

**LAPORAN TUGAS BESAR KECERDASAN BUATAN  
APLIKASI PREDIKSI CUACA BERBASIS PYQT6 DENGAN FITUR  
EVALUASI MODEL DECISION TREE**



**Disusun Oleh:**

Nastiti Dyah Ayu Gantari      122490017

**REKAYASA INSTRUMENTASI DAN AUTOMASI  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SUMATERA  
2025**

## **DAFTAR ISI**

<b>BAB I PENDAHULUAN.....</b>	<b>3</b>
1.1 Latar Belakang .....	3
1.2 Rumusan Masalah.....	4
1.3 Tujuan Penelitian .....	4
1.4 Kontribusi Penelitian .....	4
<b>BAB II Tinjauan Pustaka.....</b>	<b>5</b>
2.1 Konsep Prediksi Cuaca .....	5
2.2 Analisis Faktor Cuaca.....	5
2.3 Algoritma Machine Learning untuk Klasifikasi .....	6
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>7</b>
3.1 Desain Sistem .....	7
3.2 Arsitektur Aplikasi.....	7
3.3 Algoritma Decision Tree.....	7
<b>BAB IV Hasil Dan Pembahasan .....</b>	<b>9</b>
4.1 Tampilan GUI.....	9
4.2 Implementasi Sistem .....	13
4.3 Hasil Evaluasi Model .....	14
4.4 Evaluasi Pemilihan Algoritma dalam Implementasi Model Prediksi .....	14
<b>BAB V KESIMPULAN .....</b>	<b>16</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan pesat di bidang machine learning telah membuka peluang signifikan dalam penerapan prediksi cuaca berbasis data historis dengan tingkat kecepatan dan akurasi yang tinggi. Metode pembelajaran mesin, khususnya algoritma klasifikasi seperti Decision Tree, sangat populer dalam aplikasi ini karena kemampuannya untuk menangani data numerik dan kategorikal secara efektif serta menghasilkan model yang mudah diinterpretasikan.

Implementasi model Decision Tree pada aplikasi prediksi cuaca memungkinkan klasifikasi kondisi cuaca berdasarkan sejumlah fitur relevan yang dikumpulkan dari data pengamatan sebelumnya. Proses pembuatan aplikasi ini difasilitasi dengan penggunaan framework PyQt6 sebagai antarmuka pengguna grafis (GUI), yang secara signifikan meningkatkan kemudahan penggunaan bagi pengguna akhir. Dengan GUI ini, pengguna dapat melakukan serangkaian operasi penting seperti pemilihan data pengujian, pemuatan model terlatih, serta menjalankan evaluasi model tanpa perlu memiliki kemampuan pemrograman lanjutan.

Aplikasi ini menyediakan modul evaluasi yang komprehensif, yang meliputi perhitungan metrik evaluasi seperti akurasi, presisi (precision), recall, F1-score, serta error metrics seperti mean absolute error (MAE) dan mean squared error (MSE). Selain itu, fitur visualisasi hasil evaluasi berupa confusion matrix, grafik feature importance, dan learning curve disertakan untuk memberikan gambaran mendalam mengenai performa dan karakteristik model yang digunakan. Visualisasi ini tidak hanya berfungsi untuk membantu pengguna dalam memahami efektivitas model, tetapi juga untuk melakukan analisis yang lebih mendalam terhadap kekuatan dan kelemahan model yang dihasilkan.

Penerapan Decision Tree dalam konteks prediksi cuaca memanfaatkan kemampuan algoritma ini untuk membagi data berdasarkan kriteria informasi gain atau gini index, sehingga menghasilkan pohon keputusan yang mampu mengklasifikasikan kondisi cuaca dengan tingkat akurasi yang dapat diandalkan. Dengan adanya antarmuka GUI yang intuitif, aplikasi ini menyoroti kebutuhan pengguna non-teknis maupun teknis untuk melakukan prediksi cuaca secara interaktif dan efisien, serta mendukung pengambilan keputusan berdasarkan data yang valid dan terukur.

Secara keseluruhan, aplikasi ini menyatukan aspek teknis machine learning dengan user experience yang baik, sehingga memberikan solusi yang efektif, transparan, dan mudah diakses dalam bidang prediksi cuaca. Hal ini memungkinkan

pemanfaatan teknologi AI untuk aplikasi praktis di berbagai bidang seperti pertanian, perencanaan kegiatan, dan mitigasi risiko bencana.

## **1.2 Rumusan Masalah**

1. Bagaimana membangun aplikasi GUI untuk prediksi cuaca yang mudah digunakan?
2. Bagaimana mengintegrasikan model machine learning Decision Tree dengan antarmuka aplikasi agar pengguna dapat melakukan evaluasi performa model secara interaktif?
3. Bagaimana menampilkan metrik dan visualisasi evaluasi model agar mudah dipahami oleh pengguna?

## **1.3 Tujuan Penelitian**

1. Membangun aplikasi desktop menggunakan PyQt6 yang dapat melakukan prediksi cuaca menggunakan model machine learning Decision Tree.
2. Mengimplementasikan fitur evaluasi model dengan menampilkan metrik dan visualisasi yang informatif.
3. Menyediakan antarmuka pengguna yang intuitif dan responsif untuk penggunaan yang mudah.

## **1.4 Kontribusi Penelitian**

Penelitian ini menghasilkan aplikasi GUI interaktif yang mengintegrasikan pemilihan model Decision Tree, data testing, evaluasi performa model, serta visualisasi hasil secara real-time, sehingga dapat dimanfaatkan sebagai alat bantu analisis prediksi cuaca bagi pengguna non-teknis.

## **BAB II**

### **Tinjauan Pustaka**

#### **2.1 Konsep Prediksi Cuaca**

Prediksi cuaca merupakan aspek penting dalam berbagai sektor, termasuk pertanian, transportasi, dan kegiatan luar ruang. Dalam konteks ini, pemanfaatan algoritma *machine learning* seperti Decision Tree memungkinkan klasifikasi kondisi cuaca secara otomatis berdasarkan data historis. Model ini dibangun dengan menganalisis parameter meteorologis utama, seperti suhu, kelembapan, intensitas cahaya, dan curah hujan, yang dikumpulkan menggunakan sensor DHT22, Rain Sensor, dan BH1750. Decision Tree bekerja dengan membagi data berdasarkan fitur-fitur paling informatif untuk memaksimalkan akurasi klasifikasi terhadap empat kondisi cuaca: *Cerah*, *Berawan*, *Gerimis*, dan *Hujan*. Struktur pohon yang dihasilkan dari proses pelatihan memungkinkan sistem mengenali pola kompleks dalam data lingkungan tanpa memerlukan pemrograman lanjutan dari pengguna.

Integrasi model prediksi ke dalam aplikasi GUI berbasis PyQt6 meningkatkan aksesibilitas dan efisiensi penggunaan sistem oleh pengguna umum maupun teknis. Aplikasi ini mendukung proses pemuatan data, klasifikasi, serta visualisasi hasil secara interaktif dan informatif. Dengan adanya sistem ini, pengguna dapat melakukan prediksi cuaca secara cepat dan objektif hanya dengan memasukkan data dari sensor atau file CSV. Keandalan dan kecepatan model Decision Tree menjadikannya solusi yang praktis untuk sistem monitoring cuaca berbasis desktop, sekaligus mendukung pengambilan keputusan berbasis data dalam konteks mitigasi risiko cuaca ekstrem dan perencanaan kegiatan harian yang lebih adaptif.

#### **2.2 Analisis Faktor Cuaca**

Dalam pembuatan model machine learning untuk prediksi cuaca, pengumpulan data meteorologi yang lengkap merupakan tahap fundamental. Parameter utama yang sering dianalisis meliputi suhu udara, kelembapan relatif, curah hujan, dan intensitas cahaya, karena variabel-variabel ini memiliki pengaruh langsung terhadap kondisi atmosfer. Suhu udara memberikan gambaran tentang keadaan termal di lingkungan yang berperan dalam berbagai proses atmosferik, sementara kelembapan relatif mengindikasikan jumlah uap air yang ada di udara, yang berkaitan dengan pembentukan awan dan kemungkinan hujan. Data curah hujan merepresentasikan volume air yang jatuh ke permukaan dan penting untuk analisis hidrologi serta pengelolaan risiko banjir. Sementara itu, intensitas cahaya, yang diukur dalam satuan lux, digunakan untuk menentukan kondisi pencahayaan yang berkaitan dengan cuaca seperti cerah atau mendung, serta memengaruhi aktivitas biologis seperti fotosintesis. Selain itu, parameter tambahan seperti tekanan udara, kecepatan dan arah angin, serta

faktor geografis seperti ketinggian dan posisi lokasi juga sering dimasukkan dalam dataset untuk meningkatkan akurasi prediksi.

Agar model dapat memberikan hasil yang andal, data dari berbagai sensor harus diproses terlebih dahulu melalui tahapan pra-pemrosesan yang mencakup penghilangan noise, penanganan data yang hilang, serta normalisasi agar skala data sesuai untuk pemodelan machine learning. Langkah ini penting untuk memastikan model dapat mendeteksi pola dengan lebih efektif tanpa terganggu oleh data yang tidak valid atau inkonsisten. Dengan memasukkan data meteorologi yang telah dibersihkan dan berkualitas tinggi, model machine learning mampu memahami hubungan kompleks antar variabel dan menghasilkan prediksi cuaca yang lebih presisi dan adaptif terhadap perubahan lingkungan. Prediksi yang dihasilkan sangat bermanfaat dalam berbagai bidang, seperti pertanian presisi, manajemen sumber daya air, dan mitigasi bencana, yang memerlukan keputusan berbasis data dengan tingkat ketepatan dan kecepatan tinggi.

### **2.3 Algoritma Machine Learning untuk Klasifikasi**

Algoritma Decision Tree sering digunakan untuk klasifikasi kondisi cuaca karena kemampuannya dalam mengolah berbagai jenis data, baik yang bersifat numerik maupun kategorikal, dengan efektif. Algoritma ini bekerja dengan membagi data secara bertahap berdasarkan fitur-fitur yang paling signifikan untuk membedakan kelas target, seperti suhu, kelembapan, kecepatan angin, dan jenis cuaca. Proses pemisahan data tersebut dilakukan dengan menggunakan metrik seperti Information Gain atau Gini Index untuk memilih fitur dan nilai ambang yang terbaik, sehingga data terbagi menjadi kelompok-kelompok yang lebih homogen. Hasil akhirnya berupa struktur pohon keputusan yang mudah diikuti, di mana setiap cabang mewakili aturan berdasarkan fitur tertentu dan setiap daun menunjukkan prediksi kelas cuaca.

Selain memiliki akurasi yang baik dalam klasifikasi, Decision Tree juga unggul karena hasil modelnya mudah dipahami. Model ini menghasilkan aturan yang jelas sehingga memungkinkan para ahli meteorologi maupun pengguna lain untuk memeriksa dan mengerti bagaimana prediksi tersebut dibuat. Selain itu, Decision Tree relatif cepat dalam proses pelatihan dan prediksi serta tidak memerlukan banyak tahapan praproses data seperti normalisasi. Meskipun rentan terhadap overfitting, masalah ini dapat diatasi dengan teknik pemangkasan pohon dan pembatasan kedalaman. Karena keunggulan-keunggulan tersebut, Decision Tree menjadi salah satu pilihan terbaik untuk aplikasi prediksi cuaca yang membutuhkan model yang transparan, efisien, dan andal.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Desain Sistem**

Sistem ini dirancang dengan modul utama berupa antarmuka pengguna grafis (GUI) yang dikembangkan menggunakan framework PyQt6. Antarmuka terdiri atas tiga halaman navigasi utama, yaitu Dashboard, Training, dan Evaluasi. Fokus utama dari penelitian ini adalah pada implementasi halaman Evaluasi yang berfungsi untuk menampilkan hasil pengujian model klasifikasi Decision Tree terhadap data uji.

Data uji dimasukkan ke dalam sistem melalui fitur unggah file Excel (.xlsx) yang telah disediakan pada antarmuka GUI. File Excel tersebut berisi data dummy yang merepresentasikan parameter-parameter cuaca yang digunakan dalam proses klasifikasi, yaitu suhu udara, kelembapan, intensitas cahaya, dan curah hujan. Data ini kemudian diproses oleh sistem untuk dilakukan evaluasi terhadap performa model machine learning yang telah dilatih sebelumnya

#### **3.2 Arsitektur Aplikasi**

Arsitektur aplikasi prediksi cuaca ini dirancang modular dengan tiga fitur utama: Dashboard, Training, dan Evaluasi, yang diatur menggunakan `QStackedWidget` dalam antarmuka PyQt6. Setiap fitur memiliki fungsi spesifik dan navigasi antar halaman dilakukan melalui tombol menu.

Fungsi Evaluasi memungkinkan pemuatan model, encoder, dan data uji, lalu menampilkan hasil evaluasi berupa metrik akurasi, presisi, recall, F1-score, MAE, dan MSE. Aplikasi juga menyediakan visualisasi Confusion Matrix, Feature Importance, dan Learning Curve, sehingga mendukung analisis performa model secara komprehensif.

#### **3.3 Algoritma Decision Tree**

Decision Tree adalah metode pembelajaran mesin yang membentuk model berupa pohon keputusan, di mana setiap node menguji atribut tertentu untuk membagi data menjadi kelompok yang lebih homogen. Model ini menggunakan aturan sederhana untuk mengklasifikasikan data berdasarkan fitur yang ada, sehingga mudah dipahami dan diinterpretasikan.

Pada sistem prediksi cuaca, Decision Tree memetakan hubungan antara data sensor seperti suhu, kelembapan, dan kondisi hujan dengan kategori cuaca. Model dilatih menggunakan data berlabel dan mengoptimalkan pembagian data agar mampu mengenali pola cuaca. Pengaturan parameter seperti kedalaman pohon membantu menghindari overfitting dan menjaga kemampuan model dalam memprediksi data baru.

Keunggulan Decision Tree adalah kemampuannya menangani data numerik dan kategorikal tanpa perlu pra-pemrosesan kompleks serta kemudahannya dalam

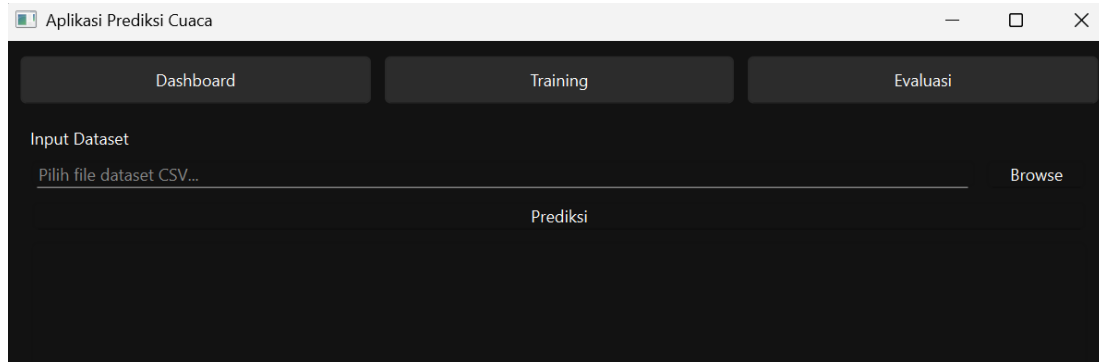
menampilkan pentingnya fitur. Model ini memungkinkan evaluasi performa secara komprehensif dan memberikan hasil prediksi yang akurat dan transparan, sehingga cocok digunakan dalam aplikasi prediksi cuaca berbasis sensor.



## BAB IV

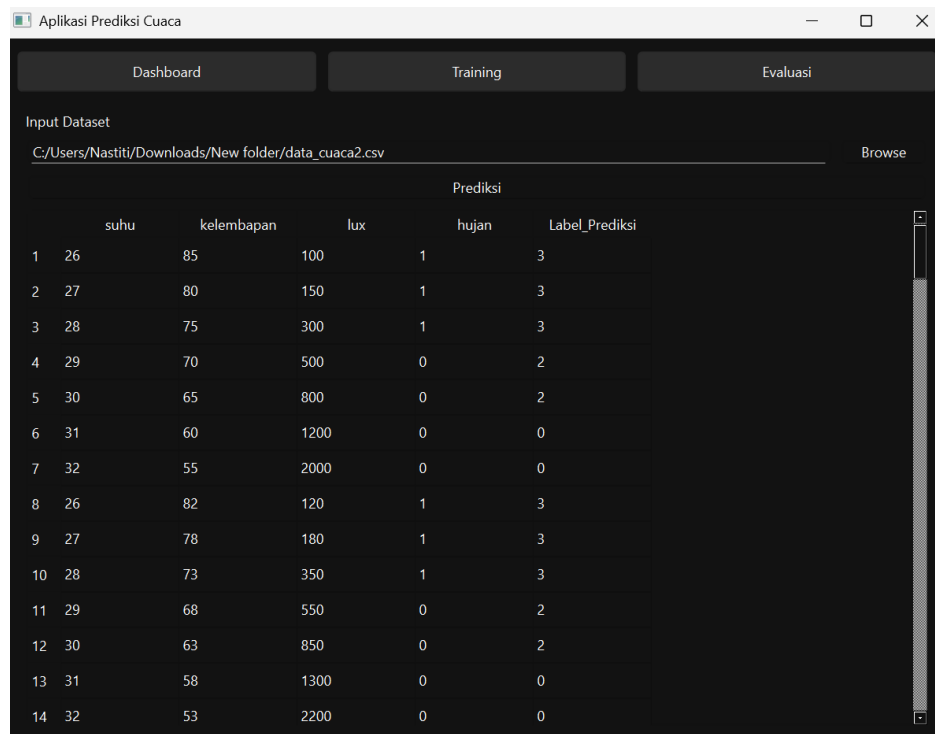
### Hasil Dan Pembahasan

#### 4.1 Tampilan GUI



**Gambar 1. Tampilan Dashboard**

Pada tampilan GUI aplikasi prediksi cuaca terdapat beberapa fitur utama yang memudahkan pengguna dalam mengoperasikan sistem. Fitur Pilih file dataset CSV digunakan untuk memasukkan file data cuaca dalam format CSV sebagai bahan dasar untuk melakukan prediksi. Setelah data dimuat, pengguna dapat menekan tombol Prediksi untuk menjalankan proses prediksi cuaca berdasarkan data yang telah diinput.

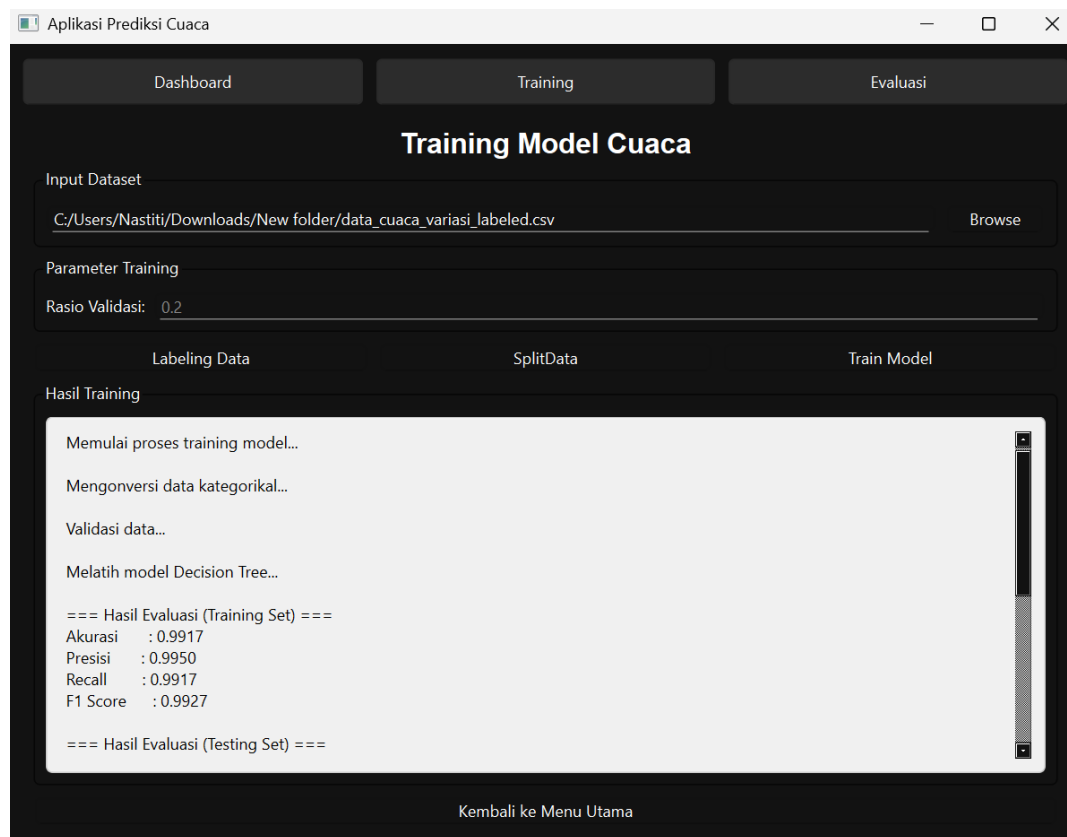


	suhu	kelembapan	lux	hujan	Label_Prediksi
1	26	85	100	1	3
2	27	80	150	1	3
3	28	75	300	1	3
4	29	70	500	0	2
5	30	65	800	0	2
6	31	60	1200	0	0
7	32	55	2000	0	0
8	26	82	120	1	3
9	27	78	180	1	3
10	28	73	350	1	3
11	29	68	550	0	2
12	30	63	850	0	2
13	31	58	1300	0	0
14	32	53	2200	0	0

**Gambar 2. Tampilan Hasil prediksi**

Pada tampilan prediksi cuaca ini, sistem telah berhasil memuat dataset dari file CSV yang ditampilkan di bagian Input Dataset. Setelah file dimuat, sistem secara otomatis menampilkan tabel berisi data dan hasil prediksi. Tabel ini terdiri dari beberapa kolom yaitu suhu, kelembapan, lux, dan hujan sebagai fitur input, serta kolom Label\_Prediksi yang menunjukkan hasil prediksi dari model.

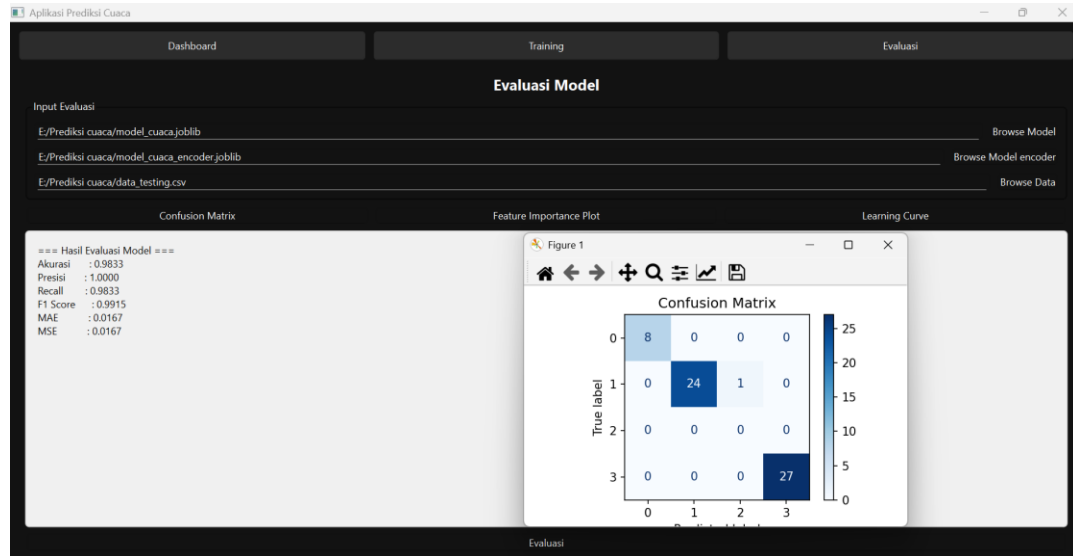
Setiap baris pada tabel merepresentasikan satu set data cuaca, di mana nilai-nilai pada kolom input digunakan oleh model untuk menghasilkan label prediksi. Misalnya, data dengan suhu 26°C, kelembapan 85%, lux 100, dan hujan 1 menghasilkan Label\_Prediksi bernilai 3. Label ini kemungkinan mengacu pada kategori kondisi cuaca tertentu (misalnya: 0 = cerah, 1 = berawan, 2 = mendung, 3 = hujan).



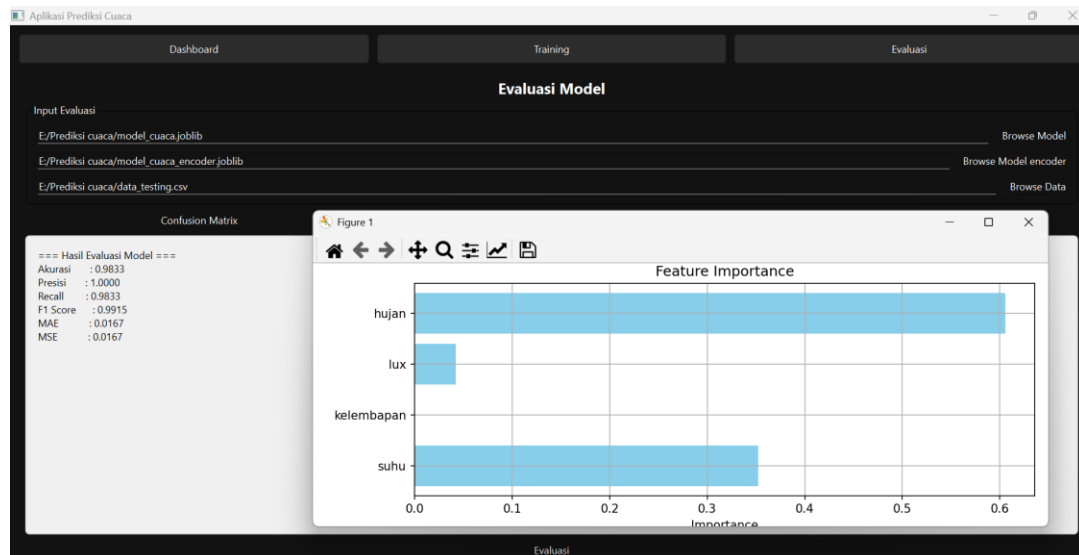
**Gambar 3. Hasil Training Model**

Halaman Training Model Cuaca berfungsi untuk melakukan pelatihan model AI berdasarkan dataset cuaca yang telah diberi label. Pengguna dapat mengunggah file dataset melalui tombol Browse, kemudian menentukan rasio validasi yang digunakan untuk membagi data antara pelatihan dan pengujian. Dalam tampilan ini, rasio validasi ditetapkan sebesar 20%. Setelah konfigurasi dilakukan, sistem akan memulai proses pelatihan dengan mengonversi data kategorikal, melakukan validasi, dan melatih model menggunakan algoritma Decision Tree. Hasil pelatihan ditampilkan

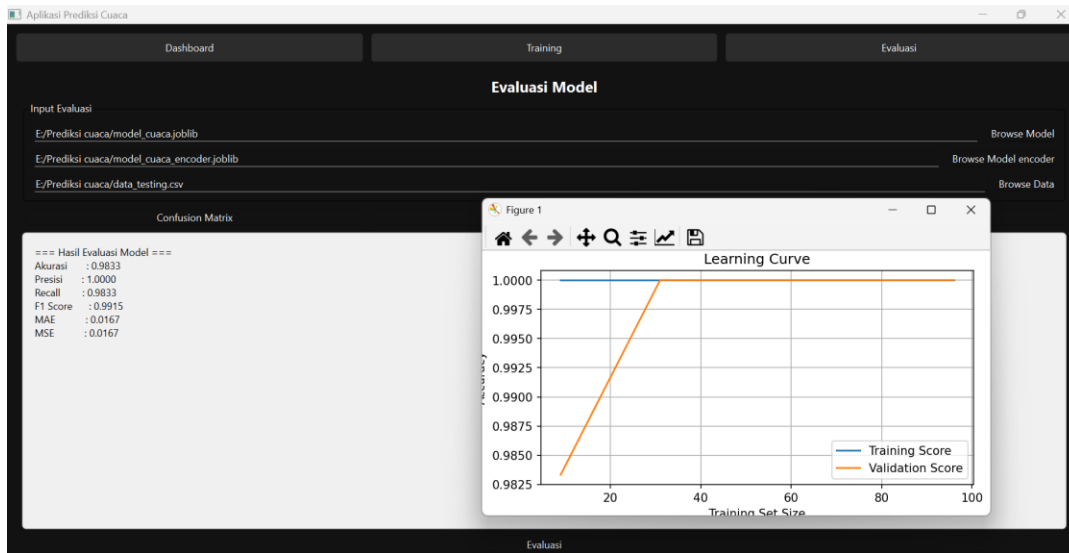
dalam bentuk metrik performa seperti akurasi sebesar 0.9917, presisi 0.9950, recall 0.9917, dan F1 Score 0.9927. Angka-angka ini menunjukkan bahwa model memiliki performa yang sangat baik pada data pelatihan. Tampilan ini memberikan kemudahan bagi pengguna untuk memahami proses pelatihan dan mengevaluasi hasilnya sebelum digunakan dalam proses prediksi.



**Gambar 4. Hasil Evaluasi model dengan confusion Matrix**



**Gambar 5. Hasil Evaluasi model dengan Feature Importance**



**Gambar 6. Hasil Evaluasi model dengan Learning Curve**

Halaman Evaluasi Model pada aplikasi prediksi cuaca ini dirancang untuk memberikan analisis lengkap terhadap performa model machine learning yang telah dilatih sebelumnya. Di bagian atas terdapat navigasi berupa tombol "Dashboard", "Training", dan "Evaluasi", dengan halaman Evaluasi yang sedang aktif ditandai secara visual. Judul besar "Evaluasi Model" berada di tengah halaman untuk menandai fokus pengguna saat ini.

Bagian utama halaman terdiri dari form input evaluasi yang memungkinkan pengguna untuk memilih tiga file penting: model .joblib, file encoder label .joblib, dan data pengujian .csv. Setiap input dilengkapi tombol "Browse" untuk mempermudah pemilihan file melalui antarmuka grafis. Di bawah form ini terdapat tiga tombol tab: "Confusion Matrix", "Feature Importance Plot", dan "Learning Curve" yang ketika diklik akan menampilkan grafik evaluasi model pada jendela terpisah menggunakan matplotlib.

Tepat di bawah tab grafik, terdapat area hasil evaluasi yang menampilkan metrik performa model seperti Akurasi, Presisi, Recall, F1 Score, MAE, dan MSE. Nilai-nilai tersebut menunjukkan bahwa model memiliki performa sangat baik, dengan akurasi tinggi mencapai 98.33% dan presisi sempurna (100%). Grafik Confusion Matrix memperlihatkan distribusi prediksi benar dan salah antar kelas, dengan kesalahan prediksi yang sangat minimal. Plot Feature Importance menyoroti bahwa fitur hujan memiliki kontribusi paling besar dalam prediksi model, sementara lux paling rendah. Sementara itu, grafik Learning Curve menunjukkan bahwa performa model konsisten dan stabil, dengan akurasi tinggi baik pada data pelatihan maupun

validasi, mengindikasikan bahwa model tidak mengalami overfitting maupun underfitting. Secara keseluruhan, halaman ini memberikan gambaran evaluasi model yang lengkap, informatif, dan mudah dipahami.

## 4.2 Implementasi Sistem

```
self.results_text.append("\nMelatih model Decision Tree...")

model = DecisionTreeClassifier(

criterion='entropy',

max_depth=3, # Lebih konservatif untuk hindari overfitting

min_samples_split=10,

min_samples_leaf=5,

random_state=42

)

model.fit(X_train, y_train)
```

Proses pelatihan model dilakukan menggunakan algoritma Decision Tree Classifier yang diambil dari pustaka scikit-learn. Model ini dikonfigurasi dengan parameter tertentu untuk menghasilkan model yang akurat namun tidak terlalu kompleks. Penggunaan `criterion='entropy'` bertujuan agar pemilihan fitur saat membentuk pohon keputusan didasarkan pada perolehan informasi terbaik. Selain itu, pengaturan `max\_depth=3` digunakan untuk membatasi kedalaman maksimum pohon guna menghindari overfitting. Parameter `min\_samples\_split=10` dan `min\_samples\_leaf=5` memastikan bahwa sebuah node hanya akan dipecah jika terdapat cukup banyak data, sehingga pohon keputusan tidak terlalu menyesuaikan diri dengan data pelatihan.

Pelatihan model dimulai dengan menjalankan fungsi `fit(X\_train, y\_train)` yang memasukkan data fitur dan label ke dalam model. Sebelum proses ini dimulai, sistem akan memberikan notifikasi berupa teks “Melatih model Decision Tree...” pada antarmuka aplikasi, sebagai indikator bahwa pelatihan sedang berlangsung. Konfigurasi yang digunakan ini dirancang agar model mampu mengenali pola-pola penting dalam data pelatihan secara efisien dan tetap mempertahankan kemampuannya untuk melakukan prediksi terhadap data yang belum pernah dilihat sebelumnya. Hasil dari pelatihan ini kemudian dapat dievaluasi lebih lanjut menggunakan data uji untuk mengukur performa model secara kuantitatif melalui metrik seperti akurasi, presisi, recall, dan lainnya.

### 4.3 Hasil Evaluasi Model

Berikut ialah penjelasan mengenai hasil evaluasi model yang terdapat pada bagian evaluasi:

- **Akurasi (Accuracy)**  
Akurasi mengukur proporsi prediksi yang benar dibandingkan dengan seluruh data uji. Nilai 0.9833 berarti 98.33% dari seluruh data berhasil diprediksi dengan benar oleh model. Ini menunjukkan performa model yang sangat baik secara keseluruhan.
- **Presisi (Precision): 1.0000**  
Presisi adalah rasio antara jumlah prediksi positif yang benar dibandingkan dengan seluruh prediksi positif yang dibuat oleh model. Nilai 1.0000 (atau 100%) menandakan bahwa semua prediksi positif model adalah benar, artinya tidak ada kesalahan dalam memprediksi kelas positif. Presisi yang sempurna sangat penting jika kesalahan prediksi positif memiliki dampak serius.
- **Recall: 0.9833**  
Recall mengukur seberapa baik model dapat menemukan seluruh data aktual positif dari dataset. Nilai 0.9833 berarti model mampu mengidentifikasi 98.33% dari seluruh data aktual positif, artinya hanya sebagian kecil kasus positif yang tidak terdeteksi (false negatives).
- **F1 Score: 0.9915**  
F1 Score merupakan rata-rata harmonis dari presisi dan recall, dan sangat berguna saat terdapat ketidakseimbangan antar kelas. Dengan nilai 0.9915, ini menunjukkan keseimbangan yang sangat baik antara presisi dan recall, menandakan model sangat andal dalam mendeteksi kelas yang benar secara konsisten.
- **MAE (Mean Absolute Error): 0.0167**  
MAE mengukur rata-rata dari selisih absolut antara nilai aktual dan prediksi. Nilai 0.0167 menunjukkan bahwa rata-rata kesalahan prediksi model adalah sekitar 1.67% dari nilai sebenarnya, yang sangat kecil dan menunjukkan ketelitian tinggi.
- **MSE (Mean Squared Error): 0.0167**  
MSE adalah rata-rata dari kuadrat selisih antara nilai aktual dan prediksi. Meskipun mirip dengan MAE, MSE lebih sensitif terhadap kesalahan besar. Nilai 0.0167 menunjukkan bahwa kesalahan model sangat kecil dan tidak ada prediksi yang meleset jauh dari nilai sebenarnya.

### 4.4 Evaluasi Pemilihan Algoritma dalam Implementasi Model Prediksi

Pada implementasi aplikasi prediksi cuaca ini, digunakan algoritma Decision Tree Classifier dari pustaka scikit-learn untuk membangun model prediktif. Model ini

dilatih menggunakan parameter tertentu seperti `criterion='entropy'`, `max_depth=3`, serta batas minimum jumlah sampel pada pemisahan node guna menghindari overfitting. Proses pelatihan dilakukan melalui antarmuka pengguna pada halaman Training, di mana pengguna dapat memuat dataset, memberi label otomatis pada data, membagi data menjadi training dan testing, serta melatih model hingga menyimpan hasilnya. Evaluasi performa dilakukan dengan metrik klasifikasi seperti akurasi, presisi, recall, dan F1-score. Selanjutnya, model yang telah dilatih digunakan pada halaman Dashboard untuk memprediksi kondisi cuaca berdasarkan input dari pengguna.

Namun setelah dilakukan peninjauan, ditemukan bahwa algoritma yang digunakan tidak sepenuhnya sesuai dengan karakteristik target yang diprediksi. Karena model ditujukan untuk memperkirakan nilai-nilai cuaca yang cenderung bersifat numerik dan kontinu (seperti suhu atau intensitas cahaya), maka algoritma yang tepat seharusnya adalah Decision Tree Regressor. Penggunaan classifier secara tidak sengaja menyebabkan sistem hanya mengelompokkan data ke dalam kategori diskrit, yang berpotensi mengurangi ketepatan prediksi jika data sebenarnya bersifat kontinu. Kesalahan ini menjadi catatan penting dalam proses pengembangan, dan menunjukkan bahwa pemilihan algoritma harus mempertimbangkan dengan tepat jenis data target yang akan diprediksi agar sistem dapat bekerja secara optimal.

## **BAB V**

### **KESIMPULAN**

Aplikasi prediksi cuaca berbasis machine learning dengan algoritma Decision Tree yang dikembangkan menggunakan PyQt6 berhasil memudahkan pengguna dalam melakukan pelatihan dan evaluasi model secara interaktif. Aplikasi ini mampu mengolah data seperti suhu, kelembapan, intensitas cahaya, dan curah hujan untuk mengklasifikasikan kondisi cuaca dengan tingkat akurasi yang memuaskan, serta menampilkan visualisasi penting seperti confusion matrix dan feature importance yang membantu pemahaman performa model.

Namun, penggunaan Decision Tree Classifier kurang sesuai jika target prediksi berupa data numerik kontinu, sehingga Decision Tree Regressor lebih dianjurkan untuk kasus tersebut agar hasil prediksi lebih akurat. Secara keseluruhan, aplikasi ini memberikan solusi praktis dan efektif dalam memprediksi cuaca berdasarkan data historis, yang berpotensi berguna untuk berbagai keperluan seperti pertanian, perencanaan kegiatan, dan mitigasi risiko.