

## BAB IV PERANCANGAN DAN IMPLEMENTASI

### 4.1 Perancangan

Tahap perancangan ini bertujuan untuk memastikan model memprediksi beban dan suhu transformator berjalan dengan baik. Pada tahap ini, dilakukan perancangan model *Long Short Term Memory* (LSTM) untuk memprediksi sisa umur transformator Gardu Induk Cianjur. Tahap perancangan model LSTM mencakup pengumpulan data beban dan suhu pada transformator Gardu Induk Cianjur, perhitungan sisa umur transformator, *preprocessing* data, pemodelan LSTM, prediksi sisa umur, kemudian evaluasi model dengan metrik evaluasi.

#### 4.1.1 Pengumpulan Data Beban dan Suhu Transformator

Pada penelitian ini data yang digunakan merupakan data beban (MW) dan data suhu atau temperatur (°C) harian transformator di Gardu Induk 150 kV Cianjur. Transformator yang digunakan merupakan transformator 2 dan transformator 5 Gardu Induk Cianjur. Spesifikasi transformator 2 dan transformator 5 Gardu Induk Cianjur tertera pada Tabel 4.1.

Tabel 4. 1 Spesifikasi Trafo 2 Gardu Induk Cianjur

Spesifikasi	Transformator 2	Transformator 5
Merek	GEC Alsthom	UNINDO – Alsthom
Serial	31G3929/6	P060LEC757-34
Status Alat	Operasi	Operasi
Daya Transformator	60 MW	60 MW
Tegangan Operasi	150/20 kV	150/20 kV
Suhu Ligkungan	30 °C	30 °C
Kenaikan Suhu Top Oil	53 °C	50 °C
Tahun Pembuatan	1995	2013
Tanggal Operasi	22 November 2001	19 Februari 2016
Pendingin	ONAN/ONAF	ONAN/ONAF

Transformator 2 dan transformator 5 mempunyai spesifikasi yang hampir serupa, mulai dari daya transformator, tegangan operasi, suhu lingkungan dan pendingin yang digunakan. Namun, kedua transformator mempunyai perbedaan tahun pembuatan dan tahun operasional, transformator 2 beroperasi pada tahun 2001 sedangkan transformator 5 beroperasi pada tahun 2016.

Sumber data pada penelitian ini merupakan data yang diperoleh secara langsung melalui hasil pengukuran dan pengamatan di lapangan oleh pihak Gardu Induk Cianjur. Data beban dan suhu transformator yang digunakan sebagai data latih dan data uji pada model LSTM merupakan data dari transformator 2 dan transformator 5. Data beban dan suhu transformator 2 merupakan data harian transformator pada dari 1 Januari 2019 hingga 16 Juli 2024 sebanyak 2012 data. Sedangkan, data beban dan suhu transformator 5 merupakan data harian transformator dari 1 Januari 2019 hingga 16 Juli 2024 sebanyak 2018 data. Data beban dan suhu transformator 2 tertera pada Tabel 4.2 dan data beban dan suhu transformator 5 tertera pada tabel 4.3.

Tabel 4. 2 Data Beban dan Suhu Transformator 2

<b>Tanggal</b>	<b>Beban (MW)</b>	<b>Suhu Minyak (°C)</b>
1/1/2019	28.41	56
1/2/2019	27.44	56
1/3/2019	26.43	56
1/4/2019	26.75	56
1/5/2019	26.5	56
...	...	...
7/12/2024	19.27	50
7/13/2024	19.72	50
7/14/2024	18.79	46
7/15/2024	18.61	47
7/16/2024	18.96	48

Tabel 4. 3 Data Beban dan Suhu Transformator 5

<b>Tanggal</b>	<b>Beban (MW)</b>	<b>Suhu Minyak (°C)</b>
1/1/2019	32.66	65
1/2/2019	33.67	65
1/3/2019	34.18	65
1/4/2019	34.98	69
1/5/2019	35.74	68
...	...	...
7/12/2024	27.85	62
7/13/2024	27.47	60
7/14/2024	26.23	60
7/15/2024	27.78	60
7/16/2024	28.03	60

#### 4.1.2 Perhitungan Sisa Umur Transformator

Salah satu parameter yang diperlukan untuk menghitung sisa umur transformator adalah data pembebanan. Tahapan dalam menghitung pembebanan adalah menggunakan persamaan untuk menghitung rata-rata beban dalam satu tahun dengan satuan MVA, menghitung kapasitas presentase pembebanan yang digunakan, dan menghitung rasio pembebanan. Data beban (MW) yang digunakan pada transformator 2 dan transformator 5 tertera pada Tabel 4.4.

Tabel 4. 4 Data rata-rata beban (MW) transformator Gardu Induk Cianjur

<b>Tahun</b>	<b>Rata-Rata Beban Transformator 2 (MW)</b>	<b>Rata-Rata Beban Transformator 5 (MW)</b>
2019	23.75	37.84
2020	26.31	34.12
2021	24.91	38.11
2022	27.5	33.34
2023	22.33	27.4
2024	20.57	27.6

Tahapan yang dilakukan untuk menghitung sisa umur transformator setelah data pembebanan didapatkan meliputi menghitung suhu *top oil*, menghitung selisih antara suhu *hotspot* dan suhu *top oil*, menghitung suhu *hotspot*, menghitung laju penuaan (V), menghitung susut umur transformator (L), dan menghitung sisa umur transformator. Hasil perhitungan sisa umur transformator 2 tertera pada Tabel 4.5, dan hasil perhitungan sisa umur transformator 5 tertera pada Tabel 4.6. Salah satu contoh perhitungan sisa umur setiap transformator disertakan setelah tabel tersebut, dan perhitungan lengkapnya disajikan di Lampiran.

Tabel 4. 5 Hasil perhitungan sisa umur transformator 2

Tahun	Pembebanan (%)	Suhu <i>Hotspot</i> (°C)	Susut Umur (p.u)	Sisa Umur (Tahun)
2019	46.57	58.21	0.0101	11
2020	51.59	61.27	0.0144	10
2021	48.84	59.56	0.0118	9
2022	53.92	62.81	0.0171	8
2023	43.78	56.66	0.0084	7
2024	40.33	54.89	0.0069	6

Contoh perhitungan tahun 2019

- a. Rata-rata beban dalam satu tahun (MVA)

$$\begin{aligned}
 MVA &= \frac{MW}{\cos\phi} \\
 &= \frac{23.75}{0.85} \\
 &= 27.94 \text{ MVA}
 \end{aligned}$$

- b. Kapasitas pembebanan

$$\begin{aligned}
 \text{Kapasitas} &= \frac{\text{Kapasitas Terpakai}}{\text{Kapasitas Maksimal}} \times 100\% \\
 &= \frac{27.94}{60} \times 100\%
 \end{aligned}$$

$$= 46.57\%$$

c. Rasio Pembebanan

$$\begin{aligned} K &= \frac{S}{Sr} \\ &= \frac{46.57}{100} \\ &= 0.47 \end{aligned}$$

d. Kenaikan temperature hotspot

$$\begin{aligned} \Delta\theta_{hr} &= \Delta\theta_b + 1.3 \Delta\theta_{w0} \\ &= 53 + 26 \\ &= 79^\circ\text{C} \end{aligned}$$

e. Perbandingan rugi

$$\begin{aligned} d &= \frac{\text{Rugi Tembaga Daya Pengenal}}{\text{Rugi Beban Nol}} \\ &= \frac{140kW}{38kW} \\ &= 3.68kW \end{aligned}$$

f. Kenaikan *temperature top oil*

$$\begin{aligned} \Delta\theta_b &= \Delta\theta_{or} \left( \frac{1 + dK^2}{1 + d} \right)^x \\ &= 53 \left( \frac{1 + 3.68 \times 0.47^2}{1 + 3.68} \right)^{0.8} \\ &= 24.65^\circ\text{C} \end{aligned}$$

g. Selisih antara kenaikan *temperature hotspot* dan kenaikan *temperature top oil*

$$\begin{aligned} \Delta\theta_{td} &= (\Delta\theta_{hr} - \Delta\theta_{or})K^{2y} \\ &= (79 - 53)0.47^{2 \times 1.3} \\ &= 3.56^\circ\text{C} \end{aligned}$$

h. *Temperature hostpot*

$$\begin{aligned} \theta_h &= \theta_a + \Delta\theta_b + \Delta\theta_{td} \\ &= 30 + 24.65 + 3.56 \\ &= 58.21^\circ\text{C} \end{aligned}$$

i. Laju penuaan *thermal* ( $V$ )

$$\begin{aligned} V &= 10^{(\theta h - 98)/19.93} \\ &= 10^{(58.21 - 98)/19.93} \\ &= 0.0101 \text{ p.u} \end{aligned}$$

j. Susut Umur

$$\begin{aligned} L &= \frac{1}{N} \sum_n^N x V \\ &= \frac{1}{365} (365 \times 0.0101) \\ &= 0.0101 \end{aligned}$$

k. Sisa Umur

$$\begin{aligned} \text{Sisa umur} &= \frac{8760 - (L \times 365)}{8760} \times (30 - \text{tahun operasi}) \\ &= \frac{8760 - (0.0101 \times 365)}{8760} \times (30 - 19) \\ &= 11 \text{ Tahun} \end{aligned}$$

Tabel 4. 6 Hasil perhitungan sisa umur transformator 5

Tahun	Pembebanan (%)	Suhu <i>Hotspot</i> (°C)	Susut Umur (p.u)	Sisa Umur (Tahun)
2019	74.2	77.24	0.0908	26
2020	66.9	70.83	0.0433	25
2021	74.73	77.73	0.0961	24
2022	65.37	69.58	0.0375	23
2023	53.73	61.12	0.0141	22
2024	54.12	61.37	0.0145	21

Contoh perhitungan tahun 2019 transformator 5

a. Rata-rata beban dalam satu tahun (MVA)

$$\begin{aligned} MVA &= \frac{MW}{\cos\phi} \\ &= \frac{37.84}{0.85} \end{aligned}$$

$$= 44.52 \text{ MVA}$$

b. Kapasitas pembebanan

$$\begin{aligned} \text{Kapasitas} &= \frac{\text{Kapasitas Terpakai}}{\text{Kapasitas Maksimal}} \times 100\% \\ &= \frac{44.52}{60} \times 100\% \\ &= 74.20\% \end{aligned}$$

c. Rasio pembebanan

$$\begin{aligned} K &= \frac{S}{Sr} \\ &= \frac{74.20}{100} \\ &= 0.74 \end{aligned}$$

d. Kenaikan *temperature hotspot*

$$\begin{aligned} \Delta\theta_{hr} &= \Delta\theta_{or} + 1.3 \Delta\theta_{w0} \\ &= 50 + 26 \\ &= 76^\circ\text{C} \end{aligned}$$

e. Perbandingan rugi

$$\begin{aligned} d &= \frac{\text{Rugi Tembaga Daya Pengenal}}{\text{Rugi Beban Nol}} \\ &= \frac{140kW}{38kW} \\ &= 3.68kW \end{aligned}$$

f. Kenaikan *temperatur top oil*

$$\begin{aligned} \Delta\theta_b &= \Delta\theta_{or} \left( \frac{1 + dK^2}{1 + d} \right)^x \\ &= 50 \left( \frac{1 + 3.68 \times 0.74^2}{1 + 3.68} \right)^{0.8} \\ &= 35.27^\circ\text{C} \end{aligned}$$

g. Perbandingan antara kenaikan *temperature hotspot* dan kenaikan *temperature top oil*

$$\begin{aligned} \Delta\theta_{td} &= (\Delta\theta_{hr} - \Delta\theta_{or}) K^{2y} \\ &= (76 - 50) 0.74^{2 \times 1.3} \end{aligned}$$

$$= 11.97^{\circ}\text{C}$$

h. *Temperature hotspot*

$$\begin{aligned}\theta_h &= \theta_a + \Delta\theta_b + \Delta\theta_{td} \\ &= 30 + 35.27 + 11.97 \\ &= 77.24^{\circ}\text{C}\end{aligned}$$

i. Laju penuaan *thermal* (V)

$$\begin{aligned}V &= 10^{(\theta_h - 98)/19.93} \\ &= 10^{(77.24 - 98)/19.93} \\ &= 0.0908 \text{ p.u}\end{aligned}$$

j. Susut umur

$$\begin{aligned}L &= \frac{1}{N} \sum_n^N x V \\ &= \frac{1}{365} (365 \times 0.0908) \\ &= 0.0908\end{aligned}$$

k. Sisa umur

$$\begin{aligned}\text{Sisa umur} &= \frac{8760 - (L \times 365)}{8760} \times (30 - \text{tahun operasi}) \\ &= \frac{8760 - (0.0908 \times 365)}{8760} \times (30 - 4) \\ &= 26 \text{ Tahun}\end{aligned}$$

#### 4.1.3 Penentuan Dataset

Dataset yang digunakan dalam penelitian ini terdiri dari data dengan empat kolom, yaitu tanggal, beban (MW), suhu minyak, dan sisa umur. Rentang data yang digunakan dimulai dari 1 Januari 2019 hingga 16 Juli 2024. Kolom sisa umur berfungsi sebagai target prediksi dalam model ini yang didapat dari hasil perhitungan. Dengan menggunakan dataset ini, model prediksi akan dibangun untuk memperkirakan sisa umur transformator berdasarkan pola beban dan suhu minyak.

#### 4.1.4 Preprocessing Data

*Preprocessing* data merupakan langkah awal yang penting digunakan untuk mempersiapkan data sebelum digunakan untuk melatih model. Tujuan utama dari *preprocessing* data adalah untuk meningkatkan kualitas data agar model dapat belajar dan membuat prediksi yang akurat. Langkah-langkah yang dilakukan pada saat *preprocessing* data antara lain pengubahan kolom tanggal menjadi tipe *datetime* untuk memastikan data diurutkan berdasarkan waktu. Selain itu, kolom yang akan digunakan untuk prediksi, yaitu kolom beban, suhu minyak dan sisa umur. Selanjutnya, data dinormalisasi menggunakan '*MinMaxScaler*' dari *scikit-learn*, yang mengubah nilai-nilai dari kolom beban, suhu minyak dan sisa umur ke dalam rentang 0 hingga 1. Normalisasi ini membantu mempercepat model dalam proses belajar dan mencegah bias yang disebabkan oleh skala yang berbeda antara fitur-fitur.

Setelah data dinormalisasi, data kemudian dibagi menjadi dua bagian, yaitu data *training* dan *testing*, dengan rasio 80:20. Pembagian ini dilakukan untuk memastikan bahwa model LSTM dapat melakukan prediksi dengan baik pada data yang belum pernah dilihat sebelumnya. Data *training* digunakan untuk melatih model, sementara data pengujian digunakan untuk mengevaluasi performa model. Evaluasi performa model dilakukan dengan menggunakan *Mean Square Error* (MSE), yang memberikan indikasi seberapa baik model memprediksi nilai sebenarnya. Pustaka *Keras* digunakan untuk membangun dan melatih model LSTM, dengan menggunakan *Sequential*, *Dense*, dan *LSTM*.

#### 4.1.5 Model LSTM

Dalam konstruksi model LSTM pada penelitian ini, dua jenis lapisan digunakan yaitu lapisan LSTM dan lapisan *Dense* dengan menggunakan *library* Keras. Pada model ini menggunakan satu lapisan LSTM. Setelah lapisan LSTM, model menambahkan satu lapisan *Dense* dengan 1 unit *neuron*. Model ini kemudian diatur menggunakan fungsi *loss Mean Squared Error* (MSE) dan *optimizer Adam*. Parameter yang digunakan dalam model ini meliputi *learning rate*, *step size*, *epoch*, dan *batch size*.

LSTM mempunyai diagram arsitektur seperti pada Gambar 2.2. Langkah pertama dalam arsitektur model adalah lapisan *input* yang menerima data dari dataset dan memformatnya agar sesuai untuk diproses oleh lapisan LSTM. Data input diubah menjadi urutan dengan langkah waktu tertentu (*time steps*) dengan tiga fitur, yaitu data beban, suhu minyak, dan sisa umur. Selanjutnya, lapisan LSTM memproses urutan input tersebut. LSTM dirancang untuk mengatasi masalah *long-term dependencies* dengan menggunakan sel memori, pintu masukan (*input gate*), pintu keluaran (*output gate*), dan pintu penghapusan (*forget gate*). Setelah itu, proses *training* model dilakukan hingga model mencapai nilai kesalahan yang terkecil atau mencapai jumlah maksimum iterasi yang telah ditentukan. Setelah *training* model, model dievaluasi menggunakan data *testing* untuk mengukur kinerjanya. Model yang telah dilatih digunakan untuk melakukan prediksi pada data baru kemudian hasil prediksi di denormalisasi ke skala aslinya. Berikut contoh perhitungan LSTM, *weight* (bobot) dan bias ditentukan secara acak dari program.

Diketahui :

Data Input ( $x_i$ ) yang Dinormalisasi:

- a. Load: 0.472327
- b. Suhu Minyak: 0.470588
- c. Sisa Umur: 1.0

Bobot dan Bias yang Digunakan:

- a. *Forget Gate*

Bobot:  $[-0.0271844, -0.14340611, 0.02547715]$

Bias: 0.014475249

- b. *Input Gate*

Bobot:  $[0.10250288, -0.03319829, 0.12237673]$

Bias: 1.0133802

- c. *Calon Cell State*

Bobot:  $[-0.05838871, -0.04251594, 0.10412718]$

Bias: 0.01491982

- d. *Output Gate*

Bobot:  $[0.08530803, -0.15841146, 0.04283256]$

Bias: 0.011117922

Untuk  $timesteps=1$

Diketahui  $h_{t-1} = 0, c_{t-1} = 0$

a. *Forget gate*

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

$$f_1 = \sigma((-0.0271844 \cdot 0.472327) + (-0.14340611 \cdot 0.470588) \\ + (0.02547715 \cdot 1.0) + (0 \cdot 0) + 0.014475249)$$

$$f_1 = \sigma(-0.05486022 + 0.014475249)$$

$$f_1 = \sigma(-0.04038497)$$

$$f_1 = 0.489904$$

b. *Input gate*

$$i_t = \sigma(W_i \cdot [h_t = 1, x_t] + b_i)$$

$$i_1 = \sigma((0.10250288 \cdot 0.472327) + (-0.03319829 \cdot 0.470588) \\ + (0.12237673 \cdot 1.0) + (0 \cdot 0) + 1.0133802)$$

$$i_1 = \sigma(0.15517878 + 1.0133802)$$

$$i_1 = \sigma(1.16855898)$$

$$i_1 = 0.762883$$

c. *Calon Cell State*

$$c_t \sim = \tanh(W_c \cdot [h_{t-1}, x_t] + bc)$$

$$c_1 \sim = \tanh((-0.05838871 \cdot 0.472327) + (-0.04251594 \cdot 0.470588) \\ + (0.10412718 \cdot 1.0) + (0 \cdot 0) + 0.01491982)$$

$$c_1 \sim = \tanh(0.05655728 + 0.01491982)$$

$$c_1 \sim = \tanh(0.0714771)$$

$$c_1 \sim = 0.071339$$

d. *Cell State*

$$c_t = f_i \cdot c_{t-1} + i_t \cdot c_t \sim$$

$$c_1 = 0.489904 \cdot 0 + 0.762883 \cdot 0.071339$$

$$c_1 = 0.054424$$

e. *Output Gate*

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

$$O_1 = \sigma((0.08530803 \cdot 0.472327) + (-0.15841146 \cdot 0.470588) + (0.04283256 \cdot 1.0) + (0 \cdot 0) + 0.011117922)$$

$$O_1 = \sigma(0.019693342)$$

$$O_1 = 0.504923$$

f. *Update Hidden State*

$$h_t = o_t \cdot \tanh(C_t)$$

$$h_1 = 0.504923 \cdot \tanh(0.054424)$$

$$h_1 = 0.504923 \cdot 0.054375$$

$$h_1 = 0.027428$$

Nilai hasil *output*  $h_1$  dan *cell state*  $c_1$  akan digunakan pada langkah waktu (*iimesteps*) berikutnya. Langkah-langkah tersebut diulangi untuk setiap langkah waktu dengan memperbarui nilai *output*  $h_t$  dan *cell state*  $c_t$ . Proses perhitungan akan berlanjut sampai model yang sesuai dengan parameter yang telah ditentukan tercapai. Setelah itu, dilakukan proses *testing* untuk mendapatkan nilai prediksi.

## 4.2 Implementasi

Implementasi pada penelitian ini dilakukan untuk mewujudkan perancangan model LSTM dalam melakukan prediksi sisa umur transformator berdasarkan data beban dan suhu minyak. Implementasi model LSTM dilakukan menggunakan platform Google Colab dengan Bahasa pemrograman *Python*. Berikut kode-code yang digunakan.

---

### Program 1 *Import Library*

---

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from keras.models import Sequential
from keras.layers import Dense, LSTM
```

---

---

### Program 2 *Input* Dataset Transformator GI Cianjur

---

```
# Import dataset
dataset = pd.read_csv('Trafo 5 - Cianjur.csv')
dataset['Date'] = pd.to_datetime(dataset['Date'])
dataset = dataset.sort_values(by='Date')
dataset = dataset.set_index('Date')

# Menentukan kolom yang digunakan untuk prediksi
load_series = dataset['Load']
suhu_series = dataset['Suhu Minyak']
sisa_umur_series = dataset['Sisa Umur']
```

---

### Program 4 Normalisasi Data

---

```
# Normalisasi data
scaler_load = MinMaxScaler(feature_range=(0, 1))
load_series = scaler_load.fit_transform(load_series.values.reshape(-1, 1))
scaler_suhu = MinMaxScaler(feature_range=(0, 1))
suhu_series = scaler_suhu.fit_transform(suhu_series.values.reshape(-1, 1))
scaler_sisa_umur = MinMaxScaler(feature_range=(0, 1))
sisa_umur_series = scaler_sisa_umur.fit_transform(sisa_umur_series.values.reshape(-1, 1))
```

---

### Program 4 Membagi training dan testing data dan mengatur time step

---

```
# Membagi data train dan test dengan rasio 80:20
train_size = int(len(dataset_scaled) * 0.80)
test_size = len(dataset_scaled) - train_size
train_data, test_data = dataset_scaled[0:train_size],
dataset_scaled[train_size:len(dataset_scaled)]
```

---

---

### Program 5 Melatih Model

---

```
# Membuat dan melatih model LSTM
model = Sequential()
model.add(LSTM(NEURON, activation='relu', input_shape=(step_size, 3)))
model.add(Dense(1))
optimizer = Adam(learning_rate=LEARNING_RATE)
model.compile(optimizer=optimizer, loss='mse')
history = model.fit(x_train, y_train, epochs=EPOCHS,
batch_size=BATCH_SIZE, verbose=VERBOSE, validation_data=(x_test,
y_test))
```

---

### Program 6 Prediksi

---

```
# Prediksi
trainPredict = model.predict(x_train)
testPredict = model.predict(x_test)
# Inverse transform predictions
trainPredict = scaler_sisa_umur.inverse_transform(trainPredict)
y_train = scaler_sisa_umur.inverse_transform(y_train.reshape(-1, 1))
testPredict = scaler_sisa_umur.inverse_transform(testPredict)
y_test = scaler_sisa_umur.inverse_transform(y_test.reshape(-1, 1))
```

---

### Program 7 Denormalisasi Data

---

```
Inverse transform predictions
pred_sisa_umur = scaler_sisa_umur.inverse_transform(predictions)
```

---

### Program 8 Plotting Prediksi 2024

---

```
# Plot Sisa Umur predictions
plt.figure(figsize=(15, 5))
plt.plot(dataset.index,
scaler_sisa_umur.inverse_transform(sisa_umur_series), 'g', label='Sisa
Umur Aktual')
```

---

---

```

plt.plot(future_predictions.index,
future_predictions['Sisa_Umur_Prediction'], 'r', label='Prediksi Sisa Umur
Hingga 2025')
plt.legend(loc='upper right')
plt.xlabel('Tahun')
plt.ylabel('Sisa Umur')
plt.show()

```

---

### Program 10 Ploting Training dan Testing Data

---

```

# Plotting results with training and testing predictions
plt.figure(figsize=(15, 5))
plt.plot(dataset.index,
scaler_sisa_umur.inverse_transform(sisa_umur_series), 'g', label='Sisa
Umur Aktual')
# Menambahkan prediksi training ke plot
trainPredictPlot = np.empty_like(sisa_umur_series)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[step_size:len(trainPredict) + step_size, :] = trainPredict
# Menambahkan prediksi testing ke plot
testPredictPlot = np.empty_like(sisa_umur_series)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(trainPredict) + (step_size * 2) + 1:len(sisa_umur_series)
- 1, :] = testPredict
plt.plot(dataset.index[:len(trainPredictPlot)], trainPredictPlot, 'r',
label='Training Prediksi Sisa Umur')
plt.plot(dataset.index[:len(testPredictPlot)], testPredictPlot, 'b',
label='Testing Prediksi Sisa Umur')
plt.legend(loc='upper right')
plt.xlabel('Tahun')
plt.ylabel('Sisa Umur')

```

---

---

```
plt.show()
```

---

---

### Program 11 Evaluasi Model LSTM

---

```
# Compute and display MSE and RMSE for training and testing data
train_mse = mean_squared_error(y_train, trainPredict)
test_mse = mean_squared_error(y_test, testPredict)
print(f'Train MSE (Sisa Umur): {train_mse:.4f}')
print(f'Test MSE (Sisa Umur): {test_mse:.4f}')
```

---

Program 1 merupakan tahap pertama dengan memasukkan berbagai *library* atau pustaka yang diperlukan untuk pemrosesan data, seperti *pandas* untuk membaca dan mengatur data, *numpy* untuk komputasi numerik, *matplotlib* untuk visualisasi data, *sklearn* untuk normalisasi dan evaluasi model, serta *keras* untuk membangun model pembelajaran LSTM. Pada program 2, file dataset dalam bentuk *.csv* dimasukkan, di mana kolom 'Date' diubah menjadi tipe *datetime* dan diurutkan sesuai tanggal. Kolom yang digunakan untuk memprediksi adalah 'Beban' dan 'Suhu Minyak', sedangkan kolom target yang diprediksi adalah 'Sisa Umur'. Selanjutnya, data dinormalisasi menggunakan *MinMaxScaler* dari *scikit-learn* untuk memastikan bahwa semua nilai berada dalam rentang 0 sampai 1.

Pada program 4, dataset dibagi menjadi data *training* dan *testing* dengan rasio 80:20, dan *step size* digunakan untuk menentukan jumlah langkah waktu atau *time steps* sebelumnya yang digunakan untuk memprediksi. Kemudian, model LSTM dibuat dan dilatih dengan beberapa lapisan, dikompilasi dengan fungsi *loss mean\_squared\_error* dan *optimizer Adam*, serta parameter seperti *epoch* dan *batch size* ditentukan. Setelah itu, dilakukan prediksi pada data *training* dan data *testing*, kemudian data hasil prediksi dilakukan denormalisasi data dengan *MinMaxScaler*.

Program 6 digunakan untuk melakukan prediksi terhadap data *training* dan *testing* lalu hasil prediksi di visualisasikan untuk dianalisis. Setelah itu, menghitung *Mean Squared Error (MSE)* untuk data *training* dan *testing* untuk mengevaluasi performa model prediksi.