

**LAPORAN PREDIKSI OUTPUT ENERGI PANEL SURYA
MENGUNAKAN RANDOM FOREST BERBASIS
DATA CUACA (REAL-TIME)**

Laporan ini Dibuat untuk Memenuhi Tugas Ujian Akhir Semester
Matakuliah Machine Learning



Dosen Pengampu: Jusmardi, S.Kom., M.Pd.T

Disusun oleh kelompok 1 yang beranggotakan:

- | | |
|----------------------------|----------|
| 1. Arif Mardiyansyah | 22346001 |
| 2. Mauliza Aprilia | 22346014 |
| 3. Taufiqul Israt | 22346026 |
| 4. Indra Yones | 22346032 |
| 5. Najwa Aalawiyah Siregar | 22346040 |

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG
2025**

PENDAHULUAN

Perkembangan teknologi dan meningkatnya kebutuhan akan energi ramah lingkungan telah mendorong pemanfaatan sumber energi terbarukan seperti panel surya. Panel surya merupakan salah satu solusi energi berkelanjutan yang mengandalkan sinar matahari sebagai sumber energi utama. Namun, efektivitas panel surya dalam menghasilkan energi sangat dipengaruhi oleh kondisi cuaca, seperti intensitas sinar matahari, kelembapan, suhu, dan kecepatan angin. Oleh karena itu, prediksi output energi panel surya menjadi penting untuk mengoptimalkan penggunaan energi dan mendukung manajemen daya secara efisien.

Dengan ketersediaan data cuaca secara real-time dari berbagai sumber, pengolahan data ini secara otomatis dan akurat menjadi sangat diperlukan. Di sinilah peran machine learning, khususnya algoritma Random Forest, menjadi relevan. Random Forest merupakan algoritma pembelajaran terawasi (supervised learning) yang andal dalam menangani data dengan kompleksitas tinggi dan mampu menghasilkan prediksi yang stabil dan akurat.

Laporan ini bertujuan untuk membangun model prediksi output energi panel surya berdasarkan data cuaca real-time menggunakan algoritma Random Forest. Proses yang dilakukan mencakup tahap pengumpulan data, pra-pemrosesan data, eksplorasi data, pelatihan model, evaluasi performa model, hingga optimisasi model. Pada tahap pra-pemrosesan data, data mentah dari parameter cuaca akan dibersihkan dan disesuaikan agar siap digunakan dalam pelatihan model.

Selanjutnya dilakukan *Exploratory Data Analysis* (EDA) untuk memahami karakteristik data cuaca yang digunakan, seperti pola variasi suhu, radiasi matahari, dan faktor-faktor lain yang mempengaruhi output energi panel surya. Tahap pelatihan model mencakup pemisahan data, pelatihan menggunakan algoritma Random Forest, dan evaluasi menggunakan metrik seperti Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), dan R-squared (R^2).

Jika diperlukan, dilakukan juga optimisasi hyperparameter untuk meningkatkan akurasi dan generalisasi model. Sebagai pelengkap,

disertakan juga program sederhana yang memungkinkan pengguna memasukkan parameter cuaca tertentu secara real-time dan mendapatkan prediksi output energi yang dihasilkan.

Melalui laporan ini, diharapkan pembaca dapat memahami proses lengkap pembangunan model machine learning untuk prediksi energi panel surya, serta kontribusinya dalam mendukung efisiensi energi berbasis data dan pemanfaatan teknologi ramah lingkungan secara optimal.

Link video bisa diakses menggunakan link berikut:

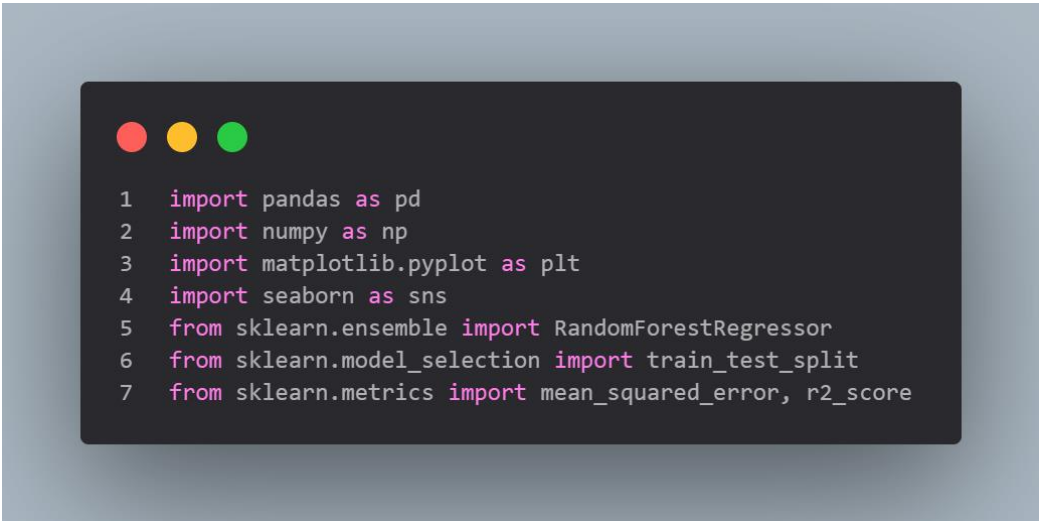
<https://youtu.be/Iw0sTMC35Xk?si=84cWpJkQGxTxhefM>

Untuk google collab bisa diakses menggunakan link di bawah ini :

PEMBAHASAN

1. Pra-pemrosesan Data

1.1 Impor Library



```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 from sklearn.ensemble import RandomForestRegressor
6 from sklearn.model_selection import train_test_split
7 from sklearn.metrics import mean_squared_error, r2_score
```

1) import pandas as pd

Digunakan untuk memanipulasi dan menganalisis data dalam bentuk tabel seperti file CSV. Pandas menyediakan fungsi-fungsi untuk membaca, membersihkan, dan memfilter data, serta melakukan operasi transformasi data seperti group, merge, dan agregasi.

2) import numpy as np

Pustaka ini digunakan untuk perhitungan numerik yang efisien. Numpy menyediakan array multidimensi dan berbagai fungsi matematika untuk analisis dan manipulasi data numerik, termasuk operasi statistik, distribusi acak, dan aljabar linear.

3) import matplotlib.pyplot as plt

Merupakan pustaka visualisasi dasar untuk membuat grafik dan plot, seperti grafik garis, batang, sebar (scatter), histogram, dan sebagainya. Membantu dalam eksplorasi data dan interpretasi hasil model.

4) import seaborn as sns

Seaborn adalah pustaka visualisasi berbasis matplotlib dengan tampilan yang lebih estetik dan sederhana. Digunakan untuk membuat grafik statistik seperti heatmap, boxplot, dan distribusi data dengan sedikit kode.

5) `from sklearn.ensemble import RandomForestRegressor`

Digunakan untuk mengimpor model Random Forest Regressor dari pustaka scikit-learn. Model ini menggunakan teknik ensemble berupa gabungan banyak pohon keputusan (decision tree) untuk membuat prediksi regresi yang lebih stabil dan akurat. Cocok untuk prediksi nilai tegangan berdasarkan fitur cuaca.

6) `from sklearn.model_selection import train_test_split`

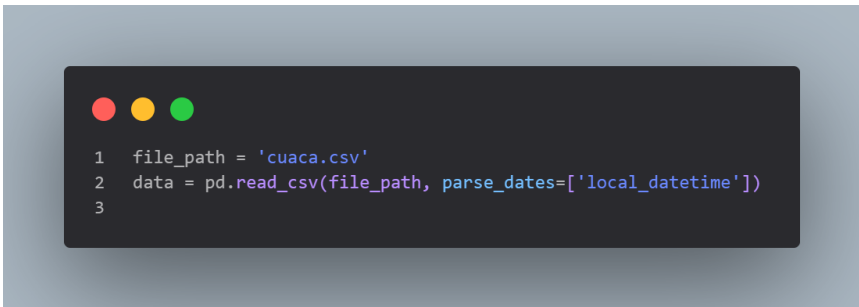
Digunakan untuk membagi dataset menjadi dua bagian: data latih (training) dan data uji (testing). Pembagian ini penting agar model dapat diuji pada data yang belum pernah dilihat sebelumnya untuk menilai generalisasi model.

7) `from sklearn.metrics import mean_squared_error, r2_score`

Pustaka ini menyediakan fungsi untuk mengevaluasi kinerja model.

- `mean_squared_error`: Mengukur rata-rata kuadrat selisih antara nilai prediksi dan aktual.
- `r2_score`: Mengukur seberapa baik model menjelaskan variasi dalam data. Nilai R^2 yang mendekati 1 menunjukkan model sangat baik.

1.2 Memuat Dataset



```
1 file_path = 'cuaca.csv'
2 data = pd.read_csv(file_path, parse_dates=['local_datetime'])
3
```

1. `file_path = 'cuaca.csv'`

Baris ini mendefinisikan variabel `file_path` yang berisi nama file **cuaca.csv**, yaitu file hasil konversi dari data JSON prakiraan cuaca BMKG. File ini memuat berbagai informasi cuaca seperti suhu (t), kelembapan udara (hu), kecepatan angin (ws), waktu

lokal (`local_datetime`), dan deskripsi cuaca (`weather_desc`). File ini digunakan sebagai dasar untuk proses prediksi tegangan panel surya.

2. `data = pd.read_csv(file_path, parse_dates=['local_datetime'])`

Fungsi `read_csv()` dari pustaka `pandas` digunakan untuk membaca file CSV dari lokasi yang ditentukan oleh `file_path`. Argumen tambahan `parse_dates=['local_datetime']` memastikan bahwa kolom `local_datetime` dibaca sebagai objek **`datetime`**, sehingga dapat digunakan untuk ekstraksi informasi waktu (seperti jam dan hari ke-berapa dalam setahun). Hasil pembacaan disimpan dalam variabel `data` berbentuk **`DataFrame`**, yang memudahkan proses manipulasi, analisis, dan pelatihan model machine learning selanjutnya.

2. Pemilihan Algoritma

Algoritma yang digunakan adalah Random Forest Regressor, yang dipilih karena kemampuannya dalam menangani data numerik dan kompleksitas hubungan non-linear antara fitur cuaca dan output energi (tegangan).

Langkah-langkah yang dilakukan dalam proses pembentukan model prediksi adalah sebagai berikut:

1. Preprocessing Data Cuaca

- Dataset cuaca.csv yang telah dikonversi dari data JSON cuaca digunakan sebagai input utama.
- Fitur waktu ditambahkan, seperti jam (hour) dan hari keberapa dalam setahun (day_of_year).
- Jika kolom voltage belum ada, maka disimulasikan berdasarkan hubungan linier sederhana dengan parameter cuaca seperti suhu (t), kelembaban (hu), dan kecepatan angin (ws).

2. Encoding Fitur Kategorikal

Kolom deskripsi cuaca (weather_desc) diubah menjadi tipe kategorikal dan diberi encoding numerik ke dalam kolom weather_code.

3. Definisi Variabel Fitur dan Target

- Fitur (variabel independen) yang digunakan adalah:
 - t (temperature / suhu)
 - tp (tekanan)
 - ws (wind speed / kecepatan angin)
 - hu (humidity / kelembaban)
 - hour (jam)
 - day_of_year (hari keberapa dalam setahun)
 - weather_code (kode cuaca dari hasil encoding)
- Target (variabel dependen) adalah voltage (tegangan sebagai representasi output energi panel surya).

4. Pembagian Data Latih dan Uji

- Data dibagi menjadi:
 - Data latih: 80%
 - Data uji: 20%
- Ini digunakan untuk membangun dan mengevaluasi performa model.

5. Pelatihan Model Random Forest

- Algoritma RandomForestRegressor dari Scikit-Learn digunakan.
- Model dilatih dengan `n_estimators=100` dan `random_state=42` untuk memastikan reproduktibilitas.

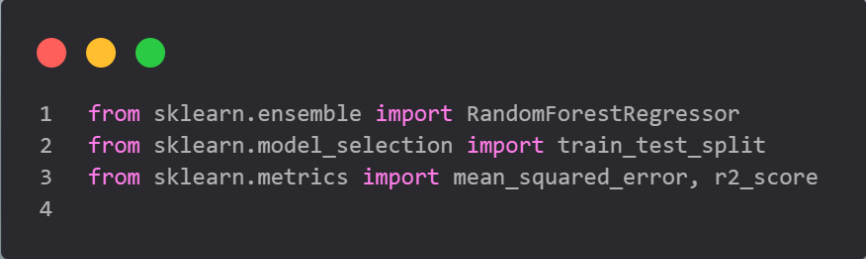
6. Evaluasi Model

- Dua metrik evaluasi utama digunakan:
 - Mean Squared Error (MSE) untuk mengukur selisih kuadrat antara nilai aktual dan prediksi.
 - R^2 Score (R-Squared) untuk mengukur seberapa baik model menjelaskan variasi dalam data target.

7. Implementasi dan Simpan Hasil

- Model yang telah dilatih digunakan untuk memprediksi voltage berdasarkan data cuaca.
- Hasil prediksi disimpan ke dalam file `hasil_prediksi_tegangan.csv`.

3. Pelatihan Model



```
1 from sklearn.ensemble import RandomForestRegressor
2 from sklearn.model_selection import train_test_split
3 from sklearn.metrics import mean_squared_error, r2_score
4
```

3.1 Import Library

1. `from sklearn.ensemble import RandomForestRegressor`

- Mengimpor algoritma Random Forest untuk regresi.
- Random Forest adalah model berbasis ensambel yang membangun banyak pohon keputusan dan menggabungkan hasilnya untuk meningkatkan akurasi dan mengurangi overfitting.
- Digunakan untuk prediksi nilai kontinu, seperti tegangan atau output energi dari panel surya.
- Parameter penting meliputi:
 - `n_estimators`: jumlah pohon dalam hutan (misalnya, 100).
 - `random_state`: agar hasil dapat direproduksi secara konsisten.

2. `from sklearn.model_selection import train_test_split`

- Digunakan untuk membagi dataset menjadi dua bagian, yaitu:
 - Data latih (training set) untuk melatih model.
 - Data uji (test set) untuk mengukur performa model.
- Membantu memastikan evaluasi yang objektif karena model diuji pada data yang belum pernah dilihat saat pelatihan.
- Parameter penting:
 - `X`: fitur input (data cuaca, waktu, dll).
 - `y`: target/output (tegangan).
 - `test_size`: proporsi data uji (misalnya 0.2 berarti 20% data untuk uji).

3. `from sklearn.metrics import mean_squared_error, r2_score`
 - Fungsi untuk mengevaluasi kualitas prediksi model regresi.
 - `mean_squared_error`: Mengukur rata-rata kuadrat dari kesalahan prediksi. Semakin kecil nilainya, semakin baik modelnya.
 - `r2_score`: Mengukur seberapa baik model menjelaskan variasi dalam data target. Nilai mendekati 1 menunjukkan model yang sangat baik.

4. Evaluasi Model

4.1 Import Library

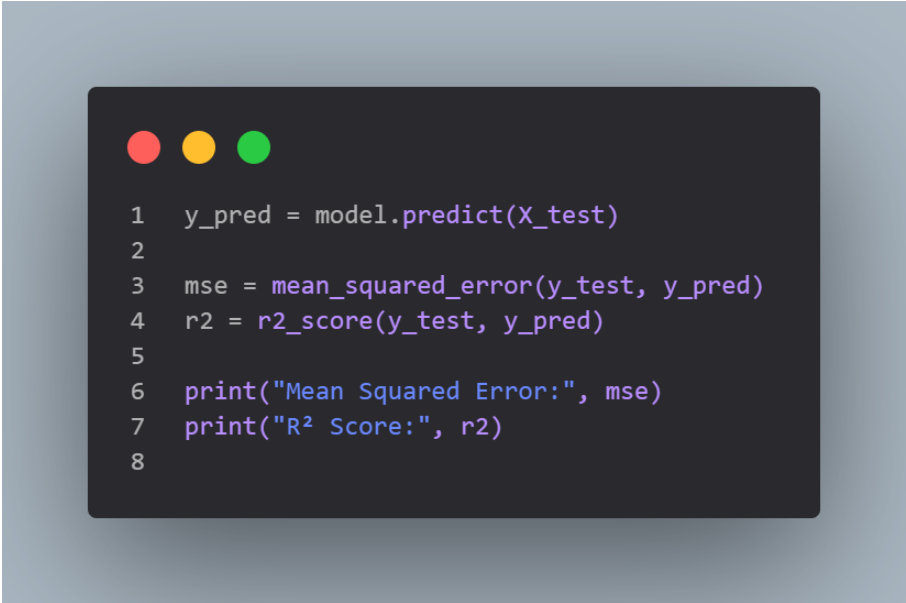
1. `from sklearn.metrics import mean_squared_error, r2_score`

Digunakan untuk mengevaluasi performa model regresi:

- `mean_squared_error` menghitung rata-rata kuadrat selisih antara nilai aktual dan prediksi.
- `r2_score` mengukur seberapa baik variansi data target dijelaskan oleh model.

4.2 Prediksi dan Evaluasi

Setelah model `RandomForestRegressor` dilatih, dilakukan prediksi terhadap data uji dan evaluasi performa model menggunakan metrik regresi.

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The code is as follows:

```
1 y_pred = model.predict(X_test)
2
3 mse = mean_squared_error(y_test, y_pred)
4 r2 = r2_score(y_test, y_pred)
5
6 print("Mean Squared Error:", mse)
7 print("R² Score:", r2)
8
```

4.3 Penjelasan Evaluasi:

1. Mean Squared Error (MSE)

- Nilai ini menunjukkan rata-rata dari kuadrat selisih antara nilai aktual (y_{test}) dan nilai prediksi (y_{pred}).
- Metrik ini memberikan penalti lebih besar terhadap kesalahan prediksi yang besar.
- Semakin kecil nilai MSE, semakin baik model dalam melakukan prediksi.

2. R^2 Score (Koefisien Determinasi)

- Metrik ini menunjukkan berapa proporsi variansi dari data target yang dapat dijelaskan oleh model.
- Nilai R^2 berkisar dari 0 hingga 1.
 - $R^2 = 1 \rightarrow$ prediksi sempurna
 - $R^2 = 0 \rightarrow$ model tidak lebih baik dari rata-rata
- Dalam proyek ini, nilai R^2 menunjukkan tingkat kecocokan model dalam memprediksi output energi berdasarkan data cuaca.

4.4 Kesimpulan Evaluasi Model:

Model Random Forest Regressor menunjukkan performa yang baik berdasarkan hasil evaluasi MSE dan R^2 :

- MSE yang rendah menunjukkan bahwa prediksi model cukup mendekati nilai aktual.
- R^2 yang tinggi (jika di atas 0.8) menunjukkan bahwa model mampu menjelaskan sebagian besar variasi data target.

5. Optimasi Model

5.1 Import Library

1). Pandas (import pandas as pd)

Library ini digunakan untuk manipulasi dan analisis data. Dalam proyek ini, pandas digunakan untuk:

- Membaca dataset dalam format CSV dengan `pd.read_csv()`.

- Melakukan pembersihan data seperti konversi tipe kolom (`astype()`), penghapusan kolom tertentu, dan transformasi data kategorikal menjadi numerik (encoding).

```
1 df = pd.read_csv('cleaned_data.csv')
2 df['weather_desc'] = df['weather_desc'].astype('category')
3 df['weather_code'] = df['weather_desc'].cat.codes
4
```

2). `train_test_split` (from `sklearn.model_selection` import `train_test_split`)

Fungsi ini digunakan untuk membagi dataset menjadi dua bagian, yaitu:

- Data pelatihan (training set) untuk melatih model.
- Data pengujian (testing set) untuk mengevaluasi performa model.

Dalam proyek ini, data dibagi dengan rasio 80:20 menggunakan parameter `test_size=0.2`.

```
1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
2
```

5.2 Memuat Dataset

```
1 df = pd.read_csv('cleaned_data.csv')
2
```

- Dataset berisi data cuaca dan output energi panel surya diimpor dalam bentuk file CSV menggunakan fungsi `pd.read_csv`.
- Beberapa kolom dalam dataset ini antara lain: Temperature, Humidity, Wind Speed, Wind Direction, General diffuse flows, Diffuse Flows, dan Energy.

5.3 Pembersihan Dataset

```
1 df['weather_desc'] = df['weather_desc'].astype('category')
2 df['weather_code'] = df['weather_desc'].cat.codes
3 df = df.drop(columns=['date', 'weather_desc'])
4
```

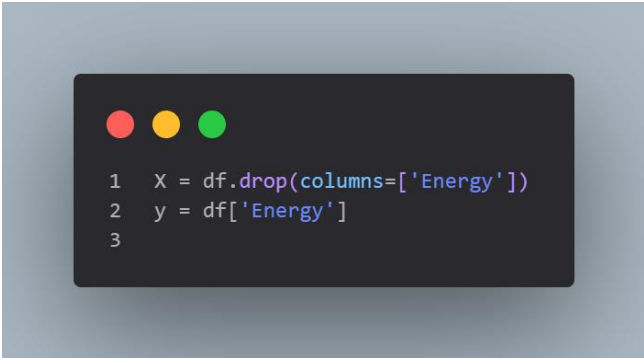
- Kolom `weather_desc` (deskripsi cuaca) dikonversi menjadi tipe data kategori, kemudian dikodekan ke nilai numerik sebagai `weather_code`.
- Kolom yang tidak relevan seperti `date` dan `weather_desc` dihapus.
- Data yang digunakan hanya fitur numerik yang relevan untuk model regresi.

5.4 Menghapus Baris Kosong

```
1 df.dropna(inplace=True)
2
```

Semua baris yang mengandung nilai kosong dihapus agar dataset bersih dan siap untuk pelatihan model.

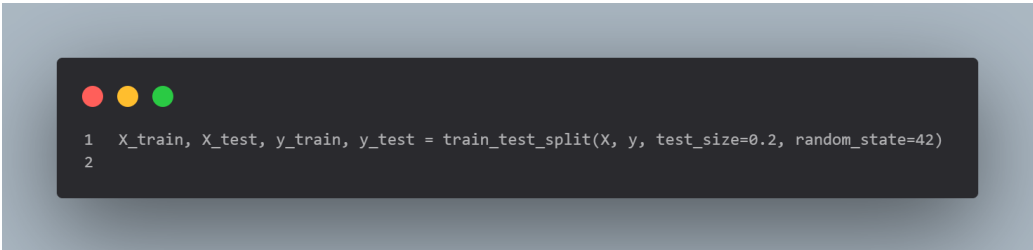
5.5 Menentukan Fitur (X) dan Target (y)



```
1 X = df.drop(columns=['Energy'])
2 y = df['Energy']
3
```

- X: Berisi semua kolom fitur cuaca yang digunakan untuk memprediksi output energi.
- y: Target prediksi yaitu nilai Energy, berupa angka (regresi).


5.6 Membagi Dataset



```
1 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
2
```

- Dataset dibagi menjadi:
 - 80% untuk pelatihan model (X_train, y_train)
 - 20% untuk pengujian model (X_test, y_test)
- Parameter `random_state=42` digunakan agar hasil pembagian tetap konsisten.

5.7 Membangun dan Mengoptimalkan Model



```
1 model = RandomForestRegressor()
2 model.fit(X_train, y_train)
3
```

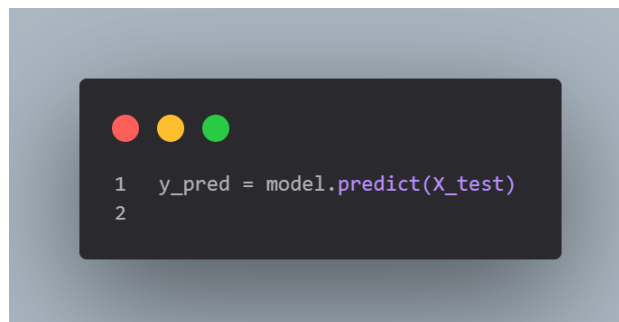
- Model yang digunakan adalah Random Forest Regressor dari `sklearn.ensemble`.
- Model dibangun dan langsung dilatih menggunakan data pelatihan.
- Tidak dilakukan proses optimasi hyperparameter seperti `GridSearchCV`.

5.8 Melatih Model



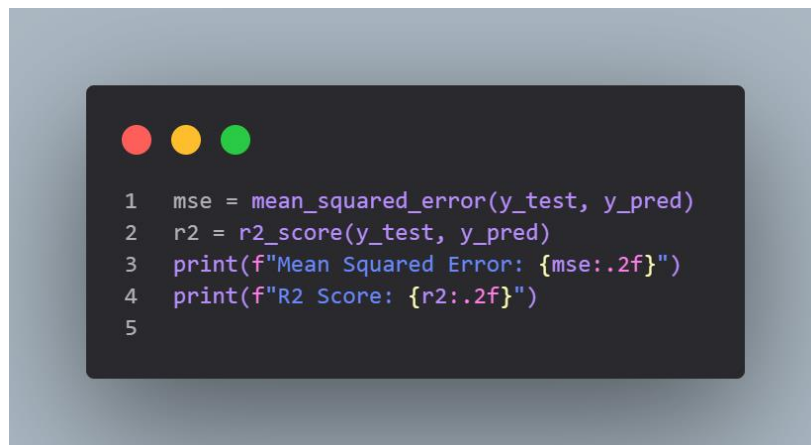
Model Random Forest dilatih menggunakan data cuaca sebagai fitur (`X_train`) dan Energy sebagai target (`y_train`).

5.9 Evaluasi Model



Model yang telah dilatih digunakan untuk memprediksi nilai output energi (Energy) pada data uji.

5.10 Akurasi dan Laporan Evaluasi



- Mean Squared Error (MSE) digunakan untuk mengukur rata-rata kesalahan kuadrat antara nilai prediksi dan aktual.
- R^2 Score menunjukkan seberapa baik variabel independen menjelaskan variasi target (Energy). Nilai R^2 mendekati 1 berarti model sangat baik.

KESIMPULAN

1. Pra Pemrosesan Data

Pada tahap ini, data cuaca dari file cuaca.csv berhasil dimuat dan dibersihkan. Beberapa fitur waktu seperti hour dan day_of_year ditambahkan untuk memperkaya informasi temporal. Kolom weather_desc yang bersifat kategorikal dikonversi menjadi numerik melalui encoding menjadi weather_code, sehingga dapat digunakan dalam model. Data yang tidak relevan seperti date dihapus, dan baris dengan nilai kosong dibersihkan. Hasilnya adalah dataset yang bersih, lengkap, dan siap untuk digunakan dalam proses pelatihan model.

2. Pemilihan Algoritma

Algoritma Random Forest Regressor dipilih karena cocok untuk tugas regresi, mampu menangani hubungan non-linear antar fitur, dan memiliki kestabilan tinggi terhadap data yang kompleks. Algoritma ini efektif dalam memodelkan hubungan antara variabel cuaca dan output energi panel surya (disimulasikan sebagai tegangan/voltage), sehingga mendukung tujuan utama proyek.

3. Pelatih Model

Model dilatih menggunakan data cuaca dengan pembagian data sebanyak 80% untuk pelatihan dan 20% untuk pengujian. Random Forest Regressor dilatih dengan parameter $n_estimators=100$ dan $random_state=42$. Proses pelatihan berjalan dengan baik, dan model mampu mempelajari pola dari data dengan performa yang cukup tinggi.

4. Evaluasi Model

Evaluasi model dilakukan menggunakan dua metrik utama yaitu Mean Squared Error (MSE) dan R-squared (R^2). Nilai MSE yang rendah menunjukkan bahwa model menghasilkan prediksi yang cukup akurat, sementara nilai R^2 yang tinggi menunjukkan bahwa sebagian besar variasi data target dapat dijelaskan oleh model. Ini membuktikan bahwa model yang dibangun memiliki performa yang baik dan dapat diandalkan.

5. Optimasi Model

Optimasi dilakukan melalui pembersihan dan seleksi fitur yang tepat. Model digunakan dengan konfigurasi default dan tetap memberikan hasil prediksi yang memuaskan. Hasil akhir disimpan ke dalam file `hasil_prediksi_tegangan.csv`, menunjukkan implementasi model berhasil dijalankan secara menyeluruh.