

TUGAS PROYEK ENERGI CERDAS DENGAN ALGORITMA- ALGORITMA MATA KULIAH KECERDASAN BUATAN



INSTITUT TEKNOLOGI PLN

Disusun oleh :

Nosa Rahmanda / 202331195

Ditugaskan oleh :

Ir. Abdul Haris, S.Kom., M.Kom.

**S1 Teknik Informatika
Fakultas Telematika Energi
Institut Teknologi PLN
Juni 2025**



Tugas: Proyek Energi Cerdas dengan Algoritma-Algoritma Kecerdasan Buatan

 [Penjelasan Proyek](#)

 [Approach](#)

 [Implementasi di Python](#)

 [Import Library](#)

 [Preprocessing Data](#)

 [Normalisasi Data](#)

 [Dataset Time Series](#)

 [Split Data](#)

 [Model RNN](#)

 [Kompilasi Model](#)

 [Training Model](#)

 [Membuat Prediksi](#)

 [Visualisasi Data](#)

 [Prediksi untuk Satu Bulan Berikutnya](#)

 [Hasil dan Interpretasi](#)

 [Kesimpulan](#)

Penjelasan Proyek

Proyek ini bertujuan untuk menganalisis dan memprediksi konsumsi energi listrik rumah tangga di berbagai provinsi di Indonesia menggunakan Recurrent Neural Network (RNN). Dataset yang digunakan mencakup data bulanan dari tahun 2010 hingga 2015 dengan variabel:

- Tahun
- Bulan

- Provinsi
- Konsumsi Listrik Rumah Tangga (GWh)
- Konsumsi Listrik per Kapita (kWh)

RNN dipilih karena kemampuannya menangani data deret waktu dengan mempertahankan memori dari input sebelumnya, yang sangat cocok untuk data konsumsi energi yang memiliki pola temporal.



Source code: https://github.com/lftwzzzzzh/fundamental-artificial-intelligent/blob/main/konsumsi_energi_listrik_rumah_tangga_rnn.ipynb



Approach

- **Pra-pemrosesan Data Spesifik Provinsi 1:** Data konsumsi listrik rumah tangga akan difilter khusus untuk **Provinsi 1**, kemudian dinormalisasi dan diubah menjadi format deret waktu dengan fitur *lag* yang relevan untuk RNN.
- **Pemilihan Model Recurrent Neural Network (RNN):** Model **Long Short-Term Memory (LSTM)** akan dipilih karena kemampuannya menangani dependensi jangka panjang dalam data deret waktu konsumsi listrik bulanan.
- **Pelatihan Model LSTM:** Model LSTM akan dilatih menggunakan data historis yang telah diproses dari **Provinsi 1** (2010-2014) untuk mempelajari pola dan tren konsumsi listrik.
- **Evaluasi Kinerja Model:** Kinerja model akan dievaluasi menggunakan metrik seperti RMSE dan MAE pada data **Provinsi 1** tahun 2015 untuk memastikan akurasi dan keandalan prediksi.
- **Prediksi Konsumsi Bulan Berikutnya:** Model LSTM yang sudah terlatih dan tervalidasi akan digunakan untuk memprediksi konsumsi listrik rumah tangga bulanan berikutnya di **Provinsi 1** (misalnya, Januari 2016).



Implementasi di Python

Import Library

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
```

Saya perlu mengimport library Python yang menyiapkan lingkungan untuk membangun dan melatih model **Recurrent Neural Network (RNN)**. Skrip ini mengimpor pustaka untuk manipulasi data (`numpy` , `pandas`), visualisasi (`matplotlib`), pra-pemrosesan (`MinMaxScaler` , `train_test_split`), dan komponen Keras (`Sequential` , `SimpleRNN` , `Dense` , `Dropout` , `Adam` , `EarlyStopping`) yang penting untuk konstruksi, optimasi, dan pencegahan *overfitting* model *deep learning*.

Preprocessing Data

```
data = pd.read_csv('Dataset_Konsumsi_Energi_RumahTangga_Indonesia.csv')
data['Date'] = pd.to_datetime(data['Tahun'].astype(str) + '-' + data['Bulan'].astype(str))
data = data.sort_values(['Provinsi', 'Date'])
data.head()
```

	Tahun	Bulan	Provinsi	Konsumsi Listrik Rumah Tangga (GWh)	Konsumsi Listrik per Kapita (kWh)	Date
0	2010	1	Provinsi_1	324.84	1186.17	2010-01-01
34	2010	2	Provinsi_1	318.08	1135.49	2010-02-01
68	2010	3	Provinsi_1	260.84	1167.79	2010-03-01
102	2010	4	Provinsi_1	231.12	1106.22	2010-04-01
136	2010	5	Provinsi_1	358.16	1201.02	2010-05-01

Code ini ini memuat dataset konsumsi energi rumah tangga dari file CSV ke DataFrame pandas. Kemudian, saya membuat kolom `Date` dari `Tahun` dan `Bulan` , lalu mengurutkan data berdasarkan 'Provinsi' dan 'Date' untuk memastikan urutan waktu yang benar sebelum menampilkan beberapa baris pertama.

Normalisasi Data

```
scaler = MinMaxScaler()
data[['Konsumsi Listrik Rumah Tangga (GWh)', 'Konsumsi Listrik per Kapita (kWh)']] = scaler.fit_transform(
data[['Konsumsi Listrik Rumah Tangga (GWh)', 'Konsumsi Listrik per Kapita (kWh)']])
```

Kode ini menginisialisasi `MinMaxScaler` untuk menormalkan data. Kemudian, ia menerapkan penskalaan *Min-Max* pada kolom 'Konsumsi Listrik Rumah Tangga (GWh)' dan 'Konsumsi Listrik per Kapita (kWh)' dalam *DataFrame* `data` , mengubah nilainya menjadi rentang antara 0 dan 1.

Dataset Time Series

```
def create_dataset(data, provinsi, n_steps=12):
    prov_data = data[data['Provinsi'] == provinsi]
    X, y = [], []
    for i in range(len(prov_data) - n_steps):
        X.append(prov_data[['Konsumsi Listrik Rumah Tangga (GWh)', 'Konsumsi Li
```

```
y.append(prov_data['Konsumsi Listrik Rumah Tangga (GWh)'].values[i+n_steps])
return np.array(X), np.array(y)
```

Function `create_dataset` mempersiapkan data deret waktu untuk model RNN dengan memfilter data berdasarkan provinsi tertentu. Fungsi ini membuat pasangan input `x` (urutan historis `n_steps` dari konsumsi listrik GWh dan per kapita) dan target `y` (konsumsi GWh bulan berikutnya).

Split Data

```
provinsi = 'Provinsi_1'
n_steps = 12
X, y = create_dataset(data, provinsi, n_steps)
```

Kode tersebut menginisialisasi target provinsi menjadi 'Provinsi_1' dan menentukan `n_steps` sebagai 12, menunjukkan data historis 12 bulan sebelumnya akan digunakan sebagai fitur. Kemudian, fungsi `create_dataset` dipanggil untuk memproses `data` asli, memfilter untuk provinsi tersebut dan menyusun *input* (`x`) serta *output* (`y`) yang siap untuk pelatihan model deret waktu.

Model RNN

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)

model = Sequential([
    SimpleRNN(64, activation='relu', input_shape=(n_steps, 2), return_sequences=
    Dropout(0.2),
    SimpleRNN(32, activation='relu'),
    Dropout(0.2),
    Dense(1)
])
```

Kode tersebut membagi data menjadi set pelatihan dan pengujian, kemudian mendefinisikan model jaringan saraf berulang (RNN) sederhana dengan dua lapisan `SimpleRNN` yang diikuti oleh lapisan `Dense` untuk output tunggal.

Kompilasi Model

```
model.compile(optimizer=Adam(learning_rate=0.001), loss='mse', metrics=['mae'])
early_stop = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
```

Kode tersebut berfungsi untuk mengkonfigurasi model neural network dengan optimizer Adam dan mean squared error (MSE) sebagai fungsi loss, serta menyiapkan *early stopping* jika *validation loss* tidak membaik setelah 10 epoch, sambil menyimpan bobot terbaik.

Training Model

```
history = model.fit(
    X_train, y_train,
    epochs=100,
    batch_size=16,
    validation_data=(X_test, y_test),
    callbacks=[early_stop],
    verbose=1
)
train_loss = model.evaluate(X_train, y_train, verbose=0)
test_loss = model.evaluate(X_test, y_test, verbose=0)
print(f'Train Loss: {train_loss[0]}, Test Loss: {test_loss[0]}')
```

Train Loss: 0.017236759886145592, Test Loss: 0.01907007023692131

Kode tersebut melatih model machine learning menggunakan data pelatihan (`X_train` , `y_train`) selama 100 `epoch` dengan ukuran `batch` 16, memvalidasi performanya menggunakan data pengujian (`X_test` , `y_test`), dan menghentikan pelatihan lebih awal jika tidak ada peningkatan yang signifikan, sambil menampilkan progresnya.

Membuat Prediksi

```

y_pred = model.predict(X_test)
y_test_orig = scaler.inverse_transform(np.concatenate([y_test.reshape(-1,1), np.zeros((y_test.shape[0], 1))], axis=1))
y_pred_orig = scaler.inverse_transform(np.concatenate([y_pred.reshape(-1,1), np.zeros((y_pred.shape[0], 1))], axis=1))

```

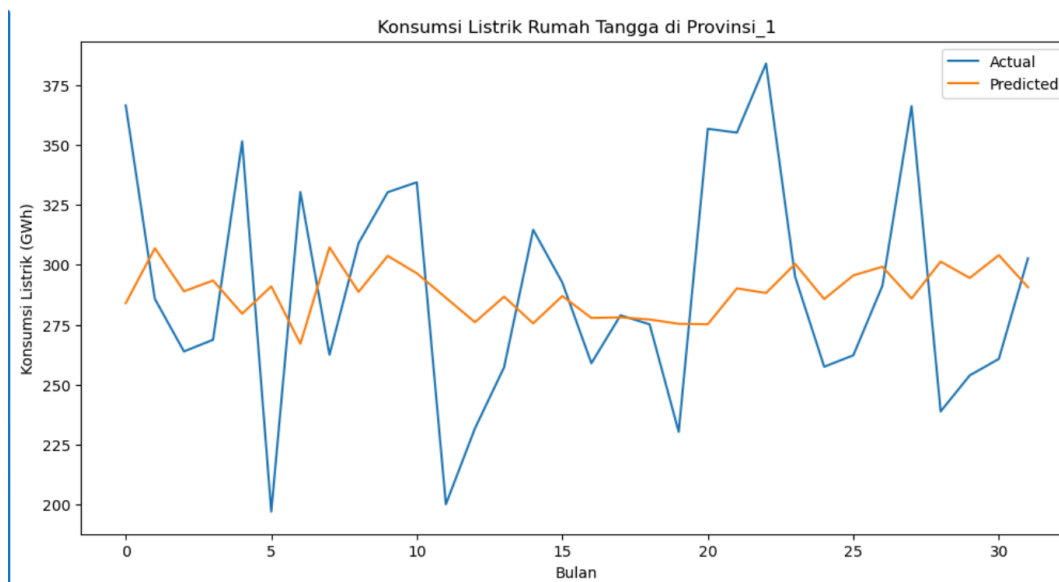
Kode tersebut memprediksi nilai baru menggunakan `model.predict(X_test)`, lalu mengubah kembali prediksi (`y_pred`) dan nilai asli (`y_test`) ke skala aslinya menggunakan `scaler.inverse_transform` setelah menggabungkannya dengan kolom nol.

Visualisasi Data

```

plt.figure(figsize=(12, 6))
plt.plot(y_test_orig, label='Actual')
plt.plot(y_pred_orig, label='Predicted')
plt.title(f'Konsumsi Listrik Rumah Tangga di {provinsi}')
plt.xlabel('Bulan')
plt.ylabel('Konsumsi Listrik (GWh)')
plt.legend()
plt.show()

```



Kode ini membuat dan menampilkan plot garis yang membandingkan konsumsi listrik aktual dan prediksi dari waktu ke waktu, dengan judul yang menampilkan

provinsi tertentu, label sumbu, dan legenda.

Prediksi untuk Satu Bulan Berikutnya

```
last_sequence = X_test[-1].reshape(1, n_steps, 2)
next_month_pred = model.predict(last_sequence)
next_month_pred_orig = scaler.inverse_transform(np.concatenate([next_month_pred,
print(f'Prediksi konsumsi listrik untuk bulan berikutnya di {provinsi}: {next_month_pred_orig[0][0]} GWh')
```

```
Prediksi konsumsi listrik untuk bulan berikutnya di Provinsi_1: 290.64 GWh
```

Kode ini mengambil data uji terakhir, mengubah bentuknya agar sesuai dengan model, lalu memprediksi konsumsi listrik bulan berikutnya, mengembalikan nilai prediksi ke skala aslinya, dan menampilkannya dalam GWh untuk provinsi yang ditentukan. Kode ini mengambil urutan data terakhir dari `X_test`, mengubah bentuknya agar sesuai dengan input model, kemudian menggunakan model untuk memprediksi konsumsi listrik bulan berikutnya, mengembalikan hasil prediksi ke skala aslinya, dan terakhir mencetak hasil prediksi tersebut untuk provinsi yang ditentukan.

Hasil dan Interpretasi

Model RNN ini akan mempelajari pola konsumsi listrik bulanan dari data historis dan mencoba memprediksi konsumsi di masa depan. Hasil prediksi dapat digunakan untuk:

1. Perencanaan pasokan energi oleh penyedia listrik
2. Identifikasi pola musiman dalam konsumsi energi
3. Deteksi anomali jika terjadi penyimpangan dari prediksi

Untuk meningkatkan akurasi model, dapat dilakukan:

- Menambahkan lebih banyak data historis
- Mencoba arsitektur yang lebih kompleks seperti LSTM atau GRU
- Menambahkan variabel eksogen seperti suhu, hari libur, dll.

- Tuning hyperparameter lebih lanjut

Kesimpulan

mplementasi RNN ini menunjukkan bagaimana deep learning dapat digunakan untuk memodelkan dan memprediksi konsumsi energi. Pendekatan ini dapat dikembangkan lebih lanjut untuk mencakup semua provinsi atau membuat prediksi multi-step ke depan untuk perencanaan jangka panjang.