

BAB III

METODOLOGI PENELITIAN

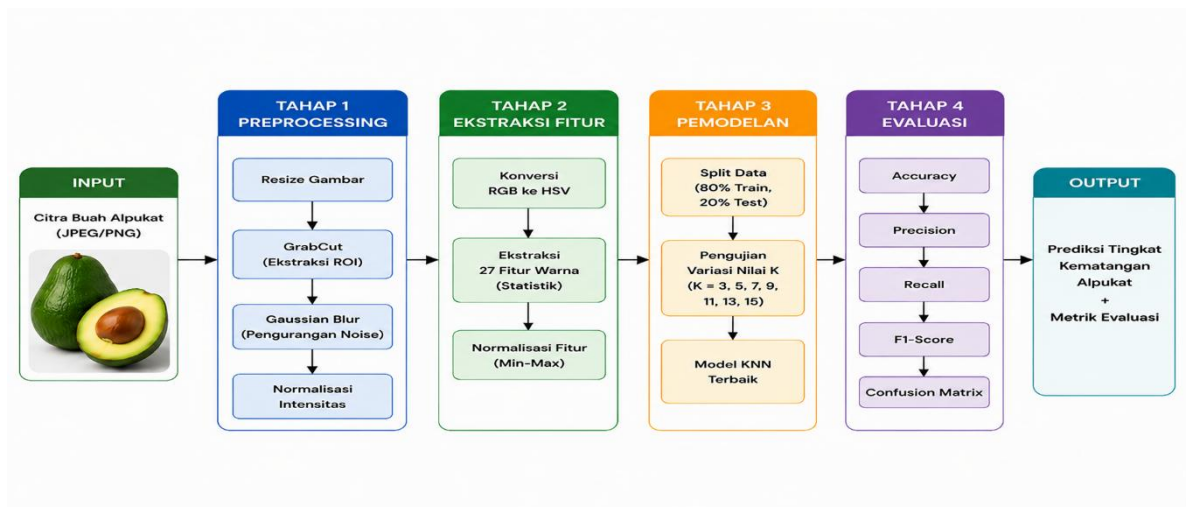
3.1 Jenis Penelitian

Penelitian ini merupakan penelitian terapan (applied research) dengan pendekatan kuantitatif eksperimental. Penelitian terapan bertujuan untuk mengembangkan solusi praktis terhadap permasalahan nyata dengan mengaplikasikan teori dan metode ilmiah yang sudah ada. Pendekatan kuantitatif digunakan karena penelitian ini melibatkan pengukuran numerik dan analisis statistik terhadap performa sistem yang dikembangkan.

Metode eksperimental diterapkan melalui variasi parameter k pada algoritma KNN ($k = 1, 2, 3, 5, 7, 9, 11, 13, 15$) untuk mengidentifikasi konfigurasi optimal yang menghasilkan performa klasifikasi terbaik. Setiap eksperimen dievaluasi menggunakan metrik kuantitatif yang terukur untuk membandingkan hasil antar konfigurasi secara objektif.

3.2 Desain Sistem Penelitian

Desain sistem penelitian mengikuti tahapan pengolahan citra digital dan machine learning yang terdiri atas akuisisi data, preprocessing citra, konversi ruang warna RGB ke HSV, ekstraksi fitur warna, normalisasi fitur, pembagian dataset, pemodelan menggunakan algoritma K-Nearest Neighbor (KNN), evaluasi performa model, serta menghasilkan prediksi tingkat kematangan buah alpukat. Kerangka kerja penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Desain Sistem Penelitian

Tahapan-tahapan dalam desain sistem penelitian adalah sebagai berikut:

1. Input: Citra buah alpukat dalam format digital (JPEG/PNG).
2. Preprocessing: Meliputi proses resize citra menjadi ukuran 100×100 piksel, ekstraksi Region of Interest (ROI) menggunakan metode GrabCut, pengurangan noise menggunakan Gaussian Blur, serta normalisasi intensitas citra agar nilai piksel berada pada rentang 0–1 sebelum dilakukan ekstraksi fitur.
3. Konversi Ruang Warna: Mengubah citra dari ruang warna RGB ke HSV.
4. Ekstraksi Fitur: Menghitung 27 fitur statistik warna (9 parameter statistik pada setiap kanal H, S, dan V).
5. Normalisasi Fitur: Melakukan normalisasi menggunakan metode Min-Max Scaling.
6. Pemodelan: Dataset telah dipisahkan secara manual oleh peneliti ke dalam folder data training dan data testing dengan proporsi sekitar 80%:20%, kemudian melakukan klasifikasi menggunakan algoritma K-Nearest

Neighbor (KNN) dengan variasi nilai $k = 1, 2, 3, 5, 7, 9, 11, 13$, dan 15 untuk memperoleh nilai k terbaik.

7. Evaluasi: Menghitung nilai Accuracy, Precision, Recall, dan F1-Score, serta menganalisis hasil klasifikasi menggunakan Confusion Matrix.
8. Output: Prediksi tingkat kematangan buah alpukat dan hasil evaluasi performa model menggunakan Accuracy, Precision, Recall, F1-Score, serta Confusion Matrix.

Seluruh tahapan penelitian diimplementasikan menggunakan bahasa pemrograman Python pada platform Google Colaboratory. Konfigurasi penelitian meliputi ukuran citra sebesar 100×100 piksel, penggunaan dataset yang telah dipisahkan secara manual ke dalam data latih dan data uji dengan proporsi sekitar $80\% : 20\%$, serta pengujian algoritma K-Nearest Neighbor (KNN) menggunakan variasi nilai $k = 1, 2, 3, 5, 7, 9, 11, 13$, dan 15 untuk memperoleh konfigurasi model terbaik.

3.3 Fokus Penelitian

Fokus utama penelitian ini adalah pengembangan sistem klasifikasi otomatis untuk menentukan tingkat kematangan buah alpukat berdasarkan analisis fitur warna HSV menggunakan algoritma K-Nearest Neighbor. Penelitian ini menekankan pada beberapa aspek kunci:

1. Ekstraksi dan analisis fitur statistik dari ruang warna HSV yang mampu menangkap karakteristik perubahan warna selama proses pematangan alpukat.

2. Optimasi parameter k pada algoritma KNN untuk mencapai keseimbangan optimal antara bias dan variance.
3. Evaluasi komprehensif performa sistem menggunakan multiple metrics untuk memberikan penilaian holistik terhadap kualitas klasifikasi.
4. Analisis pola kesalahan klasifikasi melalui confusion matrix untuk memahami limitation sistem dan potential improvement.

3.4 Sumber Data

Data yang digunakan dalam penelitian ini berupa citra digital buah alpukat yang diperoleh dari dua sumber, yaitu diperoleh dari sumber sekunder berupa dataset citra yang tersedia secara daring (online). Sebagian kecil data juga diperoleh dari dokumentasi citra alpukat secara langsung sebagai pelengkap dataset.

Dataset terdiri dari 210 citra buah alpukat yang dibagi ke dalam tiga kategori kematangan:

1. Mentah (unripe): citra buah alpukat dengan kulit berwarna hijau terang, tekstur keras, belum siap konsumsi.
2. Setengah matang (semi-ripe): citra buah alpukat dengan kulit berwarna hijau gelap hingga kecoklatan, tekstur mulai lunak, dalam proses pematangan.
3. Matang (ripe): citra buah alpukat dengan kulit berwarna coklat kehitaman hingga hitam, tekstur lunak, siap konsumsi optimal.

Sebelum digunakan dalam proses klasifikasi, seluruh citra dilakukan tahap seleksi dan preprocessing untuk memastikan kualitas data, menghilangkan citra

yang memiliki duplikasi atau kemiripan tinggi, serta menyesuaikan jumlah data pada setiap kelas. Dataset akhir terdiri dari 70 citra untuk masing-masing kelas, yaitu 70 citra mentah, 70 citra setengah matang, dan 70 citra matang.

3.5 Pengumpulan data

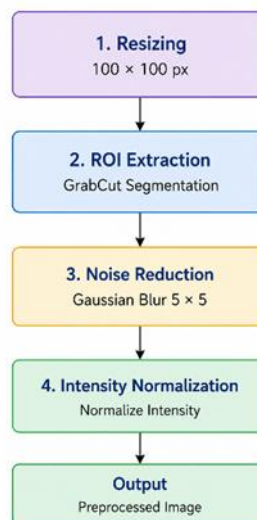
Pengumpulan data pada penelitian ini dilakukan melalui pengumpulan citra digital buah alpukat dari sumber daring (online) berupa dataset publik yang tersedia. Dataset yang diperoleh kemudian diseleksi berdasarkan kesesuaian dengan tujuan penelitian, yaitu klasifikasi tingkat kematangan buah alpukat. Proses pengumpulan data dilakukan melalui beberapa tahapan, yaitu:

1. Pengumpulan dataset: Citra buah alpukat dikumpulkan dari dataset publik yang tersedia secara daring serta beberapa citra tambahan yang diperoleh melalui dokumentasi langsung.
2. Seleksi citra: Tahap seleksi dilakukan untuk memilih citra yang memiliki kualitas baik dan sesuai dengan kategori tingkat kematangan. Citra yang memiliki kualitas rendah, gambar tidak jelas, atau memiliki kemiripan yang terlalu tinggi dengan citra lainnya akan dikeluarkan dari dataset.
3. Pelabelan data: Setiap citra dikelompokkan berdasarkan tingkat kematangan buah alpukat menjadi tiga kelas, yaitu: mentah (unripe), setengah matang (semi-ripe), dan matang (ripe). Pengelompokan kelas dilakukan berdasarkan karakteristik visual warna kulit alpukat.
4. Penyeimbangan Dataset: Setelah proses seleksi dan pelabelan, jumlah citra pada setiap kelas disesuaikan agar memiliki jumlah data yang seimbang.

Dataset akhir terdiri dari 210 citra dengan rincian: 70 kelas mentah (unripe), 70 kelas setengah matang (semi-ripe), dan 70 kelas matang (ripe).

3.6 Pra-pemrosesan Data

Pra-pemrosesan data (preprocessing) merupakan tahap awal yang bertujuan untuk mempersiapkan citra sebelum dilakukan ekstraksi fitur. Tahap ini bertujuan untuk meningkatkan kualitas citra, menghilangkan bagian yang tidak diperlukan, serta menyeragamkan ukuran dan intensitas citra sehingga proses ekstraksi fitur dapat dilakukan secara optimal. Tahapan preprocessing dapat dilihat pada Gambar 3.2.



Gambar 3.2 Flowchart Preprocessing Data

1. Resizing: Citra buah alpukat yang diperoleh dari berbagai sumber memiliki ukuran dan resolusi yang berbeda. Oleh karena itu, dilakukan proses resizing untuk menyeragamkan ukuran seluruh citra menjadi 100×100 piksel. Penyeragaman ukuran ini bertujuan untuk mengurangi kompleksitas

komputasi serta memastikan seluruh citra memiliki dimensi yang sama sebelum diproses lebih lanjut.

2. ROI Extraction: Setelah proses resizing, dilakukan ekstraksi Region of Interest (ROI) menggunakan metode GrabCut yang tersedia pada pustaka OpenCV. Metode GrabCut digunakan untuk memisahkan objek buah alpukat dari latar belakang. Hasil mask dari GrabCut kemudian dibersihkan melalui operasi morfologi (opening dan closing), lalu kontur terbesar (Largest Contour) pada mask tersebut diambil sebagai area objek utama, sehingga bagian latar belakang yang tersisa dapat dihilangkan sebelum dilanjutkan ke tahap berikutnya. Tahap ini bertujuan untuk mengurangi pengaruh latar belakang terhadap proses ekstraksi fitur warna.
3. Noise Reduction: Pengurangan noise menggunakan Gaussian Blur dengan ukuran kernel 5×5 . Proses ini bertujuan untuk mengurangi gangguan berupa noise atau variasi kecil pada citra tanpa menghilangkan informasi penting yang diperlukan dalam proses ekstraksi fitur.
4. Normalisasi Intensitas: melakukan normalisasi intensitas untuk menyeragamkan nilai intensitas citra sehingga perbedaan pencahayaan antar citra dapat diminimalkan. Proses normalisasi dilakukan menggunakan fungsi normalisasi pada OpenCV sehingga citra memiliki distribusi intensitas yang lebih konsisten sebelum memasuki tahap ekstraksi fitur.

Seluruh tahapan preprocessing diimplementasikan menggunakan pustaka OpenCV pada bahasa pemrograman Python. Hasil dari tahap preprocessing berupa citra buah alpukat yang telah memiliki ukuran seragam, objek utama telah

dipisahkan dari latar belakang, noise telah dikurangi, serta intensitas citra telah dinormalisasi sehingga siap digunakan pada tahap ekstraksi fitur.

3.7 Ekstraksi Fitur Warna HSV

Ekstraksi fitur warna HSV dilakukan melalui beberapa tahap. Pertama, citra dikonversi dari ruang warna RGB ke HSV menggunakan transformasi matematis standar yang tersedia di OpenCV dengan fungsi `cv2.cvtColor()`. Konversi ini menghasilkan tiga channel: Hue (H), Saturation (S), dan Value (V), masing-masing dalam bentuk matriks 2D dengan ukuran 100×100 piksel.

Dari setiap channel HSV, diekstraksi 9 parameter statistik yang mendeskripsikan distribusi nilai piksel dalam channel tersebut. Total fitur yang diekstraksi adalah $3 \text{ channels} \times 9 \text{ parameters} = 27$ fitur. Parameter statistik yang dihitung meliputi:

1. Mean (μ): Nilai rata-rata piksel dalam channel, menunjukkan central tendency distribusi warna.
2. Standard Deviation (σ): Mengukur dispersi nilai piksel dari mean, mengindikasikan variabilitas warna.
3. Variance (σ^2): Kuadrat dari standard deviation, memberikan ukuran spread yang lebih sensitif terhadap outliers.
4. Skewness: Mengukur asimetri distribusi, menunjukkan apakah distribusi condong ke kiri atau kanan.
5. Kurtosis: Mengukur ketajaman atau flatness distribusi, membedakan distribusi yang peaked vs flat.

6. Minimum: Nilai piksel terkecil dalam channel.
7. Maximum: Nilai piksel terbesar dalam channel.
8. Median: Nilai tengah setelah seluruh piksel diurutkan, robust terhadap outliers.
9. Interquartile Range (IQR): Rentang antara kuartil pertama (Q1) dan ketiga (Q3), mengukur spread middle 50% data.

Setelah ekstraksi, seluruh 27 fitur dinormalisasi menggunakan Min-Max Scaling untuk menstandarkan rentang nilai ke $[0, 1]$. Normalisasi ini penting untuk: (1) Menghindari dominasi fitur dengan skala nilai besar dalam perhitungan jarak Euclidean, (2) Mempercepat konvergensi proses komputasi, (3) Memastikan kontribusi yang seimbang dari setiap fitur dalam proses klasifikasi.

3.8 Implementasi Algoritma K-Nearest Neighbor

Implementasi algoritma K-Nearest Neighbor dilakukan menggunakan library scikit-learn 1.3.0, yang menyediakan implementasi KNN yang efisien dan well-optimized. Konfigurasi parameter KNN yang digunakan dalam penelitian ini adalah:

1. `n_neighbors (k)`: Variasi dari 1, 2, 3, 5, 7, 9, 11, 13, hingga 15 untuk menemukan nilai optimal.
2. `metric`: menggunakan Euclidean distance sebagai ukuran kedekatan.
3. `weights`: memberikan bobot sama untuk semua tetangga (tanpa pembobotan berdasarkan jarak).

4. algorithm: scikit-learn secara otomatis memilih algoritma terbaik (brute force, KD-tree, atau Ball-tree) berdasarkan karakteristik data.

Proses klasifikasi dilakukan dengan langkah-langkah berikut: (1) Model KNN di-fit dengan data training beserta label kelasnya, (2) Untuk setiap data testing, model menghitung jarak Euclidean ke seluruh data training, (3) Model memilih k data training terdekat, (4) Model menentukan kelas mayoritas dari k tetangga, (5) Kelas mayoritas dijadikan prediksi untuk data testing tersebut.

3.9 Evaluasi dan Validasi Model

Evaluasi performa model dilakukan menggunakan data testing yang telah disisihkan (42 citra, 20% dari total 210 citra dataset). Untuk setiap variasi parameter k, dihitung empat metrik evaluasi utama: Accuracy, Precision, Recall, dan F1-score. Untuk klasifikasi multi-class, metrik Precision, Recall, dan F1-score dihitung menggunakan weighted-averaging, yaitu menghitung metrik untuk setiap kelas kemudian merata-ratakannya dengan pembobotan berdasarkan jumlah data pada tiap kelas.

Confusion matrix dibuat untuk memvisualisasikan performa klasifikasi setiap kelas. Matriks berukuran 3×3 ini menampilkan jumlah prediksi yang benar dan salah untuk setiap kombinasi kelas aktual dan prediksi. Analisis confusion matrix memberikan insight tentang pola kesalahan klasifikasi, seperti kelas mana yang paling sering ter-misclassified dan apakah kesalahan cenderung terjadi pada adjacent classes.

Validasi model dilakukan menggunakan data testing yang telah dipisahkan secara manual dari data training sebelum proses pemodelan. Data training digunakan untuk membangun model K-Nearest Neighbor, sedangkan data testing digunakan sebagai data evaluasi. Dengan pembagian dataset yang telah ditentukan sebelumnya, seluruh pengujian variasi nilai k dilakukan menggunakan data training dan data testing yang sama sehingga hasil evaluasi antar nilai k dapat dibandingkan secara konsisten.

Pengukuran efisiensi komputasi secara rinci, seperti pencatatan waktu pemrosesan per tahapan (preprocessing, ekstraksi fitur, dan klasifikasi), belum dilakukan pada implementasi saat ini dan dapat menjadi bagian dari pengembangan penelitian selanjutnya untuk menilai kelayakan sistem secara real-time.