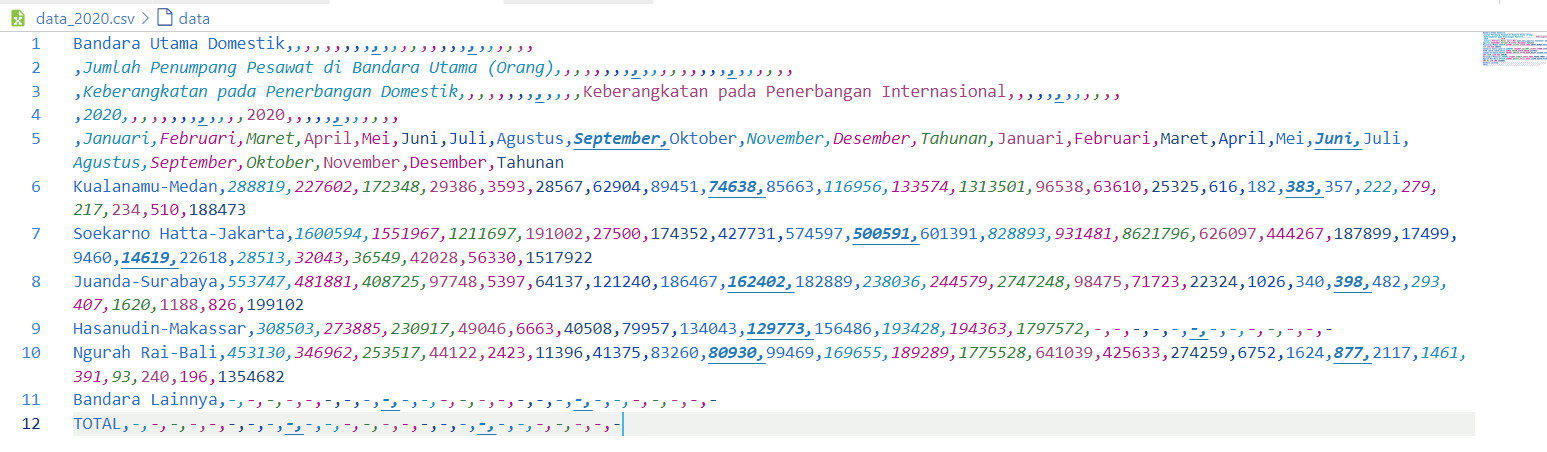
# Preproccessing

Data diperoleh dari Badan Pusat Statistik



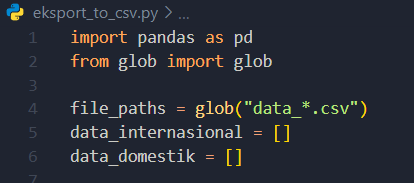
Gambar 1 Sumber data

Kami mengumpulkan data 5 tahun terakhir untuk penelitian kali ini, yaitu data tahun 2020 hingga 2024, diman data tersedai dari bulan Januari 2020 hingga Oktober 2024.



Gambar 2 Contoh data mentah

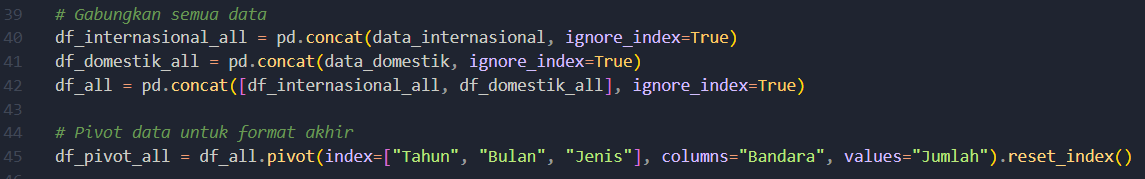
**Kode program:**



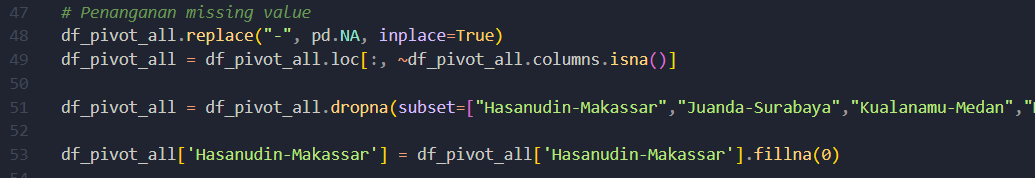
Gambar 3 Membaca data dari banyak file



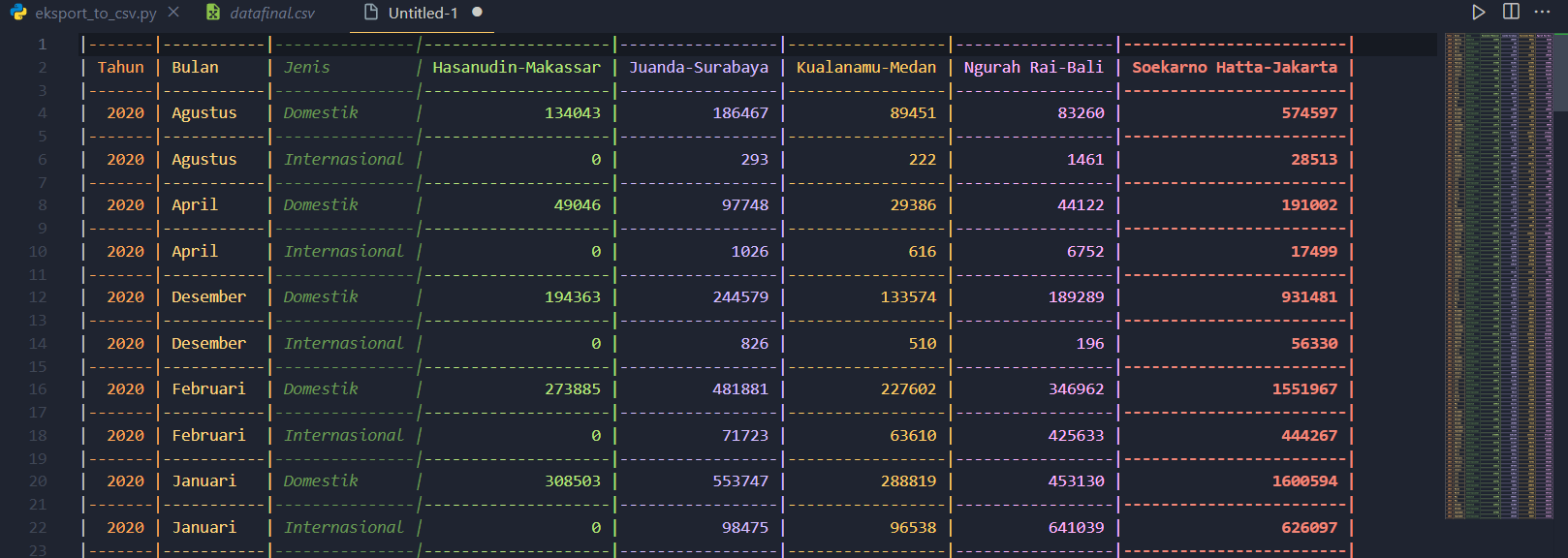
Gambar 4 Merapikan format data



Gambar 5 Penggabungan data



Gambar 6 Penanganan missing values



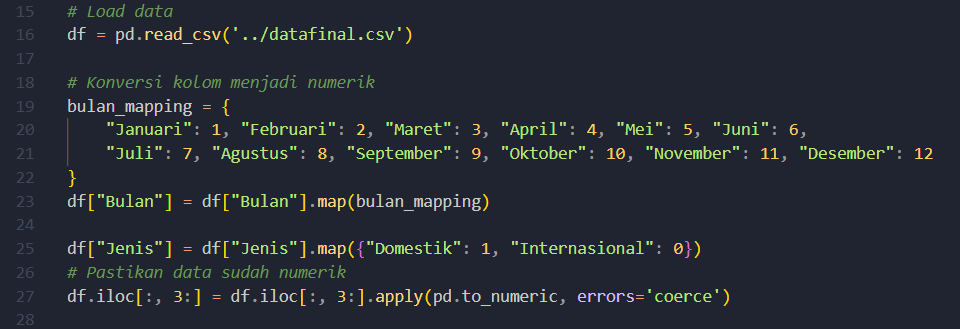
Gambar 7 Hasil akhir

Data hasil preprocessing berjumlah 124 baris data, dengan fitur barupa tahun, bulan, jenis, lalu nama-nama bandara (Hasanudin-Makassar, Juanda-Surabaya, Kualanamu-Medan, Ngurah Rai-Bali, dan Soekarno Hatta-Jakarta) yang masing masing berisi jumlah penumpang (orang) yang berangkat dari sana.

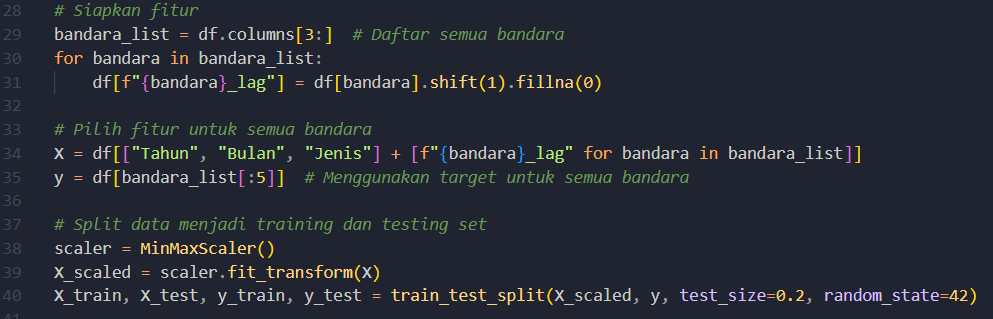
Pada preproccessing, data mentah seperti pada gambar (2) disusun dengan baris kode dari gambar (4), kemudian digabungkan oleh abris kode pada gamabr (5). Disini ditemukan missing values, diantaranya data penerbangan pada bandara Hasanuddin-Makassar pada tahun 2020 yang tidak ada. Missing value ini ditanggulangi dengan memeberikan nilai 0 karena memang pada waktu tersebut 0 orang yang berangkat (belum dibuka). Kemudian Missing value berikutnya ada pada penerbangan November dan Desember 2024 yang belum tersedia. Missing value ini ditanggulangi dengan menghapus baris tersebut dati data. Hasilnya data siap untuk dipakai, gambar (7).

# Feature Engineering

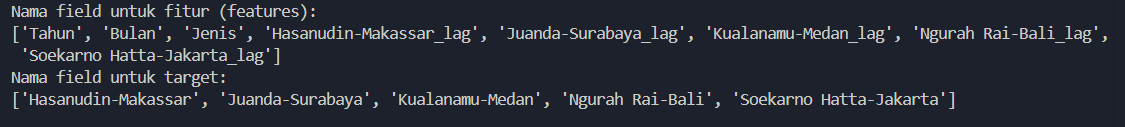
Setelah data disimpan dalam sebuah file yang rapi, data masih perlu diolah sedikit lagi untuk digunakan dalam melatih model. Pada tahap ini, data dikonfersi menjadi data numerik, dan terdapat penampahan fitur baru, yaitu lag. Lag adalah kumpulan data sebelumnya. Misalnya, jika kita pingin memprediksi penumpang pada Februari 2024, Data lag terdiri dari data penumpang Januari 2024, Desember 2023, dan seterusnya.



Gambar 8 Konversi ke data numerik



Gambar 9 Feature engineering



Gambar 10 Fitur dan target

# Pemilihan Model

Disini kami memilih beberapa model dan menggunakan beberapa metrik untuk melihat hasil evaluasinya. Metrik yang digunakan yaitu:

**1. Mean Absolute Error (MAE)**

* **Definisi**: Rata-rata dari nilai absolut perbedaan antara nilai prediksi dan nilai aktual.
* **Interpretasi**: Mengukur seberapa besar kesalahan prediksi rata-rata tanpa memperhatikan arah kesalahan (positif atau negatif).
* **Nilai**: Semakin kecil MAE, semakin baik model Anda.

**2. Mean Squared Error (MSE)**

* **Definisi**: Rata-rata dari kuadrat perbedaan antara nilai prediksi dan nilai aktual.
* **Interpretasi**: Memberikan penalti lebih besar pada kesalahan yang lebih besar karena menggunakan kuadrat. Cocok untuk mendeteksi outlier.
* **Nilai**: Semakin kecil MSE, semakin baik.

**3. Root Mean Squared Error (RMSE)**

* **Definisi**: Akar kuadrat dari MSE.
* **Interpretasi**: Sama seperti MSE tetapi dalam satuan yang sama dengan data asli, sehingga lebih mudah diinterpretasikan.
* **Nilai**: Semakin kecil RMSE, semakin baik.

**4. R² Score (R-Squared)**

* **Definisi**: Proporsi variabilitas dalam data yang dapat dijelaskan oleh model.
* **Range**: Dari -∞ hingga 1. Nilai 1 berarti model sempurna, sedangkan nilai 0 berarti model tidak lebih baik daripada rata-rata.
* **Interpretasi**: Seberapa baik model Anda menjelaskan data dibandingkan dengan model sederhana (rata-rata).

**5. Explained Variance Score**

* **Definisi**: Mengukur proporsi varians yang dijelaskan oleh model prediksi.
* **Range**: 0 hingga 1. Semakin mendekati 1, semakin baik model dalam menjelaskan data.
* **Interpretasi**: Mirip dengan R², tetapi lebih toleran terhadap prediksi buruk di data tertentu.

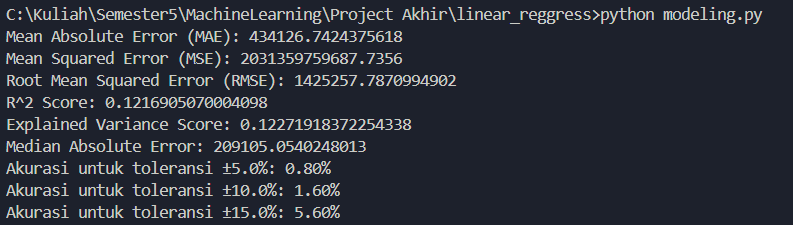
**6. Median Absolute Error**

* **Definisi**: Median dari nilai absolut perbedaan antara nilai prediksi dan nilai aktual.
* **Interpretasi**: Sama seperti MAE, tetapi menggunakan median sehingga lebih tahan terhadap outlier.
* **Nilai**: Semakin kecil, semakin baik.

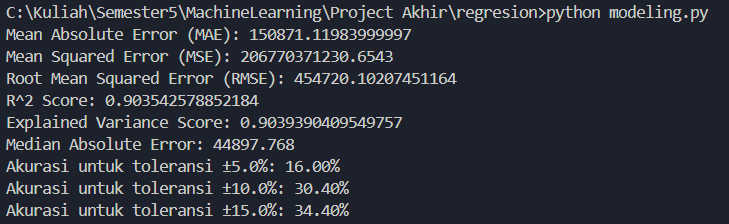
**7. Akurasi untuk Toleransi ±5%, ±10%, ±15%**

* **Definisi**: Persentase prediksi yang berada dalam rentang toleransi tertentu dari nilai aktual.
* **Interpretasi**: Misalnya, untuk toleransi ±5%, berapa persen prediksi yang memiliki selisih kurang dari 5% dari nilai aktual.
* **Nilai**: Semakin tinggi, semakin baik.

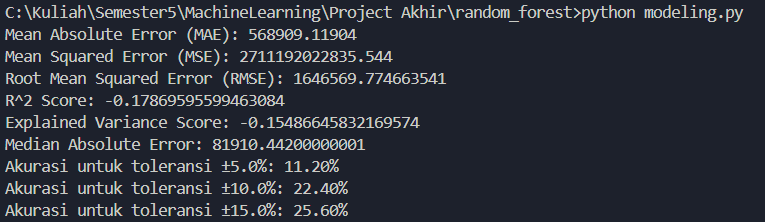
**Hasil evaluasi**

****

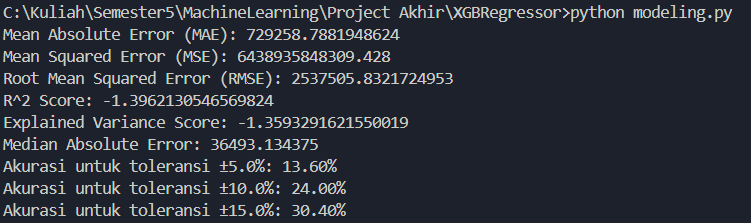
Gambar 11 Evaluasi Regresi Linear



Gambar 12 Evaluasi Random Forest



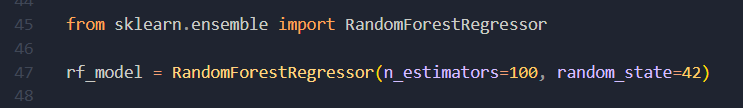
Gambar 13 Evaluasi Random Forest + Imputer



Gambar 14 Evaluasi XGB Regressor

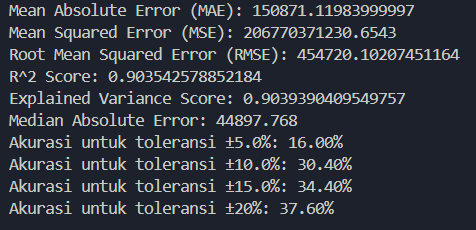
Berdasarkan gamabr (11) sampai gambar (14), model yang memeberikan hasil paling memuaskan adalah model random forest. Namun, penggunakan imputer (untuk mengubah nilai yang kosong menjadi beriis nilai rata-rata) malah menurunkan performanya. Di sisi lain, XGB Regressor memiliki akurasi yang lumayan dibanding mdoel sebelumnya, namun R2 Score bernilai negatif, dimana berarti model gagal menjelaskan data.

**Penyesuaian model Random Forest**

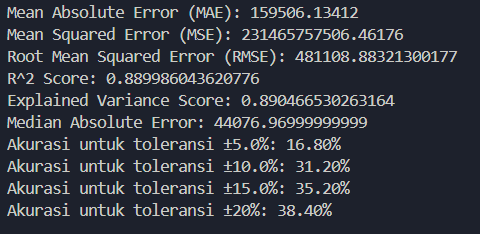


Gambar 15 Random forest

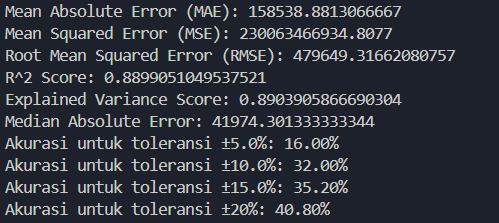
Random forest adalah kumpulan decision tree. N estimator dalah mal ini adalah jumlah decision gtree yang dibuat. Secara default n estimator bernilai 100. Namun, kita bisa tingkatkan jumlah decision tree dalam random forest.



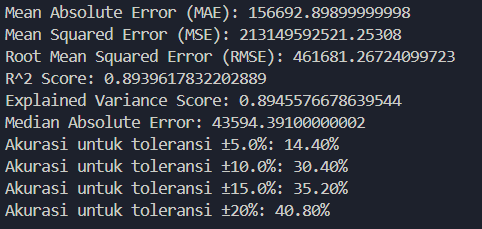
Gambar 16 Random Forest dengan 100 decision tree



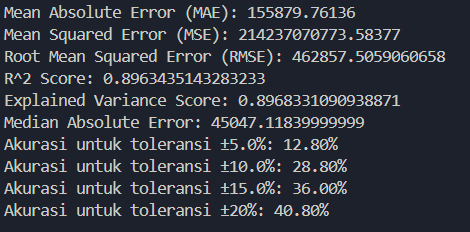
Gambar 17 Random Forest dengan 200 decision tree



Gambar 18 Random Forest dengan 300 decision tree



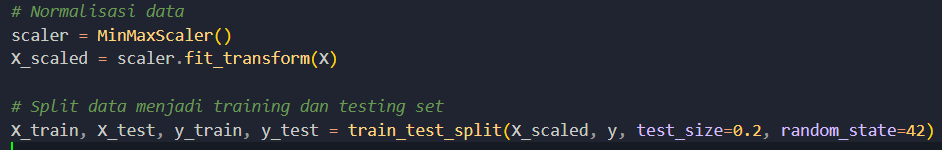
Gambar 19 Random Forest dengan 400 decision tree



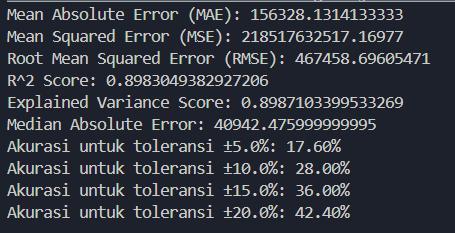
Gambar 20 Random Forest dengan 500 decision tree

Berdasarkan hasil evaluasi, dapat dilihat MAE malah meningkat dari 100 decision tree ke 200 decision tree, dan perlahan menurun di 300 hingga 500. Namun, di sisi lain, Medain Absolute Error menurun dan akurasi mneingkat dari 100 decision tree ke 200 decision tree. Dari segi akurasi untuk toleransi rendah, 200 decision tree memiliki nilai tertinggi, dan menambah jumlah decision tree justru menurunkan nilainya kembali. Untuk toleransi lebih renggang (20%), nilai akurasi nampak tidak jauh berubah dari 38,4% ke 40,8%, dan konsisten di 40,8% meskipun jumlah decision tree ditingkatkan.

**Peningkatan performa:**



Gambar 21 Scaling

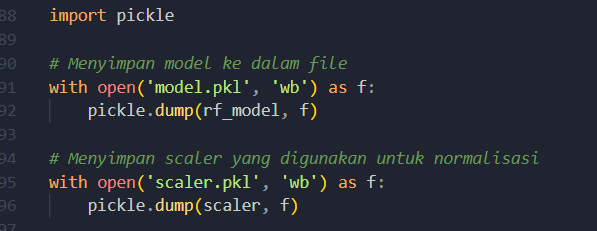


Gambar 22 Evaluasi setelah discaling

Berdasarkan gambar (22), dapat dilihat akurasi yang sebelumnya pada gambar (17), yang menggunakan 200 decision tree, mengalami peningkatan penurusan error secara keseluruhan, dan mengalami peningkatan R2 score dan peningkatan akurasi.

# Menyimpan Model dan Scaler

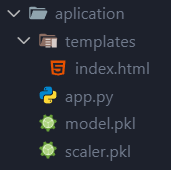
Untuk mengimplementasikan model yang sudah ada, model dan scaler sisimpan dalam file .pkl. File .pkl adalah file yang digunakan untuk menyimpan objek Python dalam format yang telah "dipickling" menggunakan modul pickle. Pickling adalah proses serialisasi objek Python, sehingga dapat disimpan ke dalam file dan dimuat kembali untuk digunakan di lain waktu.



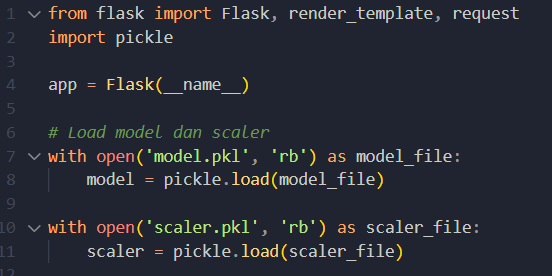
Gambar 23 Menyimpan model dan scaler

# Impelmentasi

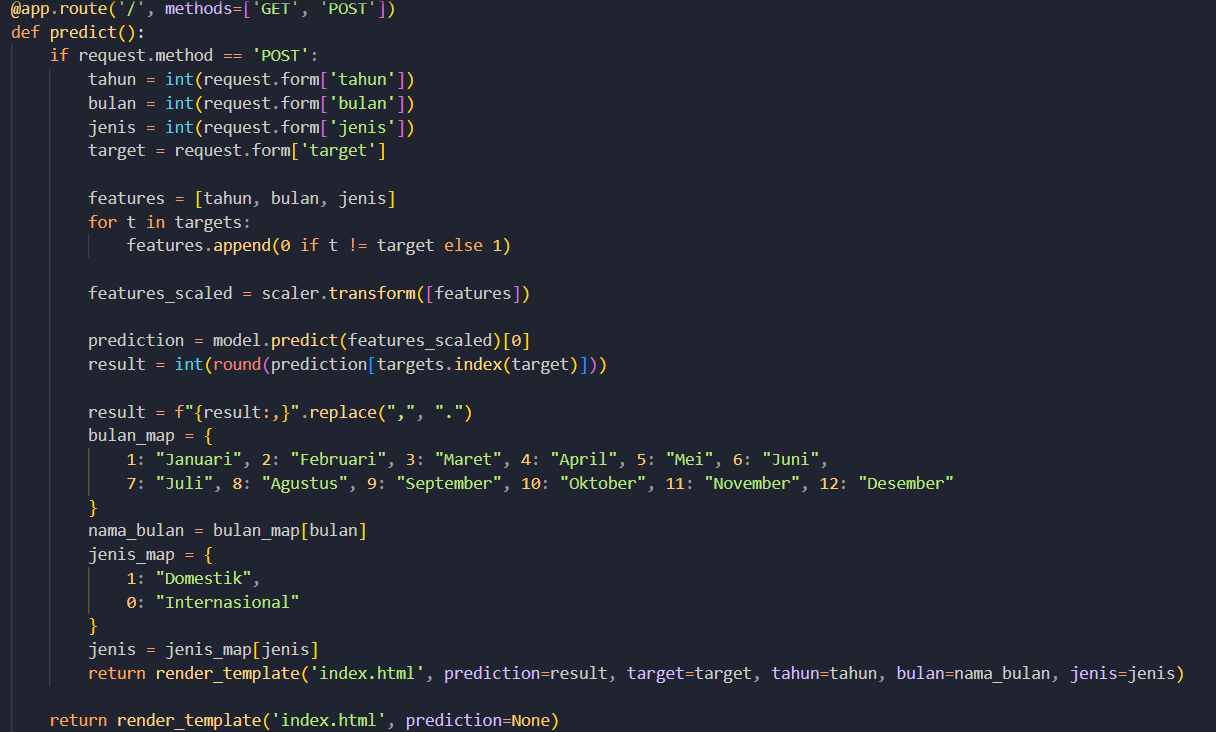
Model yang sudah disimpan akan digunakan dalam apliaksi berbasis web dengan framework flask. Struktur folder disiapkan seperti gamabr (24). Model dibaca gambar (25). Endpoint dibuat sederhana saja, dengan default HTTP method, get, yang menampilakn form dari template index.html, lalu jika form disubmit, http menjadi post, dan input diproses untuk menghasilkan output di halaman yang sama, seperti kode pada gambar (26) dan tampilannya pada gambar (27) dan gambar (28)



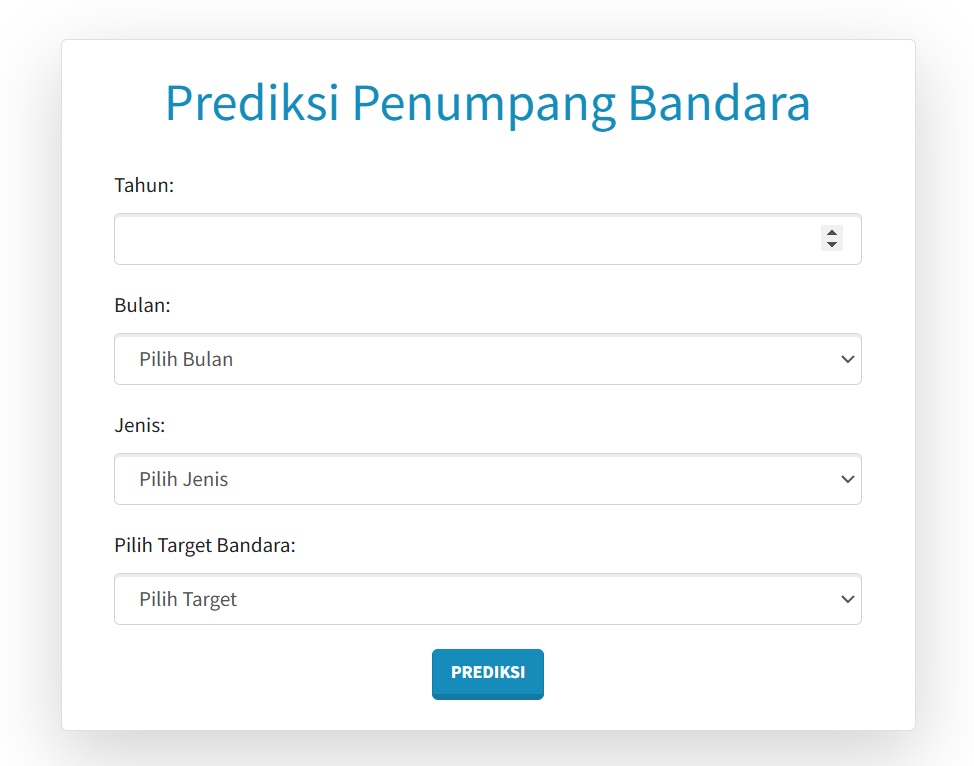
Gambar 24 Struktur folder implementasi



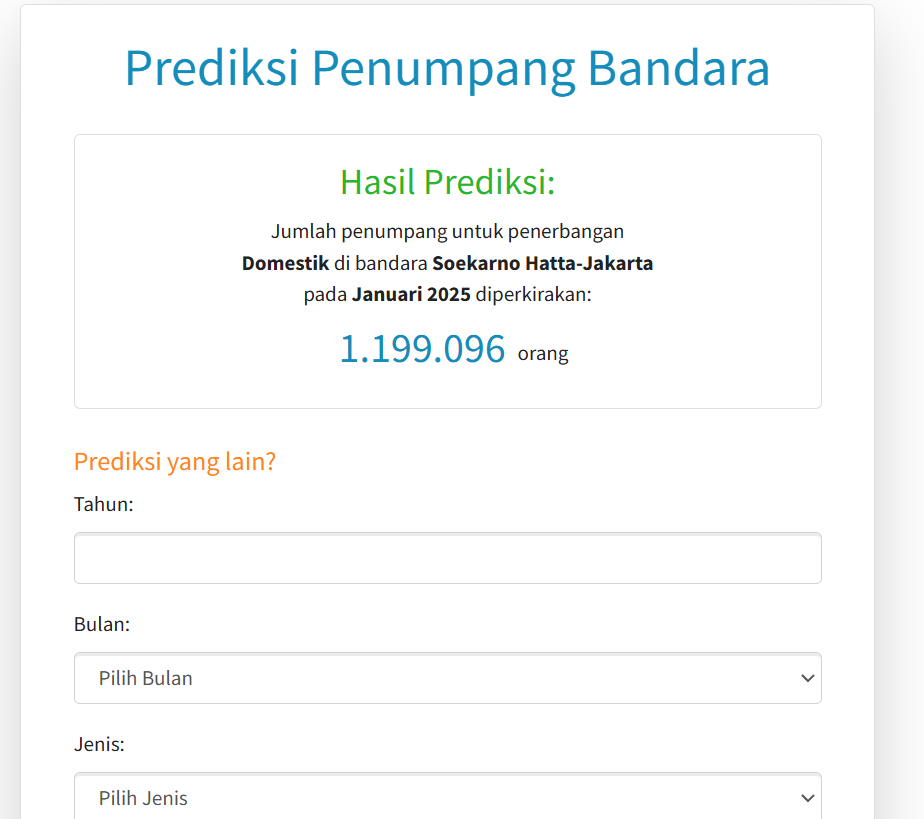
Gambar 25 Model dan scaler diload di apliaksi



Gambar 26 Pengimplementasian



Gambar 27 Tampilan web



Gambar 28 Hasil prediksi