

BAB IV

PERANCANGAN

4.1 Analisis Penelitian Terdahulu

Penelitian ini didasari sejumlah studi terdahulu yang telah mengembangkan sistem klasifikasi berbasis citra dalam bidang pertanian, khususnya beras. Beberapa pendekatan telah digunakan, seperti metode klasikal dengan ekstraksi fitur manual yang dikombinasikan dengan algoritma seperti Support Vector Machine (SVM), serta pendekatan berbasis AI lainnya.

Phan et al. (2023) menggunakan YOLOv5 untuk klasifikasi benih padi dan menunjukkan performa akurasi yang tinggi, namun data yang digunakan berasal dari citra laboratorium, bukan dari lingkungan nyata. Sementara itu, Song et al. (2025) menerapkan YOLOv8n (YOLO-Rice) untuk mendeteksi malai padi secara real-time di area pertanian terbuka, menunjukkan akurasi di atas 93%, namun hanya fokus pada deteksi objek, bukan klasifikasi varietas.

Studi lain oleh El Sakka et al. (2025) memanfaatkan CNN untuk klasifikasi beras berdasarkan warna dan tekstur, namun pendekatan ini membutuhkan komputasi tinggi dan sangat bergantung pada kondisi pencitraan yang stabil. Begitu pula penelitian Khan et al. (2024) yang menggunakan SVM, namun terbatas pada fitur visual yang diekstraksi manual dan kurang mampu beradaptasi terhadap variasi kondisi nyata.

Dibandingkan dengan penelitian-penelitian tersebut, penelitian ini menghadirkan pendekatan baru melalui penggunaan YOLOv8 untuk klasifikasi tujuh varietas beras yang citranya dikumpulkan langsung dari lingkungan nyata. Selain itu, model yang dibangun tidak hanya berdiri sendiri, tetapi juga diintegrasikan ke dalam sistem aplikasi web yang memungkinkan interaksi langsung dengan pengguna. Hal ini menjadi keunggulan utama dalam hal kepraktisan dan kontribusi nyata terhadap edukasi masyarakat dalam mengenali jenis-jenis beras lokal dan impor secara cepat dan akurat.

4.2 Spesifikasi Kebutuhan Sistem Baru

4.2.1 Spesifikasi Perangkat Lunak

Seluruh perangkat lunak digunakan dalam berbagai tahap, mulai dari perancangan sistem, pelatihan model YOLOv8, hingga pengembangan antarmuka aplikasi web. Rincian perangkat lunak dapat dilihat pada Tabel 4.1 berikut:

Tabel 4.1 Spesifikasi Kebutuhan Perangkat Lunak

No.	Perangkat	Keterangan
1.	Windows 10 (32 bit)	<i>Operating System</i> utama sebagai platform pengembangan
2.	VsCode	<i>Code editor</i> yang digunakan untuk menulis dan menjalankan skrip program
3.	Python 3.12 + PyTorch	Bahasa pemrograman dan framework utama dalam pelatihan model YOLOv8
4.	Ultralytics YOLOv8	Library deteksi objek berbasis PyTorch untuk klasifikasi varietas beras

Penjelasan lebih rinci sebagai berikut:

a) Sistem Operasi (Windows 10)

Operating-system menyediakan lingkungan pengembangan yang stabil untuk menjalankan seluruh aplikasi dan library yang terlibat dalam penelitian ini. Platform ini mendukung berbagai software yang digunakan, termasuk editor kode dan framework pembelajaran mesin.

b) Visual Studio Code (VSCode)

VsCode merupakan *code editor* kode ringan dan fleksibel yang digunakan untuk menulis skrip Python, mengelola struktur proyek, serta menjalankan server lokal saat proses integrasi sistem web. Penggunaan ekstensi pendukung seperti Python dan Live Server turut memudahkan pengembangan aplikasi secara efisien.

c) Python 3.12 dan PyTorch

Python digunakan sebagai bahasa pemrograman utama dalam penelitian ini karena dukungannya yang luas terhadap pengolahan data, pembuatan model AI, dan integrasi ke berbagai sistem. PyTorch digunakan sebagai framework *deep learning* yang menjadi fondasi dari model YOLOv8, yang bertugas melatih dan mengevaluasi performa klasifikasi citra varietas beras.

d) Ultralytics YOLOv8

Ultralytics YOLOv8 merupakan implementasi dari algoritma YOLO generasi terbaru yang digunakan dalam penelitian ini. Library ini mendukung pelatihan dan inferensi model klasifikasi objek dengan performa tinggi dan struktur yang mudah diintegrasikan. Karena dataset yang digunakan sudah tersedia dalam format YOLO, proses anotasi manual tidak dilakukan dalam penelitian ini.

4.2.2 Spesifikasi Perangkat Keras

Perangkat keras merupakan komponen penting yang mendukung proses pelatihan model dan pengembangan sistem klasifikasi berbasis citra. Masing-masing komponen memiliki peran khusus dalam mendukung efisiensi dan kelancaran proses pengolahan data, pelatihan model YOLOv8, serta integrasi sistem berbasis web. Tabel 4.2 berikut merangkum spesifikasi minimum dan aktual dari perangkat keras yang digunakan dalam proyek ini:

Tabel 4.2 Spesifikasi Kebutuhan Perangkat Keras

No.	Komponen	Spesifikasi
1.	<i>Processor</i>	4 GHz
2.	<i>Memory</i>	512 GB SSD
3.	<i>Random-access memory</i>	16 GB

a) Prosesor

Prosesor berfungsi sebagai pusat pengendali utama dari sistem komputer yang digunakan dalam proyek ini. Komponen ini menjalankan berbagai instruksi komputasi, termasuk pemrosesan data citra dan eksekusi algoritma deteksi objek berbasis YOLOv8. Dengan spesifikasi kecepatan 4 GHz, prosesor yang digunakan mendukung kelancaran pemrosesan citra secara lokal tanpa perlu bergantung pada server eksternal. Kecepatan prosesor yang tinggi juga memastikan proses pelatihan dan validasi model berjalan secara efisien.

b) Penyimpanan

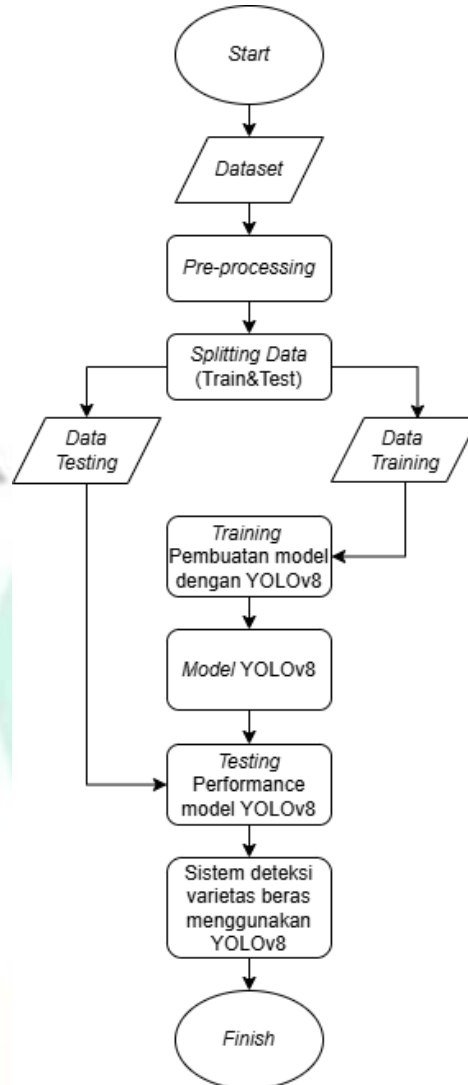
Komponen penyimpanan digunakan untuk menyimpan semua file proyek, termasuk dataset citra, model hasil pelatihan, serta file pendukung lainnya seperti konfigurasi sistem dan skrip program. Dalam penelitian ini, penyimpanan tipe SSD dengan kapasitas 512 GB digunakan untuk memastikan akses data yang cepat,

terutama saat pemanggilan gambar dan model selama proses pelatihan dan inferensi.

c) Memory (RAM)

RAM digunakan sebagai media penyimpanan sementara yang membantu sistem dalam menjalankan berbagai proses selama waktu eksekusi (*runtime*). RAM yang memadai sangat dibutuhkan dalam pelatihan dan penggunaan model YOLOv8 karena model dan data citra akan dimuat ke dalam memori selama proses berlangsung. Dengan kapasitas 16 GB, RAM yang digunakan dalam penelitian ini cukup untuk menangani proses klasifikasi citra beras tanpa mengalami hambatan signifikan dalam performa sistem.

4.3 Perancangan dan Pembuatan Sistem



Gambar 4.1 Tahap Pembuatan Model

Sistem klasifikasi varietas beras berbasis citra yang dirancang dalam penelitian ini melewati serangkaian proses yang berurutan dan terintegrasi. Proses dimulai dari pengumpulan dataset gambar berbagai jenis beras, dilanjutkan dengan tahapan *pre-processing* untuk menyiapkan data dahulu sebelum dilakukan pelatihan model. Dataset yang telah disiapkan kemudian dibagi menjadi dua bagian utama, yaitu data pelatihan (*training*) dan data pengujian (*testing*), dengan proporsi tertentu yang disesuaikan untuk menghindari overfitting.

Data pelatihan digunakan dalam proses pembentukan model menggunakan algoritma YOLOv8. Hasil pengujian ini menjadi dasar dalam menentukan tingkat akurasi dan efektivitas model dalam mengenali varietas beras berdasarkan fitur

visualnya. Model yang telah diuji dan menunjukkan performa yang baik kemudian diimplementasikan ke dalam sistem deteksi akhir, di mana pengguna dapat mengunggah citra beras dan memperoleh hasil klasifikasi secara otomatis.

4.3.1 Pengumpulan Dataset

Dataset merupakan komponen krusial dalam pengembangan sistem klasifikasi citra berbasis deep learning. Dalam penelitian ini, dataset yang digunakan berisi gambar dari tujuh varietas beras, yaitu Beras Hitam, IR42, Ketan, Merah, Basmati, Bulog, dan Japonica. Seluruh citra diperoleh dari dokumentasi langsung di lingkungan nyata, seperti pasar tradisional atau toko beras, sehingga mencerminkan kondisi pencahayaan, latar belakang, dan kemasan yang beragam.

Sebagai ilustrasi, berikut ini adalah contoh beberapa gambar dalam dataset yang digunakan beserta keterangan masing-masing varietas:

Tabel 4.3 Contoh Gambar Pada Dataset Sistem Deteksi Wajah

No.	Gambar/Citra	Keterangan
1.		Beras Hitam

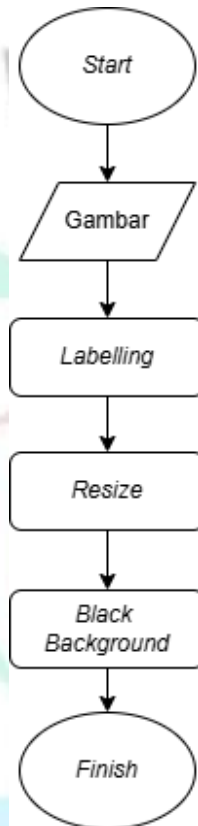
2.		Beras IR42
3.		Beras Ketan
4.		Beras Merah

5.		Beras Basmati
6.		Beras Bulog
7.		Beras Japonica

Penetapan label dilakukan secara manual berdasarkan observasi langsung terhadap karakteristik visual dari masing-masing varietas, seperti warna, bentuk butir, dan tekstur permukaan. Proses ini dilakukan untuk memastikan bahwa data yang digunakan selama pelatihan model benar-benar mewakili kelas yang dituju. Dataset yang telah dikumpulkan kemudian digunakan dalam tahap *preprocessing* dan pembagian data, sebelum akhirnya dilatih menggunakan algoritma YOLOv8.

4.3.2 *Pre-processing*

Tahap pre-processing merupakan langkah penting sebelum model YOLOv8 dapat dilatih secara optimal. Proses ini bertujuan untuk mempersiapkan dataset agar memenuhi format dan kualitas yang dibutuhkan dalam pemrosesan citra dan pelatihan model. Pada penelitian ini, proses pre-processing dilakukan secara sistematis melalui tahapan pada gambar berikut.



Gambar 4.2 Tahap Preprocessing

- a. **Start**
Proses diawali dengan menyiapkan seluruh gambar varietas beras yang telah dikumpulkan.
- b. **Gambar**
Dataset citra yang digunakan berasal dari hasil dokumentasi gambar asli yang telah diklasifikasikan berdasarkan varietasnya.
- c. **Labelling**

Setiap citra diberi label menggunakan tool anotasi seperti [LabelImg atau Roboflow], dengan menentukan posisi bounding box serta kelas varietas beras yang sesuai.

d. Resize

Ukuran gambar disesuaikan ke dimensi tertentu, misalnya 640x640 piksel, agar konsisten dengan format input yang dibutuhkan oleh model YOLOv8.

e. Black Background

Tahap ini bertujuan untuk menambahkan latar belakang hitam seragam guna menghindari noise visual dan meningkatkan kontras objek terhadap latar belakang.

f. Finish

Proses pre-processing selesai dan menghasilkan dataset yang telah terstruktur serta siap digunakan dalam tahap pelatihan model.

Dengan diterapkannya tahapan pre-processing ini, kualitas dataset akan lebih terjamin dalam melakukan pelatihan dan pengujian model YOLOv8. Hal ini diharapkan mampu meningkatkan performa model dalam melakukan deteksi dan klasifikasi varietas beras berdasarkan ciri visual seperti bentuk, warna, dan tekstur.

4.3.3 Pembagian Data

Sebelum melakukan proses pelatihan pada model, kumpulan dataset yang tersedia perlu dipisahkan, yaitu data untuk pelatihan dan data untuk pengujian. Tujuannya untuk membedakan data yang diperuntukkan dalam proses pembuatan model dengan data yang dipakai dalam evaluasi akurasi serta kemampuan model dalam mengenali data yang belum pernah dilihat sebelumnya.

Pembagian dataset dilakukan dengan membagi 80% dataset sebagai data latih dan 20% untuk data uji, bobot ini merupakan konfigurasi umum dalam praktik pembelajaran mesin. Pembagian ini dilakukan secara acak namun tetap mempertahankan distribusi kelas yang seimbang agar tidak terjadi bias dalam proses pelatihan maupun pengujian. Jumlah dan proporsi pembagian data ditampilkan pada Tabel 4.4 berikut:

Tabel 4.4 Pembagian Data Pelatihan dan Pengujian

Data Type	Images	Percentage
Data Train (Pelatihan)	240	80%

Data Test (Pengujian)	60	20%
Total	300	100%

Tabel 4.4 menunjukkan bobot pembagian data dalam proses pelatihan model. Proses pelatihan dapat dilakukan secara optimal karena model mendapatkan cukup banyak variasi data untuk belajar. Sementara itu, data pengujian yang terpisah sepenuhnya dari data pelatihan digunakan untuk mengukur performa model secara objektif terhadap data baru. Strategi ini diharapkan dapat menghasilkan model klasifikasi yang tidak hanya akurat pada data pelatihan, tetapi juga mampu melakukan prediksi secara andal pada data di luar sampel yang dilatihkan.

4.3.4 Pelatihan Model dengan YOLOv8

Pelatihan model dilakukan dengan menggunakan algoritma YOLOv8, salah satu versi terbaru dari keluarga *You Only Look Once* yang dikembangkan oleh Ultralytics. Model ini dipilih karena kemampuannya dalam mendeteksi dan mengklasifikasikan objek secara real-time dalam satu proses inferensi yang efisien. Untuk melatih model, digunakan arsitektur YOLOv8n (nano) yang lebih ringan dan cocok diterapkan pada perangkat dengan spesifikasi komputasi terbatas namun tetap memiliki akurasi yang baik.

Dataset yang telah dikumpulkan dan dibagi ke dalam subset pelatihan, validasi, dan pengujian diolah dalam lingkungan pengembangan berbasis Python dan PyTorch, dengan pelatihan dijalankan menggunakan Ultralytics YOLOv8 API. Proses pelatihan dilakukan selama 50 epoch, dengan learning rate awal sebesar 0.01, batch size sebanyak 16, dan menggunakan optimizer SGD. Untuk menghindari overfitting dan meningkatkan generalisasi model, dilakukan pula beberapa teknik augmentasi seperti *horizontal flip*, *scaling*, dan *color jittering*.

Hasil dari proses pelatihan ini menunjukkan nilai *precision*, *recall*, dan *mAP@0.5* yang meningkat secara bertahap setiap epoch. Metrik evaluasi utama yang digunakan adalah mean Average Precision (mAP) karena mampu merepresentasikan performa klasifikasi multi-kelas dengan baik. Model terbaik dipilih berdasarkan checkpoint dengan nilai mAP tertinggi pada data validasi.

Secara umum, tahapan pelatihan ini menunjukkan bahwa model mampu belajar pola visual dari berbagai jenis beras secara efektif, terutama karena

penggunaan data citra dari kondisi nyata (real-world) yang lebih bervariasi dibandingkan lingkungan laboratorium.

4.4 Skenario Pengujian

Pengujian sistem merupakan tahap penting untuk mengevaluasi performa dan kualitas model yang telah dibangun. Pada penelitian ini, pengujian dilakukan untuk memastikan bahwa sistem deteksi varietas beras berbasis algoritma YOLOv8 mampu bekerja dengan optimal juga sesuai dengan tujuan. Pengujian dilakukan melalui beberapa pendekatan, yaitu: pengujian Confusion Matrix untuk mengukur akurasi klasifikasi model, pengujian Black Box untuk menilai fungsionalitas sistem dari sisi pengguna, dan pengujian White Box untuk memvalidasi logika internal sistem serta jalur eksekusi program.

Setiap jenis pengujian disesuaikan dengan karakteristik sistem dan tujuan evaluasi masing-masing, sehingga dapat memberikan gambaran yang menyeluruh mengenai keandalan dan efektivitas sistem yang dikembangkan.

4.4.1 Skenario Pengujian *Confusion Matrix*

Confusion Matrix memberikan analisa terkait performa model YOLOv8 dalam mendeteksi dan mengklasifikasikan varietas beras. Evaluasi ini menitikberatkan pada tiga metrik utama, yaitu akurasi, presisi, dan recall. Ketiga metrik ini dihitung berdasarkan nilai *True* dan *False* dari hasil prediksi model terhadap data uji.

Semakin tinggi nilai akurasi, presisi, dan recall (mendekati 1), maka semakin baik pula performa model dalam mengenali jenis varietas beras dengan tepat.

Tabel 4.5 Skenario Pengujian *Confusion Matrix*

No.	Fokus Ukur	Perhitungan	Hasil yang diharapkan
1.	Akurasi	$\frac{TP + TN}{TP + TN + FP + FN}$	$> 0 - 1$
2.	Presisi	$\frac{TP}{TP + FP}$	$> 0 - 1$
3.	Recall	$\frac{TP}{TP + FN}$	$> 0 - 1$

4.4.2 Skenario Pengujian *Black Box*

Evaluasi diperlukan guna mengetahui sejauh mana model dapat mendeteksi varietas beras yang telah dibangun mampu mengidentifikasi gambar berdasarkan kelas yang telah dilatih sebelumnya. *Black box* digunakan dalam tahap ini, suatu metode evaluasi sistem tanpa melihat struktur internal dari model. Fokus pengujian ini adalah mengamati apakah sistem dapat mengenali varietas beras dengan benar saat diberikan input berupa gambar, baik dari dataset asli maupun dari sumber luar (*outsourse*), selama varietas tersebut masih termasuk dalam kelas yang dikenali oleh model. Dengan demikian, pengujian ini merepresentasikan skenario nyata saat sistem dioperasikan secara langsung oleh pengguna.

Tabel 4.6 Skenario Pengujian *Black Box*

No.	Skenario Pengujian	Hasil yang Diharapkan
1.	Gambar varietas beras dari dataset asli	Sistem mampu mendeteksi dan mengklasifikasikan varietas beras dengan benar
2.	Gambar varietas beras dari sumber lain (<i>outsourse</i>) tetapi masih dalam kelas model	Sistem tetap mampu mengenali varietas dengan akurat meskipun berasal dari sumber lain

4.4.3 Skenario Pengujian *White Box*

Pengujian *white box* dilakukan untuk mengevaluasi sistem dari sisi internal, terutama alur logika dan struktur kode program yang mengendalikan proses klasifikasi varietas beras. Tujuan utama dari pengujian ini adalah memastikan bahwa setiap fungsi dan alur logika dalam kode sumber aplikasi bekerja sesuai harapan, mulai dari proses unggah gambar, prediksi oleh model YOLOv8, pengambilan informasi dari basis data, hingga penyimpanan riwayat prediksi.

Dalam tahap ini, kode sumber pada file `app.py` diuji dengan memperhatikan bagaimana variabel dan fungsi saling terhubung, serta memastikan bahwa hasil keluaran sesuai dengan kondisi yang diuji. Pengujian ini melibatkan pemahaman terhadap struktur pemrograman Python, dependensi pustaka, dan integrasi model pembelajaran mesin dengan basis data MySQL. Dengan demikian, pengujian *white box* mewakili skenario evaluasi dari sisi pengembang, untuk mendeteksi

kemungkinan kesalahan logika atau bug yang tidak terdeteksi oleh metode black box.

Tabel 4.7 Skenario Pengujian White Box

No.	Skenario Pengujian	Hasil yang Diharapkan
1.	Sistem menerima file gambar dari form input pengguna (request.files['file'])	Sistem memproses file dengan benar dan tidak mengembalikan error ketika file valid diunggah
2.	Model YOLOv8 memproses gambar dan mengembalikan indeks prediksi kelas (probs.top1)	Sistem mengubah indeks menjadi label varietas beras yang benar menggunakan CLASS_MAP
3.	Sistem mengambil data varietas dari database berdasarkan hasil prediksi	Informasi yang ditampilkan sesuai dengan data beras_info dalam basis data
4.	Sistem menyimpan riwayat prediksi ke dalam tabel riwayat	Data prediksi (nama file, varietas, confidence) berhasil disimpan dan dapat ditelusuri di tabel riwayat
5.	Sistem memberikan respons dalam format JSON berisi nama varietas dan detailnya	Respons API berisi data yang sesuai struktur dan lengkap: varietas, ukuran, warna, tekstur, dan deskripsi