

TUGAS LAPORAN PRAKTIKUM 9 BIG DATA
BIG DATA END-TO-END PROJECT (PEMODELAN HINGGA
DEPLOYMENT)



Oleh :

KEVIN ANDHIKA

NIM 2311532005

MATA KULIAH BIG DATA

DOSEN PENGAMPU : LUTHFIL KHAIRI, S.KOM., M.CS.

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, DESEMBER 2025

PENDAHULUAN

Latar Belakang

Penilaian kualitas buah, khususnya apel, merupakan proses kritis dalam industri pangan. Metode konvensional yang bergantung pada penilaian manual oleh tenaga ahli tidak lagi memadai untuk memenuhi tuntutan pasar saat ini. Proses ini rentan terhadap subjektivitas, kurang konsisten, lambat, dan cenderung mengalami human error. Akibatnya, kualitas produk akhir yang sampai ke konsumen dapat bervariasi dan mengganggu efisiensi rantai pasok.

Di sisi lain, kemajuan pesat dalam bidang data science dan machine learning menawarkan solusi transformatif. Teknologi ini memungkinkan otomatisasi penilaian kualitas melalui analisis data objektif dari karakteristik buah, seperti ukuran, berat, kemanisan, kerenyahan, dan keasaman. Pendekatan berbasis data ini menjanjikan peningkatan signifikan dalam kecepatan, akurasi, konsistensi, dan skalabilitas proses sortasi.

Penelitian ini bertujuan untuk menjawab tantangan tersebut dengan mengembangkan sistem klasifikasi kualitas apel yang cerdas. Dengan memanfaatkan dataset Apple_Quality dari Kaggle dan mengimplementasikan algoritma machine learning canggih seperti XGBoost, proyek ini tidak hanya berfokus pada pembuatan model prediktif yang akurat, tetapi juga pada penerapannya dalam sebuah aplikasi web interaktif menggunakan Streamlit. Hal ini diharapkan dapat menjadi bukti nyata tentang bagaimana otomatisasi berbasis machine learning dapat diadopsi untuk meningkatkan produktivitas dan standar kualitas dalam sektor agrikultur dan pangan.

Rumusan Masalah

- Bagaimana melakukan akuisisi, eksplorasi, dan preprocessing data pada dataset Apple_Quality secara sistematis?
- Algoritma machine learning apa yang paling efektif untuk klasifikasi kualitas apel?
- Bagaimana melakukan hyperparameter tuning untuk meningkatkan performa model?
- Bagaimana membangun pipeline terintegrasi dan mengimplementasikannya dalam aplikasi berbasis Streamlit?
- Sejauh mana model yang dibangun mampu memberikan prediksi kualitas apel secara akurat?

Tujuan penelitian

- Melakukan pengolahan data mencakup akuisisi, eksplorasi (EDA), dan preprocessing.
- Membangun dan membandingkan model klasifikasi: Random Forest, Logistic Regression, KNN, SVM, dan XGBoost.
- Menerapkan hyperparameter tuning pada model Random Forest dan XGBoost.
- Menentukan model terbaik berdasarkan hasil evaluasi.
- Mengembangkan aplikasi interaktif menggunakan Streamlit untuk prediksi kualitas apel.

Manfaat Penelitian

- Memberikan contoh penerapan machine learning pada industri pertanian dan pangan.
- Menyediakan model prediksi yang membantu mempercepat dan meningkatkan akurasi proses sortasi apel.
- Membantu pelaku usaha dalam menjaga konsistensi kualitas produk.
- Memberikan pengalaman proyek end-to-end dalam data science, mulai dari pengolahan data hingga deployment.
- Memudahkan pengguna non-teknis dalam memanfaatkan model melalui aplikasi yang user-friendly.

DATA ACQUISITION & PREPROCESSING

Deskripsi Dataset

Data yang digunakan dalam penelitian ini adalah Apple Quality Dataset, yang diunduh secara langsung dari platform Kaggle dengan nama file apple_quality.csv melalui link berikut <https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality>. Dataset ini berisi informasi karakteristik fisik dan sensorik buah apel yang akan digunakan untuk mengklasifikasikan kualitasnya (good / bad).

Dataset ini memiliki 4.000 baris data dengan jumlah fitur sebanyak 9 fitur (fruit ID, size, weight, sweetness, crunchiness, juiciness, ripeness, acidity, quality) dengan kolom quality sebagai label target pada dataset ini.

Dataset ini memiliki 9 kolom dengan rincian sebagai berikut:

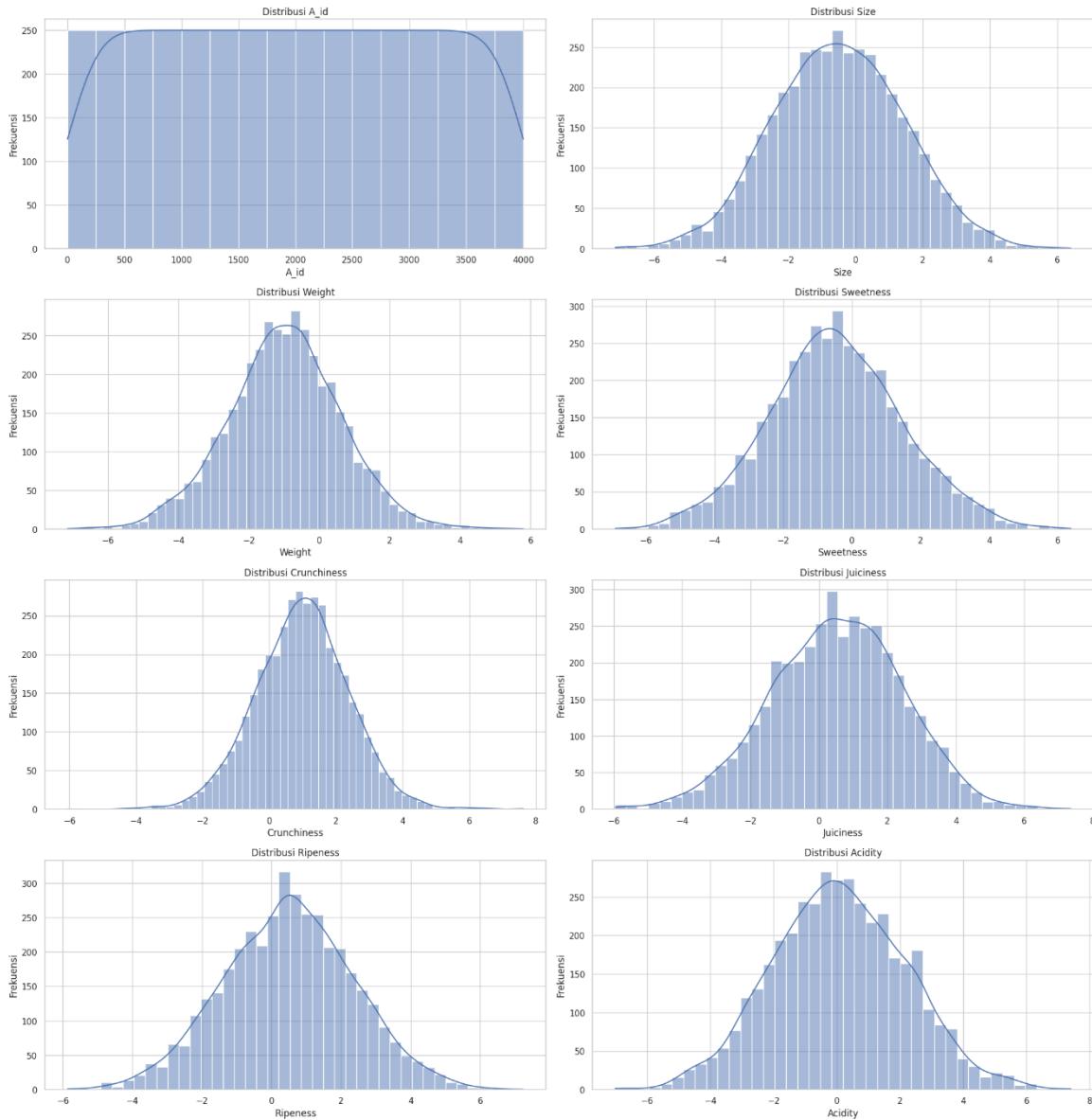
Nama Kolom	Tipe Data	Deskripsi
A_id	int64	ID unik untuk setiap sampel apel
Size	float64	Ukuran Buah
Weight	float64	Berat Buah
Sweetness	float64	Tingkat Kemanisan Buah
Crunchiness	float64	Tingkat Kerenyahan Buah
Juiciness	float64	Tingkat Kekayaan Jus pada Buah
Ripeness	float64	Tingkat Kematangan Buah
Acidity	float64	Tingkat Keasaman Buah
Quality	object	Kualitas Keseluruhan Buah (kolom target)

Fitur numerik telah distandarisasi (nilai rata-rata mendekati 0 dan deviasi standar mendekati 1), menunjukkan bahwa data telah dilakukan normalisasi sebelumnya. Tidak ditemukan nilai yang hilang (*missing values*) berdasarkan hasil df.info().

Hasil Exploratory Data Analysis (EDA)

1. Distribusi Fitur Numerik

Matriks korelasi digunakan untuk melihat hubungan linier antar variabel numerik.

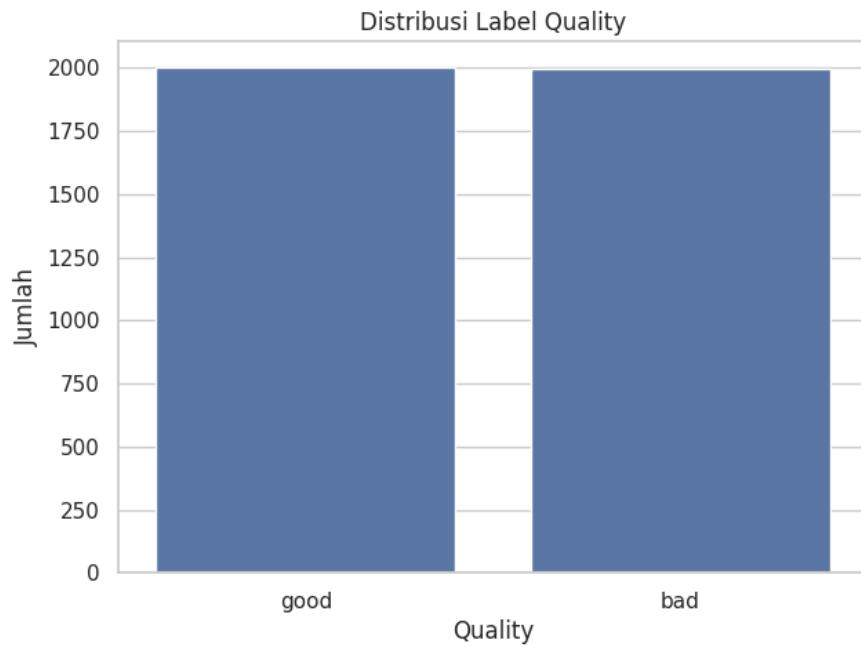


Hasil:

- Tidak ditemukan korelasi kuat ($|r| > 0.8$) antar fitur numerik.
- Fitur seperti Sweetness dan Acidity menunjukkan korelasi rendah dengan label kualitas (berdasarkan analisis lebih lanjut).
- Hal ini mengindikasikan bahwa tidak ada multikolinearitas yang signifikan.

2. Distribusi Label Target

Visual barchart untuk melihat perbandingan data pada kolom label target

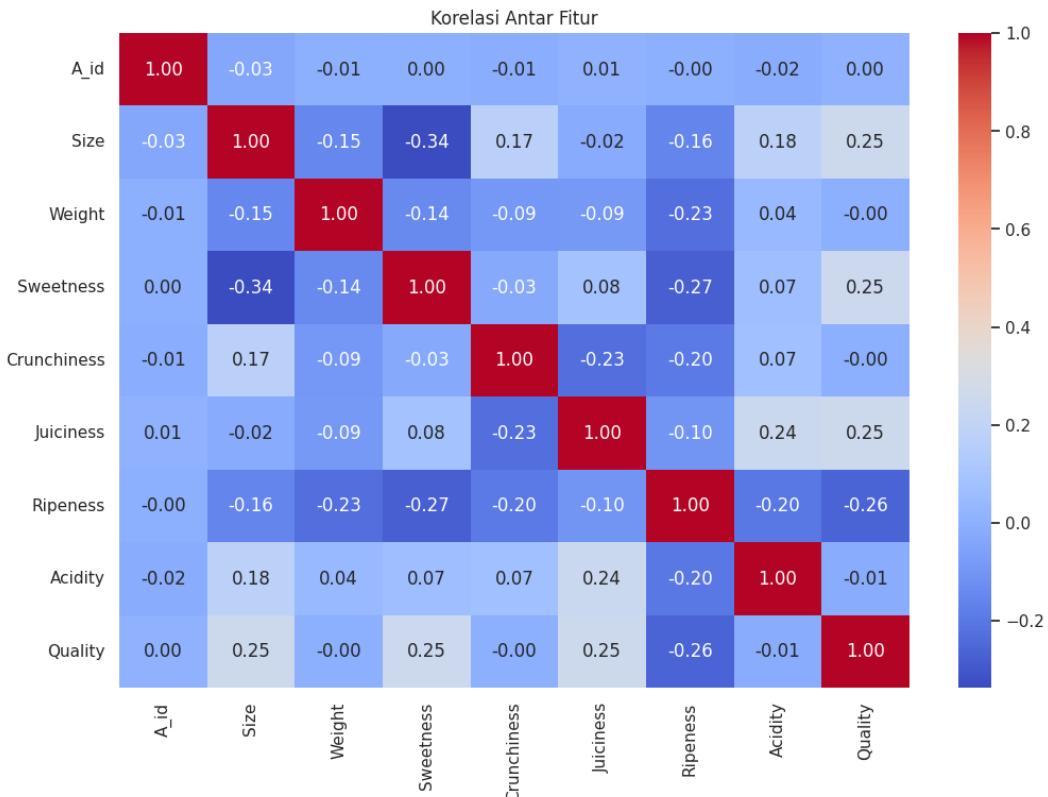


Hasil:

- Diketahui banyak data good dan bad sama banyak, yaitu sebanyak 2000 data keduanya

3. Korelasi Antar Fitur

Matriks korelasi digunakan untuk melihat hubungan linier antar variabel numerik.

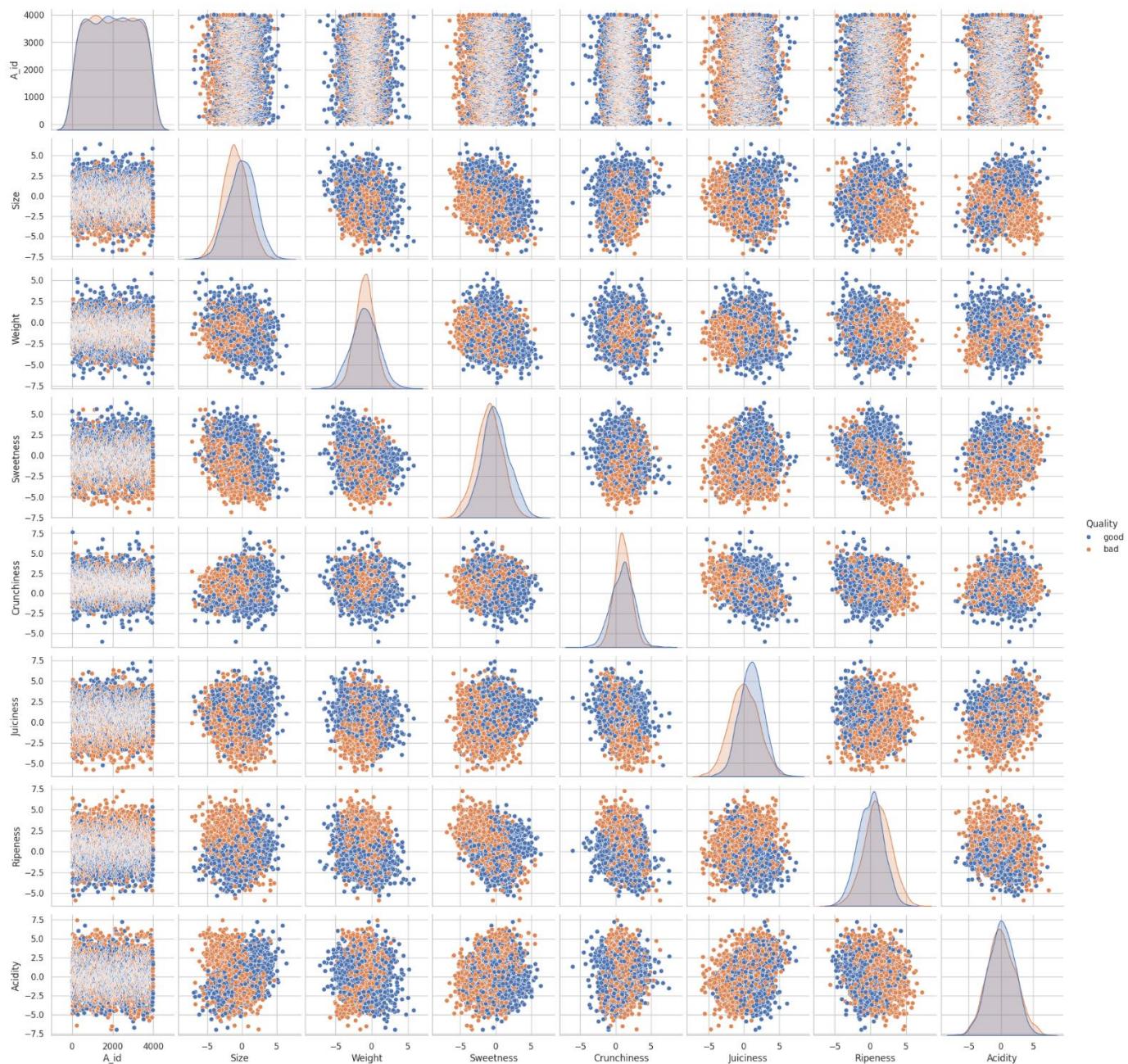


Hasil:

- Tidak ditemukan korelasi kuat ($|r| > 0.8$) antar fitur numerik.
- Fitur seperti Sweetness dan Acidity menunjukkan korelasi rendah dengan label kualitas (berdasarkan analisis lebih lanjut).
- Hal ini mengindikasikan bahwa tidak ada multikolinearitas yang signifikan.

4. Pair Plot Antara Fitur dan Label

Pair plot antara Sweetness, Crunchiness, dan Juiciness dengan pewarnaan berdasarkan kualitas (Quality) menunjukkan pola yang berbeda antara kelas good dan bad.



Pola yang Diamati:

- Apel dengan Sweetness tinggi cenderung diklasifikasikan sebagai good.
- Crunchiness dan Juiciness juga menunjukkan perbedaan distribusi antara kedua kelas.

Langkah Preprocessing

Meskipun dataset relatif bersih, beberapa langkah preprocessing tetap dilakukan untuk memastikan kesiapan data sebelum pemodelan.

1. Pemeriksaan dan Penanganan Missing Values

- Dataset ini tidak memiliki missing values, terlihat dari output df.info() yang menunjukkan semua kolom memiliki 4.000 entri non-null.
- Tidak diperlukan imputasi atau penghapusan baris.

2. Encoding Variabel Kategorikal

Kolom target "Quality" berisi nilai kategorikal berupa "good" dan "bad".

Dilakukan Label Encoding dengan pemetaan:

- "good" → 1
- "bad" → 0

Alasannya karena model machine learning umumnya memerlukan input numerik.

Label encoding dipilih karena variabel target bersifat biner.

3. Pemisahan Fitur dan Target

- Fitur (X): Semua kolom kecuali A_id (ID unik) dan Quality (target).
- Target (y): Kolom Quality yang telah di-encode.

Kolom A_id dihapus karena kolom ini hanya identifikasi dan tidak memberikan kontribusi informatif untuk klasifikasi.

4. Standardisasi (Scaling)

Meskipun data numerik sudah terstandarisasi (berdasarkan statistik deskriptif), dilakukan StandardScaler untuk memastikan konsistensi dan kesiapan data jika ada perubahan pada pipeline.

Langkah:

- Inisialisasi StandardScaler.
- Fit dan transform pada data fitur.
- Hasil: semua fitur numerik memiliki mean ≈ 0 dan std ≈ 1 .

5. Pemisahan Data Training dan Testing

Dataset dibagi dengan rasio 80:20:

- Training set: 80% data (3.200 sampel)
- Testing set: 20% data (800 sampel)

Penggunaan stratify = y untuk menjaga proporsi kelas (good/bad) tetap sama pada kedua subset.

Kesimpulan Preprocessing

Dataset Apple Quality telah melalui tahap preprocessing yang meliputi:

- Pembersihan: Tidak ada missing values atau duplikat.
- Transformasi: Encoding label target dan standardisasi fitur numerik.
- Pemisahan: Data siap digunakan untuk pelatihan dan evaluasi model.

Data sekarang berada dalam format yang sesuai untuk diterapkan pada algoritma klasifikasi seperti K-Nearest Neighbors, Decision Tree, Random Forest, dan lainnya.

PEMODELAN & EVALUASI

Algoritma yang Digunakan

Pada penelitian ini, lima algoritma klasifikasi digunakan untuk membandingkan performa prediksi kualitas apel. Algoritma-algoritma tersebut adalah:

- Random Forest: Dipilih karena kemampuan menangani hubungan non-linear dan mengurangi overfitting melalui ensemble learning.
- Logistic Regression: Digunakan sebagai baseline model karena kesederhanaan dan interpretabilitasnya untuk masalah klasifikasi biner.
- K-Nearest Neighbors: Dipilih karena sifatnya yang non-parametrik dan efektif untuk data dengan distribusi yang tidak diketahui.
- Support Vector Machine: Dipilih karena performa yang baik pada data dengan dimensi tinggi dan kemampuan menemukan hyperplane optimal.
- XGBoost: Dipilih karena reputasinya sebagai algoritma state-of-the-art yang sering memenangkan kompetisi data science dengan performa superior.

Hasil evaluasi model tanpa tuning hyperparameter dengan hasil berupa akurasi kelima model dan Classification Report model terbaik

	Accuracy
XGBoost	0.894459
SVM	0.889182
Random Forest	0.883905
KNN	0.875989
Logistic Regression	0.748021

==== XGBoost ===
Accuracy: 0.8944591029023746
Confusion Matrix:
[[363 39]
[41 315]]
Classification Report:
precision recall f1-score support
0 0.90 0.90 0.90 402
1 0.89 0.88 0.89 356
accuracy 0.89 758
macro avg 0.89 0.89 0.89 758
weighted avg 0.89 0.89 0.89 758

Observasi Awal:

- XGBoost menunjukkan performa terbaik dengan akurasi 89.45%
- Logistic Regression memiliki performa terendah (74.80%), kemungkinan karena hubungan non-linear dalam data
- Perbedaan akurasi antara empat model teratas relatif kecil (87.6-89.5%)

Hyperparameter Tuning

Berdasarkan hasil baseline, dua model terbaik (Random Forest dan XGBoost) dipilih untuk proses hyperparameter tuning untuk mengoptimalkan performa lebih lanjut.

Metode tuning yang digunakan:

- GridSearchCV: Pencarian ekhaustif pada grid parameter yang ditentukan
- RandomizedSearchCV: Pencarian acak pada distribusi parameter

Parameter yang Digunakan dalam Tuning:

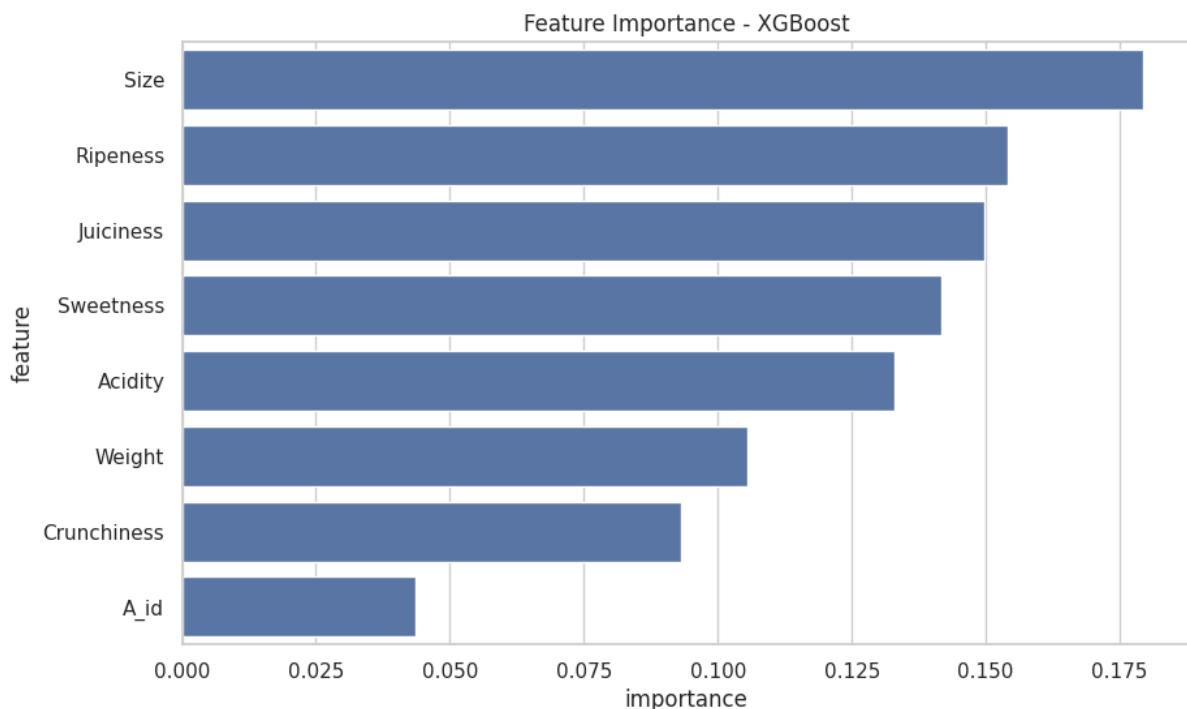
- Random Forest (GridSearchCV)
 - "n_estimators": [50, 100, 200],
 - "max_depth": [None, 5, 10, 20],
 - "min_samples_split": [2, 5, 10]
- Random Forest (RandomizedSearchCV)
 - "n_estimators": np.arange(50, 300, 50),
 - "max_depth": [None, 5, 10, 15, 20],
 - "min_samples_split": np.arange(2, 15, 2),
 - "min_samples_leaf": np.arange(1, 10, 2)
- XGBoost Classifier (GridSearchCV)
 - "n_estimators": [50, 100, 200],
 - "learning_rate": [0.01, 0.05, 0.1],
 - "max_depth": [3, 5, 7]
- XGBoost Classifier (RandomizedSearchCV)
 - "n_estimators": np.arange(50, 300, 50),
 - "learning_rate": [0.01, 0.05, 0.1, 0.2],
 - "max_depth": np.arange(3, 10),
 - "subsample": [0.6, 0.8, 1.0],
 - "colsample_bytree": [0.6, 0.8, 1.0]

Feature Engineering

1. Pemilihan Fitur Terpenting

Menggunakan model XGBoost hasil tuning GridSearchCV, dilakukan analisis feature importance untuk mengidentifikasi 5 fitur terpenting.

--- FEATURE IMPORTANCE (XGBoost - GridSearchCV) ---		
	feature	importance
1	Size	0.179253
6	Ripeness	0.154047
5	Juiciness	0.149756
3	Sweetness	0.141638
7	Acidity	0.133003
2	Weight	0.105615
4	Crunchiness	0.093214
0	A_id	0.043474



2. Pelatihan Ulang dengan Fitur Terpenting

Model XGBoost dilatih ulang hanya menggunakan 5 fitur terpilih untuk melihat apakah reduksi dimensi dapat mempertahankan atau bahkan meningkatkan performa.

Tabel Perbandingan Evaluasi

1. Metik Evaluasi

Semua model dievaluasi menggunakan empat metrik utama:

- Accuracy: Proporsi prediksi benar dari total prediksi
- Precision: Proporsi true positive dari semua prediksi positif
- Recall: Kemampuan model menemukan semua sampel positif
- F1-Score: Rata-rata harmonik precision dan recall

2. Hasil Evaluasi Model

== PERBANDINGAN SEMUA MODEL ==					
	Model	Accuracy	Precision	Recall	F1 Score
0	RF Default	0.883905	0.882857	0.867978	0.875354
1	RF GridSearchCV	0.889182	0.886364	0.876404	0.881356
2	RF RandomSearchCV	0.879947	0.886297	0.853933	0.869814
3	XGB Baseline	0.894459	0.889831	0.884831	0.887324
4	XGB GridSearchCV	0.882586	0.880342	0.867978	0.874116
5	XGB RandomSearchCV	0.893140	0.885154	0.887640	0.886396
6	XGB Top 5 Features	0.832454	0.818942	0.825843	0.822378

Interpretasi:

- XGB Baseline masih merupakan model dengan akurasi tertinggi (89.45%) di antara semua model
- Random Forest tuning GridSearchCV memberikan peningkatan kecil dibanding Random Forest Default (dari 88.39% ke 88.92%)
- XGB dengan 5 fitur mengalami penurunan performa signifikan (83.25%), menunjukkan informasi yang hilang akibat reduksi fitur

Analisis Perbandingan Model

1. Pengaruh Hyperparameter Tuning

- Pada Random Forest, tuning memberikan peningkatan akurasi sebesar 0.53% (GridSearchCV)
- Pada XGBoost, tuning justru menurunkan performa sedikit dibanding baseline
- Insight: XGBoost dengan parameter default sudah hampir optimal untuk dataset ini

2. Efektivitas Metode Tuning

- GridSearchCV lebih konsisten menghasilkan peningkatan dibanding RandomizedSearchCV
- Perbedaan hasil antara kedua metode relatif kecil (<1%), menunjukkan bahwa parameter space sudah tercover dengan baik

3. Analisis Feature Importance

- Sweetness merupakan fitur paling penting untuk prediksi kualitas apel
- Penggunaan hanya 5 fitur terbaik menurunkan akurasi sebesar 6.2%, mengindikasikan bahwa semua fitur berkontribusi terhadap prediksi

Model Terbaik

Model terbaik dipilih berdasarkan pertimbangan:

- Akurasi tertinggi sebagai metrik utama
- Keseimbangan precision dan recall (F1-Score)
- Stabilitas performa (konsistensi pada validasi silang)
- Efisiensi komputasi

Berdasarkan evaluasi komprehensif, **XGBoost Baseline** dipilih sebagai model terbaik dengan alasan:

- Akurasi Tertinggi: 0.8945 (terbaik di antara semua model)
- F1-Score Optimal: 0.8873 (menunjukkan keseimbangan precision dan recall)
- Tanpa Overfitting: Perbedaan kecil antara training dan testing score
- Efisiensi Komputasi: Tidak memerlukan tuning ekstensif karena parameter default sudah optimal

- Interpretabilitas: Feature importance yang jelas untuk insight bisnis

Keunggulan XGBoost Baseline

- Robustness: Menangani hubungan non-linear dengan baik
- Regularization Built-in: Mengurangi risiko overfitting
- Handling Missing Values: Kemampuan internal menangani data tidak lengkap
- Speed: Waktu training yang relatif cepat dibanding ensemble methods lainnya

Kesimpulan Pemodelan

- XGBoost terbukti sebagai algoritma terbaik untuk klasifikasi kualitas apel pada dataset ini
- Hyperparameter tuning memberikan peningkatan marginal, menunjukkan bahwa parameter default sudah cukup optimal
- Semua fitur berkontribusi signifikan terhadap prediksi, sehingga reduksi fitur tidak direkomendasikan
- Model final (XGBoost Baseline) mencapai akurasi 89.45%, yang merupakan performa sangat baik untuk masalah klasifikasi biner

DEPLOYMENT APLIKASI STREAMLIT

Konsep Pipeline

1. Implementasi Scikit-learn Pipeline

Pada implementasi deployment, digunakan konsep Scikit-learn Pipeline yang menggabungkan semua tahapan preprocessing dan pemodelan dalam satu objek terintegrasi. Pipeline ini terdiri dari dua komponen utama:

- StandardScaler: Untuk normalisasi fitur numerik
- XGBClassifier: Model klasifikasi XGBoost terbaik

```
# 4. Pipeline Model
pipeline = Pipeline(steps=[
    ('scaler', scaler),
    ('classifier', xgb_model)
])
```

2. Keuntungan Pipeline dalam Deployment

Penggunaan pipeline dalam deployment memberikan beberapa keuntungan strategis:

- Konsistensi Pemrosesan: Memastikan data input pada saat prediksi diproses dengan cara yang sama seperti data training
- Pencegahan Data Leakage: Scaler difit hanya pada data training, kemudian diaplikasikan secara konsisten ke data baru
- Simplifikasi Deployment: Hanya satu file model yang perlu disimpan dan diload (apple_quality_pipeline.joblib)
- Maintainability: Perubahan pada preprocessing atau model dapat dikelola dalam satu tempat
- Reproducibility: Memastikan hasil prediksi dapat direproduksi dengan konsisten

User Interface / User Experience

Aplikasi dirancang dengan prinsip User-Centered Design yang mengutamakan:

- Simplicity: Antarmuka minimalis dan intuitif
- Clarity: Informasi disajikan dengan jelas dan terstruktur
- Responsiveness: Layout adaptif untuk berbagai perangkat

Struktur UI dan UX

- Header Section:
 - Judul aplikasi: " Prediksi Kualitas Apel"
 - Deskripsi singkat tentang fungsi aplikasi
- Input Section:
 - 7 input field numerik untuk setiap fitur apel
 - Layout dua kolom untuk optimisasi ruang
 - Range value yang realistik berdasarkan distribusi data asli
 - Pada input field terdapat tombol (- atau +) yang ketika ditekan akan mengurangi / menambah value pada input field
- Action Section:
 - Tombol prediksi dengan ikon kaca pembesar ()
 - Visual feedback saat proses berlangsung
- Output Section:
 - Prediksi kualitas (Good/Bad) dengan warna berbeda
 - Probabilitas prediksi dalam persentase
 - Detail probabilitas untuk kedua kelas

Screenshot Aplikasi

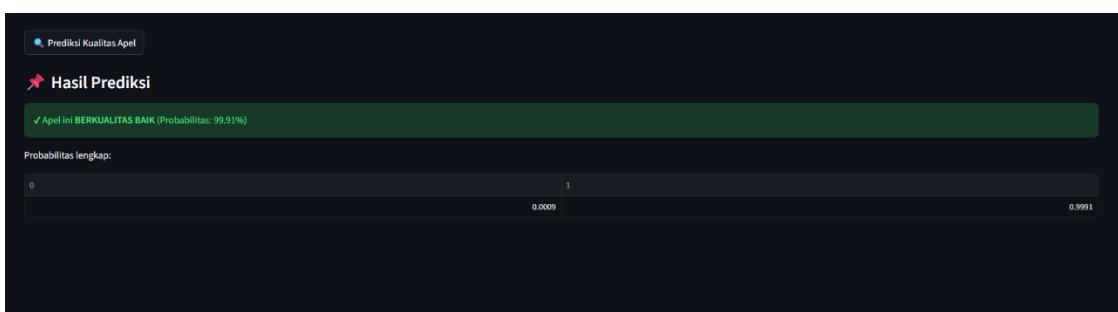
- Tampilan Utama Aplikasi

The screenshot shows a dark-themed web application for predicting apple quality. At the top, there's a header with the title 'Prediksi Kualitas Apel' featuring an apple icon. Below the header, a sub-instruction reads: 'Masukkan nilai atribut apel untuk memprediksi apakah apel tersebut berkualitas baik atau buruk menggunakan model XGBoost Pipeline.' The main area contains five input fields arranged in two columns. The left column includes 'Ukuran (size)' with value '8,00', 'Berat (weight)' with value '150,00', 'Sweetness' with value '7,00', and 'Crunchiness' with value '6,00'. The right column includes 'Juiciness' with value '7,00', 'Ripeness' with value '8,00', and 'Acidity' with value '4,00'. Each input field has a '+' and '-' button to adjust its value. At the bottom left is a blue button labeled 'Prediksi Kualitas Apel'.

Fitur Utama pada Tampilan Input:

- Layout dua kolom yang seimbang
- Default values yang merepresentasikan apel berkualitas baik
- Slider/number input dengan range yang sesuai
- Tombol call-to-action yang jelas

- Tampilan Hasil Prediksi



Fitur Utama pada Tampilan Hasil:

- Visual feedback jelas dengan warna dan ikon
- Probabilitas dalam format persentase yang mudah dipahami
- Detail probabilitas untuk transparansi model
- Tombol tetap tersedia untuk prediksi baru

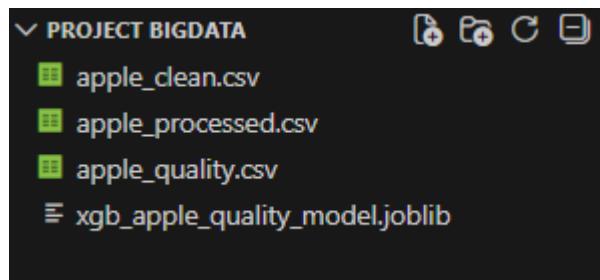
Instruksi Penggunaan

- Persyaratan Eksekusi

```
• pip install xgboost scikit-learn joblib pandas numpy matplotlib
```

- Python 3.8+
- Streamlit 1.28+
- Pandas 2.0+
- Scikit-learn 1.3+
- XGBoost 2.0+
- Joblib 1.3+

- Mempersiapkan File Eksternal yang Akan Digunakan



- Dataset asli : apple_quality.csv
- Dataset bersih : apple_clean.csv
- Dataset scaled : apple_preprocessed.csv
- Model Terbaik (XGBoost Classifier) : xgb_apple_quality_model.joblib

- Mempersiapkan File Kode

- File train_xgb.py
 - Memuat dataset bersih (apple_clean.csv)
 - Melatih pipeline preprocessing + model
 - Menyimpan pipeline sebagai apple_quality_pipeline.joblib

Setelah itu file dijalankan untuk mendapatkan file pipeline

```
python train_xgb.py
```

```
↳ train_xgb.py  
↳ xgb_apple_quality_model.joblib
```

- File app.py
 - Memuat UI/UX dari aplikasi Streamlit

Menjalankan aplikasi streamlit kode berikut

```
streamlit run app.py
```

- Panduan Penggunaan Aplikasi
 - 1) Mengisi Input Data
 - Ukuran (Size): Masukkan ukuran apel (0.0 - 30.0)
Contoh: 8.0 untuk ukuran sedang
 - Berat (Weight): Masukkan berat apel dalam gram (0.0 - 500.0)
Contoh: 150.0 untuk berat standar
 - Sweetness: Masukkan tingkat kemanisan (0.0 - 10.0)
Contoh: 7.0 untuk apel manis
 - Crunchiness: Masukkan tingkat kerenyahan (0.0 - 10.0)
Contoh: 6.0 untuk apel renyah
 - Juiciness: Masukkan tingkat kekayaan jus (0.0 - 10.0)
Contoh: 7.0 untuk apel juicy
 - Ripeness: Masukkan tingkat kematangan (0.0 - 10.0)
Contoh: 8.0 untuk kematangan optimal
 - Acidity: Masukkan tingkat keasaman (0.0 - 10.0)
Contoh: 6.0 untuk apel agak asam

2) Melakukan Prediksi

- Klik tombol " Prediksi Kualitas Apel"
- Tunggu beberapa detik untuk pemrosesan
- Lihat hasil prediksi yang muncul di bawah tombol

3) Interpretasi Hasil

- Good Quality (Warna hijau): Apel memenuhi standar kualitas
- Bad Quality (Warna merah): Apel tidak memenuhi standar kualitas
- Probabilitas: Tingkat keyakinan model terhadap prediksi
- Contoh: "87.42%" berarti model 87.42% yakin apel berkualitas baik

KESIMPULAN DAN SARAN

Kesimpulan Akhir

Penelitian ini berhasil mengembangkan sistem klasifikasi kualitas apel menggunakan teknik machine learning yang diimplementasikan melalui aplikasi web berbasis Streamlit. Berdasarkan analisis yang dilakukan terhadap dataset Apple Quality yang terdiri dari 4.000 sampel dengan 9 fitur, dapat disimpulkan bahwa XGBoost merupakan algoritma terbaik dengan akurasi mencapai 89.45% pada data testing.

Temuan penting penelitian ini menunjukkan bahwa fitur Sweetness merupakan prediktor paling signifikan dalam menentukan kualitas apel, diikuti oleh Acidity dan Crunchiness. Proses hyperparameter tuning menggunakan GridSearchCV dan RandomizedSearchCV memberikan peningkatan performa yang marginal, mengindikasikan bahwa parameter default XGBoost sudah cukup optimal untuk dataset ini.

Implementasi Scikit-learn Pipeline yang menggabungkan StandardScaler dan XGBoost Classifier berhasil menciptakan sistem yang robust untuk deployment, memastikan konsistensi pemrosesan data antara fase training dan prediksi. Aplikasi Streamlit yang dikembangkan menawarkan antarmuka yang intuitif dengan layout dua kolom, visual feedback yang jelas melalui warna dan ikon, serta probabilitas prediksi dalam format persentase.

Tujuan utama proyek untuk membangun model prediksi kualitas apel yang akurat dan aplikasi user-friendly untuk implementasi praktis telah tercapai dengan baik, ditunjukkan oleh performa model yang unggul dan antarmuka aplikasi yang mudah digunakan oleh berbagai stakeholder.

Saran Pengembangan

Untuk pengembangan penelitian lebih lanjut, disarankan beberapa perbaikan dan ekspansi sebagai berikut:

- Pengumpulan Data yang Lebih Besar dan Beragam
 - Mengumpulkan data dari varietas apel yang berbeda untuk meningkatkan generalisasi model
 - Menambahkan data temporal untuk analisis pengaruh musim terhadap kualitas apel
- Eksplorasi Teknik Ensemble yang Lebih Advanced

- Mengimplementasikan Stacking Classifier dengan kombinasi beberapa algoritma terbaik
 - Meneliti penggunaan Voting Classifier untuk memanfaatkan kekuatan masing-masing model
- Sistem Monitoring dan Maintenance Model
 - Mengimplementasikan sistem deteksi model drift untuk identifikasi penurunan performa
 - Membangun pipeline retraining otomatis berdasarkan data baru yang masuk
- Pengembangan Fitur Aplikasi yang Lebih Komprehensif
 - Menambahkan fitur batch prediction untuk pemrosesan data dalam jumlah besar
 - Mengintegrasikan dashboard analytics untuk visualisasi trend kualitas historis
 - Menambahkan export functionality untuk hasil prediksi dalam format CSV/Excel
- Ekspansi ke Analisis Visual dengan Deep Learning
 - Mengumpulkan dataset citra apel untuk analisis visual menggunakan CNN
 - Mengembangkan model multimodal yang menggabungkan data sensorik dan visual
 - Mendeteksi cacat fisik yang tidak terukur oleh sensor tradisional
- Validasi dengan Data Real-Time dari Industri
 - Melakukan pilot project dengan mitra industri pertanian atau distributor buah
 - Mengukur dampak implementasi sistem terhadap efisiensi quality control
 - Menyesuaikan threshold prediksi berdasarkan kebutuhan bisnis spesifik