecek jumlah missing values yang dimiliki oleh setiap kolom data dengan .isna().null().



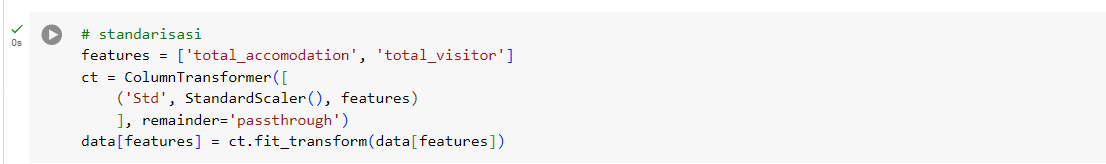
Setelah dicek ternyata data dalam kedua kolom tidak memiliki missing value. Itu berarti tidak perlu melakukan proses untuk mengisi nilai yang belum terisi tersebut. LLalu, melakukan reindexing kolom data total\_visitor dan total\_accomodation untuk memudahkan pada saat menentukan variabel x dan y-nya.



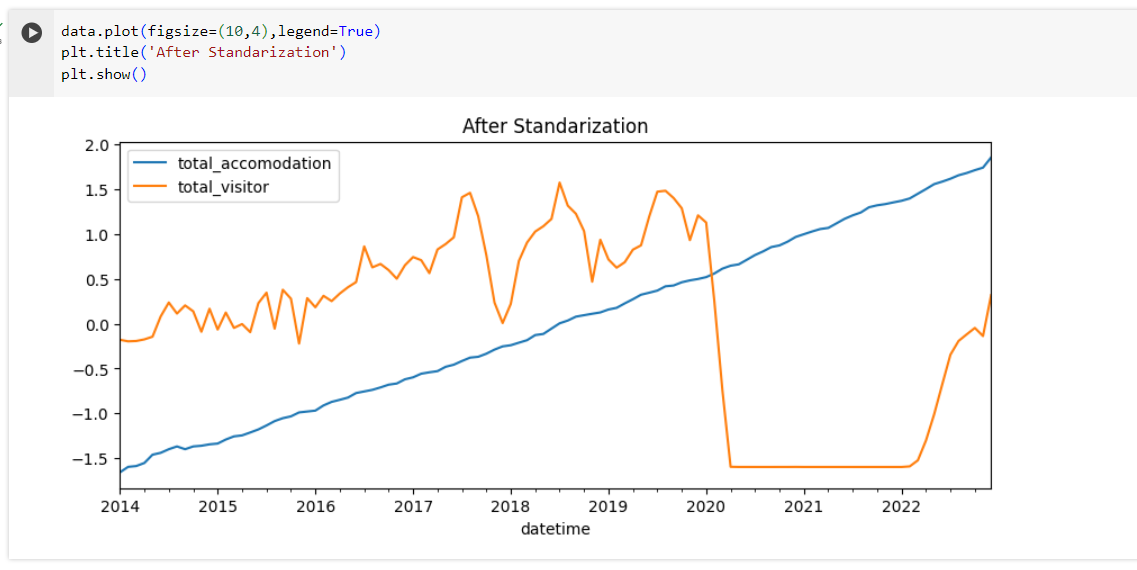
Lalu, membuat line plot pada datasetnya



Lalu, melakukan standarisasi datasetnya sebagai ganti dari normalisasi



Membuat line plot data setelah standarisasi untuk melakukan perbandingan pada data sebelum dan setelah dilakukan standarisasi

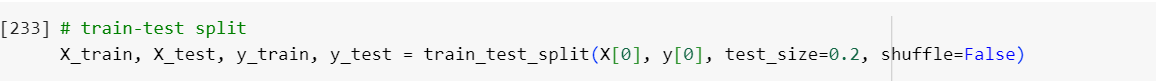


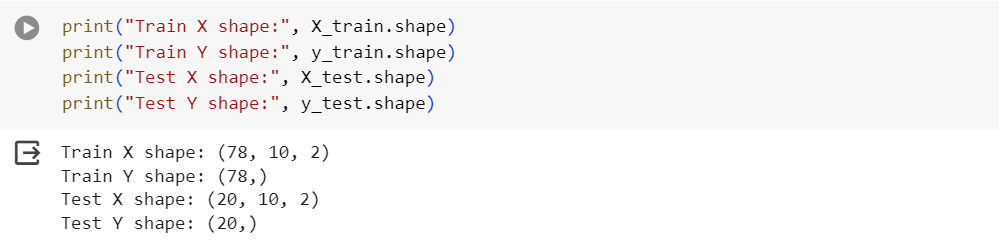
Setelah itu, membuat sliding window. Sequence length yang digunakan 3, dengan variabel yang digunakan adalah kedua kolom dalam dataset. Kolom data total\_visitor dan total\_accomodation menjadi variabel independen dan total\_visitor menjadi dependennya. Setelah itu, mengubah dataframe menjadi array 3 dimensi.

#### Splitting

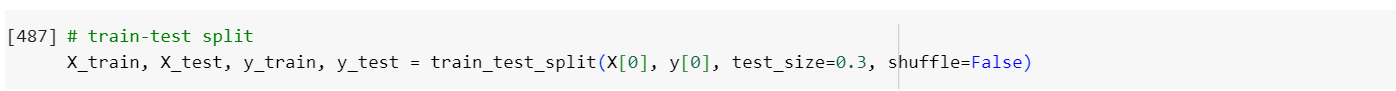
Pada saat dataset telah dipraproses, dataset akan dibagi menjadi data training-testing. Dalam tes kali ini, splitting data training-testing akan dilakukan ke dalam 3 skenario.

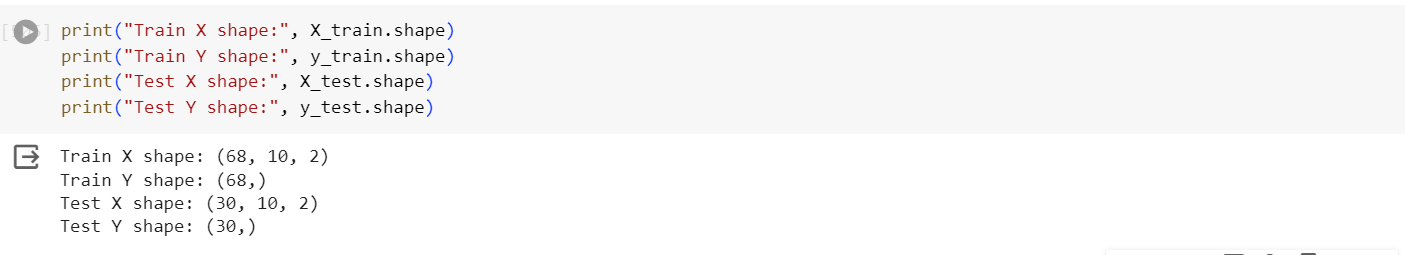
1. 80:20



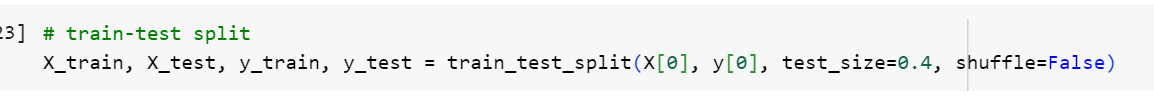


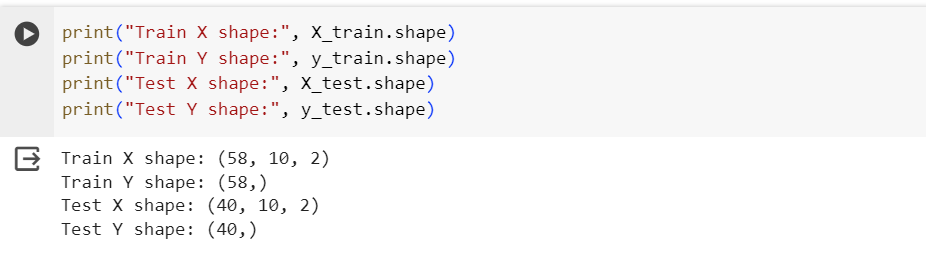
1. 70:30





1. 60:40

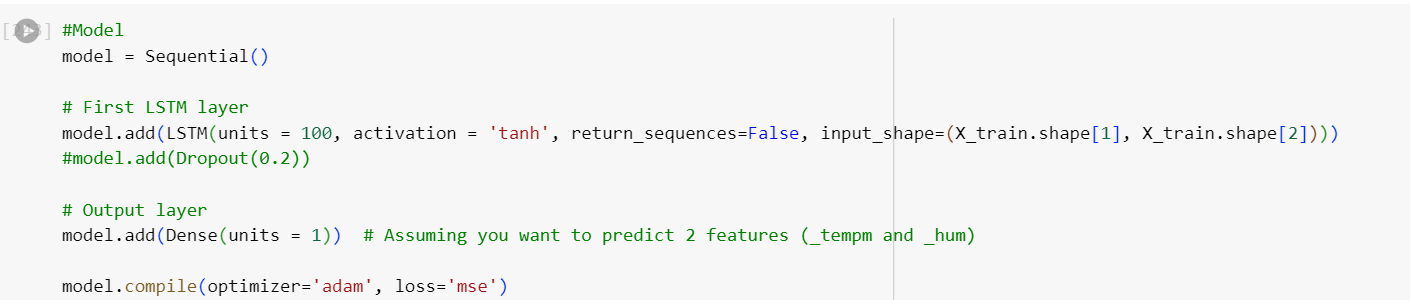


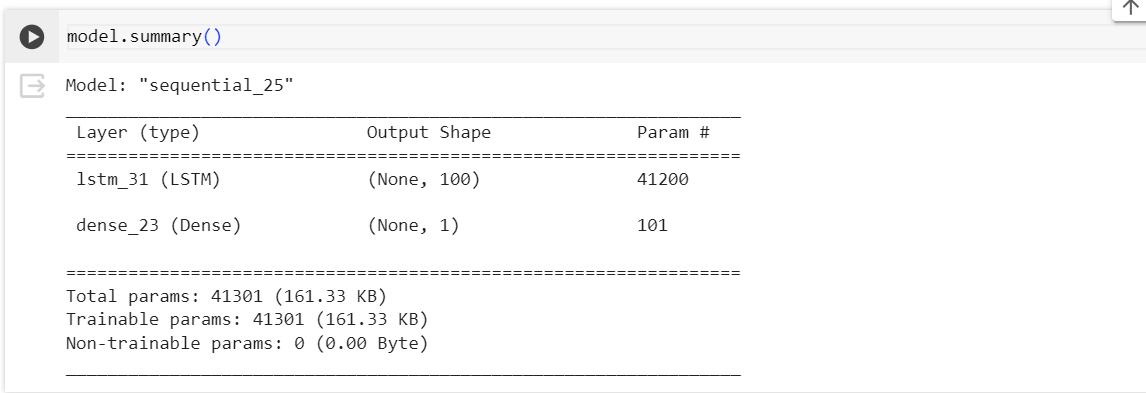


#### Model

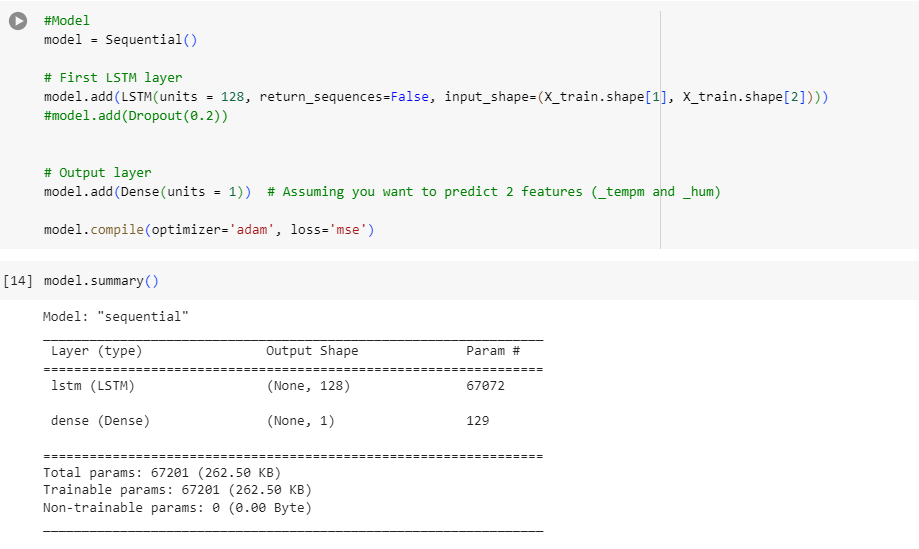
Model prediksi dibuat dengan menggunakan *library* Tensorflow-Keras. Pada tahapan ini dilakukan *hyperparameter tuning* secara manual untuk mencapai hasil R2 terbaik untuk tiap skenario. Pada model digunakan layer berupa LSTM untuk mengutilisasi fungsi RNN dan juga diimplementasikan. Berikut adalah model yang digunakan untuk tiap skenario:

1. 80:20

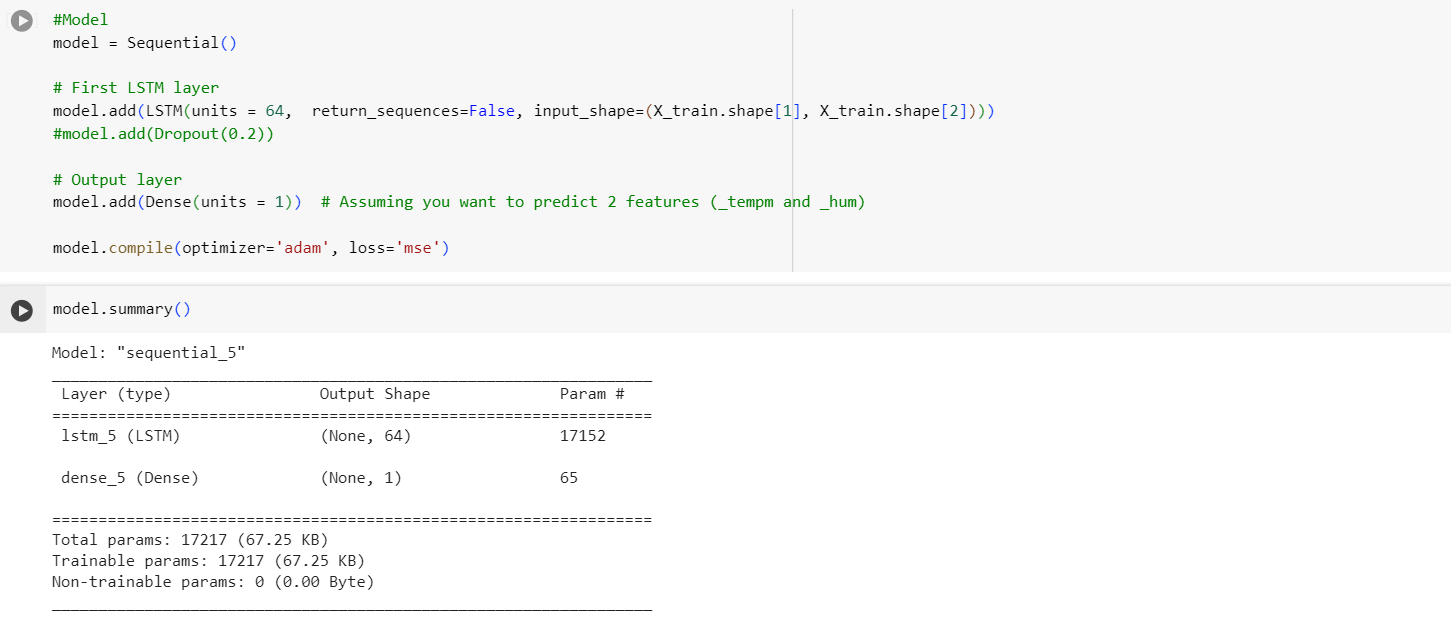




1. 70:30



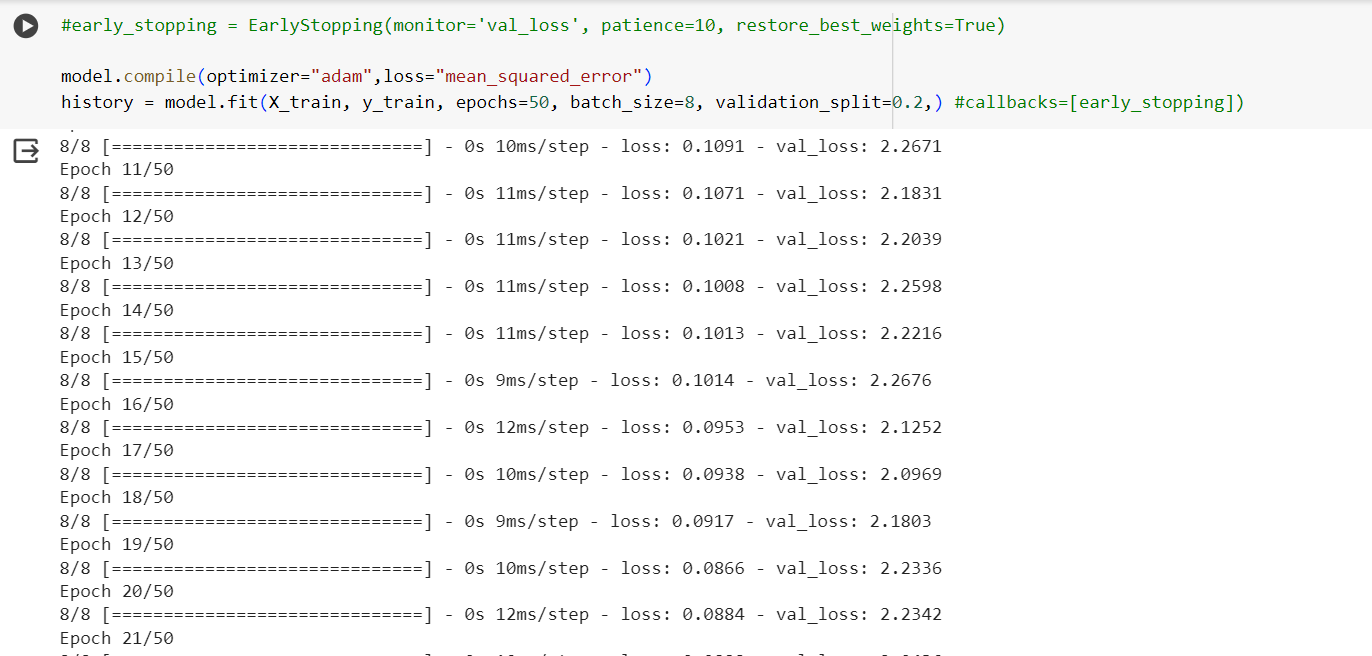
1. 60:40



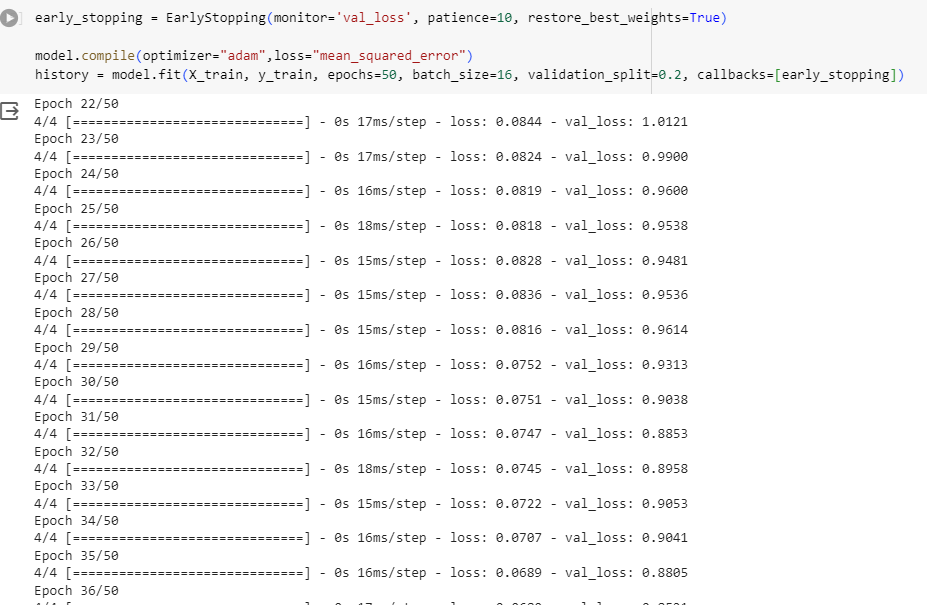
#### Pelatihan

Setelah dilakukan pembuatan model prediksi, dilakukan pelatihan untuk X\_train pada ketiga skenario dengan metode yang sama. Pelatihan dilakukan dengan jumlah epoch dan batch\_size yang diperlukan. Digunakan juga EarlyStopping() untuk mencegah terjadinya *overfitting*.

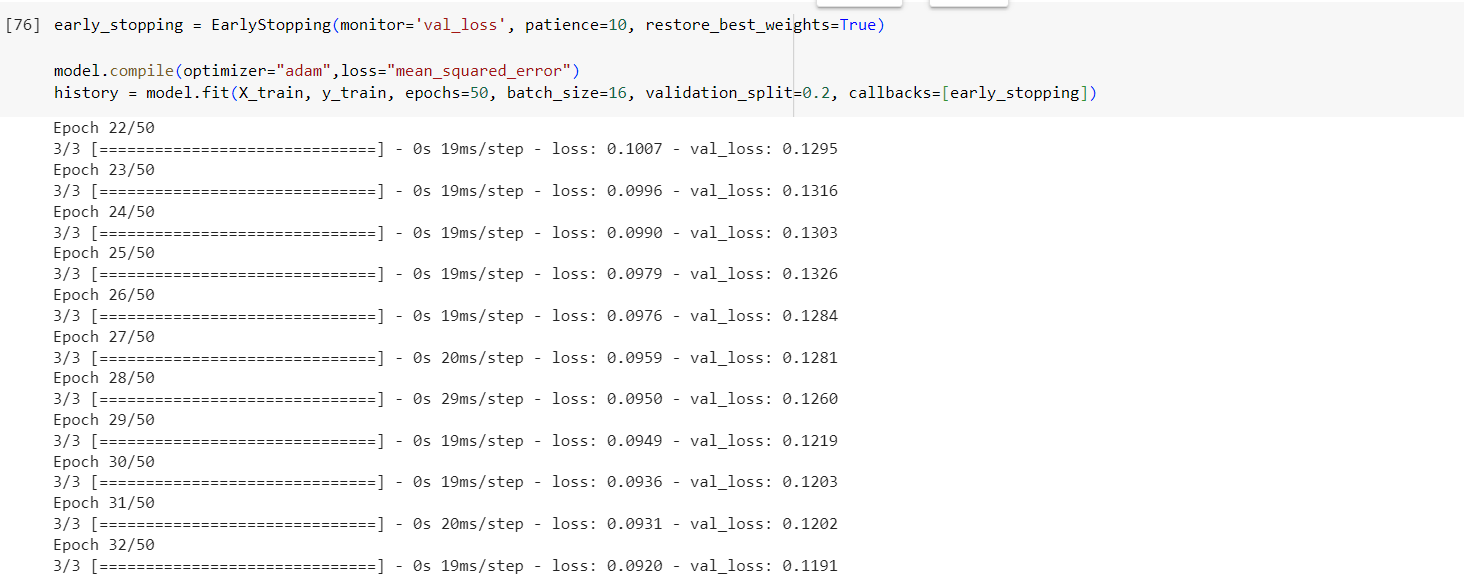
1. 80:20



1. 70:30

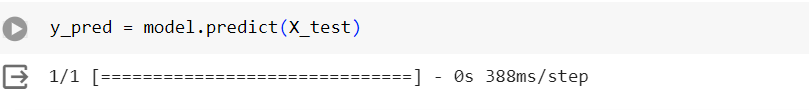


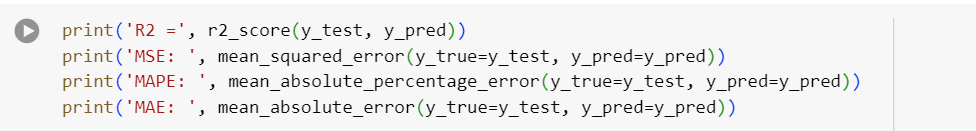
1. 60:40



#### Prediksi

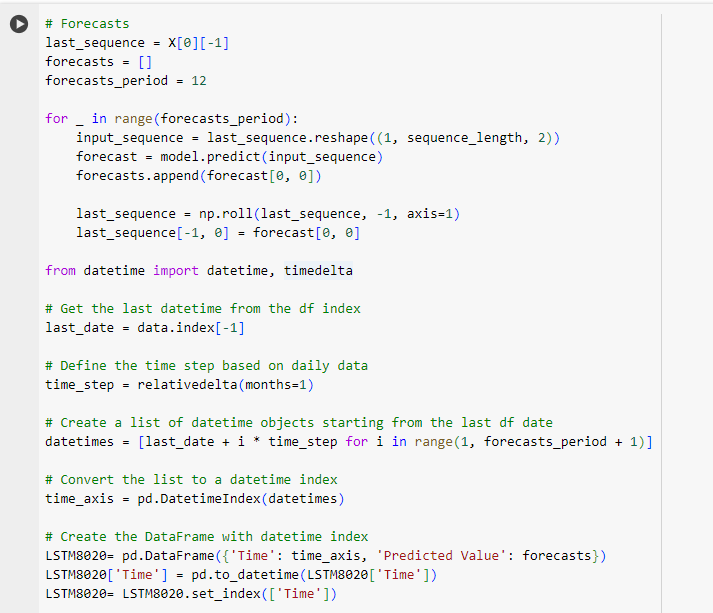
Dengan ketiga skenario model yang sudah dilatih, dilakukan prediksi untuk data X\_test. Kemudian hasil prediksi dievaluasi menggunakan empat metrik yaitu: skor R2, MSE, MAPE, dan MAE. Hasil evaluasi tersebut dijadikan bahan acuan untuk menentukan kualitas dari model dan melakukan penyesuaian *hyperparameter*.





#### Peramalan

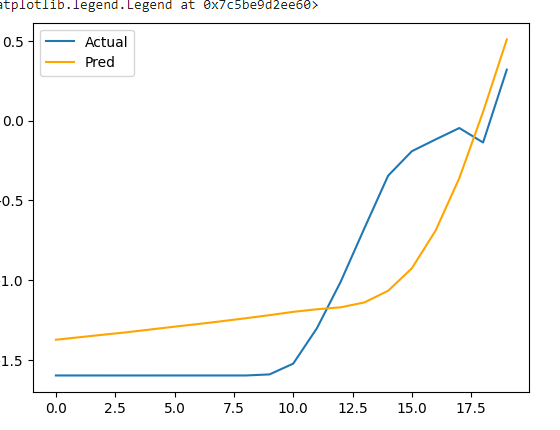
Kemudian dilakukan peramalan menggunakan ketiga model yang sudah dilatih untuk 12 periode kedepan, yaitu dari Januari 2023 hingga Desember 2023.



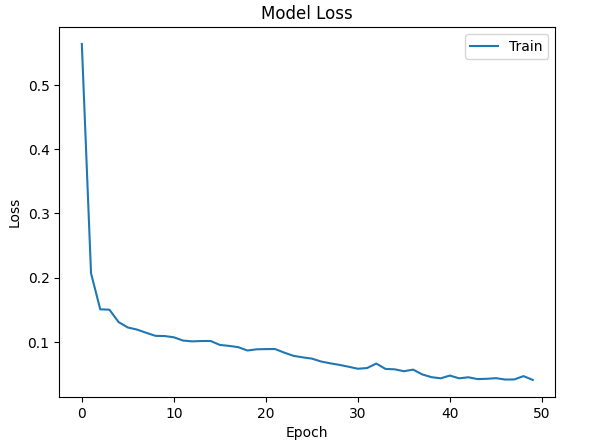
### Hasil Prediksi

Dari ketiga skenario LSTM yang telah dibuat, didapatkan hasil prediksi sebagai berikut:

1. 80:20

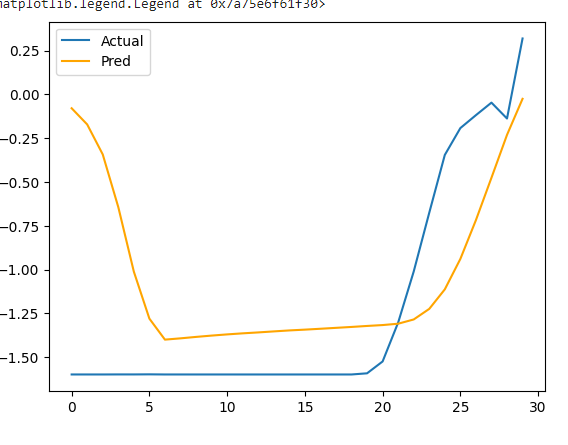


Hasil Prediksi

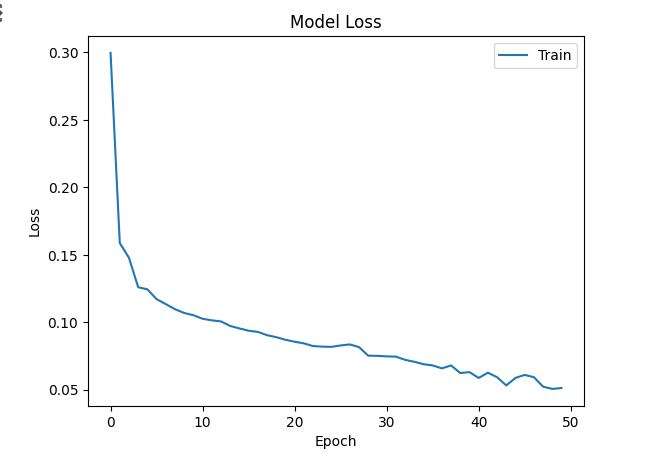


Model loss

1. 70:30

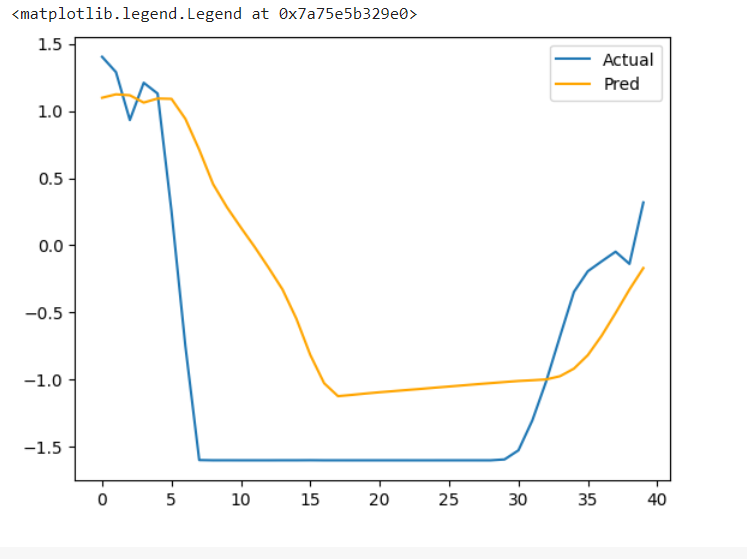


Hasil prediksi

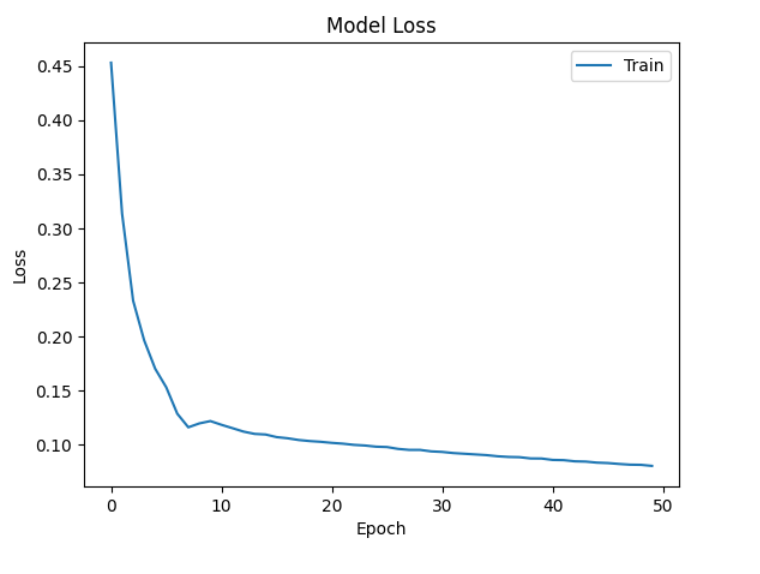


Model loss

1. 60:40



Hasil prediksi

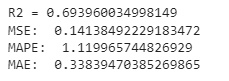


Model loss

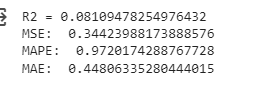
### Evaluasi

Berikut adalah hasil evaluasi prediksi X\_test dari ketiga model:

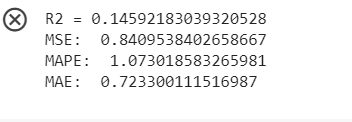
1. 80:20



1. 70:30



1. 60:40



| Evaluasi/  *Split* | R² | MSE | MAPE | MAE |
| --- | --- | --- | --- | --- |
| 80-20 | 0.694 | 0.1414 | 1.11 | 0.338 |
| 70-30 | 0.081 | 0.344 | 0.972 | 0.448 |
| 60-40 | 0.146 | 0.841 | 1.07 | 0.723 |

## GRU

### Tahapan

#### Import

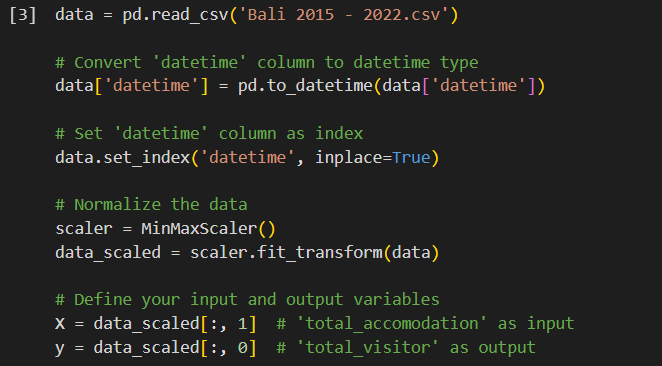
Import dataset dan library yang dibutuhkan

#### 

#### Pra-proses

Baca data yang sudah di-upload dan ubah data kolom ‘datetime’ ke bentuk date dan menjadi Index.

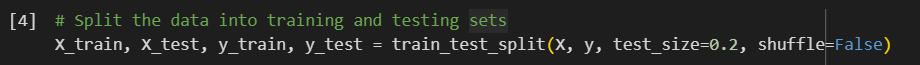
Normalize data dengan menggunakan MinMaxScaler()



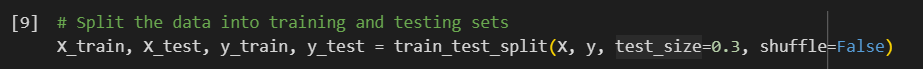
#### Splitting

Lakukan splitting data untuk training dan testing untuk ketiga scenario.

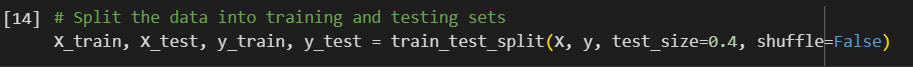
1. 80 - 20

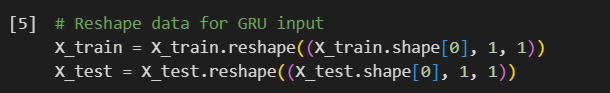


1. 70 - 30



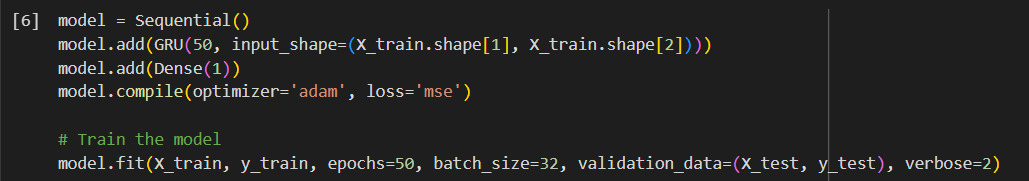
1. 60 - 40





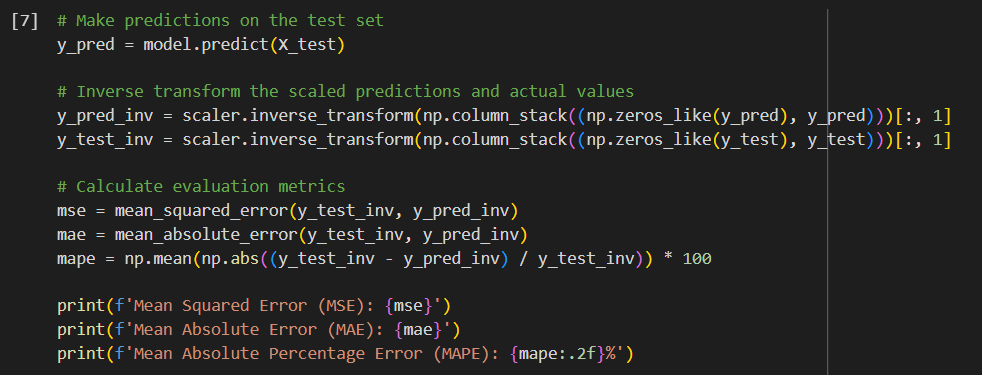
#### Model

Menggunakan model GRU, data diolah dan di training.



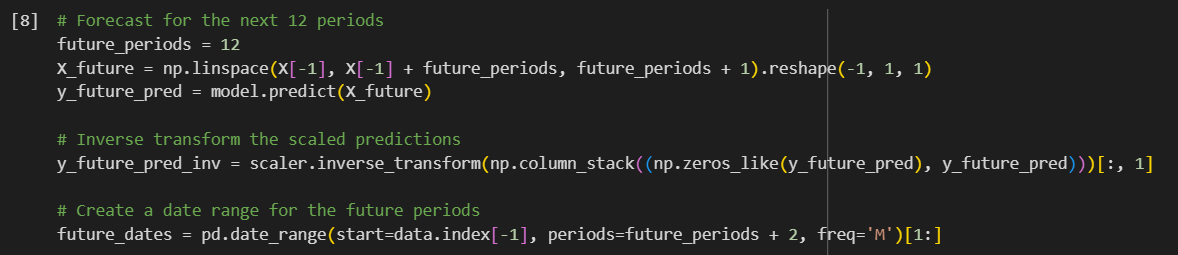
#### Prediksi

Model.fit yang sudah selesai kemudian dibuat prediction. Hasil prediksi kemudian di evaluasi (MSE, MAE, MAPE)



#### Peramalan

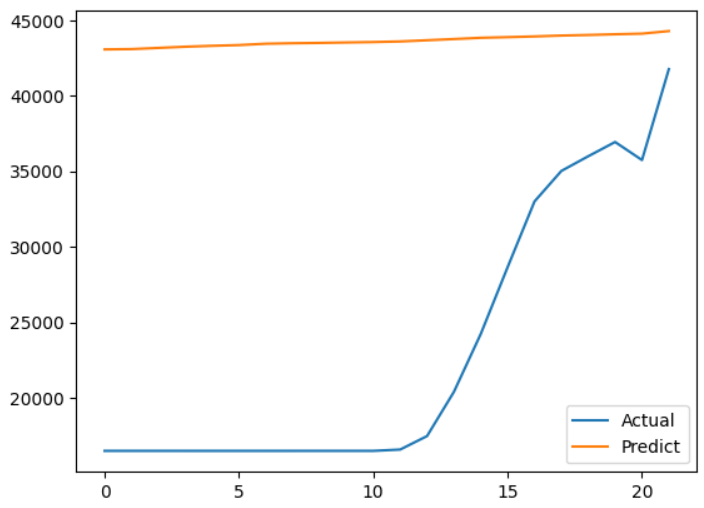
Dilakukan peramalan untuk 12 periode setelahnya.



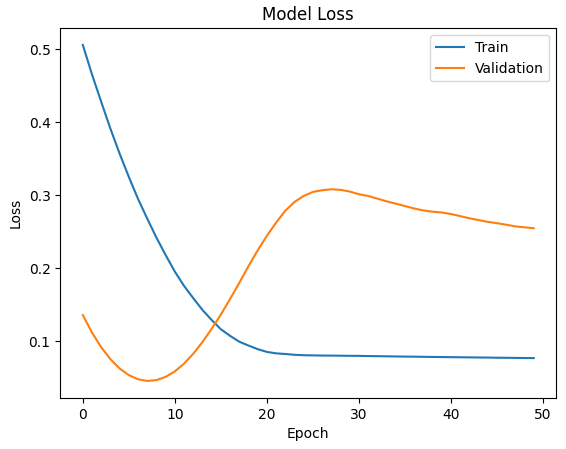
### Hasil Prediksi

Berikut merupakan hasil prediksi dari ketiga rasio

1. 80 - 20

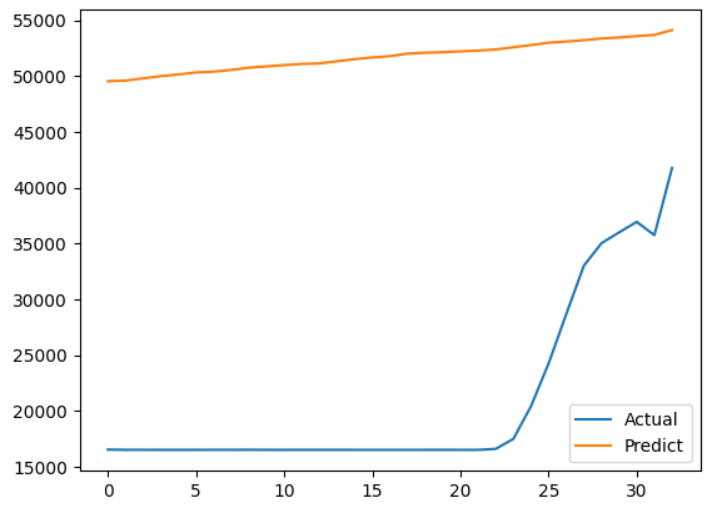


Hasil Prediksi

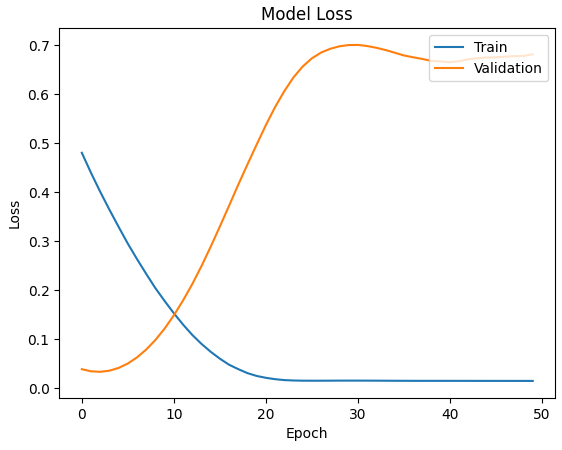


Model Loss

1. 70 - 30

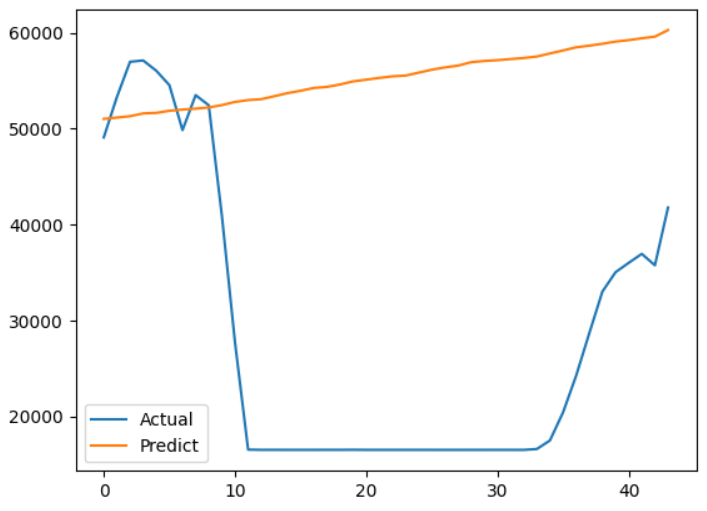


Model Predisksi

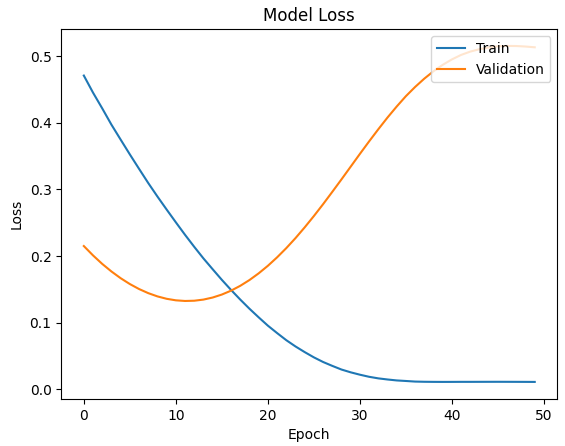


Model Loss

1. 60 - 40



Model Prediksi



Model Loss

### Evaluasi

Berikut merupakan hasil evaluasi dari ketiga rasio.

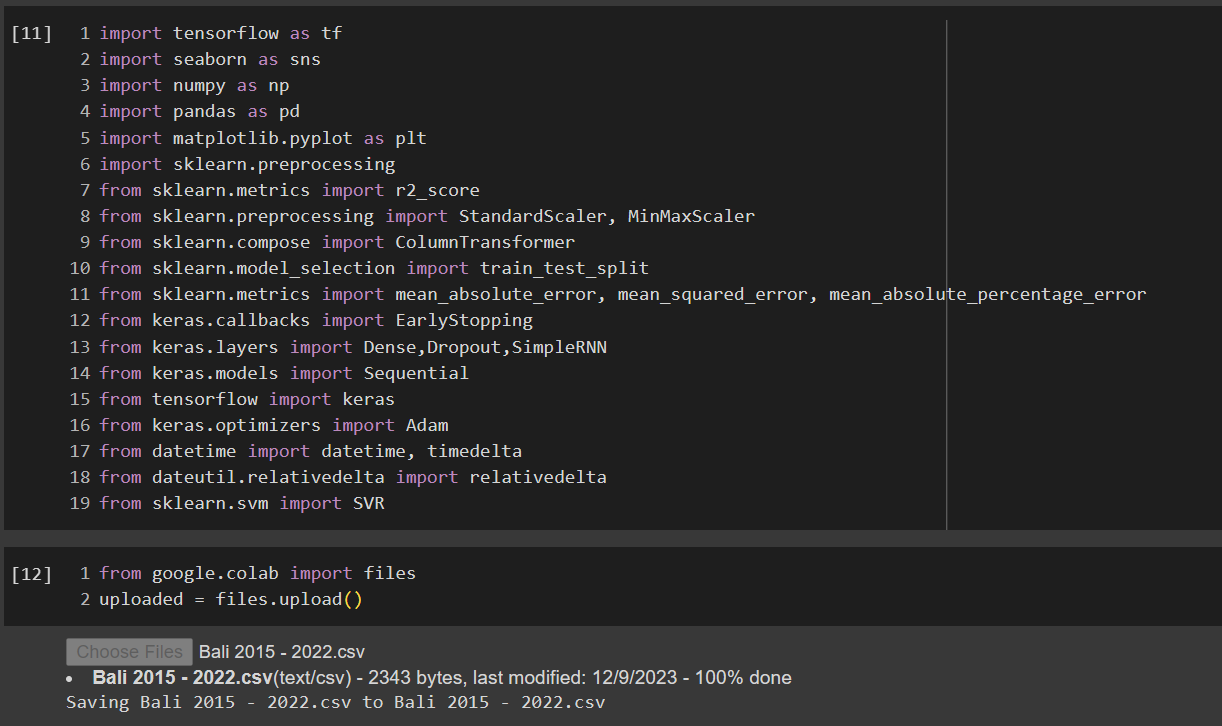
| Evaluasi/  *Split* | R² | MSE | MAPE | MAE |
| --- | --- | --- | --- | --- |
| 80-20 | 5.627 | 0.285 | 1.1312 | 0.493 |
| 70-30 | 18.431 | 0.714 | 1.9145 | 0.831 |
| 60-40 | 2.926 | 0.500 | 1.4134 | 0.630 |

## SVR

### Tahapan

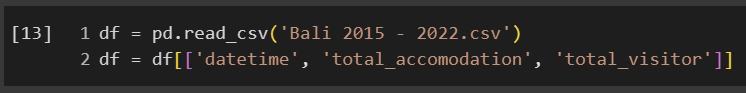
#### Import

Impor dataset dan *library* yang dibutuhkan

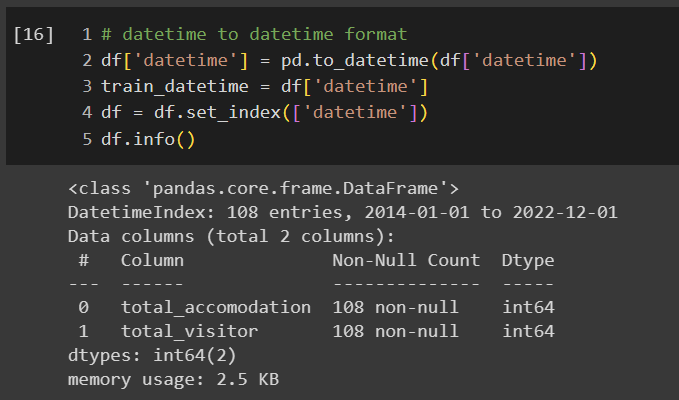


#### Pra-proses

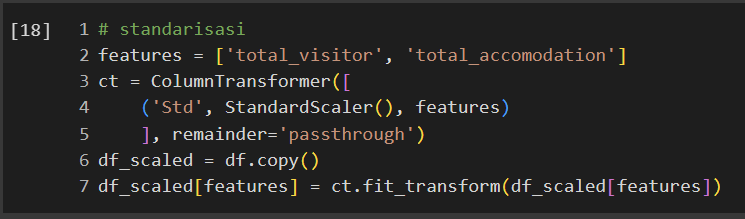
Read dataset menggunakan pd.read\_csv



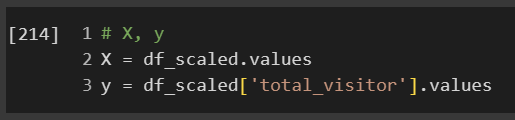
Ubah format kolom datetime menjadi format date dan jadikan datetime sebagai index



Lakukan standarisasi menggunakan StandardScaler() sehingga mean data adalah 0 dan standar deviasi nya 1.



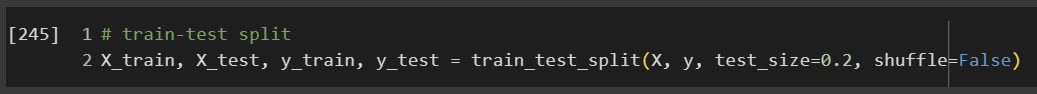
Selanjutnya bagi data menjadi X dan y dimana X adalah variabel independen yaitu total\_accomodation dan total\_visitor dan y adalah variabel dependen yaitu total\_visitor.



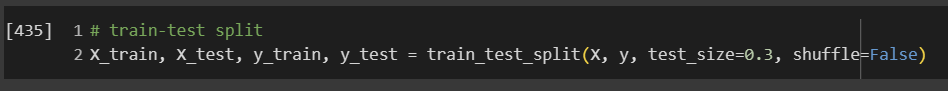
#### Splitting

Dilakukan *train-test-split* menggunakan train\_test\_split() untuk ketiga skenario, yaitu:

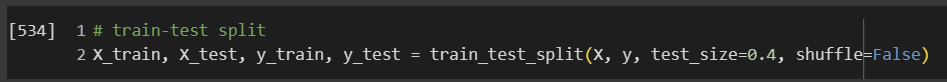
1. 80-20



1. 70-30



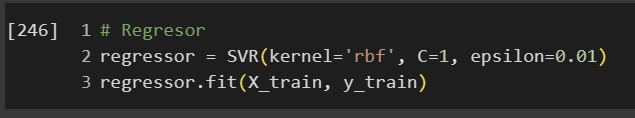
1. 60-40



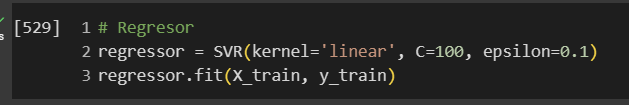
#### Model

Dibuat model menggunakan Support Vector Regression dengan parameter kernel dan nilai C dan epsilon yang optimal. Selanjutnya model dilatih pada data *training*.

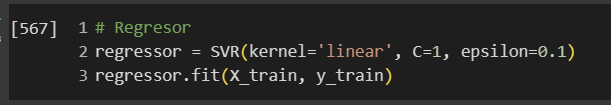
1. 80-20



1. 70-30



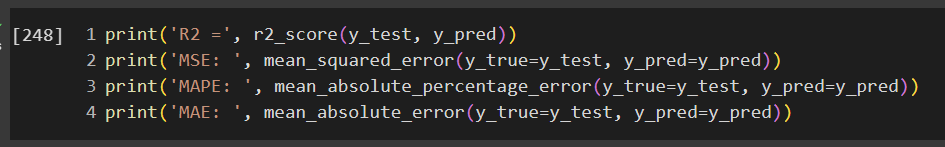
1. 60-40



#### Prediksi

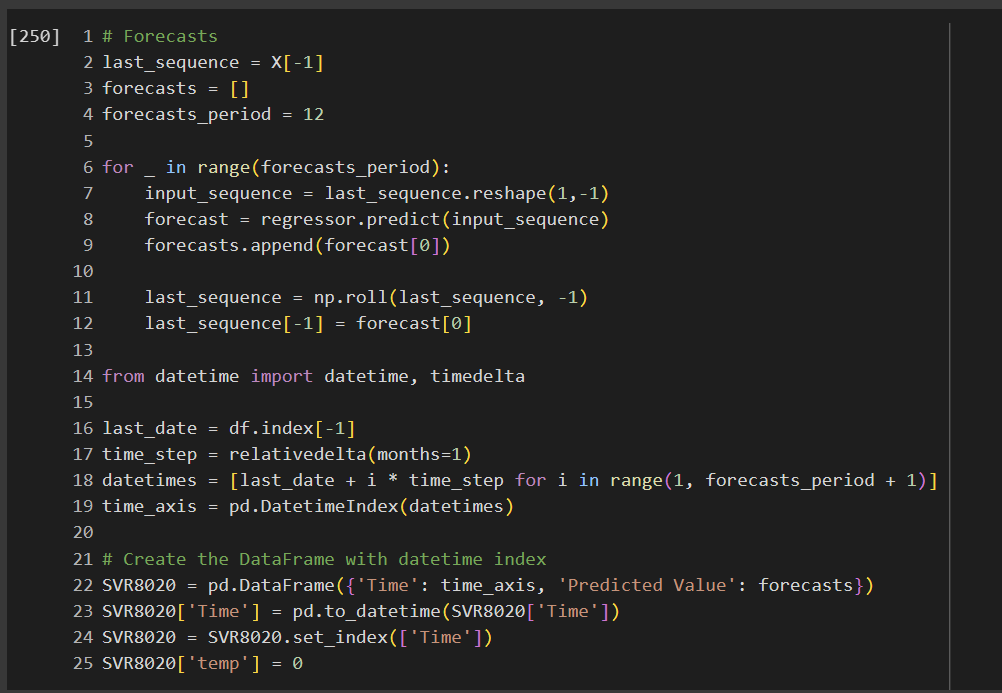
Menggunakan model yang sudah di-*fit*, dilakukan prediksi untuk data X\_test. Dari hasil prediksi dapat dievaluasi kinerja model menggunakan metrik yang sudah ditetapkan.





#### Peramalan

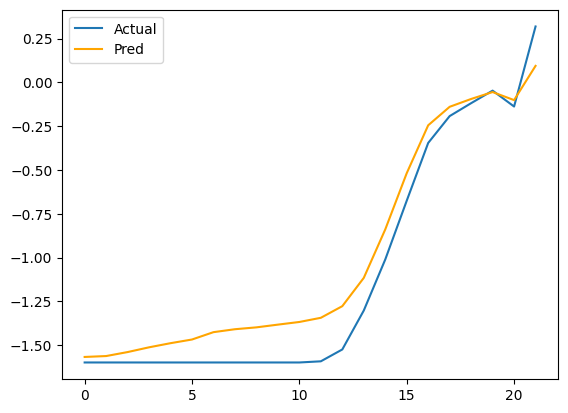
Setelah melakukan *parameter tuning* untuk model, dilakukan peramalan untuk 12 periode kedepan.



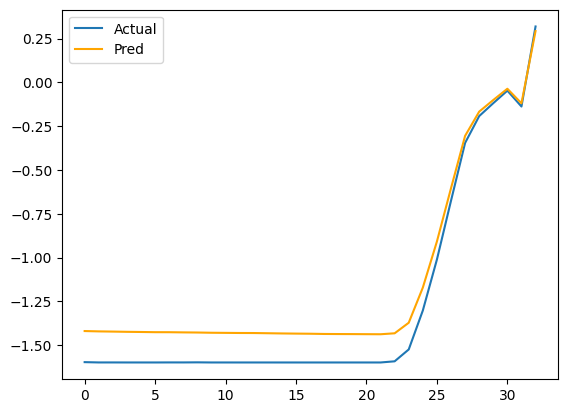
### Hasil Prediksi

Dari ketiga model RNN didapatkan hasil prediksi sebagai berikut:

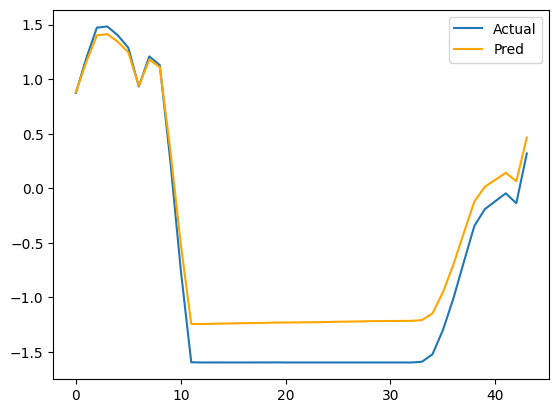
1. 80-20



1. 70-30



1. 60-40



### Evaluasi

Berikut adalah hasil evaluasi prediksi X\_test dari ketiga model:

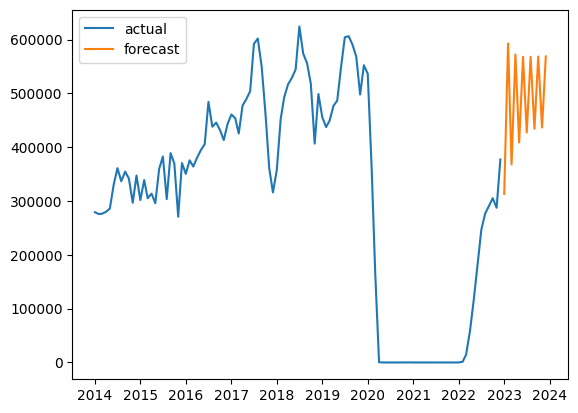
| Evaluasi/  *Split* | R² | MSE | MAPE | MAE |
| --- | --- | --- | --- | --- |
| 80-20 | 0.946 | 0.023 | 0.167 | 0.132 |
| 70-30 | 0.9386 | 0.021 | 0.111 | 0.135 |
| 60-40 | 0.931 | 0.087 | 0.381 | 0.263 |

# Hasil Forecast

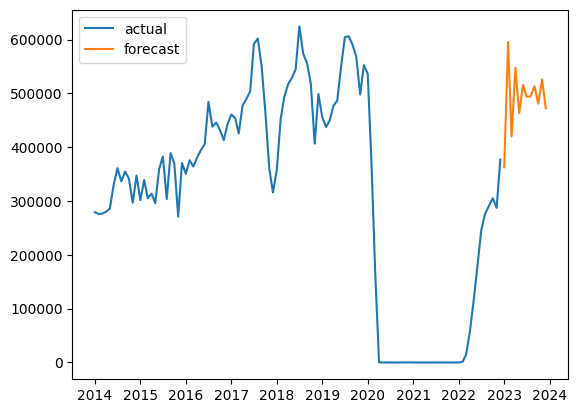
Dilakukan peramalan untuk 12 periode kedepan menggunakan tiap model yang telah dilatih dengan total\_visitor dan total\_accomodation sebagai variabel yang mempengaruhi. Berikut adalah hasil peramalan dengan tiap model:

## RNN

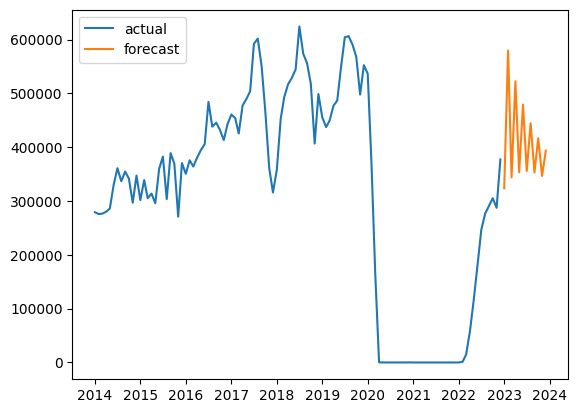
### 80-20



### 70-30

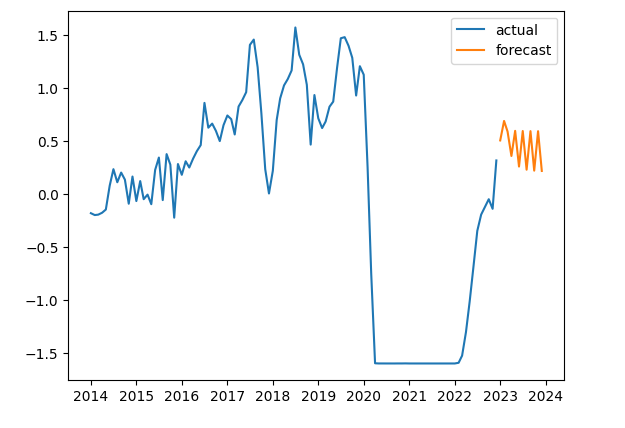


### 60-40

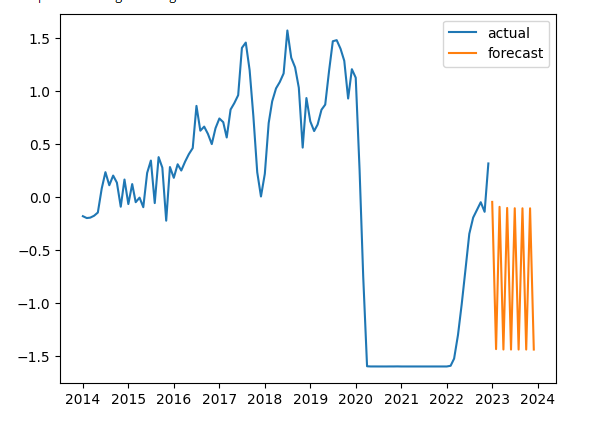


## LSTM

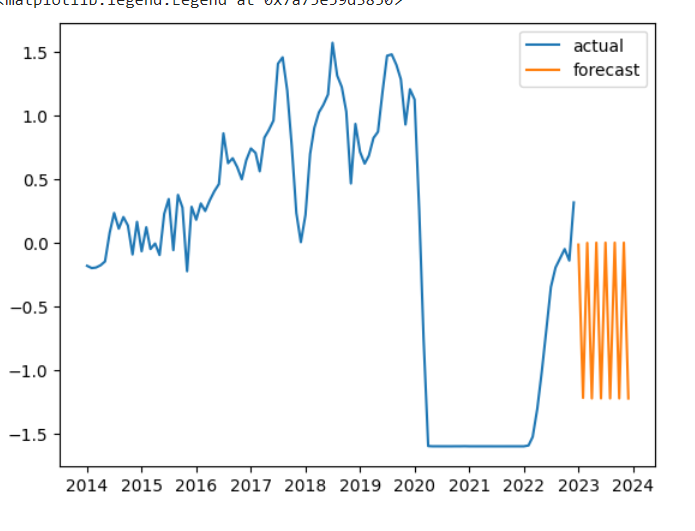
### 80-20



### 70-30

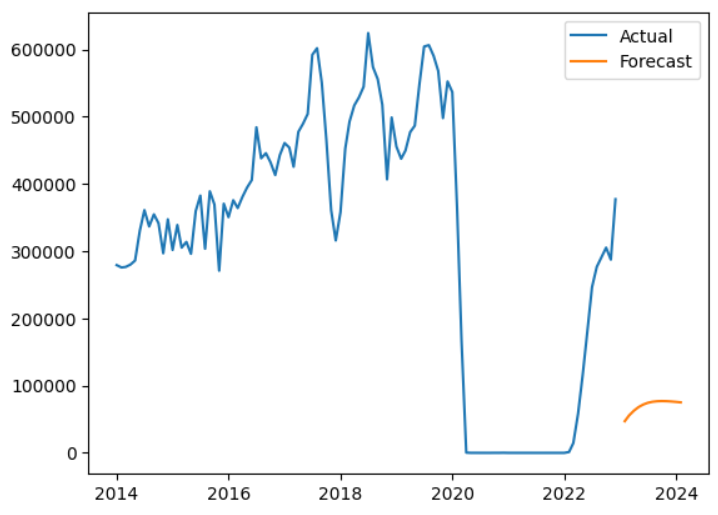


### 60-40

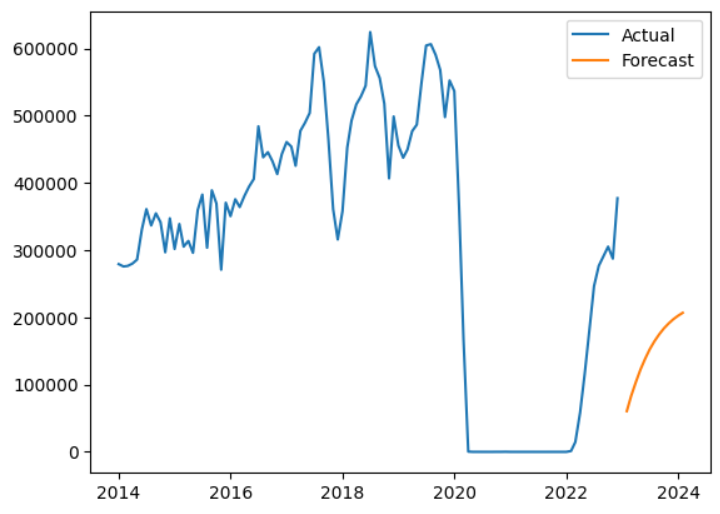


## GRU

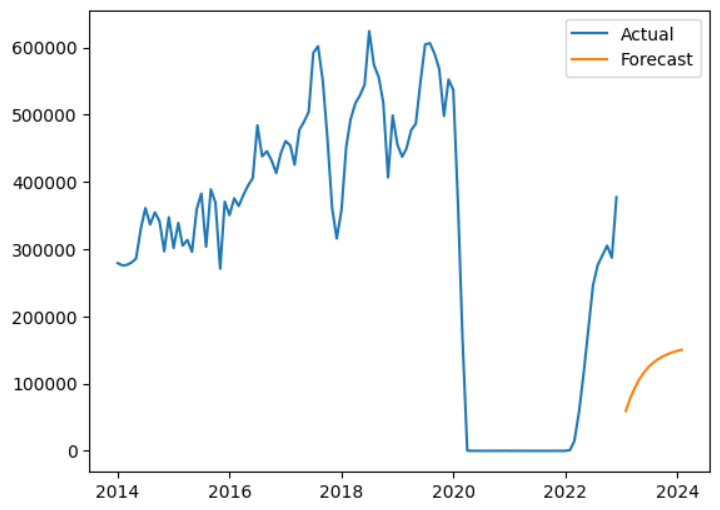
### 80-20



### 70-30

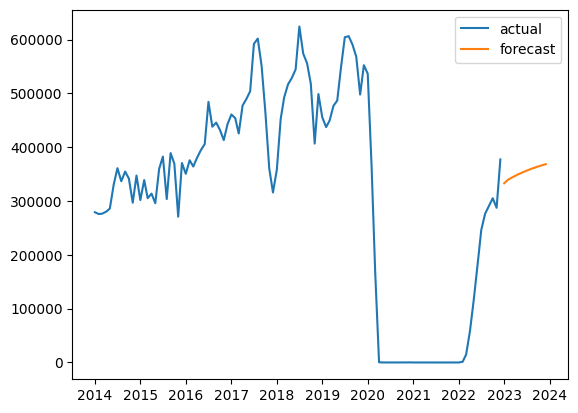


### 60-40

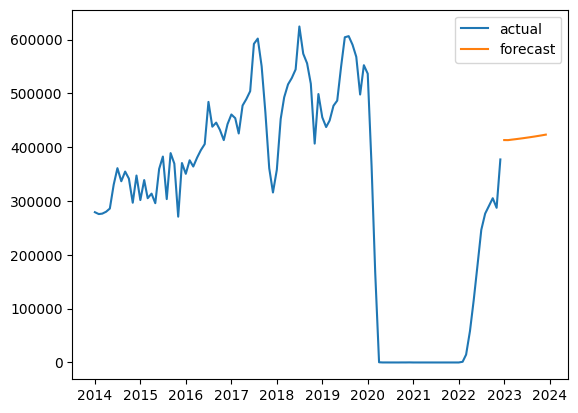


## SVR

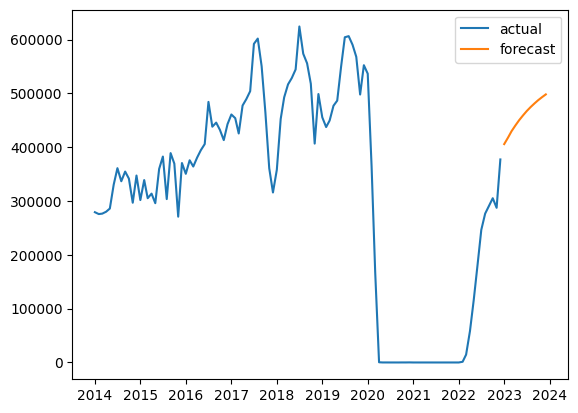
### 80-20



### 70-30



### 60-40



# Hasil analisis dan Kesimpulan

Secara umum hasil prediksi untuk tiap model pada tiap skenario memiliki performa yang kurang baik, hal ini dikarenakan jumlah data yang kurang dan terjadinya *covid* yang mempengaruhi data secara drastis. Model RNN memberikan hasil yang berbeda untuk tiap skenario *splitting*. Mengacu pada skor R² RNN memiliki performa terbaik pada *splitting* 80-20 diikuti oleh 60-40 lalu 70-30. RNN rentan mengalami *overfitting*, sehingga digunakan nilai *epoch* dan *early stopping*. Model RNN memiliki hasil prediksi dan *forecasting* yang lebih baik dengan jumlah layer serta unit pada model yang sedikit, selain itu RNN juga terlalu *overfit* ketika digunakan *activation function* yang tanh yang menyebabkan hasil peramalan berbentuk *zigzag*.

Dalam kasus model LSTM, hasil yang dihasilkan pada setiap skenario yang dilakukan memberikan nilai yang berbeda di tiap splittingnya. Sama seperti RNN, modeling yang dilakukan mengacu pada R² dimana pada kasus ini, dalam skenario 80:20 memiliki skor R² terbaik, diikuti dengan skenario 60:40 dan 70:30. Melalui percobaan ini, LSTM rentan mengalami *overfitting* pada hasilnya sehingga nilai epoch dan teknik early stopping diterapkan pada modelnya. Selain itu, Hasil analisis menunjukkan bahwa model LSTM cenderung memberikan hasil yang lebih baik ketika jumlah layer dan unit pada model sedikit. Ini akan menyebabkan kemampuan LSTM untuk menangkap dan memahami pola-pola data yang kompleks tanpa perlu struktur yang terlalu kompleks.

Model GRU menunjukkan hasil yang berbeda-beda pada tiap skenario splitting, dan performa terbaiknya terdapat pada skenario 80:20, diikuti oleh 60:40 dan 70:30, mengacu pada skor R² dan MAPE. GRU memiliki kecenderungan mengalami overfitting, sehingga diperlukan pengaturan nilai epoch.

Prediksi menggunakan model SVR memberikan nilai R² yang secara konsisten paling bagus. SVR juga sangat rentan untuk terjadi *overfitting*, hal ini menyebabkan hasil peramalan 12 periode kedepan menjadi tidak akurat. Pada model SVR digunakan nilai C yang tinggi dan epsilon yang rendah sehingga margin SVR kecil dan prediksi menjadi akurat.

Berikut adalah matriks error untuk tiap kombinasi model skenario:

1. RNN

| Evaluasi/  *Split* | R² | MSE | MAPE | MAE |
| --- | --- | --- | --- | --- |
| 80-20 | 0.873 | 0.057 | 0.357 | 0.221 |
| 70-30 | 0.590 | 0.146 | 0.579 | 0.356 |
| 60-40 | 0.828 | 0.205 | 0.530 | 0.405 |

1. LSTM

| Evaluasi/  *Split* | R² | MSE | MAPE | MAE |
| --- | --- | --- | --- | --- |
| 80-20 | 0.694 | 0.1414 | 1.11 | 0.338 |
| 70-30 | 0.081 | 0.344 | 0.972 | 0.448 |
| 60-40 | 0.146 | 0.841 | 1.07 | 0.723 |

1. GRU

| Evaluasi/  *Split* | R² | MSE | MAPE | MAE |
| --- | --- | --- | --- | --- |
| 80-20 |  | 0.285 | 1.1312 | 0.493 |
| 70-30 |  | 0.714 | 1.9145 | 0.831 |
| 60-40 |  | 0.500 | 1.4134 | 0.630 |

1. SVR

| Evaluasi/  *Split* | R² | MSE | MAPE | MAE |
| --- | --- | --- | --- | --- |
| 80-20 | 0.946 | 0.023 | 0.167 | 0.132 |
| 70-30 | 0.9386 | 0.021 | 0.111 | 0.135 |
| 60-40 | 0.931 | 0.087 | 0.381 | 0.263 |

Dari keempat hasil model di atas, model SVR memberikan nilai R² terbaik dibandingkan dengan model lainnya. Hal ini menunjukan SVR masih dapat bekerja dengan cukup baik ketika data berjumlah sedikit. Sementara itu, ketiga model *deep learning* memiliki kemampuan yang baik dalam menangkap pola jangka panjang. Dengan demikian pada ketiga model ini digunakan jumlah layer dan unit yang sedikit untuk mencegah *overfitting* dan membuat hasil peramalan lebih baik.