BAB II

TINJAUAN LITERATUR

2.1 Penelitian Terkait

Decision Tree adalah metode pembelajaran mesin yang menggunakan struktur pohon untuk merepresentasikan keputusan dan kemungkinan konsekuensinya. Setiap node internal merepresentasikan sebuah fitur, setiap cabang merepresentasikan aturan keputusan, dan setiap node daun merepresentasikan hasil klasifikasi. Algoritma ini termasuk dalam kategori supervised learning dan dapat digunakan untuk tugas klasifikasi maupun regresi (Irwan *et al.*, 2022). Dalam konteks pendidikan, Decision Tree terbukti menghasilkan model yang mudah dipahami karena menghasilkan aturan IF-THEN yang dapat diinterpretasikan secara langsung oleh pengguna non-teknis

Penelitian tentang prediksi kelulusan mahasiswa telah banyak dilakukan menggunakan pendekatan *data mining* dan *machine learning*, di mana kelulusan tepat waktu menjadi indikator penting mutu perguruan tinggi yang dipengaruhi oleh faktor akademik (seperti IPK dan IPS) maupun non-akademik (seperti usia dan status pekerjaan). Algoritma *Decision Tree*, khususnya *Classification and Regression Tree* (CART), terbukti efektif karena mampu menghasilkan model klasifikasi dengan akurasi tinggi sekaligus mudah diinterpretasikan melalui struktur pohon keputusan berbasis *Gini Index*. Penelitian sebelumnya menunjukkan bahwa optimasi hiperparameter CART, seperti pengaturan kedalaman pohon dan jumlah minimum sampel per node, dapat meningkatkan performa prediksi tanpa

mengorbankan interpretabilitas, dengan akurasi mencapai 92,1% serta keseimbangan *precision* dan *recall* yang baik pada kedua kelas kelulusan. Pendekatan ini dinilai relevan untuk dikembangkan sebagai sistem peringatan dini guna mendukung intervensi akademik yang lebih terarah, meskipun masih diperlukan pengayaan variabel dan perbandingan dengan algoritma lain untuk menghasilkan model yang lebih komprehensif (Alfiani Abdilah *et al.*, 2025).

Penelitian tentang prediksi kelulusan mahasiswa menggunakan pendekatan *data mining* telah banyak diterapkan di lingkungan pendidikan tinggi, khususnya melalui algoritma klasifikasi C4.5 yang merupakan pengembangan dari ID3 dengan menggunakan *gain ratio* untuk menghindari bias atribut dan mampu menangani data kontinu, *missing values*, serta melakukan *pruning* untuk mencegah *overfitting*. Algoritma C4.5 terbukti efektif dalam menganalisis data akademik seperti Indeks Prestasi Kumulatif (IPK), jumlah Satuan Kredit Semester (SKS) yang telah ditempuh, dan tingkat kehadiran mahasiswa, di mana IPK menjadi atribut paling dominan dalam menentukan prediksi kelulusan, diikuti oleh kehadiran dan progres SKS. Penelitian terdahulu menunjukkan bahwa C4.5 memiliki akurasi yang konsisten, dengan studi terkait mencapai akurasi 84,6% serta nilai *precision* dan *recall* yang baik, sekaligus menghasilkan pohon keputusan yang mudah diinterpretasikan oleh pihak akademik non-teknis. Metode ini dinilai relevan sebagai sistem peringatan dini untuk mendeteksi mahasiswa berisiko keterlambatan studi, dengan evaluasi menggunakan *confusion matrix* dan *cross-validation* untuk memastikan stabilitas model, serta berpotensi dikembangkan lebih lanjut dengan penambahan variabel non-akademik dan perbandingan dengan

algoritma lain seperti *Random Forest* atau *SVM (Nuari Anisa Sivi, Rudi Hartono and Putra Hanafi, 2023)*.

Penelitian tentang penentuan prestasi siswa telah menerapkan pendekatan data mining dengan menggabungkan algoritma clustering K-Means dan klasifikasi Decision Tree untuk mengidentifikasi faktor-faktor yang memengaruhi prestasi belajar. Algoritma K-Means digunakan untuk mengelompokkan siswa berdasarkan kemiripan nilai akademik (matematika, bahasa Indonesia, IPA, agama) dan prestasi sekolah, dengan pembagian menjadi tiga kluster yaitu "Cukup Baik", "Baik", dan "Sangat Baik" melalui perhitungan Euclidean Distance dan penentuan centroid. Berdasarkan pengujian pada dataset 28 siswa SDS Kartika X-6, model mencapai akurasi prediksi sebesar 71%, dengan atribut nilai IPA dan agama terbukti paling berpengaruh dalam menentukan prestasi siswa, sementara nilai matematika, bahasa Indonesia, dan prestasi kelas juga memberikan kontribusi. Penelitian ini menunjukkan bahwa kombinasi K-Means dan Decision Tree efektif untuk mengelompokkan dan memprediksi prestasi siswa, serta membedakan dengan penelitian sebelumnya yang menggunakan variabel pendapatan orang tua dan jarak rumah ke sekolah, sehingga disarankan untuk penelitian selanjutnya menambahkan atribut lain dan memperbesar jumlah data untuk meningkatkan akurasi (Jevintya, Darussalam and Abdullah, 2024).

Penelitian tentang klasifikasi mahasiswa yang mengikuti ujian perbaikan/remidial (her) telah menerapkan algoritma Support Vector Machine (SVM) dan Decision Tree (DT) sebagai metode klasifikasi supervised

learning untuk mengidentifikasi faktor-faktor yang memengaruhi mahasiswa dalam mengikuti her, seperti usia, jenis kelamin, status kuliah sambil bekerja, dan jarak tempuh ke kampus. SVM memiliki keunggulan dalam menangani data berdimensi tinggi dan sampling kecil melalui pemetaan non-linier dengan fungsi kernel, serta mampu meningkatkan akurasi dan mengurangi kesalahan klasifikasi, sementara Decision Tree dikenal karena kemudahan interpretasinya melalui aturan keputusan yang dapat dipahami manusia, meskipun memiliki kelemahan dalam hal kecepatan pemrosesan data berukuran besar. Penelitian terdahulu menunjukkan bahwa SVM efektif untuk klasifikasi multi-kelas dan berbagai aplikasi seperti prediksi beban listrik dan analisis citra otak, sedangkan Decision Tree seperti J48 telah digunakan untuk prediksi performa akademik siswa dengan hasil yang kompetitif. Berdasarkan pengujian pada 2.264 data mahasiswa, Decision Tree menghasilkan akurasi 76,84% dan AUC 0,748, sedikit lebih unggul dibandingkan SVM dengan akurasi 76,80% dan AUC 0,584, meskipun SVM memiliki recall lebih tinggi (99,94%) dibandingkan Decision Tree (99,48%). Hasil ini menunjukkan bahwa kedua algoritma dapat digunakan sebagai alat evaluasi bagi perguruan tinggi untuk mengantisipasi mahasiswa berisiko mengikuti her, dengan saran penelitian selanjutnya untuk menambahkan fitur atau optimasi guna meningkatkan performa model (Purnama *et al.*, 2020).

Penelitian tentang prediksi masa studi mahasiswa telah menerapkan metode *Decision Tree* dengan algoritma *Classification and Regression Tree* (CART) untuk mengidentifikasi parameter yang paling berpengaruh terhadap kelulusan tepat waktu, berdasarkan data akademik dan data wisudawan dari

Fakultas Teknik Universitas Bengkulu. Algoritma CART berbeda dengan ID3 dan C4.5 karena menggunakan perhitungan *Gini Index* dan *Gini Gain* untuk pembentukan cabang pohon keputusan, serta mampu menangani atribut numerik dan kategorikal. Penelitian terdahulu menunjukkan bahwa CART memiliki akurasi lebih baik dibandingkan C5.0 (84,63% vs 79,17%) dan telah digunakan untuk prediksi kelulusan mahasiswa dengan atribut seperti IPK, jalur masuk, dan jenis kelamin. Dalam penelitian ini, data melalui tahap *preprocessing* menggunakan sistem *data warehouse* dengan proses ETL (*Extract, Transform, Load*) untuk membersihkan data dari *missing values* dan mengintegrasikan berbagai sumber data. Hasil pengujian menunjukkan bahwa parameter paling berpengaruh terhadap masa studi adalah IPK (akurasi 88,52%), diikuti oleh jalur masuk, jurusan, domisili, asal sekolah, dan jenis kelamin, dengan akurasi keseluruhan model mencapai 80,22% pada penggunaan 100% dataset, yang membuktikan bahwa semakin banyak data maka semakin tinggi akurasi prediksi. Penelitian ini juga menghasilkan aplikasi prediksi masa studi yang dapat digunakan untuk mengelompokkan data alumni dan memprediksi kelulusan mahasiswa baru, serta disarankan untuk pengembangan selanjutnya menggunakan metode lain seperti KNN atau C5.0 untuk perbandingan performa (Rizalno, Johar and Coastera, 2022).

2.2 Data Mining

Data mining merupakan proses penggalian pola, hubungan, atau informasi yang bermanfaat dari sekumpulan data berukuran besar melalui pendekatan statistik, matematika, dan kecerdasan buatan. Proses data mining umumnya mencakup tahapan pembersihan data (data cleaning), integrasi data, transformasi

data, penggalian pola (pattern mining), evaluasi pola, dan presentasi pengetahuan yang dihasilkan (Ridwansyah *et al.*, 2024). Dalam konteks penelitian ini, data mining diterapkan pada data kelulusan mahasiswa untuk menemukan pola yang membedakan mahasiswa yang lulus tepat waktu dengan yang mengalami keterlambatan.

Data mining merupakan salah satu tahapan inti dalam proses yang lebih luas dikenal sebagai Knowledge Discovery in Databases (KDD), yang secara umum terdiri atas tahapan (Wicandi, Utami and Dewi, 2026) :

1. Pemilihan data (data selection), yaitu memilih data yang relevan dari sumber data yang tersedia;
2. Pra-pemrosesan (pre-processing), yaitu membersihkan data dari nilai kosong, duplikat, atau tidak konsisten;
3. Transformasi data (transformation), yaitu mengubah data ke dalam format yang sesuai untuk proses penambangan, misalnya normalisasi atau pembentukan variabel turunan;
4. Penambangan data (data mining), yaitu penerapan algoritma untuk menemukan pola;
5. Evaluasi pola (pattern evaluation), yaitu menilai kualitas dan kegunaan pola yang ditemukan; dan
6. Presentasi pengetahuan (knowledge presentation), yaitu menyajikan hasil dalam bentuk yang mudah dipahami, misalnya visualisasi atau laporan.

Secara umum, teknik data mining dapat dikelompokkan menjadi beberapa kategori, yaitu klasifikasi (classification), klasterisasi (clustering), asosiasi

(association rule mining), dan regresi (regression). Penelitian ini menggunakan teknik klasifikasi, karena tujuan utamanya adalah memprediksi label kategorikal (Status Kelulusan) berdasarkan sejumlah atribut input. Keenam tahapan KDD di atas diterapkan secara langsung pada penelitian ini: mulai dari pemilihan atribut yang relevan dari berkas kelulusan mentah, pembersihan baris data yang tidak valid, transformasi menjadi variabel kategori (Kategori IPK, Status Kelulusan), penerapan algoritma Decision Tree, evaluasi menggunakan confusion matrix dan cross validation, hingga presentasi hasil melalui visualisasi pohon keputusan pada antarmuka aplikasi.

2.3 Klasifikasi Dalam Data Mining

Klasifikasi merupakan salah satu teknik data mining yang termasuk dalam kategori pembelajaran terarah (supervised learning), yaitu proses membangun model dari data yang telah memiliki label/kelas target untuk kemudian digunakan memprediksi label dari data baru yang belum diketahui kelasnya. Proses klasifikasi umumnya terbagi menjadi dua tahap: tahap pembelajaran (training), di mana model mempelajari pola dari data latih (training set) yang telah diketahui labelnya dan tahap pengujian (testing), di mana model yang telah terbentuk diuji kemampuannya memprediksi label pada data uji (testing set) yang belum pernah dilihat sebelumnya, untuk mengukur kemampuan generalisasi model terhadap data baru (Rahmadeyan and Mustakim, 2023).

Beberapa algoritma klasifikasi yang umum digunakan dalam data mining di antaranya Decision Tree, Naive Bayes, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), dan Artificial Neural Network (ANN). Masing-masing

algoritma memiliki karakteristik, kelebihan, dan kelemahan yang berbeda. Decision Tree misalnya unggul dalam interpretabilitas namun rentan overfitting pada data yang kompleks, sementara ANN umumnya memiliki akurasi lebih tinggi pada pola yang rumit namun bersifat black box (sulit diinterpretasikan) (Nur Sofiyanto *et al.*, 2025). Pada penelitian ini, kelas target yang diprediksi adalah Status Kelulusan, dengan dua kemungkinan nilai yaitu "Tepat Waktu" dan "Terlambat", sehingga permasalahan ini termasuk dalam kategori klasifikasi biner (binary classification).

2.4 Decision Tree (Pohon Keputusan)

Decision Tree adalah salah satu metode klasifikasi yang merepresentasikan pengetahuan dalam bentuk struktur pohon, terdiri atas simpul akar (root node) yang merupakan atribut dengan daya pisah terbaik terhadap keseluruhan data, simpul internal (internal node) yang merepresentasikan pengujian terhadap suatu atribut, cabang (branch) yang merepresentasikan hasil pengujian tersebut, dan simpul daun (leaf node) yang merepresentasikan label kelas akhir (Nasrullah, 2021). Keunggulan utama Decision Tree dibandingkan metode klasifikasi lain adalah kemudahan interpretasi, karena aturan keputusan yang terbentuk dapat divisualisasikan dan dipahami secara langsung tanpa memerlukan pengetahuan statistik yang mendalam sebuah pertimbangan penting pada penelitian ini karena hasil analisis ditujukan untuk digunakan oleh pihak akademik yang tidak selalu memiliki latar belakang teknis.

Proses pembentukan pohon keputusan dilakukan secara rekursif melalui algoritma yang dikenal sebagai Top-Down Induction of Decision Trees, dengan langkah umum sebagai berikut (Setio, Saputro and Bowo Winarno, 2020):

1. memilih atribut terbaik sebagai simpul menggunakan ukuran impuritas tertentu (Gini Index atau Entropy);
2. membagi data pada simpul tersebut berdasarkan nilai atribut yang dipilih;
3. mengulangi proses tersebut secara rekursif pada setiap cabang yang terbentuk hingga mencapai kondisi berhenti (stopping criteria), misalnya seluruh data pada simpul telah homogen (satu kelas saja), tidak ada atribut tersisa untuk dipisahkan, atau telah mencapai batas kedalaman maksimum (max depth) yang ditentukan.

Penelitian ini menggunakan implementasi Decision Tree dari pustaka Scikit-learn, yang menerapkan algoritma CART (Classification and Regression Trees) — varian Decision Tree yang secara khusus membentuk pohon biner (binary tree), yaitu setiap simpul internal hanya memiliki tepat dua cabang.

Salah satu tantangan utama dalam pembentukan Decision Tree adalah risiko overfitting, yaitu kondisi ketika pohon tumbuh terlalu dalam dan kompleks sehingga terlalu menyesuaikan diri dengan data latih, namun berkinerja buruk pada data baru. Untuk mengatasi hal ini, diterapkan teknik pemangkasan (pruning), yang terbagi menjadi dua pendekatan: pre-pruning, yaitu membatasi pertumbuhan pohon sejak awal melalui parameter seperti kedalaman maksimum (max_depth) dan jumlah sampel minimum untuk pemisahan simpul (min_samples_split); dan post-pruning, yaitu membiarkan pohon tumbuh penuh terlebih dahulu kemudian memangkas cabang-cabang yang tidak signifikan setelah pohon terbentuk. Penelitian ini menerapkan pendekatan pre-pruning melalui parameter Max Depth

dan Min Samples Split yang dapat diatur langsung oleh pengguna pada antarmuka aplikasi.

2.5 Gini Index, Entropy, dan Information Gain

Gini Index dan Entropy merupakan dua ukuran impuritas (ketidakmurnian) yang umum digunakan sebagai kriteria pemilihan atribut pemisah (split) terbaik pada setiap simpul pohon keputusan. Gini Index mengukur seberapa sering suatu elemen yang dipilih secara acak akan salah diklasifikasikan apabila diberi label secara acak sesuai distribusi kelas pada simpul tersebut, dirumuskan sebagai (Banafshah Shafa *et al.*, 2024):

$$Gini(t) = 1 - \sum_{i=1}^{n} \big(p(i|t)\big)^2$$

Dengan p(i|t) adalah proporsi data kelas i pada simpul t. Entropy mengukur tingkat ketidakteraturan (disorder) suatu kumpulan data, dirumuskan sebagai:

$$Entropy(t) = -\sum_{i=1}^{n} (p(i|t) \log_2 p(i|t))$$

Semakin rendah nilai Gini Index atau Entropy suatu simpul (mendekati 0), semakin murni (homogen) komposisi kelas pada simpul tersebut — nilai 0 menandakan seluruh data pada simpul berasal dari satu kelas yang sama. Untuk menentukan atribut mana yang digunakan sebagai pemisah pada suatu simpul, algoritma menghitung penurunan impuritas (impurity decrease) yang dihasilkan oleh setiap kandidat atribut, dikenal sebagai Information Gain apabila menggunakan Entropy, dirumuskan sebagai:

$$IG(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Dengan S adalah himpunan data pada simpul saat ini, A adalah atribut kandidat pemisah, dan S_v adalah subset data hasil pemisahan berdasarkan nilai v dari atribut A. Atribut dengan nilai Information Gain (atau penurunan Gini Index) terbesar dipilih sebagai pemisah pada simpul tersebut. Penelitian ini menggunakan kriteria Gini Index sebagai pengaturan bawaan (default) karena perhitungannya lebih ringan secara komputasi dibandingkan Entropy (tidak melibatkan operasi logaritma), dengan opsi Entropy juga disediakan sebagai parameter yang dapat dipilih pengguna pada antarmuka aplikasi

2.6 Feature Importance

Feature importance merupakan ukuran yang menunjukkan seberapa besar kontribusi setiap atribut (fitur) terhadap kemampuan prediksi model Decision Tree secara keseluruhan. Pada implementasi Scikit-learn, nilai feature importance suatu atribut dihitung berdasarkan total penurunan impuritas yang dihasilkan oleh atribut tersebut di seluruh simpul tempat atribut itu digunakan sebagai pemisah, dibobotkan berdasarkan proporsi jumlah sampel yang melewati simpul tersebut, dirumuskan secara umum sebagai (Nugraha, Supriadi and Setiadi, 2026):

$$Importance(A) = \sum_{t \in A} \frac{N_t}{N} \Delta Impurity(t)$$

Dengan t adalah simpul yang menggunakan atribut A sebagai pemisah, N_t adalah jumlah sampel pada simpul t, N adalah jumlah total sampel, dan ΔImpurity(t) adalah penurunan nilai impuritas (Gini/Entropy) akibat pemisahan

pada simpul t. Nilai feature importance kemudian dinormalisasi sehingga jumlah seluruh atribut bernilai 1. Metrik ini digunakan pada penelitian ini untuk mengidentifikasi variabel akademik yang paling berpengaruh terhadap ketepatan waktu kelulusan mahasiswa, dan divisualisasikan dalam bentuk grafik batang pada tab Confusion Matrix aplikasi.

2.7 Ketidakseimbangan Kelas (Class Imbalnace) dan Class Weight

Ketidakseimbangan kelas (class imbalance) terjadi ketika proporsi jumlah data pada masing-masing kelas target tidak seimbang secara signifikan. Salah satu solusi untuk mengatasi hal ini adalah dengan memberikan bobot lebih besar pada kelas minoritas selama proses pelatihan model (class weighting), sehingga kesalahan klasifikasi pada kelas minoritas diberi penalti lebih besar dibandingkan kesalahan pada kelas mayoritas. Pada penelitian ini digunakan skema pembobotan "balanced" sebagaimana diimplementasikan pada Scikit-learn, yang menghitung bobot tiap kelas secara otomatis berdasarkan rumus (Rafi Jonathan Siger, Muhammad Naufal and Farrikh Alzami, 2026):

$$w_j = \frac{n}{k \times n_j}$$

dengan w_j adalah bobot untuk kelas j, n adalah jumlah total sampel data latih, k adalah jumlah kelas target, dan n_j adalah jumlah sampel pada kelas j. Dengan rumus ini, kelas dengan jumlah sampel lebih sedikit (Terlambat) secara otomatis memperoleh bobot lebih besar dibandingkan kelas mayoritas (Tepat Waktu), sehingga fungsi impuritas pada proses pembentukan pohon turut memperhitungkan bobot tersebut, bukan hanya jumlah sampel mentah.

2.8 Confusion Matrix dan Matrik Evaluasi

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi performa model klasifikasi dengan membandingkan hasil prediksi model dengan label aktual, terdiri atas empat komponen: True Positive (TP), yaitu data positif yang diprediksi benar sebagai positif; True Negative (TN), yaitu data negatif yang diprediksi benar sebagai negatif; False Positive (FP), yaitu data negatif yang salah diprediksi sebagai positif; dan False Negative (FN), yaitu data positif yang salah diprediksi sebagai negative (AlShammari, 2024). Dari keempat komponen tersebut, dihitung beberapa metrik evaluasi sebagai berikut.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Presisi = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

Akurasi mengukur proporsi prediksi yang benar secara keseluruhan terhadap seluruh data; Presisi mengukur ketepatan model saat memprediksi kelas positif, yaitu dari seluruh data yang diprediksi positif, berapa proporsi yang benar-benar positif; Recall (disebut juga sensitivity) mengukur kemampuan model mendeteksi seluruh data positif yang sebenarnya, yaitu dari seluruh data yang benar-benar positif, berapa proporsi yang berhasil terdeteksi oleh model; dan F1-

Score merupakan rata-rata harmonik antara Presisi dan Recall, yang berguna sebagai ukuran tunggal ketika terdapat trade-off antara keduanya — sebagaimana terjadi pada penelitian ini akibat penanganan ketidakseimbangan kelas. Pada kasus klasifikasi dengan lebih dari satu kelas seperti Tepat Waktu dan Terlambat, metrik-metrik tersebut dapat dihitung per kelas (as shown pada classification report), maupun dirata-ratakan secara macro average (rata-rata sederhana antar kelas, memperlakukan setiap kelas dengan bobot sama) atau weighted average (rata-rata yang dibobotkan berdasarkan jumlah sampel/support tiap kelas).

2.9 K-Fold Cross Validation

K-Fold Cross Validation merupakan teknik validasi model dengan cara membagi data menjadi K bagian (fold) berukuran relatif sama. Model dilatih dan diuji sebanyak K kali, di mana pada setiap putaran ke-i, fold ke-i digunakan sebagai data uji dan K−1 fold sisanya digunakan sebagai data latih. Skor evaluasi (umumnya akurasi) dari seluruh putaran kemudian dirata-ratakan untuk memperoleh estimasi performa model yang lebih stabil, dirumuskan sebagai (Wijiyanto *et al.*, 2024):

$$CV_{score} = \frac{1}{K}\sum_{i=1}^{K} score_i$$

dengan $score_i$ adalah skor evaluasi pada putaran (fold) ke-i. Selain nilai rata-rata, standar deviasi antar fold juga dihitung untuk menilai stabilitas model nilai standar deviasi yang kecil menunjukkan performa model relatif konsisten pada berbagai kombinasi pembagian data, sedangkan nilai yang besar mengindikasikan

model sensitif terhadap data tertentu (berpotensi overfitting pada kombinasi data tertentu). Penelitian ini menggunakan skema Stratified K-Fold, yaitu varian K-Fold yang menjaga proporsi kelas target pada setiap fold tetap sebanding dengan proporsi pada keseluruhan data hal ini penting mengingat kondisi ketidakseimbangan kelas pada data penelitian, agar setiap fold tetap memuat representasi kelas Terlambat yang memadai. Penelitian ini menggunakan 10-Fold Cross Validation (K=10) sebagai metode validasi utama.

2.10 Python dan Pustaka Pendukung

Python merupakan bahasa pemrograman tingkat tinggi yang banyak digunakan dalam pengembangan aplikasi data science dan machine learning karena sintaksnya yang sederhana dan didukung oleh ekosistem pustaka yang luas (Santoso and Priyadi, 2024). Penelitian ini menggunakan beberapa pustaka pendukung sebagai berikut. Tkinter digunakan untuk membangun antarmuka grafis (GUI) desktop, termasuk komponen tab navigasi (Notebook), tabel data (Treeview), tombol interaktif, dan dropdown parameter. Pandas digunakan untuk manipulasi dan pra-pemrosesan data tabular, meliputi pembacaan berkas Excel/CSV, penggabungan data multi-sheet, pembersihan baris tidak valid, dan pembentukan variabel turunan. Scikit-learn digunakan untuk implementasi algoritma Decision Tree (DecisionTreeClassifier), pembagian data latih/uji (train_test_split), validasi silang (cross_val_score, StratifiedKFold), pengkodean label kategorikal (LabelEncoder), serta perhitungan metrik evaluasi (accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report). Matplotlib digunakan untuk visualisasi pohon keputusan

(plot_tree), confusion matrix, dan grafik feature importance, yang kemudian disematkan ke dalam antarmuka Tkinter melalui FigureCanvasTkAgg. OpenPyXL digunakan secara internal oleh Pandas untuk membaca dan menulis berkas Excel (.xlsx), termasuk penataan gaya (styling) pada berkas hasil ekspor.

2.11 Pengujian Black Box

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada fungsionalitas sistem tanpa memeriksa struktur kode program secara internal (diperlakukan sebagai "kotak hitam" yang hanya diamati dari masukan dan keluarannya) (Ardiansyah and Sela, 2025). Pengujian dilakukan dengan memberikan input tertentu pada sistem dan memeriksa apakah keluaran (output) yang dihasilkan sesuai dengan yang diharapkan (expected result). Salah satu teknik perancangan skenario pengujian Black Box yang umum digunakan adalah equivalence partitioning, yaitu mengelompokkan kemungkinan input ke dalam kelas-kelas setara (misalnya "data valid" dan "data tidak valid") untuk memastikan cakupan pengujian yang representatif tanpa perlu menguji seluruh kemungkinan input satu per satu.

Setiap skenario pengujian Black Box dinyatakan berstatus PASS apabila keluaran aktual (actual output) sesuai dengan keluaran yang diharapkan (expected output), dan berstatus FAIL apabila terdapat ketidaksesuaian. Metode ini dipilih untuk memvalidasi bahwa seluruh fitur aplikasi mulai dari pemuatan data, pelatihan model, visualisasi, prediksi, hingga ekspor hasil berfungsi sebagaimana mestinya dari sudut pandang pengguna akhir.