

# **FARMSHIELD – ROBOT PENYEMPROT PESTISIDA OTOMATIS DENGAN DETEKSI KUTU PUTIH**

Tim Penyusun :

Kelas : TRK F1

Luthfi Alviani	(J0404231023)
Fikri Binaul Umah	(J0404231038)
Muhammad Rifki Munawar	(J0404231072)
Alicia Maharani	(J0404231090)



**TEKNOLOGI REKAYASA KOMPUTER  
SEKOLAH VOKASI IPB UNIVERSITY  
2025**

## Lembar Pernyataan

Dokumen ini dibuat untuk memenuhi Ujian Tengah Semester Mata Kuliah Rekayasa Perangkat Lunak TA 2025/2026.

Karya di dalamnya adalah murni ide dan karya dari tim penyusun. Jika ada rujukan materi yang memuat ide/karya orang lain dan dicantumkan dalam dokumen ini, maka sudah dituliskan dalam daftar pustaka. Karya ini bebas dari plagiarisme dan tim penyusun bersedia menerima segala konsekuensi hukum jika hal tersebut terbukti terjadi.

Tim yang menyatakan :

Tanggal :24 September 2025

NO	Nama	NIM	Status	Tanda Tangan
1	Fikri Binaul Umah	J0404231038	Ketua	
2	Luthfi Alviani	J0404231023	Anggota	.
3	Muhammad Rifki Munawar	J0404231072	Anggota	
4	Alicia Maharani	J0404231090	Anggota	

## 1. Dokumen Perencanaan

### a. Inisiasi Proyek Berupa *Project Charter*

#### Project Charter - FarmShield

Project Title	FarmShield – Robot Penyemprot Pestisida Otomatis dengan Deteksi Kutu Putih
Start Date	19/08/2025
End Date	10/11/2025
Project Manager	Fikri Binaul Umah
Project Mentor	Ardi
Project Sponsor	Program Studi Teknologi Rekayasa Komputer, Sekolah Vokasi IPB
Project Owner	Tim FarmShield (Luthfi Alviani, Fikri Binaul Umah, Muhammad Rifki Munawar, Alicia Maharani)

Business Issue/Problem (Why do we need to do this project?)	Project Description (What do we want to achieve?)
<ul style="list-style-type: none"><li>- Petani hortikultura (cabai, singkong, terong) sering mengalami serangan hama kutu putih hingga 30–80%.</li><li>- Penyemprotan manual dengan knapsack sprayer ±15 kg tidak efisien, melelahkan, dan berisiko kesehatan.</li><li>- Belum ada solusi yang memadukan AI deteksi hama + robot penyemprot otomatis + dashboard monitoring.</li></ul>	<ul style="list-style-type: none"><li>- Mengembangkan robot penyemprot pestisida otomatis berbasis ESP32, ESP32-CAM, dan Machine Learning.</li><li>- Penyemprotan presisi hanya pada area yang terinfestasi.</li><li>- Menyediakan dashboard monitoring untuk memantau deteksi, log penyemprotan, dan laporan infestasi.</li><li>- Mengurangi penggunaan pestisida berlebihan sekaligus melindungi kesehatan petani.</li></ul>

In-Scope (Included in this project)	Out of Scope (Excluded in this project)
<ul style="list-style-type: none"><li>- Pembuatan robot penyemprot otomatis (chassis, motor, pompa, nozzle).</li><li>- Implementasi AI deteksi hama dengan dataset citra daun singkong.</li><li>- Integrasi kontrol pompa dengan hasil deteksi AI.</li><li>- Pembangunan dashboard monitoring (backend + frontend).</li><li>- Uji coba lapangan pada tanaman cabai/terong.</li></ul>	<ul style="list-style-type: none"><li>- Produksi massal robot untuk komersialisasi.</li><li>- Integrasi dengan drone penyemprot.</li><li>- Deteksi jenis hama selain kutu putih.</li></ul>

#### Budget

Rp 3.000.000 – Rp 5.000.000 (perkiraan untuk hardware, server, dan operasional uji coba)

b. Struktur Sumber Daya Manusia (Tim)

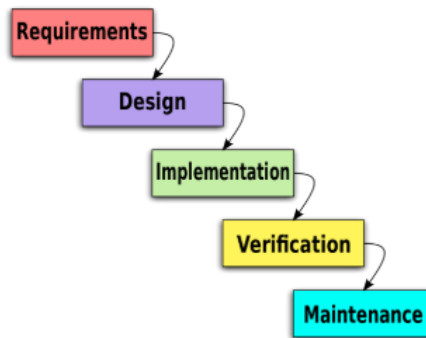
### FarmShield Team



No	Nama	Posisi
1	Fikri Binaul Umah	Manajer Proyek
2	Luthfi Alviani	Dokumenter & Tester
3	Muhammad Rifki Munawar	Programmer (Front End & Back End)
4	Alicia Maharani	Analisis (Analisis Sistem & Data)

c. Metodologi Pengembangan Sistem

- Gambar dan Penjelasan Alur Metodologi



1. Requirements

Mengidentifikasi dan mendefinisikan seluruh kebutuhan sistem di awal proyek sebagai dasar pengembangan.

2. Design

Menyusun desain arsitektur, alur proses, dan rancangan antarmuka sistem secara lengkap sebelum mulai dibangun.

3. Implementation

Melakukan pembangunan sistem sesuai desain, dengan mengembangkan setiap modul hingga menjadi satu kesatuan.

4. Verification

Mengujian sistem untuk memastikan semua fungsi berjalan sesuai kebutuhan dan memperbaiki kesalahan yang ditemukan.

5. Maintenance

Memelihara sistem setelah digunakan, melakukan perbaikan, peningkatan, dan penyesuaian sesuai kebutuhan pengguna.

- Pertimbangan Pemilihan Metodologi (penjelasan pemilihan berisi alasan kenapa metodologi ini dipilih, dikaitkan dengan karakteristik proyek)

1. FarmShield membutuhkan alur kerja yang terencana dan terstruktur sejak awal.
2. Kebutuhan sistem dapat didefinisikan secara jelas untuk meminimalkan perubahan.
3. Desain dapat disiapkan matang sebelum implementasi sehingga integrasi lebih stabil.
4. Cocok untuk proyek dengan waktu tetap ( $\pm 2$  bulan) karena tiap tahap memiliki output yang jelas.
5. Alur linear memudahkan kontrol proses dan mengurangi risiko kesalahan.

d. *Work Breakdown Structure (WBS)*

WBS dibuat berdasarkan alur metodologi yang dipilih, kemudian dibreakdown secara detail untuk melihat tasks dalam proyek

## 1. Analisis Kebutuhan

### 1.1 Observasi lapangan (kebutuhan petani)

Dari hasil observasi dan wawancara dengan petani hortikultura (cabai, singkong, terong), ditemukan beberapa permasalahan utama:

- Penyemprotan pestisida masih manual dengan knapsack sprayer ( $\pm 15$  kg) → melelahkan dan butuh banyak tenaga.
- Penyemprotan tidak presisi, pestisida mengenai seluruh tanaman, termasuk bagian yang sehat.
- Petani sering terpapar langsung pestisida → berisiko bagi kesehatan (gangguan pernapasan, kulit, dll).
- Pemborosan pestisida → biaya operasional meningkat, serta dampak negatif bagi lingkungan.
- Petani berharap adanya alat bantu otomatis yang bisa menyemprot dengan lebih efisien, presisi, dan aman.

### 1.2 Studi literatur (robotik, AI, IoT)

1. Ernawati et al. (2024) – *Inovasi Pertanian: Scarecrow Pintar Berbasis ESP32 Untuk Mengusir Hama Secara Otomatis*  
Penelitian ini mengembangkan alat pengusir hama otomatis berbasis ESP32. Hasilnya menunjukkan ESP32 dapat dimanfaatkan untuk otomasi di bidang pertanian secara murah dan efektif, sekaligus mengurangi tenaga manual petani.
2. Fachri et al. (2019) – *Rancang Bangun Robot Penyemprot Pestisida (Robot Sida) Pada Tanaman Hortikultura*  
Merancang robot penyemprot pestisida berbasis Bluetooth HC-05 dengan kapasitas 5 liter dan jarak efektif <40 meter. Robot ini mampu menyemprot dengan akurasi tinggi serta mengurangi paparan pestisida pada petani.
3. Hariyanto et al. (2020) – *Identifikasi Keanekaragaman Hama Kutu Putih (Mealybug) Pada Tanaman Singkong*  
Mengidentifikasi dua spesies kutu putih utama (*Ferrisia virgata* dan *Paracoccus marginatus*) pada singkong dengan tingkat infestasi tinggi. Hasil penelitian menegaskan perlunya teknologi pengendalian hama yang efektif.
4. Riki & Sutabri (2024) – *Perancangan Aplikasi Pendeteksi Hama*

- Tanaman Padi Berbasis Android*  
 Aplikasi Android yang memanfaatkan pengolahan citra digital dan AI untuk mendeteksi hama padi. Akurasi deteksi mencapai 85%, menunjukkan potensi integrasi AI dalam pengendalian hama lapangan.
5. Prakoso et al. (2024) – *Edukasi Pengendalian Kutu Putih Pada Tanaman Alpukat*  
 Penelitian ini berfokus pada edukasi pengendalian kutu putih dengan biopestisida yang ramah lingkungan. Menunjukkan pentingnya pendekatan non-kimia yang dapat diintegrasikan dengan sistem otomatisasi.
  6. Sari et al. (2024) – *Rancang Bangun Alat Pendeteksi dan Pengusir Hama Tanaman Kangkung Berbasis IoT*  
 Menggunakan sensor PIR untuk mendeteksi pergerakan hama, lalu melakukan penyemprotan otomatis. Sistem ini dapat diintegrasikan dengan aplikasi Telegram untuk notifikasi real-time.
  7. Sabinus et al. (2018) – *Keragaman Jenis Hama Kutu Putih pada Tanaman Singkong*  
 Menemukan bahwa serangan kutu putih menyebabkan kerusakan fisiologis serius pada tanaman singkong. Penelitian ini memperkuat urgensi solusi berbasis teknologi untuk pengendalian hama.
  8. Setiawan et al. (2022) – *Implementasi ESP32-CAM dan Blynk Pada WiFi Door Lock System*  
 Menunjukkan bahwa ESP32-CAM dapat digunakan untuk pemrosesan citra dan kontrol jarak jauh. Konsep ini dapat diadaptasi untuk mendukung deteksi hama berbasis kamera pada robot pertanian.
  9. Simarmata et al. (2021) – *Beberapa Aspek Biologi Kutu Putih (Paracoccus marginatus) Pada Terung*  
 Mengkaji siklus hidup kutu putih di rumah kaca. Hasilnya menunjukkan reproduksi tinggi dan siklus hidup cepat (31–37 hari), menjadikan hama ini ancaman besar bagi tanaman hortikultura.
  10. Toscano et al. (2025) – *Recent Developments and Future Prospects in the Integration of Machine Learning in Mechanised Systems for Autonomous Spraying*  
 Review internasional yang membahas penerapan machine learning dalam penyemprotan otonom. Artikel ini menegaskan prospek teknologi AI untuk meningkatkan akurasi dan efisiensi pertanian presisi.
  11. Tusubira et al. – *Improving In-field Cassava Whitefly Pest Surveillance with Machine Learning*  
 Penelitian menggunakan dataset citra daun singkong (7.500 gambar) untuk melatih model deteksi hama (Haar Cascade & Faster-RCNN). Hasilnya akurat dalam mendeteksi infestasi kutu putih secara real-time.
  12. Widiyanti et al. (2023) – *Rancang Bangun Robot Penyemprot Pestisida dengan Sistem Wall-Follower pada Tanaman Cabai*  
 Mengembangkan robot dengan sensor ultrasonik berbasis metode wall-

follower untuk navigasi di lahan cabai. Robot dapat menyemprot otomatis mengikuti jalur tanaman.

### 1.3 Identifikasi kebutuhan fungsional & non-fungsional

- Kebutuhan Fungsional:
  1. Login & registrasi pengguna: autentikasi via email/password, pemeriksaan brute-force, pemberian cookie role.
  2. Manajemen peran admin/user: routing otomatis ke dashboard atau dashboard-user, akses CRUD konten dibatasi admin.
  3. Streaming deteksi kutu putih: menampilkan video MJPEG dari FastAPI dan overlay bounding box hasil Firebase.
  4. Pencatatan riwayat deteksi: menyimpan dan menampilkan daftar detections (timestamp, kelas, confidence, koordinat).
  5. Dashboard statistik: menghitung total/today/akurasi, menampilkan grafik deteksi dengan filter harian/mingguan/bulanan.
  6. Ensiklopedia pertanian: admin dapat tambah/edit/hapus artikel + unggah gambar; user membaca artikel.
  7. Radio/podcast edukasi: CRUD konten audio/video dengan validasi jenis URL (YouTube, Spotify, audio langsung, dsb.) dan pemutar ter-embed.
  8. Forum komunitas: membuat post, upload foto, balas, like, pagination, modal edit/hapus, sanitasi konten, realtime replies.
  9. Chat antar pengguna: daftar user, pembuatan thread otomatis bila belum ada, pengiriman pesan teks/foto realtime.
  10. Chat AI assistant: percakapan dengan Gemini (riwayat disimpan lokal), tanggapan konteks pertanian.
  11. Logout & sesi: tombol logout yang menghapus cookie dan mengembalikan ke halaman login.
- Kebutuhan Non-Fungsional (contoh inti):
  1. Keamanan: rate limiting login, sanitasi input, hash password (register), headers keamanan (API).
  2. Kinerja realtime: penggunaan Firebase Realtime Database untuk streaming data deteksi, forum replies, dan chat.
  3. Skalabilitas UI: Next.js dengan dynamic import (mis. Three.js), responsive layout Tailwind, dark mode context.
  4. Reliabilitas data: fallback penanganan snapshot kosong, validasi struktur array/object Firebase, penanganan error upload.
  5. Kemudahan pemeliharaan: modul UI reusable (Card, Button, PageHeader, layout admin/user), pemisahan layanan (upload helper, aiChat).
  6. Integrasi eksternal: koneksi ke FastAPI streaming, Firebase (Firestore + Realtime DB), Supabase opsional, Gemini API.



7. Pengalaman pengguna: animasi (LeafAnimation, framer-motion), loading states, toast feedback, modal konfirmasi.
8. Aksesibilitas: penggunaan ikon/label jelas, teks alternatif gambar, warna kontras pada elemen penting.

#### 1.4 Penyusunan dokumen SKPL

Berdasarkan hasil observasi lapangan dan studi literatur, maka disusunlah dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini sebagai acuan pengembangan proyek FarmShield. Dokumen SKPL mencakup:

- Deskripsi umum sistem.
- Tujuan pembuatan sistem.
- Daftar kebutuhan fungsional & non-fungsional.
- Diagram fungsional (Use Case/DFD), ERD, Class Diagram.
- Spesifikasi software & hardware.

## 2. Desain Sistem (Arsitektur Umum)

### 2.1 Desain arsitektur sistem (robot, AI, dashboard)

Arsitektur FarmShield dibagi menjadi 3 komponen utama:

- Robot → terdiri dari ESP32/Arduino untuk kontrol motor & pompa, kamera ESP32-CAM untuk mengambil citra daun, sensor ultrasonik untuk navigasi.
- AI/ML Processing Unit → menggunakan Raspberry Pi/Jetson Nano untuk menjalankan model machine learning (Yolov8) untuk deteksi hama secara real-time.
- Dashboard Monitoring (Web) → berbasis React.js/Next.js, terhubung dengan backend Node.js & database. Menampilkan laporan deteksi, log penyemprotan, heatmap area infestasi, serta notifikasi.

### 2.2 Desain database (ERD)

Entity utama dalam sistem FarmShield antara lain:

- User (id\_user, nama, email, password\_hash, role, created\_at)
- DetectionRecord (id\_detection, timestamp, kelas\_hama, confidence, bbox\_coordinates, image\_url)
- DashboardStats (id\_stats, total\_detection, today\_detection, accuracy, periode)
- KnowledgeArticle (id\_article, judul, isi, image\_url, author\_id, created\_at)
- PodcastMedia (id\_media, judul, url, tipe\_media, thumbnail, created\_at)
- ForumPost (id\_post, id\_user, isi, foto\_url, tanggal)
- ForumReply (id\_reply, id\_post, id\_user, isi, foto\_url, tanggal)
- ChatThread (id\_reply, id\_post, id\_user, isi, foto\_url, tanggal)
- ChatMessage (id\_message, id\_thread, sender\_id, text, photo\_url, timestamp)
- AiChatHistory (id\_aichistory, user\_id, message, role, timestamp)

### 2.3 Desain UI dashboard

Dashboard dirancang sederhana agar mudah dipahami petani. Fitur utama:

- Halaman Login & Registrasi
- Halaman Monitoring → menampilkan live data robot, status baterai, posisi GPS.
- Halaman Deteksi Hama → menampilkan daftar foto daun + bounding box hama terdeteksi.
- Laporan & Statistik → grafik jumlah serangan, heatmap lokasi infestasi, penggunaan pestisida.
- Ensiklopedia Hama & Forum Petani

## 2.4 Desain hardware (chassis, pompa, sensor)

Komponen utama hardware FarmShield:

- Chassis Robot → 4 roda, berbahan aluminium ringan, motor DC dengan driver L298N.
- ESP32 → pengendali motor, pompa, sensor.
- ESP32-CAM → menangkap citra daun.
- Pompa DC + Nozzle → menyemprot pestisida hanya ke area terdeteksi.
- Tangki Pestisida → kapasitas  $\pm 5$  liter.
- Sensor Ultrasonik → deteksi jarak untuk navigasi & wall-follower.
- Baterai 12V Li-Ion → untuk motor & pompa.
- Raspberry Pi / Jetson Nano → pemrosesan AI.

## 3. Waterfall 1 – Modul Deteksi Hama (AI + Kamera)

### 3.1 Requirements (Kebutuhan Sistem)

- Sistem membutuhkan dataset kutu putih untuk melatih model deteksi.
- Model harus mampu mendeteksi lokasi hama dan jumlahnya pada daun
- Kamera harus dapat menangkap gambar secara real-time untuk diproses AI.
- Output deteksi harus digunakan sebagai pemicu (trigger) penyemprotan otomatis.

### 3.2 System Design (Perancangan Sistem)

#### 3.2.1 Perancangan Dataset & Label

- Dataset diambil dari penelitian Tusubira JF (7.500 citra daun.)
- Citra diambil dengan kamera smartphone 13 MP, format JPG.
- Dipilih  $\pm 2.000$  gambar untuk pelabelan bounding box menggunakan LabelImg.
- Format label: PASCAL VOC (XML).
- Label kategori infestasi: *low* ( $< 50$  ekor) dan *high* ( $> 50$  ekor).

#### 3.2.2 Perancangan Model ML

Perancangan model Machine Learning pada sistem deteksi hama dilakukan menggunakan algoritma YOLOv8 (Ultralytics), yang merupakan model object detection modern dengan performa cepat dan akurat. Perancangan ini mencakup pemilihan arsitektur, persiapan dataset, konfigurasi training, serta perencanaan evaluasi performa model.

##### 1. Pemilihan Model

Model yang dipilih adalah YOLOv8s, dengan alasan:

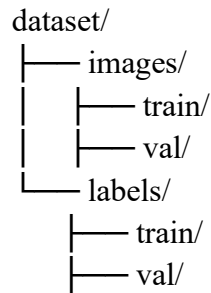
- Memiliki keseimbangan antara akurasi dan kecepatan inferensi.
- Cocok untuk implementasi di perangkat Jetson Nano, Raspberry Pi, atau server lokal.
- Mampu mendeteksi objek kecil seperti kutu putih pada daun.

Model lain seperti YOLOv8n atau YOLOv8m tetap dipertimbangkan:

- YOLOv8n → untuk perangkat sangat terbatas (low-power).
- YOLOv8m → untuk akurasi lebih tinggi pada dataset besar.

## 2. Persiapan Dataset

Dataset disiapkan dalam format YOLO dengan struktur:



Langkah-langkah persiapan:

- Mengumpulkan gambar daun yang mengandung hama *kutu putih*.
- Melakukan anotasi bounding box menggunakan LabelImg atau Roboflow.
- Membagi dataset menjadi 80% data training dan 20% data validation.
- Membuat file konfigurasi dataset `.yaml` yang berisi path dataset, class, dan struktur direktori.

### 3.2.3 Perancangan Integrasi Kamera –AI

- Kamera: ESP32-CAM atau Raspberry Pi Camera.
- Kamera mengirim frame ke Raspberry Pi/Jetson Nano untuk inferensi AI.
- Output berupa: bounding box, jumlah hama.

## 3.3 Training model ML (deteksi hama)

Satu model utama digunakan:

YOLOv8 (Ultralytics)

Model ini dipilih karena:

- Lebih ringan dan cepat, cocok untuk deployment di hardware seperti Jetson Nano, Raspberry Pi 4, ESP32-CAM + server inference.
- Memiliki akurasi tinggi untuk deteksi objek kecil seperti kutu putih.
- Mendukung training, validation, dan deployment dalam satu framework.
- Training dilakukan di Python menggunakan Ultralytics YOLOv8 dengan dataset VOC/YOLO format.

### 3.4 Integrasi kamera dengan Raspberry Pi/Jetson Nano

- Kamera yang digunakan: ESP32-CAM atau Pi Camera.
- Alur integrasi:
  1. Kamera menangkap frame citra daun.
  2. Frame dikirim ke Raspberry Pi/Jetson Nano untuk diproses oleh model ML.

3. Output berupa bounding box + jumlah hama.
4. Jika hama terdeteksi → hasil dikirim ke ESP32 (kontrol robot) sebagai trigger penyemprotan.

### 3.5 Pengujian akurasi deteksi

- Uji coba dilakukan dengan membandingkan hasil deteksi robot dengan pengamatan manual (ground truth).
- Variasi kondisi pengujian:
  1. Intensitas cahaya berbeda (terang, teduh).
  2. Sudut kamera berbeda ( $0^{\circ}$ – $45^{\circ}$ ).
  3. Tingkat infestasi berbeda (rendah, sedang, tinggi).
- Target pengujian:
  1. Akurasi deteksi  $\geq 85\%$ .
  2. Kecepatan inferensi  $\leq 2$  detik/frame agar bisa digunakan real-time.
- Hasil uji dicatat dalam confusion matrix untuk evaluasi performa model.

### 3.6 Implementation (Implementasi)

#### 3.6.1 Training Model ML

Proses pelatihan dijalankan pada perangkat dengan dukungan GPU, seperti:

- Jupyter dengan NVIDIA SSMI

Di dalam proses ini, YOLOv8 memanfaatkan kemampuan PyTorch untuk:

- melakukan forward & backward pass,
- menghitung loss (box loss, cls loss, dan dfl loss),
- serta memperbarui bobot neural network menggunakan optimizer bawaan (SGD/AdamW).

#### 3.6.2 Implementasi Integrasi Kamera–Model

- Kamera menangkap frame → dikirim ke Jetson Nano/Raspberry Pi.
- Model melakukan deteksi → mengembalikan koordinat bounding box.

#### 3.6.3 Implementasi Komunikasi ke ESP32

- Jika deteksi positif, Raspberry Pi mengirim trigger ke ESP32 untuk penyemprotan.

### 3.7 Testing (Pengujian)

#### 3.7.1 Pengujian Akurasi Model

- Variasi Uji:
  - Cahaya: terang/teduh.
  - Sudut kamera:  $0$ – $45^{\circ}$ .
  - Tingkat infestasi: rendah–tinggi.
- Target:
  - Akurasi  $\geq 85\%$ .
  - Kecepatan inferensi  $\leq 2$  detik/frame.

### 3.7.2 Dokumentasi Hasil Uji

- Menggunakan confusion matrix.

## 4. Waterfall 2 –Modul Penyemprotan Otomatis

### 3.1 Requirements (Kebutuhan Sistem)

- Robot harus dapat menyemprot otomatis berdasarkan output AI.
- Sistem butuh pompa DC, nozzle, dan mikrokontroler ESP32.
- Harus ada mode manual override.

### 3.2 System Design (Perancangan Sistem)

#### 3.2.1 Rancang Rangkaian Pompa & Nozzle

- Pompa DC 12V (3–5 L/menit).
- Nozzle low-pressure agar daun tidak rusak.
- Pompa dikendalikan oleh relay/MOSFET ke ESP32.
- Tangki 5 liter di bagian belakang robot.

#### 3.2.2 Perancangan Alur Kontrol

- AI mendeteksi → kirim sinyal → ESP32 menyalakan pompa.
- Menyediakan mode manual via dashboard.

### 3.3 Implementation (Implementasi)

#### 3.3.1 Integrasi ESP32-AI

- ESP32 menerima sinyal "spray ON".
- Pompa aktif 1–2 detik sesuai posisi target.

#### 3.3.2 Pencatatan Log Penyemprotan

- Data disimpan ke *SprayLog*: waktu, volume, lokasi, status.

### 3.4 Testing (Pengujian)

#### 3.4.1 Pengujian di Lahan Uji

- Skenario Uji:
  - Deteksi → Semprot otomatis.
  - Cakupan semprotan.
  - Perbandingan pestisida otomatis vs manual.
- Target:
  - Akurasi semprot  $\geq 80\%$ .
  - Efisiensi pestisida  $\geq 30\%$ .
  - Waktu respon  $\leq 3$  detik.

## 5. Waterfall 3– Modul Dashboard Monitoring

### 5.1 Requirements (Kebutuhan Sistem)

- Output deteksi harus digunakan sebagai pemicu (trigger) penyemprotan otomatis.
- Dashboard harus menerima data deteksi & penyemprotan real-time.
- User harus bisa melihat grafik, lokasi, dan log aktivitas.
- Sistem membutuhkan API untuk pengiriman data dari robot.

## 5.2 System Design (Perancangan Sistem)

### 5.2.1 Backend (Node.js/Express)

Fungsi:

- Menerima data dari robot (deteksi, log penyemprotan, status device).
- Simpan data ke database.
- Menyediakan endpoint untuk dashboard.
- Menggunakan autentikasi JWT.

### 5.2.2 Database (Firestore)

Tabel yang dirancang

- DetectionRecord
- SprayLog
- DeviceStatus

### 5.2.3 UI Dashboard (Next.js)

Fitur tampilan:

- Grafik serangan hama.
- Heatmap GPS.
- Log aktivitas.
- Mode responsif.

## 5.3 Modul Ensiklopedia (PestInfo)

Berisi informasi hama, gejala, dan solusi.

## 5.4 Notifikasi (opsional)

Notif ketika serangan hama melewati ambang batas.

## 5.5 Implementasi (implementasi)

### 5.5.1 Pengembangan Backend API

- CRUD data deteksi, semprot, device status.

### 5.5.2 Pengembangan Dashboard Web

- Grafik menggunakan chart library.
- Heatmap GPS.
- Log aktivitas realtime.

## 5.6 Testing (Pengujian)

- Uji endpoint API.
- Uji tampilan grafik & heatmap.
- Uji performa data realtime.

## 6. Tahap Integrasi & Finalisasi (Waterfall: Integration + Maintenance)

### 5.1 Integration (Integrasi Sistem)

Menggabungkan semua modul hasil tahap sebelumnya:

Alur lengkap:

- Kamera menangkap citra → AI mendeteksi hama
- AI mengirim data → ESP32 menyalakan pompa
- Pompa menyemprot → ESP32 mengirim log ke server
- Backend menyimpan → Dashboard menampilkan grafik & laporan
- Komunikasi: WiFi / HTTP / MQTT.

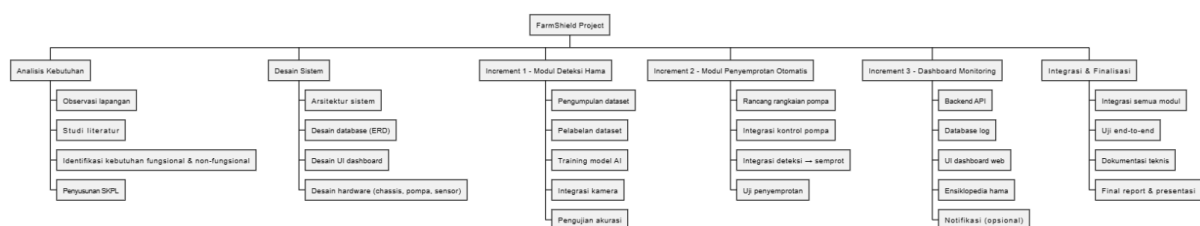
### 5.2 Documentation (Dokumentasi Teknis)

Berisi:

- Kamera menangkap citra → AI mendeteksi hama
- Akurasi deteksi.
- Respons penyemprotan
- Stabilitas koneksi.
- Ketepatan data dashboard
- Manual pengguna robot & dashboard
- Dokumentasi kode AI, backend, frontend, ESP32
- Skema rangkaian
- Diagram arsitektur

### 5.3 Final Report & Presentation

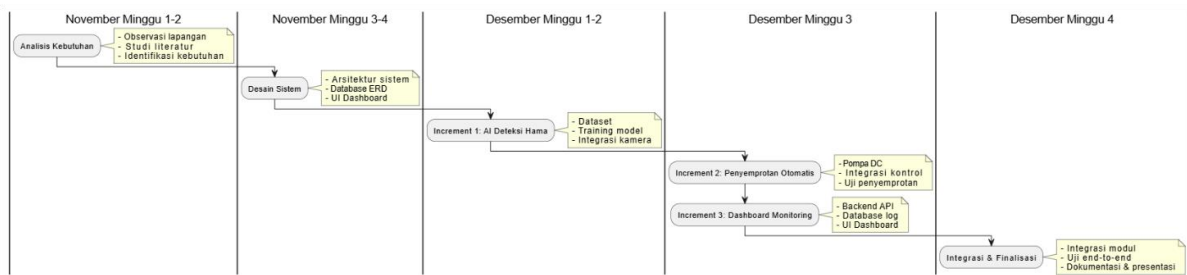
Termasuk laporan akhir, presentasi, dan demonstrasi robot.



#### e. Gant Chart

(jadwal dibuat dengan pemetaan tasks yang ada di WBS dengan SDM yang ada, silahkan bebas menggunakan tools apa saja untuk pembuatan gant chart - penjadwalan proyek selama 2 bulan November-Desember)





## 2. Dokumentasi Spesifikasi Kebutuhan Perangkat Lunak

### a. Deskripsi Umum Sistem

Sistem FarmShield adalah sistem yang dibuat untuk memberikan solusi atas permasalahan petani hortikultura dalam mendeteksi serangan hama kutu putih dan melakukan penyemprotan pestisida secara manual yang melelahkan, tidak presisi, serta berisiko bagi kesehatan petani.

Sistem yang dibangun terdiri dari robot penyemprot otomatis berbasis AI dan aplikasi berbasis web yang berfungsi sebagai dashboard monitoring. Pengguna dalam sistem ini adalah petani/ operator lapangan dan kelompok tani/koperasi, yang dapat melakukan patroli otomatis di lahan, deteksi hama berbasis kamera & machine learning, penyemprotan pestisida presisi, serta pemantauan laporan serangan hama melalui dashboard.

Dengan demikian, sistem ini dapat membantu dalam proses perlindungan tanaman hortikultura secara efisien, mengurangi penggunaan pestisida berlebihan, serta meningkatkan produktivitas hasil panen.

### b. Tujuan Pembuatan Sistem

Pembuatan sistem FarmShield bertujuan untuk memfasilitasi proses deteksi hama kutu putih dan penyemprotan pestisida secara otomatis sehingga dapat meningkatkan akurasi pengendalian hama, mengurangi kontak langsung petani dengan bahan kimia berbahaya, menghemat penggunaan pestisida, dan menyediakan laporan berbasis data yang bermanfaat bagi petani maupun koperasi.

### c. Daftar Kebutuhan fungsional

No	Nama Fungsi	Deskripsi Fungsi
1	Registrasi & Login	Fungsi untuk mendaftarkan pengguna dan melakukan autentikasi dengan email/password, termasuk pemeriksaan brute-force dan pemberian cookie role.
2	Manajemen Peran (Admin/User)	Fungsi untuk mengatur hak akses pengguna, melakukan routing otomatis ke dashboard sesuai role, serta membatasi akses CRUD hanya untuk admin.
3	Streaming Deteksi Kutu Putih	Fungsi menampilkan video MJPEG dari FastAPI dan overlay bounding box hasil deteksi hama dari Firebase.
4	Riwayat Deteksi	Fungsi menyemprot pestisida secara otomatis hanya pada area yang terdeteksi hama. Fungsi menyimpan dan menampilkan daftar hasil deteksi seperti timestamp, kelas, confidence, dan koordinat.

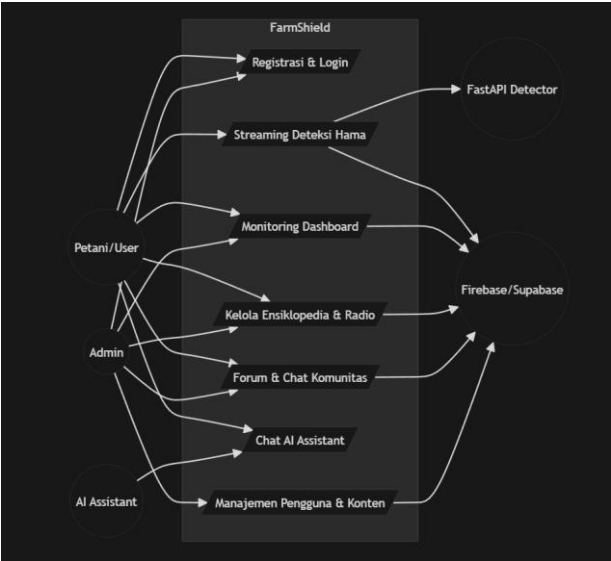
5	Dashboard Statistik	Fungsi menampilkan total deteksi, deteksi harian, akurasi model, serta grafik deteksi dengan filter harian/mingguan/bulanan.
6	Ensiklopedia Pertanian	Fungsi menyediakan artikel pertanian; admin dapat menambah, mengedit, menghapus artikel dan mengunggah gambar, sementara user dapat membaca artikel.
7	Radio / Podcast Edukasi	Fungsi CRUD konten audio/video, melakukan validasi URL (YouTube, Spotify, dsb.), serta menampilkan pemutar media ter-embed.
8	Forum Komunitas Petani	Fungsi membuat postingan, upload foto, balas, like, edit/hapus, pagination, sanitasi konten, serta mendukung balasan realtime.
9	Chat Antar Pengguna	Fungsi menampilkan daftar user, membuat thread otomatis jika belum ada, serta mengirim pesan teks/foto secara realtime.
10	Chat AI Assistant (Gemini)	Fungsi menyediakan percakapan dengan AI, menyimpan riwayat secara lokal, dan memberikan jawaban terkait pertanian.
11	Logout & Manajemen Sesi	Fungsi menghapus cookie sesi dan mengembalikan user ke halaman login

d. Daftar Kebutuhan Non-fungsional

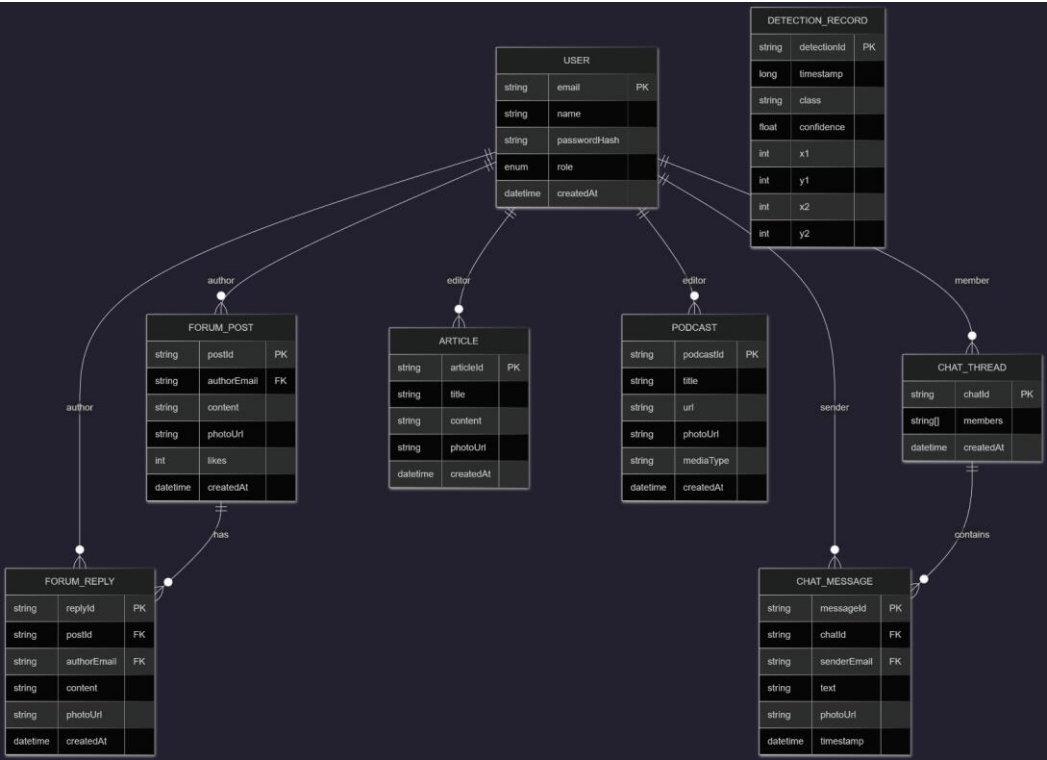
No	Nama Fungsi Non-Fungsional	Deskripsi Fungsi
1	Keamanan Sistem	Fungsi menjamin keamanan melalui rate limiting login, sanitasi input, hash password saat registrasi, serta penggunaan security headers pada API.
2	Kinerja Realtime	Fungsi memastikan aplikasi dapat memperbarui data secara realtime menggunakan Firebase Realtime Database untuk deteksi, forum replies, dan chat.
3	Skalabilitas UI	Fungsi mendukung antarmuka yang dapat diperluas seperti penggunaan Next.js dynamic import (Three.js), desain responsif Tailwind, serta dark mode.
4	Reliabilitas Data	Fungsi menjaga konsistensi data dengan fallback untuk snapshot kosong, validasi struktur array/object Firebase, serta penanganan error upload.
5	Kemudahan Pemeliharaan	Fungsi menyediakan modul UI reusable (Card, Button, Header), pemisahan service (upload helper, aiChat service) untuk memudahkan perawatan sistem.
6	Integrasi Eksternal	Fungsi mengelola koneksi ke FastAPI streaming, Firebase (Firestore + Realtime DB), Supabase opsional, dan Gemini API.
7	Pengalaman Pengguna (UX)	Fungsi memberikan pengalaman nyaman melalui animasi (LeafAnimation, framer-motion), loading states, toast feedback, dan modal konfirmasi.
8	Aksesibilitas	Fungsi memastikan akses mudah bagi semua pengguna melalui ikon/label jelas, alt-text pada gambar, dan penggunaan warna berkontras tinggi.

**Catatan :** no fungsi pada kebutuhan non fungsional biasanya melanjutkan no fungsi yang sudah ada di kebutuhan fungsional.

e. Usecase Diagram

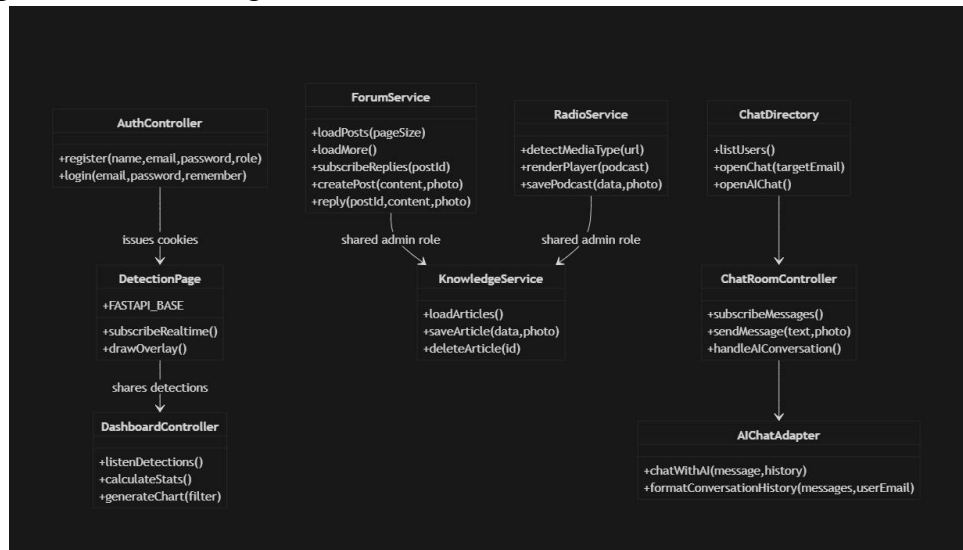


f. Diagram analisis data (entity relationship diagram- ERD)



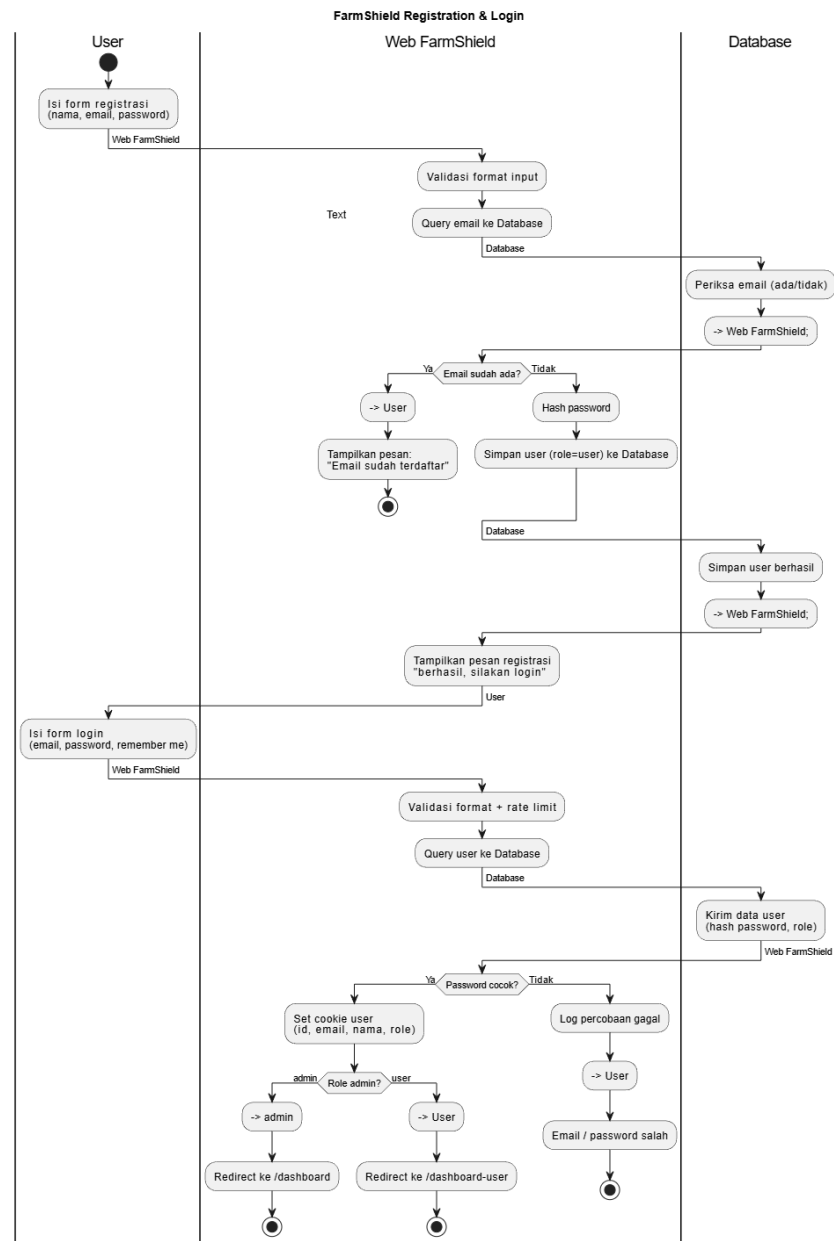


## g. Model Class Diagram



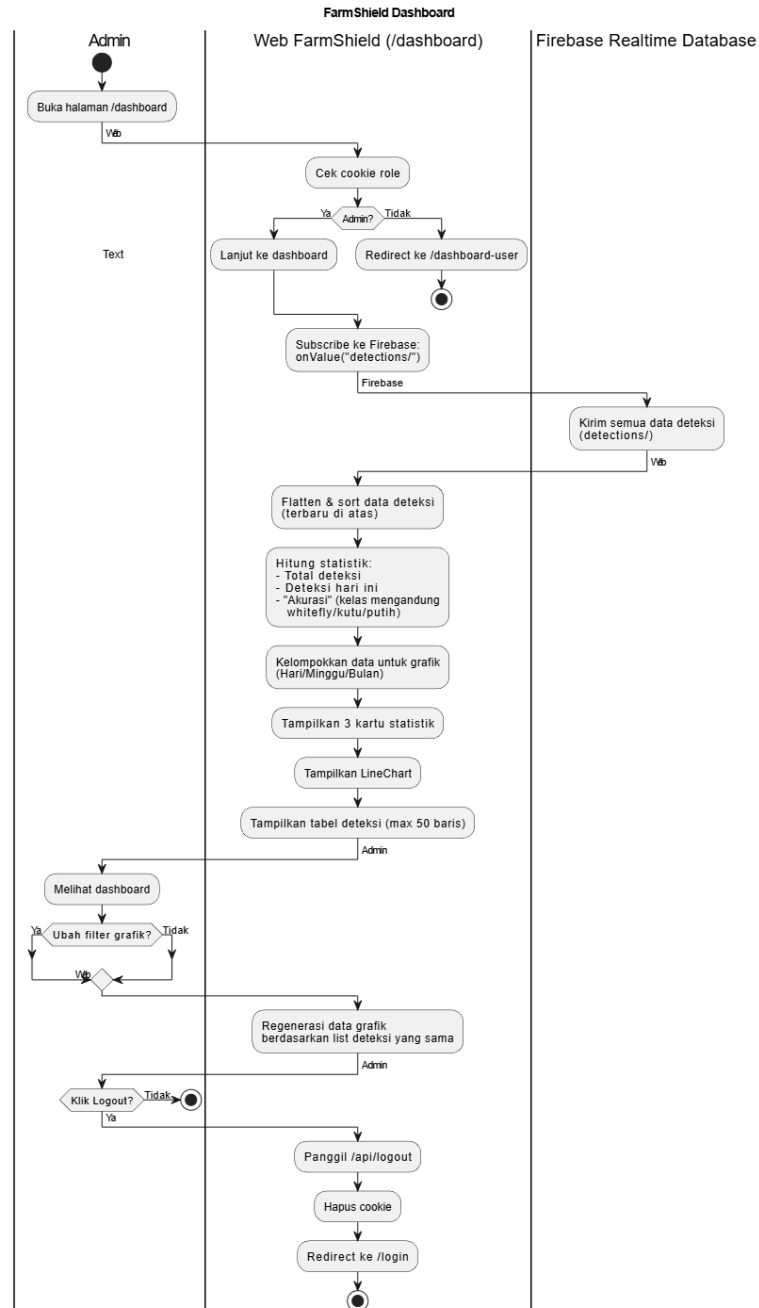
## h. Activity Diagram

- Activity Registrasi & Login

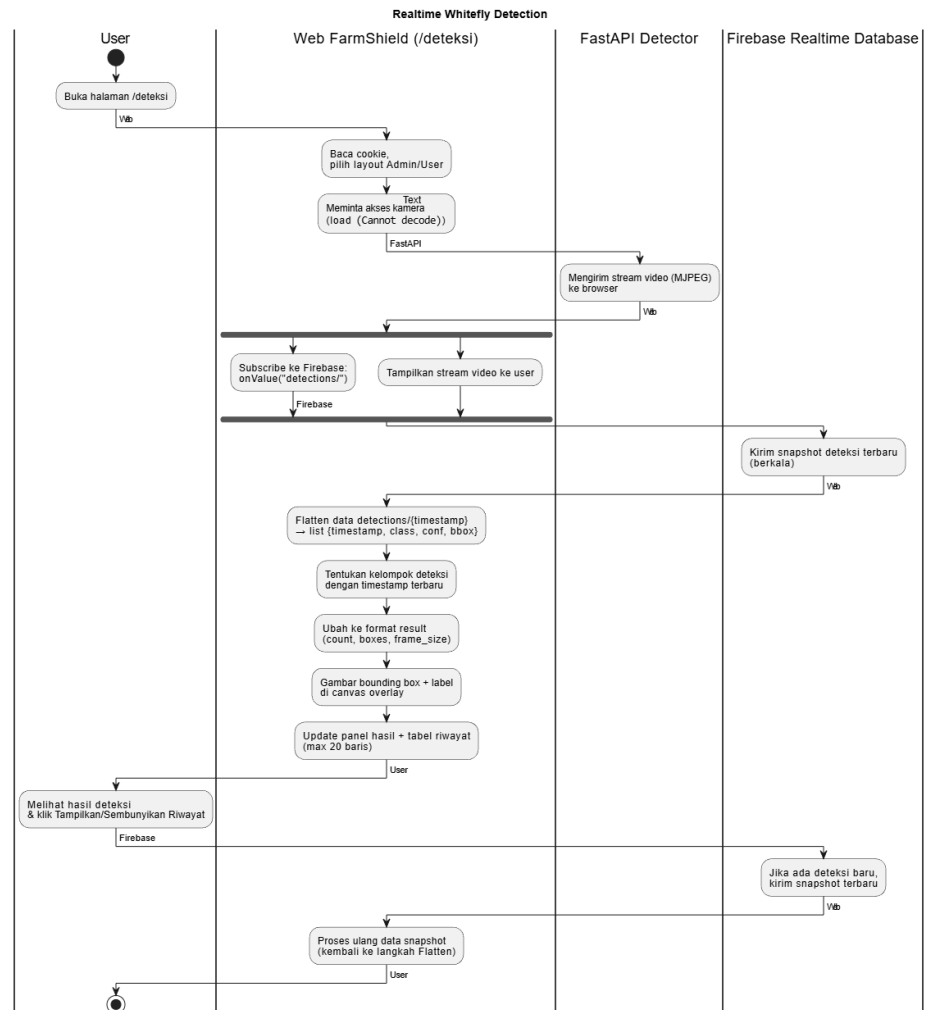


- Dashboard Statistik (Admin)

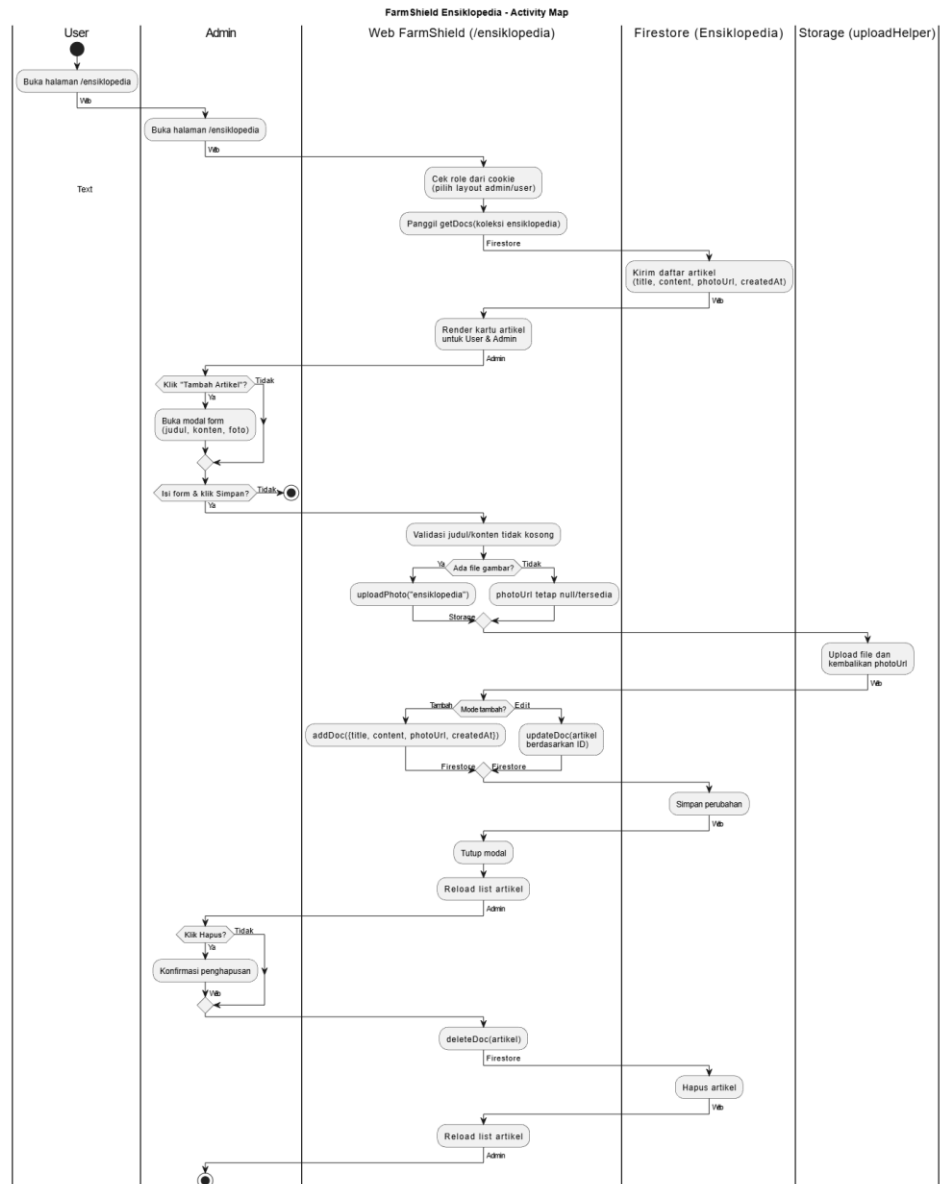




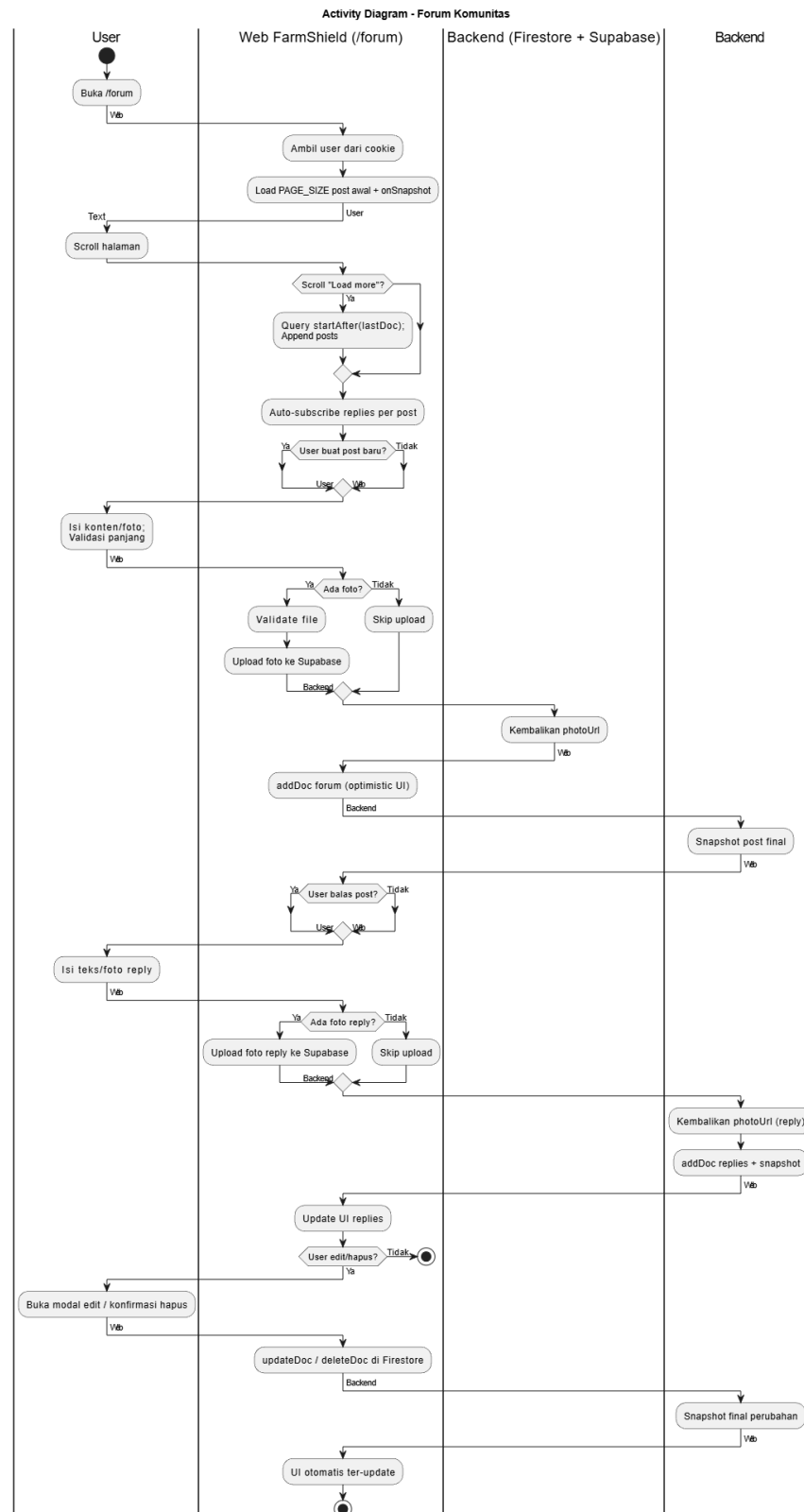
- Deteksi Kutu Putih Real-Time



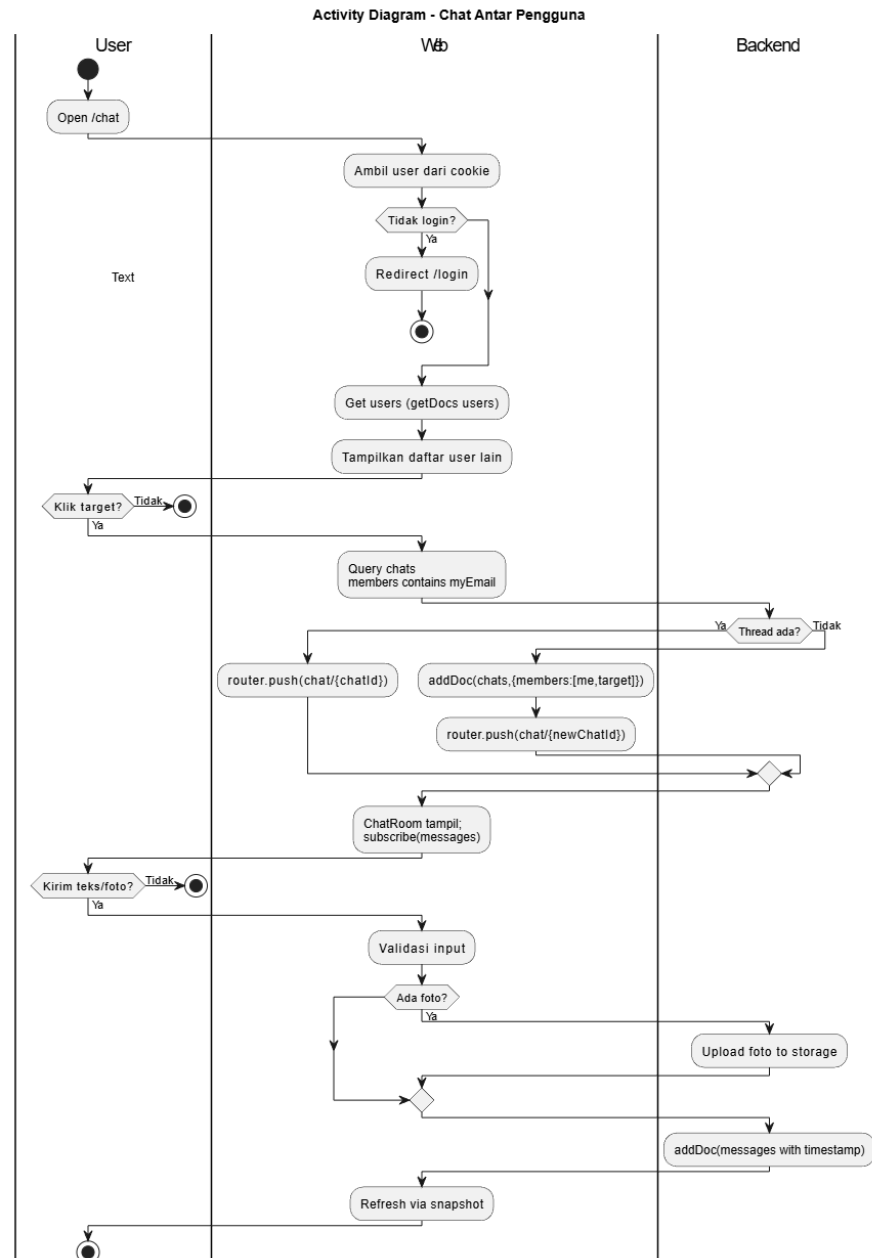
- Ensiklopedia



- Forum Komunitas



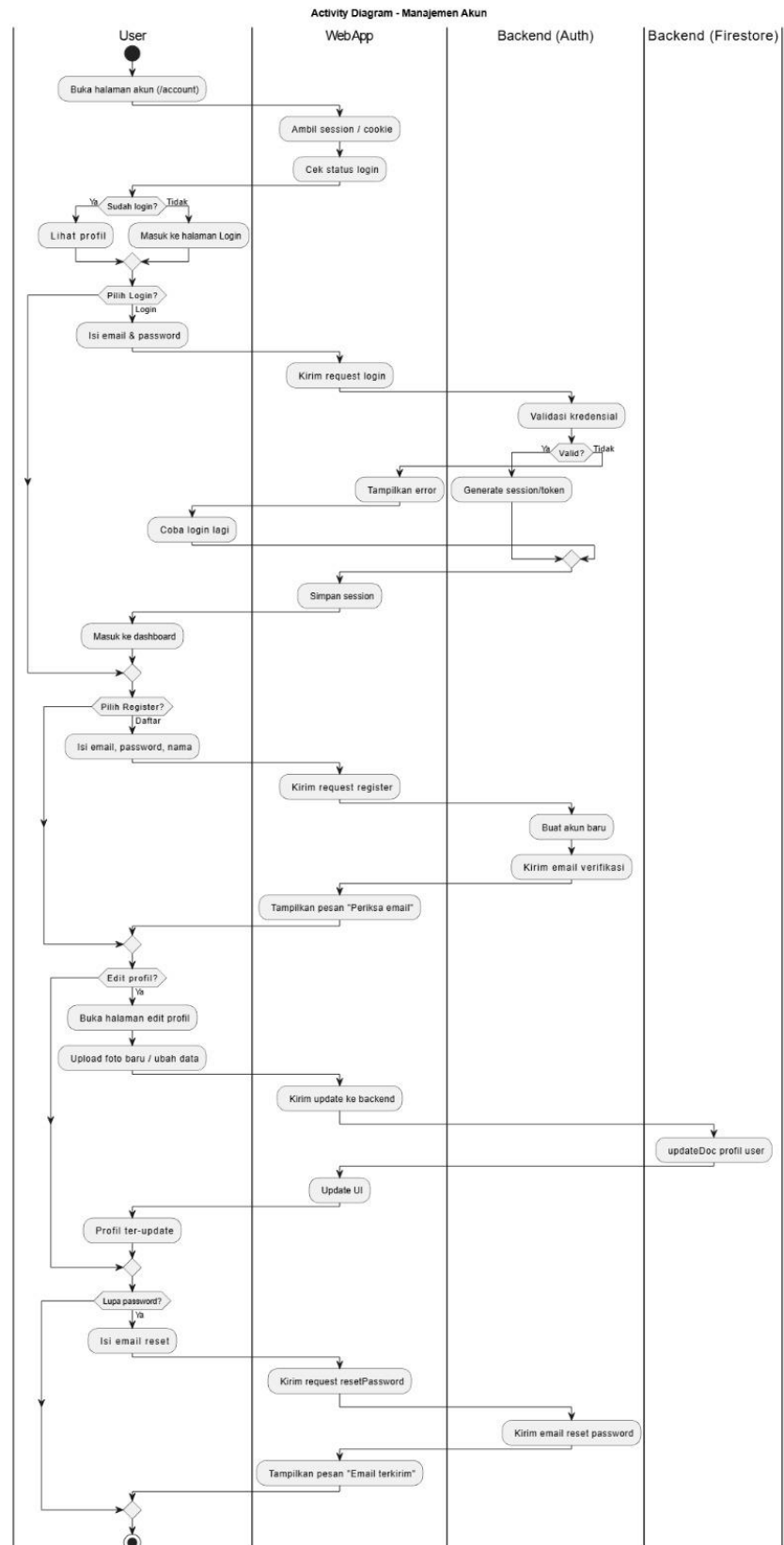
- Chat Antar Pengguna



- Chat AI Assistant

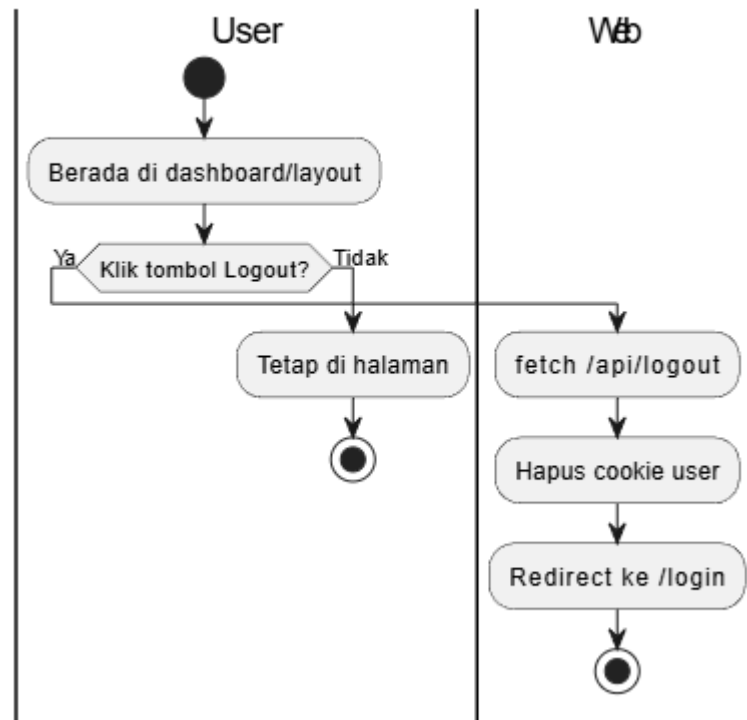


- Akun



- Logout

### Logout Flow (User | Web)





## Spesifikasi Software dan Hardware

### Daftar Software yang digunakan :

No	Nama Software	Digunakan Untuk
1	Python (TensorFlow, OpenCV)	Training dan implementasi AI/ML untuk deteksi hama.
2	Node.js + Express	Backend API untuk komunikasi robot ↔ dashboard.
3	MySQL / PostgreSQL	Database untuk menyimpan DetectionRecord, SprayLog, dan DeviceStatus.
4	React.js / Next.js	Frontend dashboard berbasis web (visualisasi laporan, heatmap, notifikasi).
5	Arduino IDE / PlatformIO	Pemrograman mikrokontroler untuk kontrol pompa DC & sensor.
6	Docker	Containerisasi sistem agar mudah deployment dan maintainability.
7	GitHub / GitLab	Version control dan kolaborasi tim pengembang.
8	Draw.io / PlantUML	Pembuatan diagram (DFD, Use Case, Class, ERD, WBS, Gantt).

### Daftar Hardware yang digunakan :

No	Nama Hardware	Spesifikasi
1	Robot Base (Chassis)	4 roda, motor DC dengan driver motor, rangka besi/aluminium ringan.
2	Mikrokontroler (ESP32 / Arduino Mega)	Kontrol motor, pompa, sensor ultrasonik, komunikasi ke server.
3	Kamera (USB / Pi Camera)	Resolusi minimal 720p, digunakan untuk capture citra hama.
4	Jetson Nano / Raspberry Pi 4	Komputasi AI (menjalankan model ML deteksi hama secara real-time).
5	Pompa DC + Nozzle	Kapasitas semprot 3–5 liter/menit, terhubung ke tangki pestisida.
6	Sensor Ultrasonik	Digunakan untuk deteksi jarak & navigasi robot.
7	Baterai Li-Ion 12V	Kapasitas 10.000 mAh, untuk daya motor + pompa.
8	Laptop Developer	Intel i5/i7, RAM 8GB, SSD 256GB+, digunakan untuk coding & training model AI.
9	Server / Cloud (opsional)	Untuk penyimpanan data, training model skala besar, dan dashboard online.