

BAB III

METODE PENELITIAN

3.1 Metode Penelitian

Dalam menerapkan sistem perhitungan kelapa sawit yang menggunakan *computer vision* dengan integrasi teknologi IoT agar dapat tersusun dengan baik, diperlukan rencana atau metode yang terstruktur. Penelitian ini menggunakan pendekatan *Design and development* (D&D). Penelitian D&D bertujuan untuk menghasilkan produk atau sistem baru melalui proses perancangan serta pengembangan yang sistematis (Ellis & Levy, 2010). Metode D&D terdiri dari enam langkah utama seperti yang terlihat pada Gambar 3.1, yang memberikan kerangka kerja yang terstruktur dalam mengembangkan sebuah artefak atau sistem yang dapat diuji dan diterapkan.



Gambar 3.1 Tahapan Metode Penelitian D&D

3.1.1 Identifikasi Masalah

Industri pengolahan kelapa sawit merupakan salah satu sektor strategis di Indonesia yang berkontribusi pada perekonomian nasional. Salah satu tahap penting dalam proses produksi minyak kelapa sawit adalah pemilihan Tandan Buah Segar (TBS) yang akan diolah. Kualitas buah yang masuk ke dalam pabrik sangat menentukan mutu dan kuantitas minyak sawit yang dihasilkan. Kematangan buah kelapa sawit menentukan hasil minyak yang didapatkan. Kesalahan dalam pemilihan kelapa sawit yang mentah dapat menurunkan rendemen (*yield*) minyak, meningkatkan kadar asam lemak bebas (*Free Fatty Acid/FFA*), dapat mengakibatkan kerugian bagi perusahaan.

Permasalahan utama yang ingin dipecahkan dalam penelitian ini yaitu tidak tersedianya sistem otomatis dan akurat yang mampu mendeteksi tingkat kematangan buah kelapa sawit untuk diolah berdasarkan citra visualnya. Oleh karena itu, penelitian ini berupaya mengatasi permasalahan tersebut melalui pengembangan sistem perhitungan otomatis berbasis YOLO (*You Only Look*

Once), yang dapat melakukan deteksi objek secara *real-time* terhadap citra buah kelapa sawit serta status kematangannya serta dapat di-*monitoring* melalui *website* dan panel LED.

3.1.2 Deskripsi Tujuan

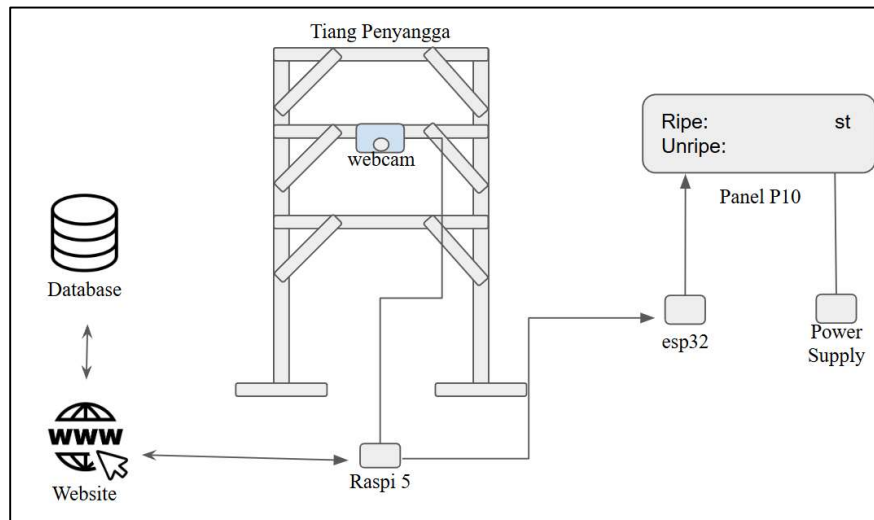
Berdasarkan permasalahan yang telah diidentifikasi pada tahap sebelumnya, penelitian ini memiliki tujuan utama untuk mengembangkan sistem perhitungan kelapa sawit secara otomatis berdasarkan citra visual. Sistem ini dirancang untuk mendukung proses seleksi kelapa sawit secara lebih cepat, akurat, dan konsisten dibandingkan metode manual. Sistem ini diharapkan dapat meningkatkan efisiensi proses panen, membantu pengambilan keputusan, serta memudahkan *monitoring* data saat pengolahan kelapa sawit.

3.1.3 Perancangan dan Pengembangan Artefak

Secara keseluruhan, penelitian ini bertujuan menghasilkan sistem klasifikasi tingkat kematangan buah kelapa sawit yang cepat, akurat, dan dapat diimplementasikan secara nyata di lapangan atau di lingkungan industri. Sistem ini diharapkan dapat meningkatkan efisiensi proses panen, membantu pengambilan keputusan, serta mengurangi ketergantungan terhadap penilaian subjektif dalam menentukan buah sawit yang siap panen.

a. Perancangan Arsitektur Sistem

Sistem yang dirancang ini terdiri dari tiga komponen utama, yaitu Raspberry Pi 5 dengan *webcam* sebagai AI server, platform web, dan modul tampilan hasil (Panel LED P10). Diagram arsitektur sistem pada Gambar 3.2 menunjukkan alur kerja sistem perhitungan tingkat kematangan buah kelapa sawit berbasis *computer vision* dengan integrasi teknologi *Internet of Things* (IoT).



Gambar 3.2 Diagram Arsitektur Sistem Perhitungan Kelapa Sawit

Dengan integrasi ini yang terlihat pada gambar 3.2, sistem mampu melakukan deteksi dan perhitungan tingkat kematangan buah kelapa sawit secara otomatis, menampilkan dan menyimpan data pada web, serta memperlihatkan hasil perhitungan melalui panel LED P10 secara *real-time*.

1. Webcam dan Raspberry Pi 5

Webcam dipasang pada tiang penyangga yang diarahkan ke jalur lewat buah kelapa sawit. Kamera ini terhubung langsung ke Raspberry Pi 5 yang berperan sebagai pusat pengolahan data. Raspberry Pi 5 menjalankan model YOLOv8 untuk mendeteksi dan menghitung jumlah buah matang dan mentah secara *real-time*.

2. Platform Web

Hasil perhitungan dari Raspberry Pi 5 dikirimkan ke platform web berbasis Django melalui jaringan. Data tersebut kemudian disimpan pada basis data SQLite yang berada di server. Website ini berfungsi untuk menampilkan data perhitungan secara *real-time*, menampilkan riwayat perhitungan, serta memberikan kontrol sistem (*start, pause, resume, stop*) kepada pengguna.

3. ESP32, Panel LED P10, dan Power Supply Unit (PSU)

Raspberry Pi 5 juga mengirimkan hasil perhitungan ke mikrokontroler ESP32 melalui komunikasi serial. ESP32 kemudian mengatur tampilan data pada Panel LED P10, yang menampilkan jumlah buah matang (*Ripe*) dan mentah (*Unripe*) secara langsung di lapangan. Panel LED P10 mendapatkan suplai daya dari Power Supply Unit.

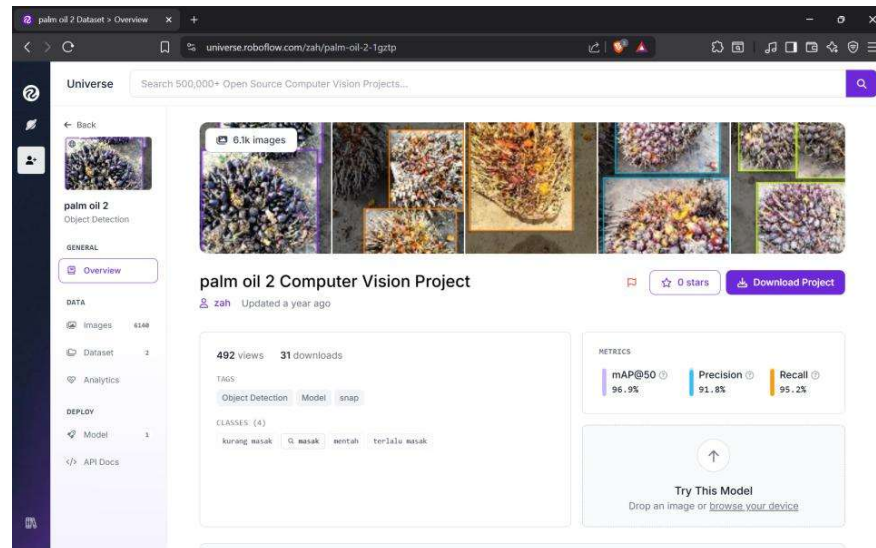
b. *AI Development Life Cycle*

Pengembangan sistem klasifikasi tingkat kematangan buah kelapa sawit dalam penelitian ini mengikuti tahapan *AI Development Life Cycle* yang umum digunakan dalam pengembangan model kecerdasan buatan (*Artificial Intelligence*). Tahapan ini mencakup proses mulai dari persiapan data, pelatihan model, evaluasi, hingga *deployment* ke perangkat keras seperti Raspberry Pi. Pendekatan ini sejalan dengan kerangka kerja yang diuraikan oleh Paleyes dkk. (2022), yang menekankan pentingnya pengelolaan *dataset* secara tepat, pemilihan arsitektur model yang sesuai, pemantauan metrik kinerja selama pelatihan, serta strategi penerapan model pada lingkungan produksi yang andal.

1. *Dataset Preparation*

Dataset yang digunakan dalam penelitian ini didapatkan dari *dataset* publik yang tersedia di Roboflow. *Dataset* yang terdapat di Roboflow telah dianotasi serta diformat agar kompatibel dengan berbagai jenis model deteksi objek termasuk YOLO. *Dataset* dibagi menjadi tiga *subset* yaitu data latih (*training*) sebesar 69%, data validasi (*validation*) sebesar 21%, dan data uji (*testing*) sebesar 10% yang dibagi oleh penyedia dataset tersebut.. Pembagian ini dilakukan agar model dapat belajar dari sebagian besar data, kemudian divalidasi untuk menghindari *overfitting*, serta diuji guna mengevaluasi performa secara objektif. Pendekatan pembagian *dataset* dengan proporsi serupa juga digunakan pada penelitian berbasis *deep learning* oleh Patil dan Kumar (2022) yang menerapkan proporsi 70% data latih, 20% validasi, dan 10% uji dalam pengembangan *framework Rice-fusion* untuk diagnosis penyakit padi.

Dataset yang digunakan dalam penelitian ini diperoleh dari platform Roboflow Universe yang terlihat pada gambar 3.3. *Dataset* tersebut bernama *palm oil 2* yang disusun oleh pengguna dengan nama zah. *Dataset* ini bersifat open *source* dan dapat digunakan untuk keperluan penelitian, khususnya dalam pengembangan model deteksi objek berbasis citra.



Gambar 3.3 Tampilan halaman *dataset* dari Roboflow (zah, 2024).

Dataset palm oil 2 berisi kumpulan gambar Tandan Buah Segar (TBS) kelapa sawit yang telah diberi anotasi *bounding box* dan diklasifikasikan ke dalam empat kelas yaitu kurang masak, masak, mentah, dan terlalu masak. Akan tetapi, pada penelitian ini, kelas yang digunakan hanya 2 yaitu matang dan mentah. Oleh karena itu kelas masak dan terlalu masak digabungkan menjadi kelas matang sedangkan kelas mentah dan kurang masak digabungkan menjadi mentah.

2. Model Training

Model dilatih menggunakan *library* Ultralytics yang mendukung YOLOv8. Pelatihan Dilakukan di Google Colab yang menyediakan penggunaan gpu secara gratis untuk mempercepat proses *training* model. Pelatihan bertujuan untuk membuat model mampu mengenali fitur visual dari buah sawit berdasarkan tingkat kematangan. Evaluasi selama pelatihan dilakukan dengan memantau metrik kinerja seperti mAP (mean *Average*

Precision), *precision*, dan *recall* yang sudah tersedia langsung dari *library* *ultralytics* dengan menghasilkan grafik kinerja model ketika pelatihan selesai dilakukan.

3. *Model Evaluation*

Setelah pelatihan, model dievaluasi menggunakan *webcam* untuk menangkap gambar serta laptop yang menampilkan *slideshow* kelapa sawit. *Webcam* ditempatkan setinggi 40 cm, 50 cm, dan 60 cm untuk melihat perbedaan kinerja model. Selain itu, model juga diuji dengan kondisi pencahayaan yang berbeda yaitu redup, normal, dan terang.

4. *Model Deployment*

Model yang telah diperoleh, di-*deploy* ke perangkat Raspberry Pi. *Deployment* dilakukan dengan meng-*install* *Ultralytics* dan dependensi pendukung di Raspberry Pi, lalu menjalankan inferensi secara langsung dari *webcam*. Raspberry Pi kemudian mengirim hasil perhitungan ke *Website* untuk ditampilkan dalam *dashboard* dan disimpan sebagai Riwayat serta ke ESP32 untuk menampilkan hasil di LED P10 secara *real-time*.

c. Pembuatan Web

Pembuatan sistem berbasis web dalam penelitian ini mengikuti model *Software Development Life Cycle* (SDLC) dengan metode *Waterfall*. Metode *Waterfall* dipilih karena karakteristik pengembangan sistem ini memiliki kebutuhan yang sudah jelas sejak awal, bersifat *linear*, serta tidak memerlukan perubahan fitur yang dinamis selama proses pengembangan. Setiap tahapan dalam model *Waterfall* dilakukan secara berurutan dan harus diselesaikan sebelum melanjutkan ke tahap berikutnya.

Model *Waterfall* cocok digunakan untuk proyek yang memiliki ruang lingkup yang jelas, dokumentasi kebutuhan yang matang, serta tidak memerlukan banyak iterasi atau perubahan di tengah proses. Hal ini sesuai dengan kebutuhan sistem dalam penelitian ini yang utamanya hanya kontrol proses perhitungan, penyimpanan riwayat data hasil perhitungan, dan *Monitoring* hasil perhitungan secara *real-time*.

Tahapan pengembangan web dengan model *Waterfall* yang diterapkan pada penelitian ini antara lain:

1. *Requirement Analysis* (Analisis Kebutuhan)

Tahapan awal ini dilakukan untuk mengidentifikasi dan mendokumentasikan seluruh kebutuhan sistem dari sisi pengguna (*user requirement*) dan teknis (*technical requirement*). Analisis dilakukan terhadap fitur-fitur utama yang dibutuhkan seperti:

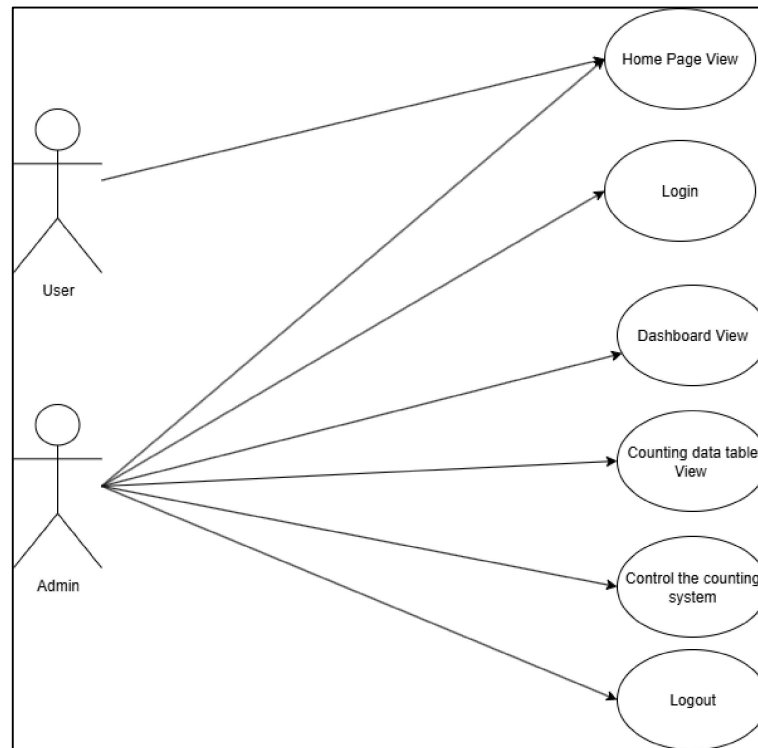
- a. Fungsi untuk memulai dan menghentikan proses klasifikasi
- b. Komunikasi antara web dengan server AI (Raspberry Pi)
- c. Penyimpanan dan penampilan riwayat hasil klasifikasi dalam bentuk tabel dan grafik.

2. *System Design* (Perancangan Sistem)

Pada tahap ini dilakukan perancangan sistem berdasarkan hasil analisis kebutuhan sebelumnya seperti perancangan arsitektur sistem secara menyeluruh, termasuk interaksi antara web, Raspberry Pi 5, dan ESP32.

- a. *Use Case Diagram*

Use case diagram ini digunakan untuk menampilkan hubungan antara aktor (*User* dan *Admin*) dengan sistem web yang dibuat. Diagram ini memperjelas fungsi-fungsi utama yang dapat diakses oleh masing-masing aktor, serta batasan hak akses dari masing-masing peran. Sistem ini memiliki dua aktor utama, yaitu *User* dan *Admin*. *Admin* memiliki akses penuh terhadap sistem, sementara *User* hanya memiliki akses yang sangat terbatas, yaitu sebatas melihat halaman utama saja. Gambar dari *use case diagram* dapat dilihat pada gambar 3.4.



Gambar 3.4 Use Case Diagram Website Sistem Perhitungan Kelapa Sawit

1) *User*

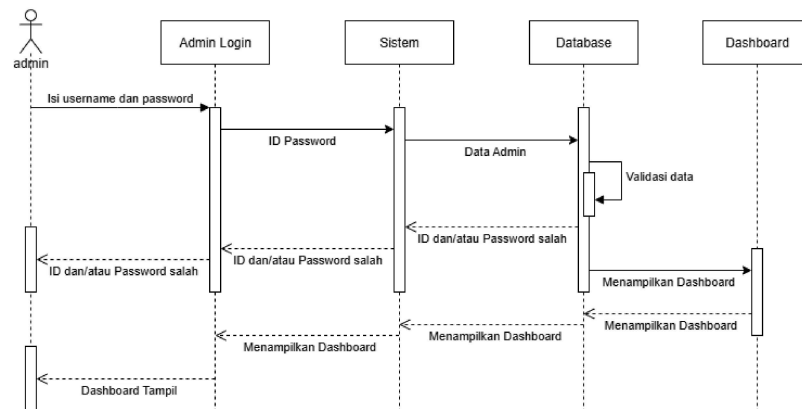
User merupakan aktor yang hanya dapat mengakses tampilan halaman utama (*home page*) dari sistem. *User* tidak memiliki hak untuk mengelola data ataupun mengontrol sistem perhitungan AI.

2) *Admin*

Admin merupakan aktor yang memiliki hak akses penuh terhadap seluruh fitur yang tersedia pada sistem. *Admin* bertanggung jawab melakukan proses verifikasi (*login*), mengelola dan memantau hasil klasifikasi, melakukan kontrol terhadap proses perhitungan sistem AI, dan melakukan *logout*.

b. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan interaksi antar komponen dalam sistem berdasarkan urutan waktu. Gambar 3.5 menunjukkan proses verifikasi *admin* hingga berhasil mengakses halaman *dashboard admin*.



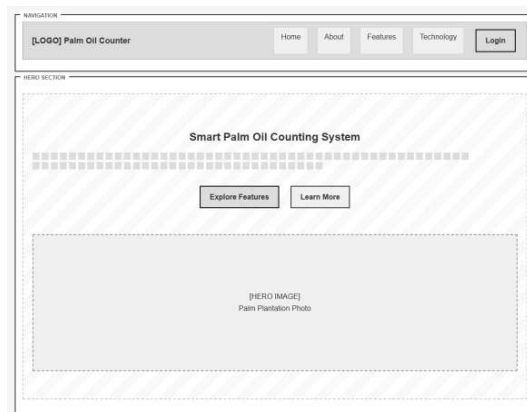
Gambar 3.5 *Sequence Diagram* Halaman *Login Website* Sistem

Perhitungan Kelapa Sawit

Sequence diagram pada gambar 3.5, menjelaskan bagaimana proses *login* dilakukan oleh *admin*, dimulai dari pengisian data *login* hingga tampilan *dashboard* ditampilkan. *Admin* melakukan *login* dengan isi *username* dan *password*. Setelah itu sistem akan melakukan pengecekan ke *database*. Ketika data benar maka *dashboard* akan ditampilkan sedangkan apabila salah maka akan mengirim kembali dan menampilkan ID dan/atau *password* salah.

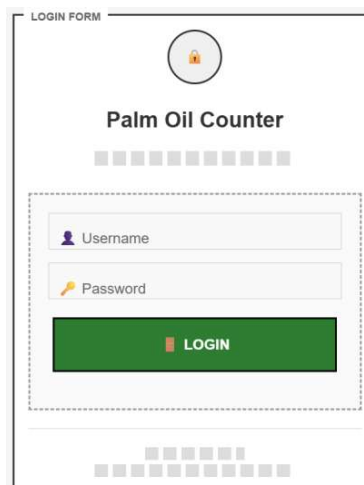
c. Rancangan Tampilan *Website*

Website dilakukan perancangan dari sisi desain dan fungsionalitas untuk mempermudah pembuatan *website* serta sebagai dasar dari pembuatan *website*. Setiap rancangan dibuat untuk menggambarkan tata letak, alur navigasi, dan fungsi dari masing-masing halaman sebelum dilakukan proses implementasi sehingga memudahkan dalam memastikan bahwa desain yang dihasilkan sesuai dengan kebutuhan pengguna dan tujuan sistem. Tampilan rancangan *website* dapat dilihat pada Gambar 3.6, Gambar 3.7, Gambar 3.8, Gambar 3.9, dan Gambar 3.10.



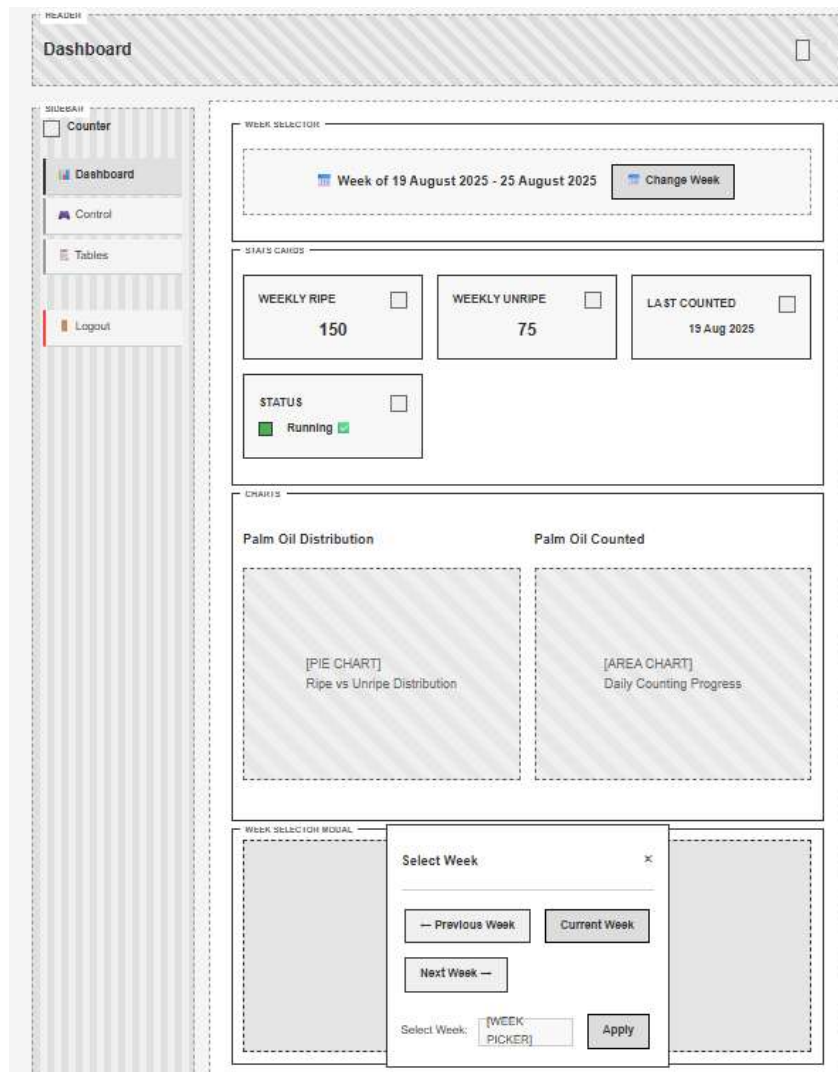
Gambar 3.6 Tampilan Rancangan *Landing Page Website* Perhitungan Kelapa Sawit

Gambar 3.6 menampilkan rancangan halaman landing page dari sistem Perhitungan Kelapa Sawit. Pada bagian atas terdapat navigasi dengan logo, menu *Home*, *About*, *Features*, dan *Technology*, serta tombol login. Terdapat *section* yang menampilkan judul utama sistem, tombol aksi (*Explore Features* dan *Learn More*), serta area untuk menampilkan gambar ilustrasi yang difokuskan untuk memperkenalkan sistem secara umum sekaligus memberikan akses cepat bagi pengguna untuk menjelajahi fitur atau masuk ke dalam sistem.



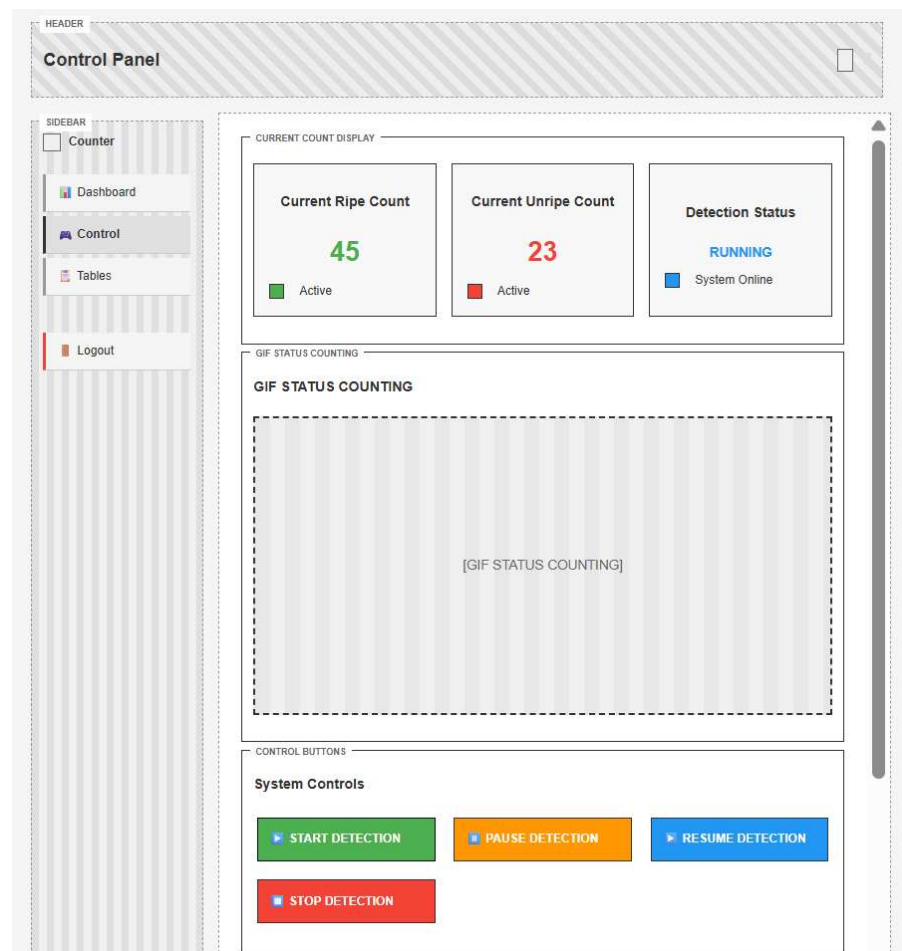
Gambar 3.7 Tampilan Rancangan Halaman *Login Website* Perhitungan Kelapa Sawit

Gambar 3.7 menampilkan rancangan halaman login dari sistem yang dirancang sederhana dengan menampilkan logo kunci di bagian atas, judul sistem, serta *form input* untuk *username* dan *password*. Tombol *login* berwarna hijau diletakkan jelas di bawah form agar pengguna dapat dengan mudah masuk ke dalam sistem. Desain ini difokuskan pada kesederhanaan dan kemudahan penggunaan, serta menjadi gerbang utama sebelum pengguna dapat mengakses fitur yang tersedia di dalam sistem.



Gambar 3.8 Tampilan Rancangan Halaman *Dashboard Website* Perhitungan Kelapa Sawit

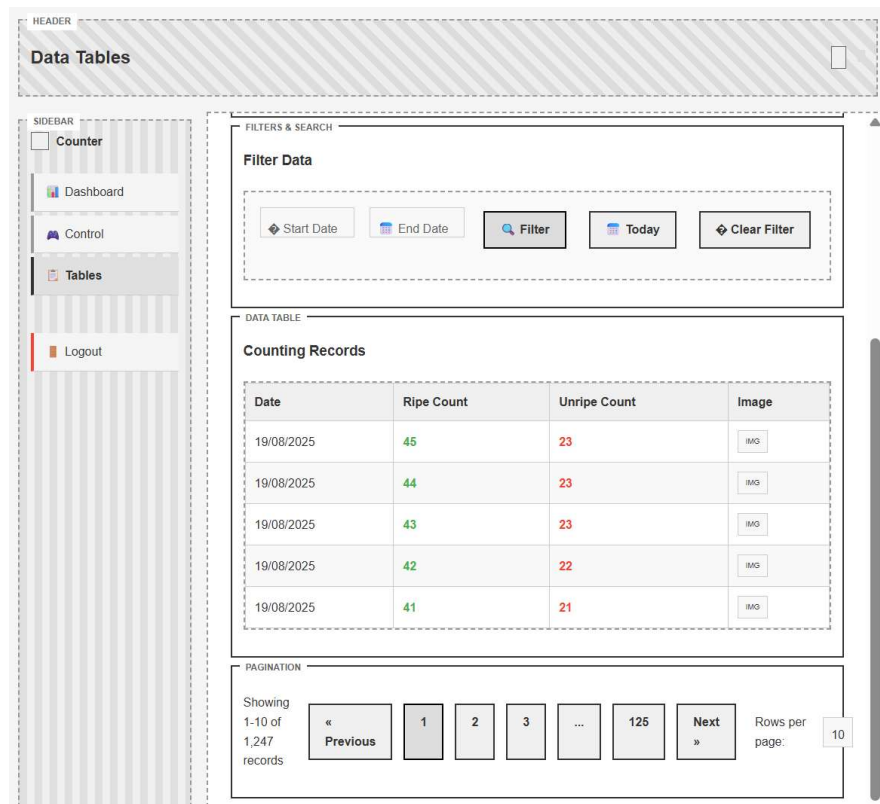
Gambar 3.8 menampilkan rancangan halaman *dashboard* yang menjadi pusat informasi dari sistem. Pada bagian atas terdapat fitur pemilihan minggu yang memungkinkan pengguna menampilkan data berdasarkan periode tertentu. Di bawahnya terdapat kartu informasi (*status cards*) yang menampilkan jumlah sawit matang dan belum matang, tanggal terakhir perhitungan, serta status sistem. Selain itu, bagian tengah menampilkan visualisasi data berupa *pie chart* dan *area chart* untuk distribusi sawit dan progres harian. Desain *dashboard* ini membantu pengguna memperoleh gambaran cepat terkait kinerja sistem dalam periode waktu tertentu.



Gambar 3.9 Tampilan Rancangan Halaman *Control Website* Perhitungan Kelapa Sawit

Gambar 3.9 merupakan rancangan halaman *control* yang berfungsi untuk mengontrol jalannya sistem deteksi. Pada bagian atas ditampilkan

jumlah terkini sawit matang, sawit belum matang, serta status deteksi yang sedang berjalan. Bagian tengah dilengkapi dengan area tampilan animasi (*GIF Status Counting*) yang menunjukkan aktivitas deteksi secara real-time. Di bagian bawah, terdapat tombol kendali untuk memulai, menghentikan, menghentikan sementara, dan melanjutkan deteksi. Halaman ini memberikan kendali kepada pengguna untuk mengatur jalannya sistem sesuai kebutuhan operasional di lapangan.



Gambar 3.10 Tampilan Rancangan Halaman *Tables Website* Perhitungan Kelapa Sawit

Gambar 3.10 merupakan rancangan halaman *control* yang berfungsi untuk mengontrol jalannya sistem deteksi. Pada bagian atas ditampilkan jumlah terkini sawit matang, sawit belum matang, serta status deteksi yang sedang berjalan. Bagian tengah dilengkapi dengan area tampilan animasi (*GIF Status Counting*) yang menunjukkan aktivitas deteksi secara real-time. Di bagian bawah, terdapat tombol kendali untuk memulai,

menghentikan, menghentikan sementara, maupun melanjutkan deteksi. Halaman ini memberikan kendali penuh kepada pengguna untuk mengatur jalannya sistem sesuai kebutuhan operasional di lapangan.

3. Implementation (Implementasi)

Pengembangan aplikasi web dengan bahasa pemrograman Python menggunakan *framework* Flask/Django. Web dikembangkan untuk dapat berkomunikasi dengan AI server (Raspberry Pi) melalui API, serta mengelola penyimpanan data riwayat hasil klasifikasi.

4. Testing (Pengujian)

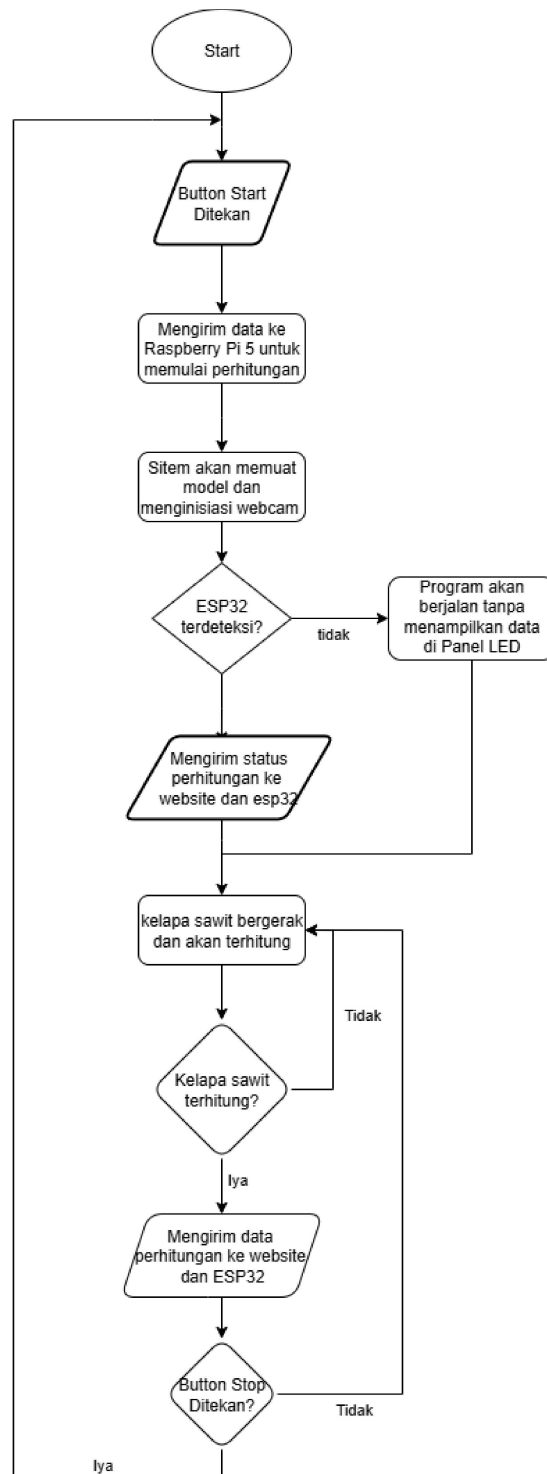
Setelah implementasi selesai, dilakukan pengujian sistem baik secara unit maupun integrasi untuk memastikan semua fitur berjalan sesuai perancangan. Pengujian meliputi koneksi dengan Raspberry Pi, *database*, dan fungsionalitas *website*.

5. Deployment (Penerapan)

Sistem yang telah selesai dan diuji kemudian di-*deploy* pada laptop sebagai server web yang siap digunakan untuk mengontrol dan memantau proses klasifikasi secara *real-time*.

d. Diagram Alir

Diagram alir merupakan diagram alur yang digunakan untuk menggambarkan tahapan proses kerja dalam sistem secara visual dan terstruktur. Diagram alir pada Gambar 3.11 menjelaskan alur kerja sistem deteksi dan perhitungan kelapa sawit dari awal hingga proses berhenti.



Gambar 3.11 Diagram Alir Sistem Perhitungan Kelapa Sawit

Diagram alir ini menggambarkan proses integrasi antara *website*, Raspberry Pi 5 yang menjalankan model YOLO, dan ESP32 yang mengontrol panel LED P10.

1) *Start*

Proses dimulai ketika sistem diaktifkan.

2) *Button Start Ditekan*

Pengguna menekan tombol Start pada halaman *website* untuk memulai proses perhitungan.

3) Kirim Perintah ke Raspberry Pi 5

Website mengirimkan perintah ke Raspberry Pi 5 agar memuat model deteksi dan menjalankan perhitungan.

4) Raspberry Pi 5 Memuat Model dan menyiapkan *Webcam*
Sistem akan memuat model YOLOv8 yang telah dilatih untuk mendeteksi kelapa sawit dan mengaktifkan kamera.

5) ESP32 Terdeteksi?

Sistem memeriksa apakah ESP32 yang bertugas menampilkan angka di panel LED P10 telah terhubung:

- a. Jika Tidak, maka program tetap berjalan, namun data tidak akan ditampilkan di panel LED.
- b. Jika Ya, sistem melanjutkan proses dan mengirimkan status perhitungan ke *website* dan ESP32.

6) Kelapa Sawit Bergerak dan Terdeteksi

Ketika kelapa sawit muncul dan terdeteksi oleh model AI, sistem akan memproses data tersebut.

7) Kelapa Sawit Terhitung?

Sistem memeriksa apakah kelapa sawit berhasil dihitung oleh algoritma:

- a. Jika Tidak, sistem menunggu input berikutnya.
- b. Jika Ya, maka sistem mengirimkan data hasil perhitungan ke *website* dan ESP32.

8) *Button Stop Ditekan?*

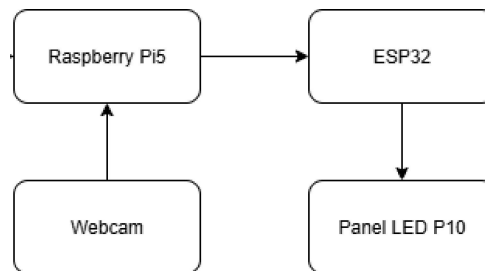
Sistem menunggu sampai pengguna menekan tombol *Stop* pada web:

- a. Jika Tidak, proses akan terus berjalan dan mendeteksi objek berikutnya.

b. Jika Ya, maka proses perhitungan dihentikan dan sistem dinonaktifkan.

e. Diagram Blok sistem

Diagram blok merupakan representasi grafis dari arsitektur sistem yang menunjukkan komponen-komponen utama dan hubungan antar komponen tersebut. Pada sistem ini, blok diagram digunakan untuk menggambarkan hubungan antara Raspberry Pi 5, ESP32, *webcam*, dan panel LED P10. Gambar dari blok diagram sistem dapat dilihat pada Gambar 3.12.



Gambar 3.12 Blok Diagram Sistem Perhitungan Kelapa Sawit

1) Raspberry Pi 5

Raspberry Pi bertindak sebagai server dari sistem deteksi. Komponen ini bertugas untuk menjalankan model deteksi objek berbasis YOLOv8 dan mengakses *webcam* secara *real-time*. Raspberry Pi menerima perintah dari *website* dan mengolah hasil deteksi untuk dikirim ke *database* dan ESP32.

2) ESP32

ESP32 berfungsi untuk menerima data hasil perhitungan dari Raspberry Pi dan menampilkannya ke Panel LED P10. ESP32 bekerja sebagai penghubung antara sistem AI dengan perangkat keras *display*.

3) *Webcam*

Webcam terhubung ke Raspberry Pi dan digunakan untuk menangkap citra secara *real-time*. Data gambar ini kemudian diolah oleh model YOLOv8 untuk mendeteksi dan menghitung jumlah kelapa sawit.

4) Panel LED P10

Panel LED P10 adalah media *output* visual yang digunakan untuk menampilkan jumlah kelapa sawit yang telah terdeteksi dan terhitung. Nilai ini dikirimkan dari ESP32 setelah menerima data dari Raspberry Pi.

3.1.4 Pengujian Artefak

a. Pengujian Akurasi Model (YOLOv8)

Pengujian Akurasi model bertujuan untuk mengukur tingkat akurasi dari model deteksi objek YOLOv8 yang telah dilatih menggunakan *dataset* publik dari Roboflow. Model ini bertugas melakukan klasifikasi tingkat kematangan buah kelapa sawit menjadi dua kelas yaitu matang dan mentah. Langkah-langkah yang dilakukan dalam pengujian ini yaitu:

1. Menyiapkan *dataset* uji yang berbeda dari data latih, validasi, dan testing yang berjumlah 14 gambar dengan 18 objek.
2. Membuat *slideshow* dari *dataset* agar dapat terlihat seperti mensimulasikan conveyor.
3. Memasang webcam ke laptop yang berperan sebagai server menjalankan AI.
4. Menjalankan program perhitungan dan *slideshow* kelapa sawit.
5. Merekam proses perhitungan dan klasifikasi yang kemudian dicatat hasilnya.

b. Pengujian Web

Pengujian ini bertujuan untuk memverifikasi fungsionalitas dari sistem web sebagai pusat kontrol dan *monitoring* data yang diperoleh. Metode pengujian yang digunakan yaitu *Black-Box Testing*. Pengujian *black box testing* merupakan pengujian yang berfokus pada *input* dan *output* tanpa memperhatikan proses internal kode program (Wulandari & Sulistyanto, 2025). Objek yang diuji pada penelitian ini dapat dilihat pada tabel 3.1.

Tabel 3.1 Tabel Pengujian *Blackbox Website* Sistem Perhitungan Kelapa Sawit

No	Fitur	Skenario	Hasil yang Diharapkan
Halaman Landing Page			
1	<i>Navbar Home</i>	Menekan <i>Button Navbar Home</i>	Bergeser ke <i>section Home</i>

No	Fitur	Skenario	Hasil yang Diharapkan
2	<i>Navbar About</i>	Menekan <i>Button Navbar About</i>	Bergeser ke <i>section About</i>
3	<i>Navbar Features</i>	Menekan <i>Button Navbar Features</i>	Bergeser ke <i>section Features</i>
4	<i>Navbar Technology</i>	Menekan <i>Button Technology</i>	Bergeser ke <i>section Technology</i>
5	<i>Button Explore Features</i>	Menekan <i>Button Explore Features</i>	Bergeser ke <i>section Features</i>
6	<i>Button Learn More</i>	Menekan <i>Button Learn More</i>	Bergeser ke <i>section About</i>
7	<i>Button Login</i>	Menekan <i>Button Login</i>	Berpindah ke halaman <i>login</i>
8	<i>Button Access System</i>	Menekan <i>Access System</i>	Berpindah ke halaman <i>login</i>
Halaman <i>Login</i>			
1	<i>Button Login</i>	Menekan <i>button login</i> dengan <i>username</i> dan/atau <i>password</i> kosong	Terdapat <i>popup</i> “please fill out this field”
		Menekan <i>button login</i> dengan <i>username</i> dan/atau <i>password</i> salah	Terdapat <i>popup</i> “Username atau password salah!”
Halaman <i>Dashboard</i>			
1	<i>Card Weekly Ripe</i>	Data <i>card weekly Ripe</i> ter- <i>update</i> ketika terdapat data baru masuk	Data <i>weekly ripe</i> tampil dan ter- <i>update</i> ketika terdapat data baru masuk.

No	Fitur	Skenario	Hasil yang Diharapkan
2	<i>Card Weekly Unripe</i>	Data <i>card weekly</i> Unipe ter- <i>update</i> ketika terdapat data baru masuk	Data <i>card weekly unripe</i> tampil dan ter- <i>update</i> ketika terdapat data baru masuk.
3	<i>Card Last Counted</i>	<i>Card</i> menampilkan data tanggal terakhir perhitungan.	Data <i>card</i> menampilkan data tanggal terakhir perhitungan dan ter- <i>update</i> otomatis.
4	<i>Card Status</i>	<i>Card</i> menampilkan data status koneksi dari web ke raspberry pi 5 ketika program berjalan berhenti <i>pause error</i> .	Data <i>card</i> menampilkan data status <i>counting</i> dari raspberry pi 5 dan akan ter- <i>update</i> sesuai dengan statusnya.
5	<i>Pie Chart</i>	<i>Pie chart</i> menampilkan data <i>ripe</i> dan <i>unripe</i> mingguan.	<i>Pie chart</i> menampilkan data mingguan dengan <i>ripe</i> hijau dan <i>unripe</i> merah dan akan ter- <i>update</i> ketika data baru masuk.
6	<i>Line Chart</i>	<i>Line chart</i> menampilkan data <i>ripe</i> dan <i>unripe</i> mingguan.	<i>Pie chart</i> menampilkan data mingguan dengan <i>ripe</i> hijau dan <i>unripe</i> merah dan akan ter- <i>update</i> ketika data baru masuk.
7	<i>Button change period</i>	Menekan <i>button change period</i>	Menampilkan <i>popup</i> pengaturan tanggal dari data yang ingin ditampilkan.

No	Fitur	Skenario	Hasil yang Diharapkan
8	<i>Popup change period</i>	Menekan <i>button previous month/week/year</i>	Mengganti menjadi bulan sebelumnya.
		Menekan <i>button next month/week/year</i>	Mengganti menjadi bulan setelahnya.
		Menekan <i>button week/month/year</i>	Mengganti pengaturan untuk melakukan penyesuaian data berdasarkan mingguan, bulanan, atau tahunan.
		Menekan <i>button icon</i> kalender	Menampilkan dan dapat memilih minggu/bulan/tahun dari data yang ingin ditampilkan.
		Menekan <i>button apply</i>	Mengubah data yang ditampilkan pada <i>dashboard</i> berdasarkan data rentang waktu yang dipilih.
Halaman <i>Control</i>			
1	<i>Button Start</i>	Menekan <i>button start</i> dan akan berubah menjadi <i>loading</i>	Ketika <i>button start</i> ditekan tampilan dan status <i>counting</i> akan berubah dan mengirimkan perintah ke Raspberry Pi 5 untuk menjalankan program. <i>Button start</i> menjadi tidak dapat ditekan dan hanya dapat menekan <i>button pause</i> dan <i>stop</i> .

No	Fitur	Skenario	Hasil yang Diharapkan
2	<i>Button Pause</i>	Menekan <i>button pause</i>	<i>Button pause</i> akan berubah menjadi <i>resume</i> dan perhitungan akan berhenti serta status di <i>dashboard</i> juga akan menjadi <i>paused</i>
3	<i>Button Resume</i>	Menekan <i>button resume</i>	Program akan melanjutkan perhitungan dari titik <i>pause</i> sebelumnya
4	<i>Button Stop</i>	Menekan <i>button stop</i>	Program akan berhenti dengan memberikan perintah ke Raspberry Pi 5 serta akan mengganti perhitungan yang tampil di halaman <i>ripe</i> Dan <i>unripe</i> kembali menjadi 0.
Halaman <i>Table</i>			
1	<i>Data Table</i>	Menampilkan data perhitungan	Data perhitungan akan tampil yang dibatasi menjadi 10 <i>row</i> terbaru.
2	<i>Button Navigasi Table</i>	Menekan <i>button</i> kiri atau kanan	Menekan <i>button</i> kiri atau kanan, menampilkan 10 data setelahnya atau sebelumnya.
3	<i>Button Filter</i>	Memasukkan data dengan rentang <i>start date</i> lebih awal dan <i>end date</i> lebih akhir yang terdapat data.	Data yang tampil akan berubah menjadi data yang telah ditentukan tanggal di bagian <i>start</i> dan <i>end date</i> .

No	Fitur	Skenario	Hasil yang Diharapkan
		Memasukkan data dengan <i>end date</i> lebih awal dibandingkan <i>start date</i>	Menampilkan pesan “No data available for selected date range.”.
4	<i>Button Today</i>	Menekan <i>button today</i>	Menekan <i>button today</i> akan otomatis mengganti <i>input</i> ke tanggal serta data ke hari saat <i>button</i> ditekan dan akan menampilkan data yang ada di hari tersebut.
5	<i>Page number</i>	Menekan <i>page number</i>	Menampilkan <i>popup</i> navigasi untuk pindah ke halaman yang diinputkan angkanya oleh <i>user</i> .
		<i>Popup page number</i>	Menekan <i>close</i> ketika di bagian <i>popup</i> akan menutup <i>popup page number</i> .
6	<i>Image</i>	Menekan <i>image</i>	Menampilkan <i>popup</i> yang menampilkan gambar lebih besar beserta dengan keterangannya.
		<i>Close popup image</i>	Menekan <i>close</i> ketika di bagian <i>popup</i> akan menutup <i>popup image</i> .
Menu <i>Sidebar Admin</i>			

No	Fitur	Skenario	Hasil yang Diharapkan
1	<i>Dashboard</i>	Menekan <i>dashboard</i>	berpindah ke halaman <i>dashboard</i> .
2	<i>Control</i>	Menekan <i>control</i>	berpindah ke halaman <i>Control</i> .
3	<i>Table</i>	Menekan <i>table</i>	berpindah ke halaman <i>table</i> .
4	<i>Logout</i>	menekan <i>logout</i>	mengalihkan <i>user</i> kembali ke halaman <i>login</i> dan sesi <i>login</i> berakhir.
5	Menu <i>admin dashboard/control/table</i>	mencoba mengakses langsung dengan url halaman <i>admin</i>	jika mengakses langsung tanpa <i>login</i> akan dialihkan ke halaman <i>login</i> .

c. Pengujian Keseluruhan Sistem (Integrasi Raspberry Pi, ESP32, LED P10)

Pengujian ini bertujuan memastikan seluruh sistem bekerja secara terintegrasi mulai dari web, AI (Raspberry Pi), hingga tampilan *real-time* pada panel LED P10. Langkah-langkah yang dilakukan dalam pengujian ini dapat dilihat pada tabel 3.2.

Tabel 3.2 Tabel Pengujian Keseluruhan Sistem Perhitungan Kelapa Sawit

No	Skenario
1	<i>User</i> menekan tombol start, mengirimkan perintah ke Raspberry Pi 5 untuk menjalankan program perhitungan dan meng- <i>update</i> status di <i>dashboard</i> web dan <i>control</i> menjadi <i>loading</i>
2	Setelah <i>loading</i> selesai status <i>counting</i> akan berubah menjadi rudi panel LED serta menjadi <i>running</i> di halaman <i>dashboard</i> dan halaman <i>control</i>
3	Kelapa sawit tersebut dipantau melalui webcam yang terhubung ke Raspberry Pi 5. ketika kelapa sawit melewati garis, data <i>ripe/unripe</i> akan bertambah di <i>popup</i> layar dan akan mengirimkan data

No	Skenario
	perhitungan ke <i>database</i> web server yang juga akan langsung <i>ter-update</i> di halaman <i>Control</i> dan panel LED yang dikirimkan melalui serial arduino.
4	Ketika di- <i>pause</i> status <i>counting</i> di panel LED akan menjadi Pa serta di halaman <i>dashboard</i> dan <i>control</i> akan menjadi <i>paused</i> .
5	Jika <i>user</i> ingin melanjutkan perhitungannya dapat menekan <i>button resume</i> atau <i>stop</i> untuk memberhentikan sesi perhitungannya.
6	Perhitungan akan selesai dan data akan kembali menjadi 0 untuk <i>ripe</i> dan <i>unripe</i> .

3.1.5 Evaluasi Hasil

Tahap evaluasi hasil pengujian dilakukan untuk menganalisis apakah artefak yang telah dikembangkan, baik dari sisi model AI, web, maupun integrasi keseluruhan sistem, telah memenuhi kriteria yang diharapkan sesuai dengan tujuan penelitian. Evaluasi dilakukan berdasarkan hasil pengujian yang telah dilaksanakan pada tahap sebelumnya, meliputi pengujian akurasi model, pengujian fungsionalitas web, dan pengujian integrasi keseluruhan sistem.

a. Evaluasi Hasil Pengujian Akurasi Model (YOLOv8)

Hasil pengujian akurasi model dievaluasi dengan membandingkan nilai metrik seperti *accuracy*, *precision*, *recall*, dan mean *Average Precision* (mAP). Model dalam mengklasifikasikan dan menghitung buah kelapa sawit menjadi dua kategori, yaitu matang dan mentah. Nilai keyakinan atau *confidence* diukur untuk melihat kinerja dari model yang dibangun dengan membandingkan hasil yang diperoleh dari setiap pengujian dari ketinggian dan pencahayaan.

b. Evaluasi Hasil Pengujian Web (Black-Box Testing)

Evaluasi hasil pengujian web dilakukan dengan mencocokkan output sistem terhadap fungsionalitas yang telah dirancang. Web dinyatakan berhasil apabila semua fungsionalitas yang ada di tabel 3.1 berhasil. Apabila seluruh fungsi berjalan sesuai dengan skenario uji (*test case*) yang telah dirancang tanpa adanya kesalahan atau kegagalan sistem, maka dapat disimpulkan bahwa web telah memenuhi kriteria keberhasilan.

c. Evaluasi Hasil Pengujian Integrasi Sistem (Raspberry Pi 5, ESP32, LED P10)

Evaluasi terhadap integrasi sistem dilakukan dengan memastikan bahwa sistem dapat berjalan tanpa adanya kendala. Jika hasil pengujian menunjukkan seluruh perangkat dapat bekerja secara terhubung dan menampilkan hasil dengan akurat sesuai proses klasifikasi AI, maka integrasi sistem dinyatakan berhasil.

3.1.6 Komunikasi Hasil Pengujian

Tahap terakhir dari siklus penelitian berbasis *Design and development* (D&D) adalah laporan hasil pengujian. Pada tahap ini, seluruh hasil pengujian yang telah diperoleh dikomunikasikan dalam bentuk laporan tertulis sebagai bagian dari penyusunan laporan skripsi. Komunikasi ini bertujuan untuk menyampaikan sejauh mana sistem yang telah dikembangkan mampu memenuhi tujuan penelitian dan dapat diimplementasikan sesuai kebutuhan.

Hasil pengujian didokumentasikan dalam bentuk tabel, grafik, dan gambar pendukung agar memudahkan pembaca dalam memahami capaian dari penelitian ini. Seluruh data pengujian ini nantinya akan disajikan lebih rinci pada Bab IV (Hasil dan Pembahasan) dalam skripsi. Dengan demikian, melalui tahapan komunikasi hasil ini, seluruh rangkaian proses penelitian yang dimulai dari identifikasi masalah hingga pengujian sistem telah tersusun secara utuh dan dapat dipertanggungjawabkan secara ilmiah.

3.2 Spesifikasi Alat dan Komponen

Penelitian ini menggunakan berbagai perangkat keras dan perangkat lunak yang saling terintegrasi untuk membangun sistem perhitungan kematangan buah kelapa sawit berbasis YOLOv8. Spesifikasi alat dan komponen yang digunakan dapat dilihat pada tabel 3.3.

Tabel 3.3 Tabel Spesifikasi Komponen Perangkat yang Digunakan

No	Komponen	Spesifikasi
Perangkat Keras		
1	Single Board Computer (Raspberry Pi 5)	Broadcom BCM2712 Quad-core Cortex-A76 @ 2.4GHz, RAM 8 GB LPDDR4X, GPU VideoCore VII, microSD 128 GB Class 10, Debian GNU/Linux 12 (Bookworm) 64-bit, Python 3.11.2
2	Laptop Pengembangan (Advan Workplus)	Windows 11, RAM 16 GB, prosesor Intel Core i5 Gen 11, Python 3.13.4
3	Mikrokontroler ESP32 DevKit V1	Dual-core Xtensa LX6 @ 240 MHz, 520 KB SRAM + 4 MB Flash, Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BLE
4	Webcam NYK Nemesis A60	Resolusi Full HD 1080p (1920×1080), 30 FPS
5	Panel LED P10	Ukuran 320×160 mm, resolusi 32×16 <i>pixel</i> , tegangan 5V DC
6	Power Supply	12V DC 5A
7	Adaptor Daya Raspberry Pi	5V DC 3A
8	Peralatan Tambahan	Kabel <i>jumper</i> , <i>breadboard</i> , <i>heatsink</i> + kipas pendingin
Perangkat Lunak		
1	Sistem Operasi Raspberry Pi	Debian GNU/Linux 12 (Bookworm) 64-bit
2	Sistem Operasi Laptop	Windows 11
3	Framework Web	Django Framework (Python)

No	Komponen	Spesifikasi
4	Bahasa Pemrograman	Python 3.13.4 (Laptop), Python 3.11.2 (Raspberry Pi), C++ (Arduino)
5	<i>Database</i>	SQLite
6	<i>Library</i> Python	Django ($\geq 4.2.0$), djangorestframework ($\geq 3.14.0$), ultralytics ($\geq 8.0.0$), opencv-python ($\geq 4.8.0$), torch ($\geq 2.0.0$), torchvision ($\geq 0.15.0$), numpy ($\geq 1.24.0$), Pillow ($\geq 10.0.0$), Flask ($\geq 2.3.0$), requests ($\geq 2.31.0$), pyserial (≥ 3.5), pandas ($\geq 2.0.0$), python-dotenv ($\geq 1.0.0$)