

***APLIKASI SMART FARMING BERBASIS FEATURE ON-
DEMAND DENGAN TEKNOLOGI DYNAMIC FEATURE
MODULE DI PLATFORM ANDROID***

SKRIPSI

Oleh:

Muhamad Syaroful Anam

NIM. 2141720108



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

2025

HALAMAN PENGESAHAN

PERNYATAAN

Dengan ini saya menyatakan bahwa pada Skripsi ini tidak terdapat karya, baik seluruh maupun sebagian, yang sudah pernah diajukan untuk memperoleh gelar akademik di Perguruan Tinggi manapun, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini serta disebutkan dalam daftar sitasi/pustaka

Malang, 28 Juli 2025

Muhamad Syaroful Anam

ABSTRAK

Syaroful Anam, Muhamad. “Aplikasi Smart Farming Berbasis Feature On-Demand Dengan Teknologi Dynamic Feature Module di Platform Android”.
Pembimbing: (1) Ir. Yan Watequlis Syaifudin, ST., M.MT., Ph.D., (2) Triana Fatmawati, S.T., M.T.

Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2025.

Pertanian digital dan penerapan smart farming di Indonesia menghadapi berbagai tantangan yang cukup kompleks, namun hal ini sangat penting untuk meningkatkan produktivitas pertanian nasional. Penelitian ini bertujuan mengembangkan aplikasi Smart Farming yang optimal dengan fitur-fitur beragam, namun tetap efisien dalam penggunaan sumber daya. Metode yang digunakan adalah pengembangan aplikasi menggunakan arsitektur Dynamic Feature Modules (DFM) berbasis on-demand. Proses pengembangan mengikuti SDLC Rapid Application Development (RAD). Hasil utama penelitian menunjukkan bahwa aplikasi berhasil mengurangi ukuran unduhan awal aplikasi hingga sekitar 31,9%, dengan hasil pengujian black-box yang membuktikan aplikasi berfungsi secara optimal. Hasil User Acceptance Test (UAT) juga menunjukkan aplikasi diterima dengan baik oleh pengguna. Aplikasi ini diharapkan membantu petani dalam mengelola lahan secara lebih efisien melalui fitur relevan yang disediakan, sehingga dapat meningkatkan produktivitas dan mengurangi pemborosan sumber daya.

Kata Kunci : Pertanian Cerdas, Modul Fitur Dinamis, Pertanian Berbasis *Mobile*.

ABSTRACT

Syaroful Anam, Muhamad. *“Smart Farming Application Based on On-Demand Features with Dynamic Feature Module Technology on the Android Platform”*. Supervisor: (1) **Ir. Yan Watequlis Syaifudin, ST., M.MT., Ph.D.**, (2) **Triana Fatmawati, S.T., M.T.**

Thesis, Informatics Management Study Program, Department of Information Technology, State Polytechnic of Malang, 2025.

Digital agriculture and the adoption of smart farming in Indonesia face various challenges that are crucial for enhancing national agricultural productivity. This study aims to develop an optimal Smart Farming application featuring diverse functionalities while maintaining resource-efficient usage. A mixed-method approach was employed, with the application built using an on-demand Dynamic Feature Modules (DFM) architecture. The development process followed the SDLC Rapid Application Development (RAD) model and was supported by interviews, surveys, and literature reviews. Key results indicate that the application reduced the initial download size by approximately 31.9%. Black-box testing confirmed functional correctness, while User Acceptance Testing (UAT) showed that users accepted the application well. This application is expected to assist farmers in managing their fields more efficiently with relevant features, thereby improving productivity and reducing resource wastage.

Keywords: *Smart Farming, Dynamic Feature Module, Mobile Agriculture*

KATA PENGANTAR

Puji Syukur kami panjatkan kehadiran Allah SWT/Tuhan YME atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul “APLIKASI SMART FARMING BERBASIS ON-DOMAND DENGAN TEKNOLOGI DYNAMIC FEATURE MODULE DI PLATFORM ANDROID”. Skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari bahwasannya dengan tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Bapak Rudy Ariyanto, ST., M.Cs., selaku Ketua Jurusan Teknologi Informasi.
2. Ibu Dr. Ely Setyo Astuti, S.T., M.T., selaku Koordinator Program Studi D-IV Teknik Informatika Politeknik Negeri Malang.
3. Bapak Ir. Yan Watequlis Syaifudin, ST., M.MT., Ph.D., selaku Dosen Pembimbing I yang telah memberikan bimbingan, arahan, dan masukan yang sangat berharga selama proses pengerjaan skripsi.
4. Ibu Triana Fatmawati, S.T., M.T., selaku Dosen Pembimbing II yang telah membimbing dan memberikan saran serta dukungan dalam menyelesaikan laporan skripsi ini.
5. Bapak Rakhmad Hardiyanto, ST., selaku *stakeholder* dan pemilik rumah kaca sebagai tempat penelitian yang telah membantu proses pengembangan.
6. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Akhir dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, 28 Juli 2025

Muhamad Syaroful Anam

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN	i
PERNYATAAN	ii
ABSTRAK	iii
<i>ABSTRACT</i>	iv
KATA PENGANTAR.....	v
DAFTAR ISI	vii
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB I . PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	3
1.4 Batasan Masalah.....	3
1.5 Manfaat	3
BAB II . LANDASAN TEORI	5
2.1 Studi Literatur	5
2.2 Dasar Teori.....	9
2.2.1 <i>Smart farming</i>	9
2.2.2 Google Play	9
2.2.3 Dynamic Feature Module.....	10
2.2.4 Modular Architecture	11
2.2.5 Feature On Demand	11
2.2.6 Android	11

2.2.7	Rapid Application Development.....	11
BAB III . METODOLOGI PENGEMBANGAN.....		13
3.1	Metode Pengembangan	13
3.1.1	Perencanaan Kebutuhan	14
3.1.1.1.	Analisis Kebutuhan.....	14
3.1.1.2.	Deskripsi Sistem	18
3.1.2	Desain Sistem.....	20
3.1.2.1.	Desain Antarmuka Aplikasi <i>Mobile</i>	21
3.1.3	Use Case Diagram.....	26
3.1.3.1.	Business Flow Diagram.....	28
3.1.4	Pengembangan	29
3.1.5	Implementasi.....	29
3.2	Skenario Pengujian.....	30
3.2.1	Black Box Testing.....	30
3.2.2	User Accpetance Testing.....	31
BAB IV . ANALISIS DAN PERANCANGAN SISTEM		32
4.1	Analisis Sistem.....	32
4.1.1	Ruang Lingkup.....	32
4.1.2	Analisis Pengguna.....	33
4.1.3	Analisis Kebutuhan Fungsional	33
4.1.4	Analisis Kebutuhan Non Fungsional	34
4.2	Perancangan Sistem	36
4.2.1	Desain Sistem.....	36
4.2.1.1.	Use Case Diagram	36
4.2.1.2.	Activity Diagram	37
4.2.1.3.	<i>Flowchart</i>	42

4.2.2	Alur Sistem.....	46
4.2.3	Arsitektur Program.....	47
BAB V IMPLEMENTASI DAN PENGUJIAN.....		50
5.1	Implementasi Sistem	50
5.2	Implementasi <i>Dynamic Feature Module</i>	50
5.2.1	Integrasi Modul Dinamis.....	50
5.2.2	Integrasi Penginstalan Fitur Secara <i>On-Demand</i>	52
5.3	Implementasi Fitur	53
5.3.1	Fitur Otentikasi.....	53
5.3.1.1.	Data Flow Model Fitur Autentikasi	53
5.3.1.2.	Implementasi Kode Fitur Autentikasi.....	54
5.3.2	Fitur <i>Monitoring</i>	54
5.3.2.1.	Data Flow Model Fitur <i>Monitoring</i>	55
5.3.2.2.	Implementasi Kode Fitur <i>Monitoring</i>	55
5.3.3	Fitur <i>Controlling</i>	56
5.3.3.1.	Data Flow Model Fitur <i>Controlling</i>	56
5.3.3.2.	Implementasi Kode Fitur <i>Controlling</i>	57
5.3.4	Fitur <i>Growth Formula</i>	58
5.3.4.1.	Data Flow Model Fitur <i>Optimal Growth Formula</i>	58
5.3.4.2.	Implementasi Kode Fitur <i>Optimal Growth Formula</i>	58
5.3.5	Fitur <i>Commodity Price Prediction</i>	59
5.3.5.1.	Data Flow Model Fitur <i>Commodity Price Prediction</i>	59
5.3.5.2.	Implementasi Kode Fitur <i>Commodity Price Prediction</i>	60
5.3.6	Fitur <i>Plant Disease Detection</i>	60
5.3.6.1.	Data Flow Model Fitur <i>Plant Disease Detection</i>	61
5.3.6.2.	Implementasi Kode Fitur <i>Plant Disease Detection</i>	61

5.4	Pengujian.....	64
5.4.1	Pengujian Black Box.....	64
5.4.1.1.	Pengujian <i>Dynamic Feature Module</i>	64
5.4.1.2.	Pengujian Fitur	64
5.4.2	User Acceptance Testing.....	67
5.4.2.1.	Instrumen Penelitian	67
BAB VI HASIL DAN PEMBAHASAN.....		71
6.1	Hasil Implementasi Sistem.....	71
6.1.1	Download Fitur Secara On-Demand.....	71
6.1.2	Fitur Otentikasi.....	71
6.1.3	Fitur <i>Monitoring</i>	72
6.1.4	Fitur <i>Control Actuator</i>	73
6.1.5	Fitur <i>Optimal Growth Formula</i>	73
6.1.6	Fitur <i>Commodity Price Prediction</i>	74
6.1.7	Fitur <i>Plant Disease Detection</i>	75
6.2	Hasil Pengujian Sistem	76
6.2.1	Hasil Black Box Testing	76
6.2.2	Hasil UAT	78
6.2.2.1.	Data Responden.....	79
6.2.2.2.	Persepsi Pentingnya Ukuran Awal Aplikasi.....	80
6.2.2.3.	Persepsi Terhadap Aplikasi Smart Farming	81
6.2.2.4.	Penilaian Terhadap Aplikasi Agrilink Vocpro	82
6.3	Pembahasan.....	84
BAB VII KESIMPULAN DAN SARAN		86
7.1	Kesimpulan	86
7.2	Saran.....	86

DAFTAR PUSTAKA.....	88
LAMPIRAN-LAMPIRAN	91

DAFTAR GAMBAR

Gambar 3.1 Model Pengembangan RAD	13
Gambar 3. 2 <i>Smart Farming Workflow Concept</i>	14
Gambar 3.3 Modular Arsitektur Aplikasi <i>Smart farming</i>	17
Gambar 3. 4 Proses <i>On-Demand Feature</i> bekerja.....	18
Gambar 3. 5 Desain UI Fitur Autentikasi	21
Gambar 3. 6 Desain UI Fitur Beranda dan <i>Monitoring</i>	22
Gambar 3. 7 Desain UI Pesan Permintaan Download Fitur	22
Gambar 3. 8 Desain UI Fitur Profil Pengguna	23
Gambar 3. 9 Desain UI Fitur Kontrol Aktuator.....	24
Gambar 3. 10 Desain UI Fitur Resep Pertumbuhan Optimal	24
Gambar 3. 11 Desain UI Fitur Prediksi Harga Koomoditas	25
Gambar 3. 12 Desain UI Fitur Deteksi Penyakit	26
Gambar 3.13 Use Case Diagram Aplikasi <i>Smart farming</i>	27
Gambar 4. 1 Use Case Diagram Aplikasi	36
Gambar 4. 2 Login Activity Diagram.....	37
Gambar 4. 3 Monitor Activity Diagram	38
Gambar 4. 4 Controlling Activity Diagram.....	38
Gambar 4. 5 Commodity Price Activity Diagram	39
Gambar 4. 6 Plant Disease Detection Activity Diagram	40
Gambar 4. 7 Optimal Growth Formula	41
Gambar 4. 8 User Profile Activity Diagram	41
Gambar 4. 9 <i>Flowchart</i> fitur otentikasi	42
Gambar 4. 10 <i>Flowchart</i> fitur <i>monitor</i>	43
Gambar 4. 11 <i>Flowchart</i> fitur <i>control</i>	43
Gambar 4. 12 <i>Flowchart</i> fitur <i>growth formula</i>	44
Gambar 4. 13 <i>Flowchart</i> fitur prediksi harga komoditas	45
Gambar 4. 14 <i>Flowchart</i> fitur deteksi penyakit tanaman	45
Gambar 4. 15 Gambar Alur Aplikasi secara umum	46
Gambar 4. 16 Arsitektur MVVM	47
Gambar 5. 1 Pembuatan modul baru di Android Studio	50
Gambar 5. 2 Model Alur data Fitur Autentikasi	53

Gambar 5. 3 Model Alir Data Fitur <i>Monitoring</i>	55
Gambar 5. 4 Model Alir Data Fitur <i>Controlling</i>	56
Gambar 5. 5 Model Alir Data Fitur Formula Pertumbuhan	58
Gambar 5. 6 Model Alir Data Fitur Prediksi Harga	59
Gambar 5. 7 Model Alir Data Fitur Deteksi Penyakit Tanaman	61
Gambar 6. 1 <i>Screenshot</i> pengunduhan fitur <i>on-demand</i>	71
Gambar 6. 2 <i>Screenshot</i> Fitur Autentikasi.....	72
Gambar 6. 3 <i>Screenshot</i> fitur <i>Monitoring</i>	72
Gambar 6. 4 <i>Screenshot</i> Fitur <i>Control</i>	73
Gambar 6. 5 <i>Screenshot</i> fitur formula pertumbuhan optimal	74
Gambar 6. 6 <i>Screenshot</i> fitur prediksi harga komoditas	75
Gambar 6. 7 <i>Screenshot</i> fitur deteksi penyakit tanman	75
Gambar 6. 8 Data pendidikan responden UAT	79
Gambar 6. 9 Hasil survei pentingnya ukuran aplikasi.....	80
Gambar 6. 10 Hasil survei seringnya pembatalan pengunduhan.....	80
Gambar 6. 11 Hasil survei pertimbangan penyimpanan perangkat.....	81
Gambar 6. 12 Hasil survei familiaritas terhadap aplikasi <i>smart farming</i>	81
Gambar 6. 13 Hasil survei harapan fitur aplikasi <i>smart farming</i>	82
Gambar 6. 14 Hasil survei tantangan dalam pertanian saat ini	82
Gambar 6. 15 Hasil survei kemudahan pengunduhan fitur <i>on-demand</i>	83
Gambar 6. 16 Hasil survei penilaian UI	84

DAFTAR TABEL

Tabel 2. 1 Studi Literatur Penelitian Terdahulu	5
Tabel 3.1 Daftar Fitur dan <i>instal time</i> Aplikasi <i>Smart farming</i>	15
Tabel 3.2 Deskripsi Sistem	18
Tabel 3. 3 Daftar Library Utama Yang Digunakan	19
Tabel 3. 4 Rencana Tabel Pengujian <i>Black Box</i>	30
Tabel 3. 5 Rencana Tabel Pengujian Instalasi <i>Dynamic Feature Module</i> ...	30
Tabel 3. 6 Rencana Tabel Pengujian UAT	31
Tabel 4. 1 Tabel Analisis Kebutuhan Fungsional.....	33
Tabel 4. 2 Tabel Daftar Library	34
Tabel 4. 3 <i>Foldering</i> Arsitektur MVVM	48
Tabel 5. 1 Tabel pengujian <i>dynamic feature module</i>	64
Tabel 5. 2 Pengujian Fitur	65
Tabel 5. 3 Tabel Pertanyaan UAT	68
Tabel 6. 1 Tabel ukuran AAB (Android App Bundle)	76
Tabel 6. 2 Ukuran ril aplikasi setelah terinstal pada perangkat.....	77
Tabel 6. 3 Ilustrasi efieinsi ukuran berdasarkan kebutuhan pengguna.....	77
Tabel 6. 4 Interpretasi nilai mean.....	78
Tabel 6. 5 Data usia responden UAT.....	79
Tabel 6. 6 Distribusi pendidikan responden UAT	79
Tabel 6. 7 Hasil survei kesesuaian fitur.....	83

DAFTAR LAMPIRAN

Lampiran 1. Business Flow Login.....	91
Lampiran 2. Business Flow Pemantauan Kondisi Green House	92
Lampiran 3. Business Flow Kontrol Sistem Irigasi Green House.....	93
Lampiran 4. Business Flow Prediksi Harga Komoditas.....	94
Lampiran 5. Business Flow Deteksi Penyakit Tanaman	95
Lampiran 6. Business Flow formula Pertumbuhan Optimal	96

BAB I. PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi *smart farming* semakin pesat seiring dengan kebutuhan untuk meningkatkan efisiensi dan produktivitas di sektor pertanian. Di Indonesia, sektor pertanian menghadapi berbagai tantangan yang signifikan. Salah satunya adalah masih dominannya penggunaan teknologi konvensional dalam praktik pertanian (Rachmawati, 2021). Oleh karena itu, diperlukan inovasi dan penerapan teknologi modern untuk mengatasi tantangan ini dan meningkatkan daya saing produk pertanian Indonesia di pasar global.

Kebutuhan dalam pertanian cerdas di era modern semakin kompleks dan beragam. Pertanian digital memiliki beragam kebutuhan seperti mendukung kepatuhan regulasi, optimasi peralatan, permainan simulator pertanian, manajemen informasi, informasi referensi agronomi, pelacakan produk, identifikasi hama, akuntansi emisi, atau tolok ukur untuk klaim pemasaran (Inwood & Dale, 2019). Dalam upaya menjawab tantangan tersebut, penulis berencana mengembangkan aplikasi *smart farming* berbasis seluler. Aplikasi seluler dalam pertanian telah mengubah cara petani beroperasi dengan berbagi informasi waktu nyata, tren pasar, dan alat pengambilan keputusan yang lebih baik (.Venkadesh, 2025).

Aplikasi Agrilink Vocpro memiliki potensi besar terhadap dukungan digitalisasi pertanian di Indonesia dengan fitur-fitur baru yang terintegrasi dengan *machine learning*. Tapi, dengan menambahkan fitur-fitur baru, aplikasi ini berpotensi meningkatkan ukuran aplikasi. Oleh karena itu, pendekatan Dynamic Feature Module dengan On-demand delivery diterapkan agar pengguna dapat mengunduh fitur yang relevan saja, sehingga ukuran aplikasi tetap optimal dan efisien.

Pengembangan aplikasi dalam penelitian ini akan menerapkan sistem modularisasi, khususnya pendekatan Feature On-Demand (fitur yang diunduh sesuai permintaan). Hal ini menjadi jawaban atas beberapa masalah seperti ukuran aplikasi dengan *scope* yang luas biasanya besar. Aplikasi dengan ukuran unduhan lebih kecil cenderung lebih banyak diunduh dan diinstal (Uniyal dkk., 2023). Ukuran aplikasi yang besar membatasi akses di perangkat low-end dan

jaringan lambat (Uniyal dkk., 2023). Aplikasi besar mengonsumsi lebih banyak memori, CPU, dan baterai, memperburuk performa (Liu dkk., 2023).

Pendekatan sistem modular ini sangat penting karena modularisasi membantu mengatasi masalah pemeliharaan dan kompleksitas dalam aplikasi berskala besar, serta meningkatkan kualitas dan skalabilitas (Shubha Ramachandran dkk., 2009). Modularisasi meningkatkan pemahaman terhadap kualitas perangkat lunak dengan memecah kode sumber menjadi pengelompokan yang koheren dan independen, menghasilkan kohesi tinggi dan keterkaitan rendah (Yuniarti dkk., 2024). *Smart farming*, dengan memanfaatkan teknologi Internet of Things (IoT), bertujuan untuk mengatasi tantangan dalam produksi pangan dengan meningkatkan akurasi diagnosis tanaman dan mencegah masalah tanaman. Teknologi ini menawarkan solusi inovatif untuk tantangan pertanian modern dengan meningkatkan produktivitas, efisiensi, dan keberlanjutan.

Penelitian ini memiliki signifikansi yang besar, terutama dalam konteks pembangunan ekonomi dan sosial di negara berkembang seperti Indonesia. Dengan memanfaatkan teknologi canggih, *smart farming* dapat mengurangi dampak lingkungan, menurunkan biaya, dan meningkatkan hasil pertanian. Selain itu, *smart farming* berpotensi menciptakan lapangan kerja baru dan meningkatkan ketahanan pangan, yang pada akhirnya mendukung pembangunan ekonomi dan sosial yang berkelanjutan. Oleh karena itu, penelitian ini akan berfokus pada pengembangan lebih lanjut aplikasi Agrilink Vocpro dengan menerapkan konsep modular dan Feature on Demand, guna meningkatkan efisiensi dan relevansi aplikasi bagi para pengguna di sektor pertanian.

1.2 Rumusan Masalah

Adapun rumusan masalah yang telah disusun oleh peneliti adalah sebagai berikut:

1. Bagaimana penerapan teknologi *Dynamic Feature Module (DFM)* dalam pengembangan aplikasi *smart farming* berbasis *Feature On-Demand* dapat menghasilkan ukuran aplikasi yang lebih kecil sehingga dapat meningkatkan efisiensi penggunaan penyimpanan pada perangkat?

1.3 Tujuan

Tujuan dari dilakukannya skripsi dengan judul “APLIKASI *SMART FARMING* BERBASIS FEATURE ON DEMAND DENGAN TEKNOLOGI DYNAMIC FEATURE MODULE DI PLATFORM ANDROID” adalah sebagai berikut:

1. Merancang arsitektur aplikasi *smart farming* berbasis Dynamic Feature Module (DFM) untuk menghadirkan fitur secara *on-demand* guna mengevaluasi dampaknya terhadap efisiensi ukuran aplikasi dan penggunaan ruang penyimpanan perangkat.

1.4 Batasan Masalah

Adapun batas permasalahan dalam penelitian ini adalah:

1. Aplikasi dikembangkan untuk platform android dengan Versi SDK minimal 29 (Android 10).
2. Aplikasi *smart farming* ini dikembangkan menggunakan bahasa kotlin dengan Jetpack Compose sebagai framework UI.
3. Aplikasi ini dirancang khusus untuk mengontrol sistem *smart farming* di area *greenhouse* Bumiaji. Data dan kontrol yang telah diintegrasikan pada area tersebut.
4. Penelitian berfokus pada pengembangan aplikasi *mobile* dengan integrasi pada perangkat IoT yang telah tersedia pada *greenhouse* di Bumiaji.

1.5 Manfaat

Manfaat dari penelitian ini antara lain

1. Aplikasi ini diharapkan akan membantu petani mengelola lahan pertanian mereka secara lebih efisien melalui fitur-fitur yang relevan, sehingga meningkatkan produktivitas dan mengurangi pemborosan sumber daya.
2. Penggunaan arsitektur DFM akan menghasilkan aplikasi yang lebih ringan dan efisien dalam penggunaan sumber daya perangkat, sehingga dapat diakses dan digunakan oleh petani dengan berbagai jenis perangkat Android, termasuk yang memiliki spesifikasi rendah.
3. Hasil penelitian ini dapat dijadikan sebagai referensi bagi peneliti lain yang tertarik untuk mengembangkan aplikasi *smart farming* dengan pendekatan

modular dan fitur *on-demand*, serta sebagai dasar untuk pengembangan fitur-fitur baru pada aplikasi *smart farming* di masa mendatang.

BAB II. LANDASAN TEORI

2.1 Studi Literatur

Tabel 2. 1 Studi Literatur Penelitian Terdahulu

Judul & Penulis	Hasil	Kesimpulan
“Rancang Bangun Sistem <i>Smart farming</i> Berbasis IoT Studi Kasus Kebun Nyoman Gumitir” (Putu dkk., 2022).	Output dari pengembangan sistem <i>smart farming</i> berbasis IoT ini adalah aplikasi android yang mencakup beberapa aspek penting: • Sistem dapat mengukur kelembaban tanah dengan tingkat keberhasilan rata-rata sebesar 93%. • Hasil pengukuran pH tanah menunjukkan rata-rata selisih perhitungan sensor pH dengan pH meter sebesar 0.18, yang menunjukkan akurasi yang baik dalam mendeteksi kadar asam tanah. • Sensor water level berhasil mendeteksi kapasitas air pada tempat penampungan air, yang penting	Penelitian ini berhasil merancang sistem <i>smart farming</i> berbasis IoT di Kebun Nyoman Gumitir. Proses dimulai dengan analisis kebutuhan dan pembuatan skematik menggunakan aplikasi Proteus, diikuti pemasangan rangkaian dengan komponen input, modul kontroler ESP32, dan output. Komponen utama meliputi sensor soil moisture, sensor pH, dan sensor water level. Aplikasi Android digunakan untuk monitoring dan kontrol penyiraman. Hasil menunjukkan tingkat keberhasilan pengukuran kelembaban tanah mencapai 93%, selisih rata-rata sensor pH dengan pH meter sebesar 0.18, dan sensor water

	untuk pengisian otomatis. • Sistem dapat menghidupkan pompa air dan memberikan pupuk secara otomatis berdasarkan data yang diterima dari sensor, sehingga mempermudah pemilik kebun dalam pengelolaan tanaman.	level berhasil mendeteksi kapasitas air.
“<i>Smart farming</i> Teknologi Monitoring Produksi dan Pemasaran Kebun Organik” (Dewi dkk., 2022).	Menghasilkan aplikasi <i>mobile</i> dan web yang memiliki 3 peran akun untuk dikelola, yaitu <i>customer</i>, mitra tani, dan mitra kebun. Aplikasi ini memiliki beberapa fitur: • Akun Konsumen merupakan bentuk pemasaran digital dalam menawarkan produk sayuran digital dan melakukan transaksi jual beli. • Mitra kebun salah satu fiturnya adalah supply inventory. Fitur ini akan memberitahukan	Rancangan aplikasi <i>smart farming</i> untuk monitoring produksi dan pemasaran kebun organik akan mempermudah transaksi penjualan dan pengelolaan inventaris antara mitra tani dan Gapoktan Jaya Alam Lestari. Konsumen dapat menggunakan aplikasi ini untuk membeli produk sayuran organik secara digital. Bagi petani, aplikasi ini menyediakan kemudahan dalam pendataan informasi

	status dari semua pengiriman barang, mulai dari status pengiriman, status on supply, dan status Receive. • Pada halaman home mitra tani terdapat beberapa fitur yang sama seperti halaman mitra kebun. Yaitu profil, riwayat transaksi, menu account mitra, inventory, kelompok tani, history, info dan laporan.	anggota kelompok tani, pembibitan, pemupukan, lokasi tanam, serta tanggal tanam dan panen. Diharapkan, aplikasi ini dapat meningkatkan efektivitas produksi pertanian organik dan mempermudah pemasaran digital, sehingga meningkatkan profit usaha pertanian organik.
"Rapid Application Development (RAD) model method for creating an agricultural irrigation system based on internet of things" (Nalendra, 2021).	Hasil dari penelitian tersebut menunjukkan bahwa metode Rapid Application Development (RAD) dapat digunakan untuk membangun sistem berbasis Internet of Things (IoT) yang efektif, khususnya dalam pengendalian irigasi tanaman. Penelitian ini berhasil mengembangkan dan menguji perangkat IoT yang dapat mengontrol pompa irigasi dengan waktu respons rata-rata sekitar 4,8	Kesimpulan dari penelitian ini adalah bahwa metode RAD tidak hanya mempercepat proses pengembangan sistem, tetapi juga meningkatkan kualitas hasil yang dicapai. Pengujian perangkat IoT menunjukkan bahwa sistem irigasi yang dikembangkan dapat beroperasi dengan baik dan memenuhi

	detik, yang menunjukkan bahwa sistem dapat berfungsi sesuai dengan kebutuhan pengguna.	kebutuhan pengguna dalam pengelolaan irigasi tanaman.
“DeltaDroid: Dynamic Delivery Testing in Android”(Ghorbani dkk., 2023).	pengembangan alat yang disebut DeltaDroid. DeltaDroid dirancang untuk membantu pengembang dengan memperluas suite pengujian yang ada, sehingga mereka dapat lebih efektif mendeteksi kesalahan dan perilaku yang tidak diinginkan yang mungkin tidak terungkap oleh alat pengujian otomatis yang ada.	Secara keseluruhan, penggunaan DFMs dalam pengembangan aplikasi Android memberikan keuntungan signifikan dalam hal efisiensi, pengalaman pengguna, dan fleksibilitas, tetapi juga memerlukan pendekatan pengujian yang lebih cermat untuk memastikan bahwa aplikasi berfungsi dengan baik dalam berbagai kondisi.

Berdasarkan studi literatur yang telah dilakukan, berbagai penelitian dan pengembangan di bidang *smart farming* umumnya berfokus pada fitur monitoring, pengendalian, dan sistem manajemen. Hal ini menunjukkan bahwa cakupan *smart farming* terus berkembang, dan seiring dengan perkembangan tersebut, fitur-fitur baru akan terus ditambahkan. Akibatnya, aplikasi *smart farming* berpotensi memiliki ukuran yang besar dan kompleks. Oleh karena itu, peneliti mengusulkan penggunaan arsitektur modular dalam pengembangan aplikasi *smart farming* untuk mengatasi tantangan ini, sehingga setiap komponen dapat dikembangkan dan dikelola secara terpisah, serta memudahkan pemeliharaan dan penambahan fitur di masa depan.

2.2 Dasar Teori

Dasar teori membahas beberapa teori maupun teknologi yang berkaitan dengan pengembangan dan penelitian ini. Berfokus pada penjelasan tentang teknologi smart farming dalam aplikasi mobile yang memiliki struktur *modular architecture*.

2.2.1 *Smart farming*

Smart farming atau dalam Bahasa indonesianya Pertanian Pintar merupakan konsep pertanian yang mengarah pada pengelolaan pertanian menggunakan teknologi berbasis IoT (*Internet of Things*) untuk meningkatkan kuantitas dan kualitas produk dan mengoptimalkan tenaga manusia yang dibutuhkan oleh produksi (Ulfada dkk., 2022). *Smart farming* juga sebuah pendekatan modern dalam bidang pertanian yang mengintegrasikan teknologi informasi dan komunikasi (TIK) untuk meningkatkan efisiensi, produktivitas, dan keberlanjutan dalam aktivitas pertanian. Dasar dari *smart farming* berpusat pada pemanfaatan teknologi seperti Internet of Things (IoT), kecerdasan buatan (AI), sensor, big data, dan cloud computing untuk mengumpulkan, menganalisis, dan memanfaatkan data dari berbagai aspek pertanian. Dengan sensor yang terpasang di ladang, petani dapat memantau kondisi tanah, cuaca, kelembapan, suhu, dan kebutuhan nutrisi tanaman secara real-time. Teknologi ini memungkinkan pengambilan keputusan yang lebih akurat, seperti kapan waktu terbaik untuk irigasi atau pemupukan, sehingga mengurangi penggunaan sumber daya yang berlebihan dan meminimalkan dampak lingkungan. Selain itu, *smart farming* juga mencakup penggunaan drone untuk pemantauan tanaman, robot untuk panen, serta aplikasi *mobile* yang memberikan petani akses langsung ke data pertanian mereka, memungkinkan manajemen pertanian yang lebih terkoordinasi dan berbasis data. Tujuan utama dari *smart farming* adalah untuk mendukung pertanian yang lebih produktif, efisien, dan berkelanjutan, sekaligus mengatasi tantangan global seperti perubahan iklim dan kebutuhan pangan yang terus meningkat.

2.2.2 Google Play

Google Play, yang sebelumnya dikenal sebagai Android Market, adalah toko online resmi untuk media digital yang didistribusikan oleh Google. Selain aplikasi Android, Google Play juga menawarkan permainan, film, acara televisi, buku

elektronik, dan buku audio (Sheldon R, 2023). Model layanan aplikasi Google Play memanfaatkan Android App Bundle (aab) untuk menghasilkan dan mengirimkan APK yang telah dioptimalkan untuk setiap konfigurasi perangkat pengguna. Dengan demikian, pengguna hanya mengunduh kode dan sumber daya yang dibutuhkan untuk menjalankan aplikasi, sehingga meminimalkan ukuran unduhan.

2.2.3 Dynamic Feature Module

Dynamic Feature Module adalah modul yang digunakan untuk memisahkan fitur dan sumber daya dari APK dasar (Darmawan LS, 2023). *Dynamic feature module* memungkinkan pengembang untuk menambahkan modul tertentu ke aplikasi utama secara dinamis. Dynamic Feature Modules (DFMs) adalah komponen dalam model pengiriman dinamis Android yang memungkinkan pengembang untuk menyertakan fitur opsional dalam aplikasi mereka (Ghorbani dkk., 2023). DFMs dapat diunduh dan diinstal setelah aplikasi utama diinstal, sesuai dengan permintaan pengguna. Ini memberikan fleksibilitas dalam pengembangan aplikasi, memungkinkan pengembang untuk mengoptimalkan ukuran aplikasi dan hanya menyertakan fitur yang relevan bagi pengguna tertentu. Ada beberapa pendekatan, *install time delivery*, *conditional delivery*, *on-demand delivery* dan *instant delivery*. Manfaat DFMs dalam pengembangan aplikasi Android meliputi:

1. DFMs memungkinkan pengembang untuk membagi aplikasi menjadi modul-modul (*Modularization*) yang lebih kecil dan lebih terfokus, sehingga memudahkan pengelolaan dan pengembangan fitur baru tanpa mempengaruhi keseluruhan aplikasi.
2. Pengguna dapat menghemat ruang penyimpanan di perangkat mereka, karena tidak semua fitur harus diinstal secara bersamaan.
3. Aplikasi dapat dimuat lebih cepat karena hanya modul yang diperlukan yang diunduh dan diinstal, mengurangi waktu yang diperlukan untuk memulai aplikasi.
4. DFMs memungkinkan pengembang untuk menawarkan pengalaman yang lebih disesuaikan kepada pengguna, dengan fitur yang hanya diunduh jika pengguna membutuhkannya atau memilih untuk menggunakannya.

5. Pengembang dapat memperbarui DFMs secara terpisah dari aplikasi utama sehingga fleksibel, memungkinkan perbaikan bug dan penambahan fitur baru tanpa memerlukan pembaruan keseluruhan aplikasi.

2.2.4 Modular Architecture

Modularisasi adalah proses membuat bagian-bagian dari sebuah aplikasi yang dapat bekerja secara terpisah. Ini berarti setiap bagian bisa berfungsi secara independen tanpa terlalu bergantung pada bagian lainnya (Zneika dkk., 2013). Dengan cara ini, pengembangan perangkat lunak bisa lebih mudah disesuaikan dan dimodifikasi secara lebih fleksibel tanpa mengurangi kinerja aplikasi.

2.2.5 Feature On Demand

Feature on demand adalah konsep yang memungkinkan pengguna untuk mengakses fitur tertentu dari aplikasi hanya ketika mereka membutuhkannya. Dengan pendekatan ini, aplikasi tidak perlu memuat semua fitur sekaligus, yang dapat mengurangi waktu muat dan penggunaan sumber daya. Dalam aplikasi *smart farming*, fitur on demand dapat digunakan untuk memberikan akses kepada pengguna ke alat atau informasi spesifik, seperti analisis cuaca atau rekomendasi pemupukan, berdasarkan kondisi saat ini atau kebutuhan pengguna. Ini meningkatkan efisiensi dan relevansi aplikasi bagi pengguna.

2.2.6 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat *mobile*, seperti smartphone dan tablet (Easttom, 2021). Android menyediakan platform yang fleksibel dan *open source* bagi pengembang untuk menciptakan aplikasi yang dapat diunduh dan digunakan oleh pengguna di seluruh dunia. Dengan dukungan untuk berbagai perangkat keras dan kemampuan untuk mengintegrasikan teknologi terbaru, Android menjadi salah satu platform paling populer untuk pengembangan aplikasi *mobile*.

2.2.7 Rapid Application Development

Rapid Application Development (RAD) adalah model pengembangan perangkat lunak yang bersifat inkremental dan sesuai untuk kebutuhan yang telah dipahami dengan baik. Model ini mampu menghasilkan perangkat lunak dalam jangka waktu yang singkat, sehingga sangat cocok diterapkan pada proyek-proyek yang memerlukan skala yang besar (B. Prashanth Kumar & Y. Prashanth, 2014).

RAD menggunakan prototyping dan umpan balik pengguna untuk mempercepat siklus pengembangan, memungkinkan pengembang untuk merespons perubahan kebutuhan dengan cepat. Dalam pengembangan ini, RAD memungkinkan peneliti untuk dengan cepat menguji ide-ide baru, mendapatkan umpan balik dari pengguna, dan melakukan iterasi pada aplikasi, sehingga menghasilkan produk akhir yang lebih sesuai dengan kebutuhan pengguna.

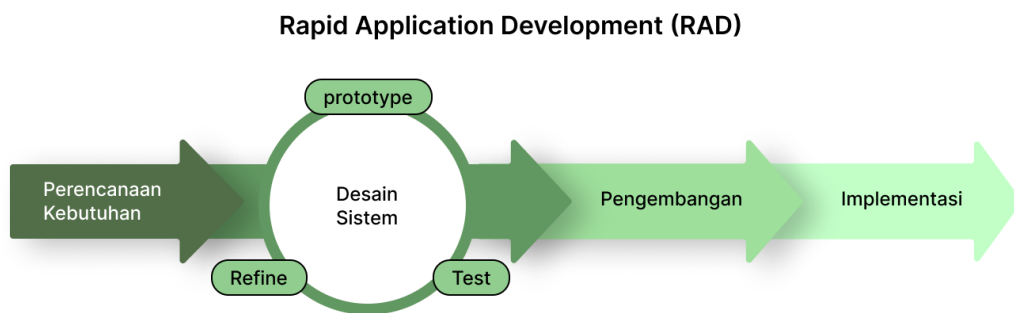
BAB III. METODOLOGI PENGEMBANGAN

Metodologi pengembangan akan membahas detail tentang metode yang akan digunakan untuk mengembangkan aplikasi. Selain itu juga akan terdapat detail dari analisis kebutuhan sistem serta rancangan aplikasi yang akan dibuat. Rancangan ini mulai dari arsitektur, desain aplikasi, hingga desain alur sistem.

3.1 Metode Pengembangan

Dengan mempertimbangkan uraian kebutuhan serta kompleksitas sistem yang sedang dikembangkan, penulis telah memutuskan untuk memilih metode Rapid Application Development (RAD) sebagai pendekatan pengembangan. Metode RAD ini memungkinkan pembuatan prototipe secara cepat, penyesuaian ulang, serta penggunaan kembali perangkat lunak yang telah ada, sambil tetap memungkinkan optimalisasi khusus perangkat keras dan siklus pengembangan yang lebih singkat.

Penggunaan metode RAD sangat sesuai untuk diterapkan dalam siklus pengembangan aplikasi yang cepat, serta mampu memberikan kualitas perangkat lunak yang lebih baik dibandingkan dengan pendekatan rekayasa perangkat lunak tradisional seperti *waterfall*. Melalui proses pengembangan aplikasi yang cepat, metode ini dapat mengurangi biaya pengembangan dan pemeliharaan perangkat lunak yang akan dibuat. Berikut ini adalah siklus pengembangan yang diterapkan dalam metode RAD.



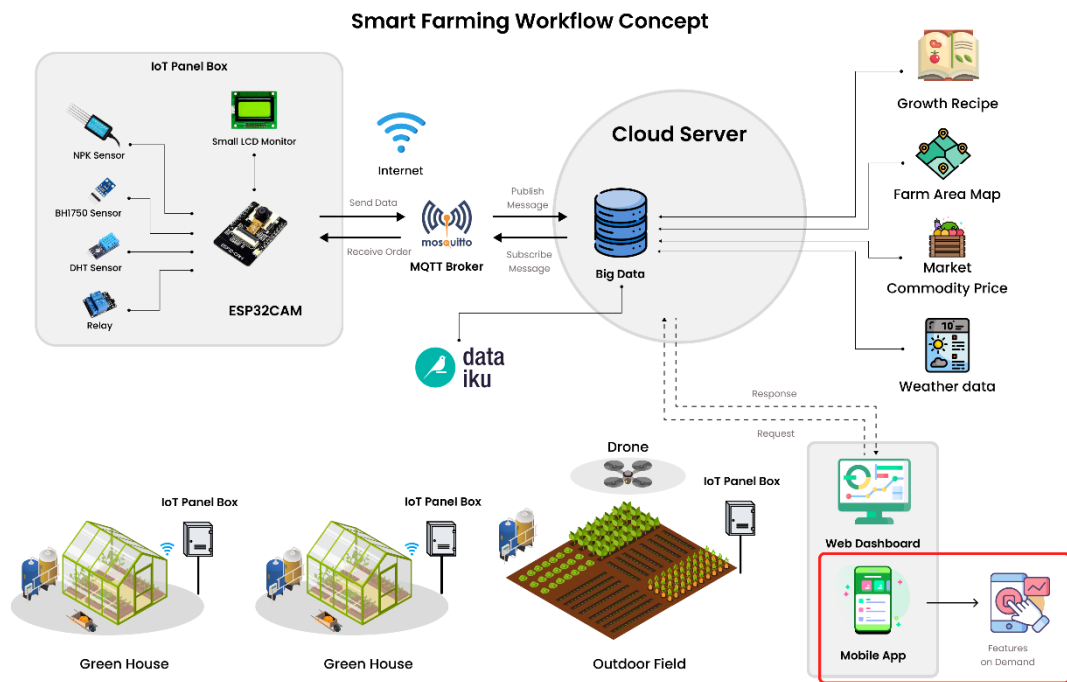
Gambar 3.1 Model Pengembangan RAD

Gambar 3.1 merupakan gambar model atau alur dari pengembangan RAD. Terdapat tahapan yang dilakukan untuk mencapai terwujudnya sebuah produk. Mulai dari perancangan kebutuhan, desain sistem, pengembangan hingga implementasi.

3.1.1 Perencanaan Kebutuhan

Pada tahap perencanaan kebutuhan, penulis akan melakukan analisis untuk mengidentifikasi kebutuhan pengguna dan fitur-fitur yang diperlukan dalam aplikasi *smart farming*. Kegiatan yang dilakukan meliputi pengumpulan data melalui studi literatur dan wawancara kepada calon pengguna, seperti petani dan pengelola agroindustri, untuk memahami kebutuhan dan harapan mereka terhadap aplikasi. Selanjutnya, penulis akan melakukan analisis kebutuhan dengan mengkategorikan informasi yang telah dikumpulkan menjadi fitur utama dan sekunder, serta menentukan prioritas pengembangan fitur berdasarkan urgensi dan dampaknya terhadap pengguna.

3.1.1.1. Analisis Kebutuhan



Gambar 3. 2 *Smart Farming Workflow Concept*

Dalam pengembangan aplikasi *smart farming* Agrilink Vocpro sebelumnya, fitur-fitur yang tersedia masih terbatas, sehingga ukuran aplikasi tetap kecil. Berdasarkan gambar 3.2 sistem *smart farming* ini akan dikembangkan dalam skala yang lebih besar dan ada penambahan fitur-fitur baru. Namun, dengan rencana penambahan fitur-fitur baru, ukuran aplikasi akan bertambah. Oleh karena itu, penulis mengusulkan konsep aplikasi yang modular, di mana fitur-fitur dipisahkan ke dalam modul-modul terpisah. Konsep ini memanfaatkan teknologi Dynamic

Feature Modules (DFM) dari Google Play, yang memungkinkan pengguna untuk mengunduh dan menggunakan hanya fitur-fitur yang dibutuhkan. Dengan cara ini, ruang penyimpanan perangkat dapat lebih dihemat, karena hanya fitur yang relevan bagi pengguna yang akan diinstal.

Tabel 3.1 Daftar Fitur dan *instal time* Aplikasi *Smart farming*

Nama Fitur	<i>Install Time</i>	Deskripsi
Autentikasi (<i>login, register, dan logout</i>)	<i>included</i>	Fitur autentikasi berfokus pada keamanan data pribadi pengguna. Semua fitur dapat dijalankan jika pengguna sudah <i>login</i> dan memiliki <i>access token</i> .
Pemantauan kondisi <i>green house</i>	<i>included</i>	Fitur ini merupakan fitur untuk <i>monitoring</i> kondisi <i>green house</i> . Ada pun paramater yang dipantau adalah nutrisi tanah dan juga kondisi udara seperti suhu dan kelembaban.
Kontrol <i>actuator</i> di <i>green house</i>	<i>on-demand</i>	Fitur ini memungkinkan pengguna untuk mengontrol sistem irigasi di <i>green house</i> .
Resep pertumbuhan optimal	<i>on-demand</i>	Fitur ini merupakan fitur untuk melihat dan memberikan rekomendasi nutrisi standar.

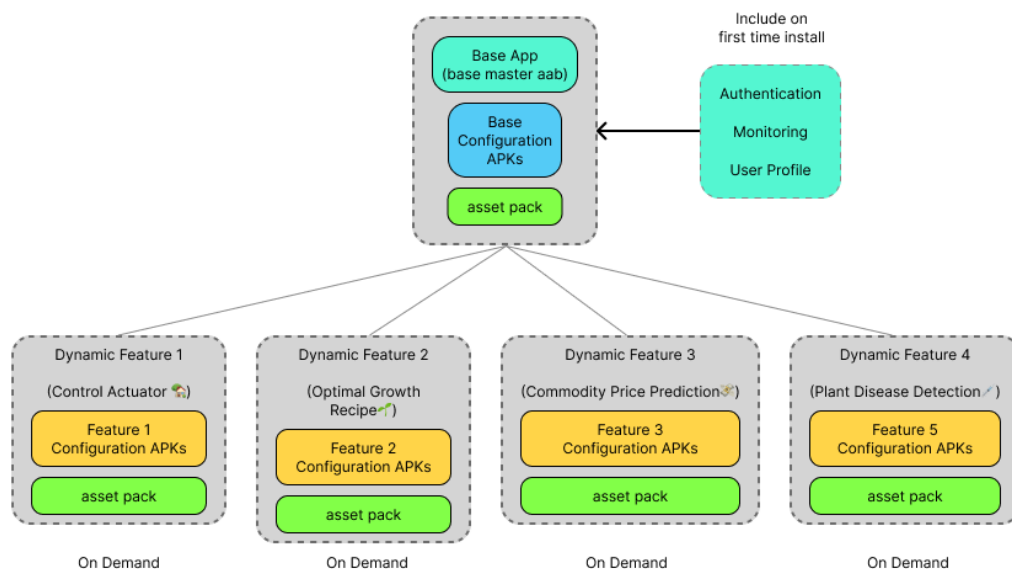
Nama Fitur	<i>Install Time</i>	Deskripsi
Prediksi harga Komoditas	<i>on-demand</i>	Fitur ini merupakan fitur untuk memprediksi harga komoditas.
Deteksi penyakit tanaman	<i>on-demand</i>	Fitur untuk mendeteksi penyakit tanaman melalui citra. Pengguna bisa mengambil gambar tanaman untuk mengetahui penyakit suatu tanaman.
Profil pengguna	<i>included</i>	Fitur untuk melihat data pengguna.

Berdasarkan Tabel 3.1, beberapa fitur utama yang akan diimplementasikan mencakup pemantauan kondisi greenhouse, pengelolaan kontrol irigasi, prediksi harga komoditas, formula pertumbuhan komoditas, serta deteksi penyakit tanaman melalui citra. Fitur pemantauan tanaman memungkinkan pengguna untuk memantau kondisi tanaman secara real-time dengan memanfaatkan sensor yang terhubung, sehingga petani dapat mengambil tindakan yang tepat berdasarkan data yang diperoleh. Selain itu, aplikasi ini juga akan dilengkapi dengan fitur pengelolaan irigasi yang memungkinkan pengguna untuk mengatur dan memantau sistem irigasi secara otomatis sesuai dengan kebutuhan tanaman.

Untuk mendukung keputusan ekonomi, aplikasi ini juga akan menyertakan fitur prediksi harga komoditas di Jawa Timur, yang menggunakan model *machine learning* yang telah dikembangkan. Selain itu, pengguna dapat menemukan formula pertumbuhan optimal tanaman sesuai dengan komoditas yang mereka tanam, sehingga dapat membantu pengguna dalam memutuskan perihal tindakan pemeliharaan.

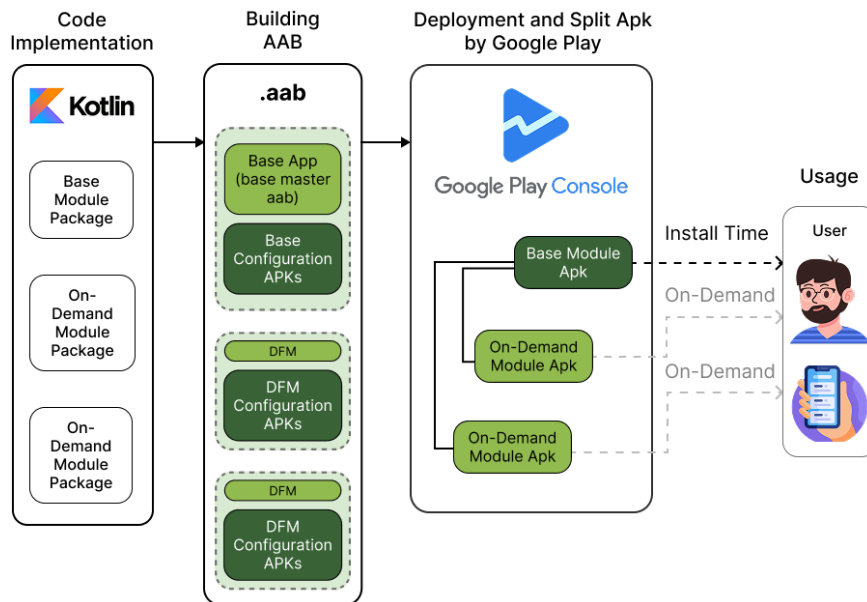
Fitur-fitur seperti *Commodity Price Prediction* dan *Plant Disease Detection* akan dikembangkan oleh mahasiswa lain sebagai bagian dari skripsi mereka. Penulis akan bertanggung jawab untuk mengintegrasikan fitur-fitur ini ke dalam

aplikasi Agrilink Vocpro setelah pengembangan selesai. Namun, terdapat potensi masalah jika model Machine Learning untuk fitur-fitur tersebut belum selesai pada saat pengembangan aplikasi. Jika model tidak tersedia, fitur-fitur seperti *Commodity Price Prediction* dan *Plant Disease Detection* tidak akan dapat berfungsi secara optimal. Untuk mengatasi hal ini, penulis akan melakukan pengujian dan evaluasi terhadap fitur-fitur yang telah dikembangkan, serta menyediakan alternatif solusi sementara, seperti menggunakan data statis, simulasi, ataupun integrasi dengan AI yang sudah ada seperti Gemini API hingga model ML siap untuk diintegrasikan.



Gambar 3.3 Modular Arsitektur Aplikasi *Smart farming*

Dynamic Feature Module (DFM) berperan penting dalam memisahkan fitur-fitur tersebut agar pengguna dapat menginstal fitur secara on-demand. Seperti pada gambar 3.3 beberapa fitur akan diinstal pada saat penginstalan pertama, seperti autentikasi, pemantauan greenhouse, dan profil pengguna, sementara fitur lainnya akan diinstal sesuai dengan kebutuhan pengguna. Penggunaan sensor dan perangkat IoT akan dibatasi pada model tertentu yang kompatibel dengan aplikasi, untuk memastikan integrasi yang optimal. Pada tahap awal, aplikasi ini akan dikembangkan khusus untuk platform Android, tanpa dukungan untuk iOS, sehingga fokus pengembangan dapat lebih terarah dan efisien.



Gambar 3. 4 Proses *On-Demand Feature* bekerja

Gambar 3.4 menggambarkan alur instalasi fitur secara *on-demand* menggunakan teknologi Dynamic Feature Modules (DFMs). Dalam tahap pengembangan, modul-modul aplikasi dipisahkan berdasarkan fitur masing-masing dan dikembangkan secara independen. Setelah aplikasi selesai dikembangkan, berkas dengan format .aab diunggah ke Google Play Console. Play Console kemudian melakukan proses pemisahan modul atau *split* APKs.

Ketika pengguna pertama kali menginstal aplikasi, hanya modul utama atau *base master app* yang terpasang. Sementara itu, fitur lain yang terdapat dalam modul terpisah dapat diunduh dan diinstal sesuai dengan kebutuhan atau permintaan pengguna.

3.1.1.2. Deskripsi Sistem

Berdasarkan permasalahan dan latar belakang yang telah penulis sampaikan sebelumnya mengenai pengembangan aplikasi *mobile smart farming* diharapkan dapat mencapai sebuah sistem yang lebih efisien dan bermanfaat bagi pengguna akhir.

Tabel 3.2 Deskripsi Sistem

Spesifikasi	Detail
Platform	Aplikasi <i>Mobile</i>
Nama Aplikasi	Agrilink Vocpro

Spesifikasi	Detail
Versi Sistem Operasi minimal	Android 10, API Level 29
Bahasa Pemrograman	Kotlin
Framework UI	Jetpack Compose
Database	PostgreSQL (<i>cloud</i>), SQLite (<i>local</i>)
API	Integrasi dengan API Green House Bumiaji
Kebutuhan Perangkat Keras	<i>Smartphone</i> atau tablet dengan OS android
Fitur Utama	Autentikasi, Profil Pengguna, Pemantauan kondisi tanaman, Pengelolaan irigasi, Prediksi harga komoditas, Resep pertumbuhan optimal, Deteksi penyakit tanaman.

Sesuai dengan tabel 3.2 diatas mengenai deskripsi aplikasi secara keseluruhan, maka ada beberapa kebutuhan aplikasi yang akan dikategorikan kedalam kebutuhan perangkat lunak dan kebutuhan perangkat keras untuk proses pengembangan aplikasi.

Dalam implementasi awal untuk menerapkan Dynamic Feature Module serta pengembangan awal dibutuhkan beberapa library yang terdapat pada tabel 3.3, diantaranya:

Tabel 3. 3 Daftar Library Utama Yang Digunakan

Nama Library	Sumber	Fungsi
Play Feature Delivey	com.google.android.play:feature-delivery	Mengelola fitur on-demand untuk Dynamic Feature Module menggunakan Play Store.
Play in App Update	com.google.android.play:app-update-ktx	Mengelola pembaruan aplikasi langsung dari Play Store.
OkHttp dan Retrofit	io.ktor:ktor-client-android	Mengambil data dari API eksternal.

Nama Library	Sumber	Fungsi
Koin	io.insert-koin:koin-android	Dependency Injection yang ringan dan sederhana.

1. Kebutuhan Perangkat Lunak

- Sistem operasi minimal Windows 11.
- Microsoft Word, Goole Spreadsheet, dan Figma sebagai *tools* untuk penulisan proposal dan laporan selama proses pengembangan.
- Android SDK, kumpulan dari beberapa *tools* atau *kit* yang digunakan untuk membantu mengembangkan aplikasi android.
- Android Studio, sebagai environment dan tempat penulisan kode untuk mengembangkan aplikasi android.
- Postman, sebagai alat untuk membantu melakukan pengujian API.
- Gitlab, sebagai layanan untuk menyimpan kode program dan juga *version control*.
- Figma, sebagai alat untuk mendesain UI dari sistem.

2. Kebutuhan Perangkat Keras

- Laptop/PC dengan spesifikasi prosesor 11th Gen Intel(R) Core(TM) i7-11370H @ 3.30GHz, GPU NVIDIA GeForce MX450, RAM 16GB, dan *storage* 1TB.
- Smartphone* android dengan spesifikasi Android 11, prosesor Mediatek MT6765 Helio P35 (12 nm), RAM 4GB, GPU PowerVR GE8320, dan *storage* 64GB.

3.1.2 Desain Sistem

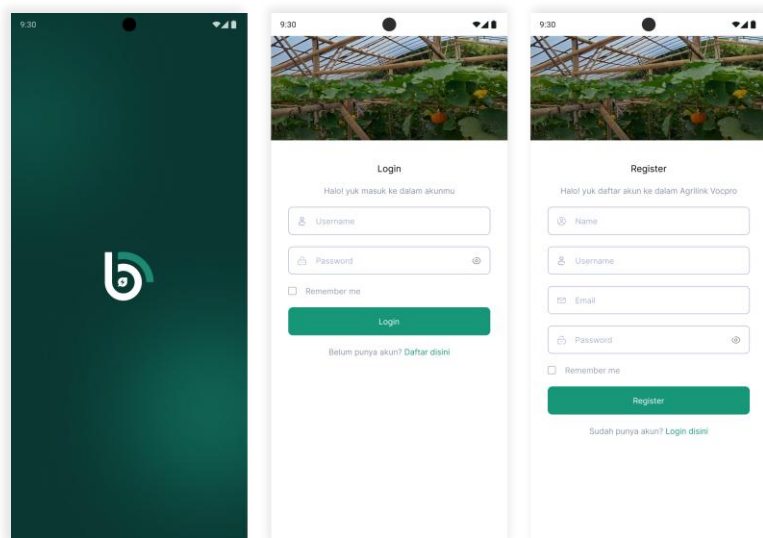
Pada tahap ini, setelah kebutuhan sistem telah ditentukan pada tahap sebelumnya, dilakukan perancangan desain sistem yang mencakup berbagai aspek penting dalam pengembangan perangkat lunak. Proses perancangan ini dimulai dengan penyusunan diagram use case untuk menggambarkan interaksi antara aktor dan sistem. Selain itu, disusun pula business flow diagram untuk memberikan

gambaran menyeluruh mengenai aliran bisnis dan proses utama yang terjadi dalam sistem.

Selain aspek perancangan sistem, tahap ini juga mencakup perancangan antarmuka pengguna (UI design) yang berfungsi sebagai acuan dalam pengembangan sistem yang sesungguhnya. Desain UI ini dibuat dengan mempertimbangkan aspek kenyamanan, kemudahan penggunaan, serta kesesuaian dengan kebutuhan pengguna. Hasil dari tahap perancangan ini akan menjadi dasar dalam implementasi sistem pada tahap pengembangan selanjutnya

3.1.2.1. Desain Antarmuka Aplikasi *Mobile*

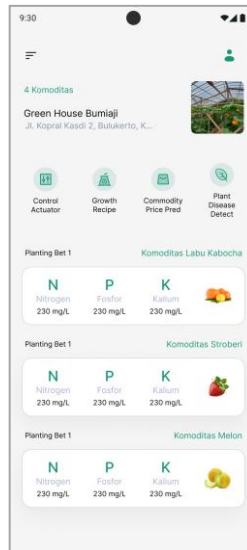
a. Autentikasi



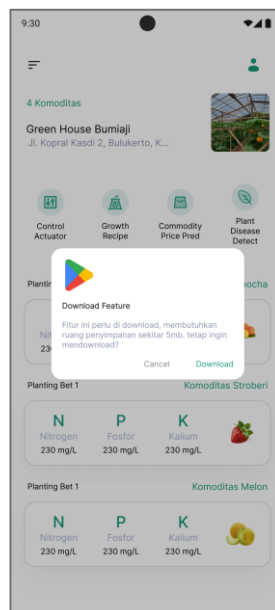
Gambar 3. 5 Desain UI Fitur Autentikasi

Pada gambar 3.5 ditunjukkan gambar *splash screen*, *login screen*, dan *register screen*. halaman ini adalah pintu masuk utama dari aplikasi ketika pengguna baru selesai menginstal aplikasi. Terdapat *splash screen* yang akan menampilkan logo aplikasi. Kemudian akan diarahkan ke halaman *login* yang berisi input teks *username* dan *password*. Jika pengguna belum memiliki akun disediakan juga halaman *register* yang telah disediakan input teks berupa nama, *username*, *email*, *password*.

b. Beranda dan Monitor



Gambar 3. 6 Desain UI Fitur Beranda dan *Monitoring*

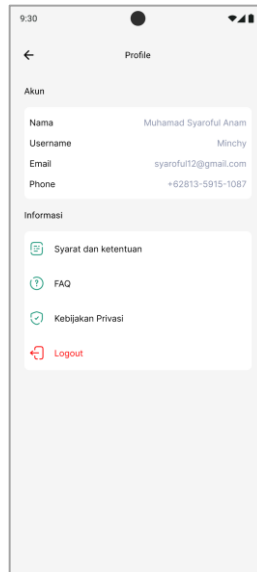


Gambar 3. 7 Desain UI Pesan Permintaan Download Fitur

Gambar 3.6 merupakan halaman beranda. Halaman beranda sebagai halaman utama untuk mengakses semua fitur. Terdapat beberapa ikon untuk mengakses fitur-fitur *on-demand*, seperti ikon *control actuator*, *growth Formula*, *commodity price prediction*, dan *plant disesase detection*. Halaman ini juga memiliki fitur untuk monitoring data kondisi *green house* secara *realtime* berupa *listview* sesuai dengan urutan planting bet. Pada *listview* ini pengguna dapat melihat kadar nitrogen, fosfor, dan kalium pada tanah. Gambar 3.7 menunjukkan pesan permintaan mengunduh fitur yang

ingin digunakan. Ini akan muncul ketika pengguna baru mengakses fitur on-demand untuk pertama kali dengan menekan ikon fitur yang disediakan sebelumnya.

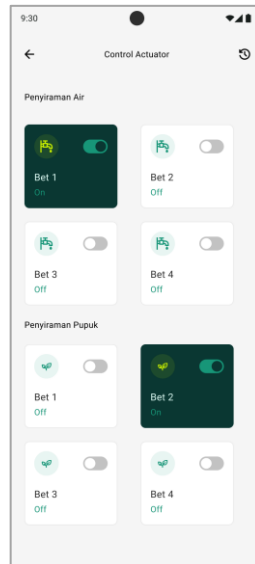
c. Profil Pengguna



Gambar 3. 8 Desain UI Fitur Profil Pengguna

Gambar 3.8 merupakan UI dari halaman profil. Halaman profil adalah halaman untuk mengetahui informasi akun pengguna seperti nama, *username*, *email* dan nomor telepon. Ada beberapa fitur lain seperti syarat dan ketentuan, FAQ, kebijakan privasi, dan tombol *logout* untuk mengakhiri *session*.

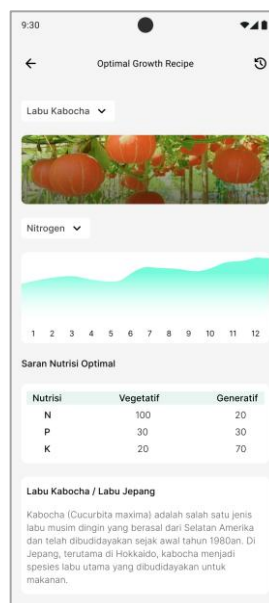
d. Fitur Kontrol Aktuator



Gambar 3. 9 Desain UI Fitur Kontrol Aktuator

Gambar 3.9 adalah UI dari fitur kontrol aktuator. Halaman kontrol aktuator merupakan salah satu fitur on-demand atau fitur yang diunduh sesuai kebutuhan. Halaman ini memungkinkan pengguna mengontrol mekanisme pemupukan dan penyiraman pada setiap *planting bet* di *green house* dengan cara menekan tombol kontrol sesuai dengan aktuator yang disediakan. Warna hijau menunjukkan aktuator dalam keadaan aktif sedangkan putih menunjukkan bahwa aktuator dalam keadaan nonaktif.

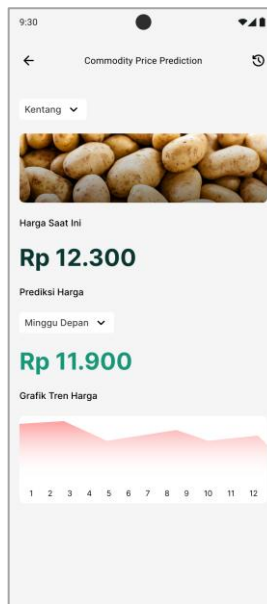
e. Fitur Formula Pertumbuhan Optimal



Gambar 3. 10 Desain UI Fitur Resep Pertumbuhan Optimal

Halaman formula pertumbuhan optimal memungkinkan pengguna memantau grafik sesuai dengan standar optimal nilai NPK pada tanah. Dilihat dari gambar 3.10 halamn ini terdapat informasi garfik harian, nilai optimal NPK, dan informasi tentang komoditas. Pengguna dapat melihat informasi tersebut sesuai dengan komoditas yang dipilih melalui *dropdown* yang terdapat pada bagian atas.

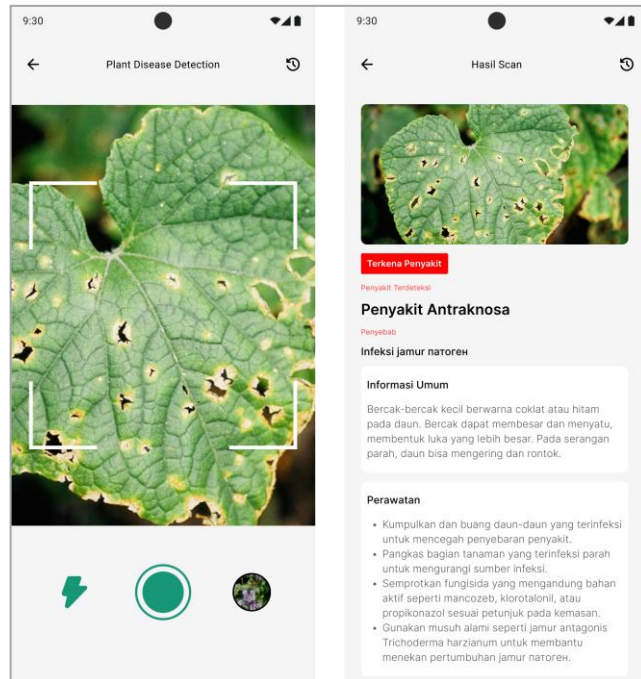
f. Fitur Prediksi Harga Komoditas



Gambar 3. 11 Desain UI Fitur Prediksi Harga Koomoditas

Gambar 3.11 menunjukan UI dari fitur prediksi harga. Halaman prediksi harga komoditas memiliki fitur untuk melihat harga komoditas pada hari ini serta harga komoditas pada minggu atau bulan depan sesuai dengan komoditas yang dipilih melalui *dropdown* yang terdapat di atas gambar.

g. Fitur Deteksi Penyakit Tanaman

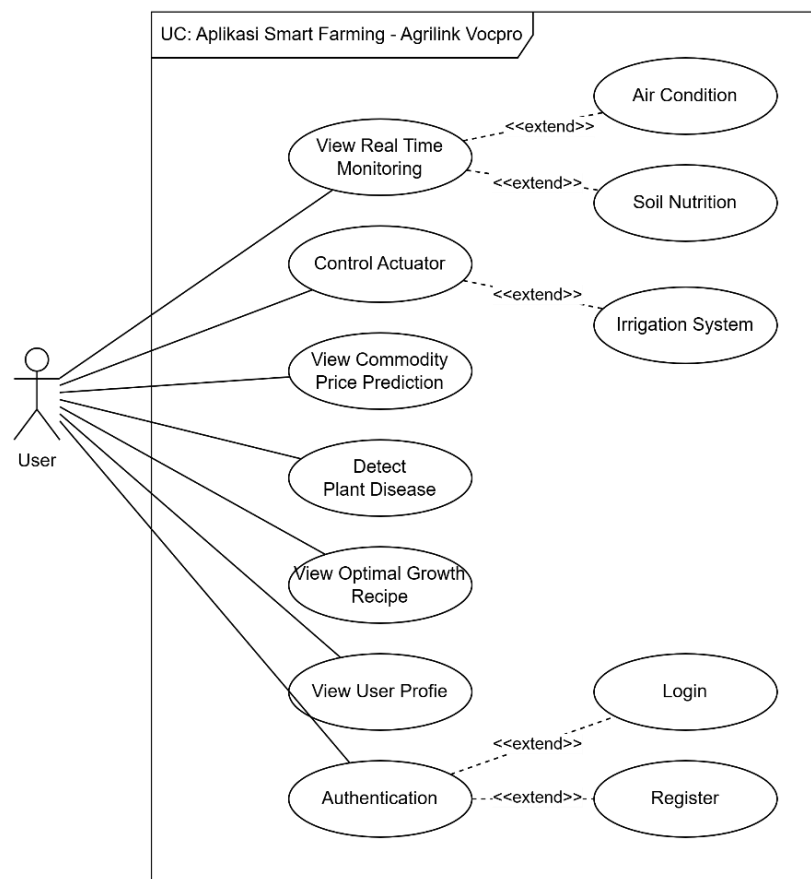


Gambar 3. 12 Desain UI Fitur Deteksi Penyakit

Gambar 3.12 merupakan UI dari fitur deteksi penyakit tanaman. Ketika mengakses fitur deteksi penyakit tanaman pertama-tama akan membuka halaman kamera. Setelah pengguna menangkap gambar dari daun, hasilnya akan ditampilkan pada halaman hasil *scan*. Pada halaman ini terdapat informasi apakah tanaman terjangkit penyakit atau tidak, serta jenis penyakit apa dan bagaimana cara mengatasinya.

3.1.3 Use Case Diagram

Diagram *Use Case* adalah diagram yang menggambarkan bagaimana pengguna yang berbeda menggunakan fungsionalitas suatu sistem untuk mencapai tujuan mereka, membantu pengembang perangkat lunak memahami perilaku sistem dari perspektif pemangku kepentingan (Kautz dkk., 2022). Use case diagram ini menggambarkan interaksi antara pengguna dan sistem dalam aplikasi *mobile smart farming* Agrilink Vocpro.



Gambar 3.13 Use Case Diagram Aplikasi *Smart farming*

Berdasarkan gambar 3.13 Diagram ini menampilkan berbagai fitur utama yang tersedia dalam aplikasi, serta hubungan antara fitur-fitur tersebut. Salah satu fitur utama adalah *Real Time Monitoring*, yang memungkinkan pengguna untuk memantau kondisi lingkungan pertanian secara langsung, termasuk parameter seperti suhu, kelembapan, dan kualitas tanah.

Selain itu, terdapat fitur *Control Actuator* yang memungkinkan pengguna untuk mengontrol perangkat aktuator yang terhubung. Fitur ini mengatur *Irrigation System*, yang memungkinkan pengguna untuk mengatur dan memantau sistem irigasi secara efisien. Pengguna juga dapat mengakses fitur *View Commodity Price Prediction*, yang memberikan informasi tentang prediksi harga komoditas pertanian, membantu mereka dalam pengambilan keputusan terkait penjualan hasil pertanian.

Fitur *Early Detect Plant Disease* memungkinkan pengguna untuk mendeteksi penyakit tanaman lebih awal, sehingga tindakan pencegahan dapat diambil untuk

mengurangi kerugian. Fitur *Optimal Growth Formula*, memungkinkan pengguna bisa melihat standar kondisi tanaman dalam masa pertumbuhan termasuk nutrisi yang dibutuhkan. Hal ini dapat menjadi pembantu pengambilan keputusan untuk pemupukan dan perawatan tanaman. Selain itu, Pengguna juga dapat melihat dan mengelola profil mereka melalui fitur *View User Profile*, yang mencakup informasi pribadi dan pengaturan preferensi. Terakhir, fitur *Login* merupakan langkah awal bagi pengguna untuk mengakses aplikasi, memastikan bahwa hanya pengguna terdaftar yang dapat menggunakan fitur-fitur yang tersedia. Dengan desain sistem yang terstruktur ini, aplikasi Agrilink Vocpro diharapkan dapat memenuhi kebutuhan pengguna dalam praktik *smart farming* secara efektif dan efisien.

3.1.3.1. Business Flow Diagram

Business Flow Diagram ini menggambarkan alur proses bisnis dalam aplikasi *mobile smart farming* "Agrilink Vocpro", yang menunjukkan interaksi antara pengguna, sistem, dan berbagai komponen yang terlibat dalam pengoperasian aplikasi. Diagram ini dimulai dengan pengguna yang melakukan Login ke dalam aplikasi (Lampiran 5.1), yang merupakan langkah awal untuk mengakses semua fitur yang tersedia. Setelah berhasil masuk, pengguna akan diarahkan ke halaman utama, di mana mereka dapat memilih berbagai fitur sesuai kebutuhan mereka.

Salah satu alur utama dalam diagram ini adalah pemantauan kondisi *green house* (Lampiran 5.2), di mana pengguna dapat memantau kondisi lingkungan pertanian secara langsung. Pengguna dapat mengakses data terkait suhu, kelembapan, dan kualitas tanah, yang diperoleh dari sensor yang terhubung. Informasi ini membantu pengguna dalam mengambil keputusan yang tepat untuk meningkatkan hasil pertanian. Selain itu, pengguna juga dapat menggunakan fitur kendali *actuator* untuk mengontrol perangkat seperti sistem irigasi (Lampiran 5.3), yang memungkinkan mereka untuk mengoptimalkan kondisi pertanian secara real-time.

Selanjutnya, pengguna dapat mengakses fitur *View Commodity Price Prediction* untuk mendapatkan informasi tentang prediksi harga komoditas pertanian (Lampiran 5.4). Informasi ini sangat penting bagi pengguna dalam merencanakan waktu penjualan hasil pertanian mereka. Selain itu, fitur deteksi penyakit melalui citra memungkinkan pengguna untuk mendeteksi potensi penyakit

pada tanaman, sehingga mereka dapat mengambil tindakan pencegahan yang diperlukan untuk mengurangi kerugian (Lampiran 5.5).

Diagram ini juga mencakup alur untuk formula pertumbuhan optimal, di mana pengguna dapat melihat analisis standar pertumbuhan yang relevan untuk komoditas yang ditanam, membantu mereka merencanakan aktivitas pertanian dengan lebih baik. Dengan alur bisnis yang terstruktur ini, aplikasi "Agrilink Vocpro" dirancang untuk dapat mendukung praktik *smart farming* secara efisien.

3.1.4 Pengembangan

Pada tahap pengembangan, penulis akan mulai mengimplementasikan desain yang telah dibuat menjadi aplikasi yang fungsional. Kegiatan yang dilakukan mulai dari menulis kode program untuk setiap fitur yang telah direncanakan, menggunakan bahasa pemrograman Kotlin dan framework UI *jetpack compose* untuk platform Android. Kemudian proses selanjutnya adalah mengintegrasikan berbagai fitur dalam aplikasi, termasuk proses pengunduhan modul, serta memastikan semua komponen berfungsi dengan baik. Terakhir melakukan pengujian untuk setiap fitur yang dikembangkan, guna memastikan bahwa setiap bagian aplikasi berfungsi sesuai dengan yang diharapkan. Pada tahap ini juga akan dilakukan pengujian *Black Box Testing* dan *UAT* secara bertahap untuk mengidentifikasi bug serta fungsionalitas dari aplikasi.

3.1.5 Implementasi

Setelah aplikasi selesai dikembangkan dan diuji, penulis akan melanjutkan ke tahap implementasi. Kegiatan yang dilakukan meliputi:

- a. Mengunggah aplikasi ke platform distribusi, seperti Google Play Store, dan memastikan bahwa aplikasi dapat diakses oleh pengguna.
- b. Menyediakan panduan penggunaan aplikasi kepada pengguna, agar mereka dapat memanfaatkan fitur-fitur yang ada dengan maksimal.
- c. Mengumpulkan umpan balik dari pengguna setelah penggunaan aplikasi, untuk melakukan perbaikan dan pengembangan lebih lanjut di masa mendatang.

3.2 Skenario Pengujian

Skenario pengujian pada pengembangan ini adalah dengan menggunakan dua pendekatan. Pertama adalah *Black-Box Testing* dan kedua adalah *User Acceptance testing* (UAT).

3.2.1 Black Box Testing

Skenario dalam pengujian kotak hitam dipaparkan pada tabel berikut ini:

Tabel 3. 4 Rencana Tabel Pengujian *Black Box*

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Status
1	Berisi tentang skenario pengujian yang akan dilakukan untuk memastikan output sesuai.	Berisi tentang kasus pengujian positif atau negatif.	Berisi output yang diharapkan sesuai dengan skenario yang ditentukan.	Berisi hasil nyata dari pengujian (berhasil atau gagal).

Black Box Testing adalah metode pengujian yang fokus pada fungsionalitas aplikasi tanpa mempertimbangkan struktur internal atau kode program. Seperti pada tabel 3.4, dalam tahap ini penulis akan menguji setiap fitur yang ada dalam aplikasi berdasarkan spesifikasi yang telah ditentukan. Setiap fitur akan diuji dengan berbagai skenario untuk memastikan bahwa aplikasi berfungsi sesuai harapan dan tidak terdapat bug yang mengganggu pengalaman pengguna.

Tabel 3. 5 Rencana Tabel Pengujian Instalasi *Dynamic Feature Module*

No	Dynamic Feature	On-demand Download	On-demand Install	Size
1	Berisi nama dari fitur modul dinamis yang diuji.	Berisi validasi apakah fitur bisa diunduh.	Berisi validasi apakah fitur berhasil diinstal.	Berisi tentang penambahan ukuran aplikasi setelah modul diinstal.

Tabel 3.5 merupakan rencana dokumentasi dari pengujian instalasi fitur dinamis. Memastikan semua fitur dinamis yang ditentukan untuk on-demand dapat diinstal oleh pengguna sesuai dengan kebutuhan.

3.2.2 User Acceptance Testing

penulis akan melanjutkan ke User Acceptance Testing (UAT), di mana pengguna akhir akan menguji aplikasi untuk memastikan bahwa aplikasi memenuhi kebutuhan dan harapan mereka. Pada tahap ini, penulis akan melibatkan sekelompok pengguna yang representatif, seperti petani dan pengelola agroindustri, untuk melakukan pengujian.

Tabel 3. 6 Rencana Tabel Pengujian UAT

No	Skenario Penggunaan	Status	Catatan
1	Berisi skenario penggunaan aplikasi yang dilakukan oleh responden.	Berisi status penerimaan fitur (diterima, diterima dengan catatan, atau ditolak).	Berisi catatan bila diperlukan.

Pengguna akan diminta untuk menggunakan semua fitur yang ada dalam aplikasi dan memberikan umpan balik mengenai fungsionalitas serta kemudahan penggunaan. Mereka juga akan mengevaluasi apakah aplikasi memenuhi kebutuhan yang telah diidentifikasi selama tahap perencanaan kebutuhan. Untuk tahap peluncuran sesuai dengan syarat dan ketentuan dari Play Console, aplikasi akan diujikan pada minimal 12 pengguna.

BAB IV. ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem membahas proses perencanaan kebutuhan hingga proses desain sistem. Perencanaan kebutuhan meliputi ruang lingkup, target pengguna hingga analisis kebutuhan fungsional maupun non fungsional. Sedangkan desain sistem mencakup pembuatan *diagram*, alur sistem dan arsitektur program.

4.1 Analisis Sistem

Analisis sistem membahas cakupan pengembangan beserta batasannya. Selain itu juga membahas pengguna dari aplikasi beserta kebutuhan fungsional dan non-fungsional.

4.1.1 Ruang Lingkup

Aplikasi Smart Farming Agrilink Vocpro merupakan sebuah aplikasi *mobile* berbasis Android yang dikembangkan dengan menggunakan teknologi utama Kotlin Jetpack Compose, yang memungkinkan pengembangan antarmuka pengguna yang interaktif dan dinamis. Aplikasi ini menerapkan Dynamic Feature Module dari Google Play Feature Delivery, yang memungkinkan beberapa fitur utama diunduh secara on-demand sehingga menghemat ruang penyimpanan dan meningkatkan pengalaman pengguna. Pengembangan elemen-elemen UI yang responsif dan interaktif menjadi fokus agar aplikasi mudah digunakan oleh berbagai kalangan pengguna. Untuk tahap awal pengembangan, fokus utama aplikasi adalah pada lingkup *green house*, yaitu pemantauan dan pengendalian lingkungan rumah kaca secara digital. Fitur-fitur inti seperti Autentikasi dan *Monitoring Sensor Green House* sudah termasuk dalam instalasi awal aplikasi, sementara fitur on-demand lainnya seperti *Control Actuator*, *Optimal Growth Formula*, *Commodity Price Prediction*, dan *Plant Disease Detection* dapat diunduh sesuai kebutuhan pengguna. Sistem ini memungkinkan pemantauan data sensor secara real-time dan kontrol otomatis rumah kaca, serta menyediakan prediksi harga komoditas dan deteksi penyakit tanaman menggunakan teknologi AI. Dengan demikian, ruang lingkup sistem mencakup proses autentikasi, monitoring, dan pengendalian lingkungan pertanian khususnya di dalam Green House, serta analisis data pendukung bagi pengguna.

4.1.2 Analisis Pengguna

Pengguna aplikasi SmartFarming terdiri dari beberapa kelompok utama yang memiliki kebutuhan berbeda namun saling berkaitan dalam mengelola Green House Bumi Aji.

1. Pertama, pengelola Green House yang bertanggung jawab atas pemantauan dan pengendalian lingkungan rumah kaca secara keseluruhan.
2. Kedua, para petani yang secara langsung mengelola tanaman dan menggunakan aplikasi untuk memonitor kondisi tanaman serta melakukan kontrol otomatisasi penyiraman dan pemupukan.
3. Ketiga, kelompok siswa dan mahasiswa yang melakukan magang atau penelitian di Green House tersebut dan memanfaatkan aplikasi sebagai alat bantu dalam pengumpulan data dan analisis kondisi pertanian untuk keperluan akademik.

Semua pengguna melakukan registrasi dan login melalui email dan password, kemudian memiliki hak akses yang sama dalam menggunakan fitur-fitur aplikasi. Pendekatan ini memungkinkan aplikasi berfungsi tidak hanya sebagai sarana manajemen pertanian modern tetapi juga sebagai media edukasi dan penelitian praktis di lingkungan Green House Bumi Aji.

4.1.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional adalah proses mengidentifikasi dan mendefinisikan fungsi-fungsi yang harus disediakan sistem untuk memenuhi kebutuhan pengguna. Kebutuhan fungsional aplikasi ini mencakup berbagai fungsi utama yang mendukung tujuan pemantauan dan pengendalian pertanian secara digital.

Tabel 4. 1 Tabel Analisis Kebutuhan Fungsional

No	Fitur	Deskripsi
1	Autentikasi	Registrasi, Login, dan Logout pengguna
2	Monitoring Sensor Green House	Menampilkan data sensor tanah (NPK) data sensor ruang terkini serta grafik detail.
3	Control Actuator	Kontrol penyiraman air dan pupuk pada bet tanaman, riwayat penyiraman.

4	Growth Formula	Grafik nutrisi tanah, rekomendasi penyiraman berdasarkan komoditas.
5	Commodity Price Prediction	Grafik harga komoditas di 3 daerah dengan perbandingan dan prediksi harga.
6	Plant Disease Detection	Diagnosis penyakit tanaman menggunakan AI, hasil diagnosis tersimpan pada database local.

4.1.4 Analisis Kebutuhan Non Fungsional

Dalam pengembangan aplikasi ini, kebutuhan non fungsional menjadi faktor penting yang menjamin kualitas, performa, dan kemudahan penggunaan sistem.

1. Library yang digunakan

Beberapa library utama yang digunakan dalam aplikasi dapat dilihat pada tabel 4.2

Tabel 4. 2 Tabel Daftar Library

Library	Deskripsi
Play Core Delivery	Mendukung Dynamic Feature Module agar fitur aplikasi bisa diunduh sesuai kebutuhan pengguna.
AndroidX Core KTX	Menyediakan ekstensi Kotlin untuk Android API, mempermudah pengembangan aplikasi Android.
Lifecycle KTX	Mendukung pengelolaan siklus hidup komponen seperti ViewModel dan Activity.
Jetpack Compose	Framework modern untuk membangun UI deklaratif secara efisien dan reaktif.
Material 3 Compose	Komponen UI berbasis Material Design 3 yang konsisten dan adaptif di Jetpack Compose.
Navigation Compose	Navigasi deklaratif antar layar dalam aplikasi menggunakan Jetpack Compose.
Retrofit	Library utama untuk melakukan request ke REST API dengan cara yang efisien dan terstruktur.

Library	Deskripsi
OkHttp	Klien HTTP tingkat rendah untuk Retrofit, dilengkapi dengan logging interceptor untuk debugging.
DataStore Preferences	Alternatif modern untuk SharedPreferences untuk penyimpanan data lokal.
Koin	Framework Dependency Injection yang ringan untuk mengelola ViewModel dan service lainnya.
JWT (Java JWT)	Digunakan untuk encoding dan decoding token autentikasi berbasis JSON Web Token (JWT).

2. Kebutuhan Perangkat Keras

- Sistem operasi minimal Windows 11.
- Microsoft Word, Goole Spreadsheet, dan Figma sebagai tools untuk penulisan proposal dan laporan selama proses pengembangan.
- Android SDK, kumpulan dari beberapa tools atau kit yang digunakan untuk membantu mengembangkan aplikasi android.
- Android Studio, sebagai environment dan tempat penulisan kode untuk mengembangkan aplikasi android.
- Postman, sebagai alat untuk membantu melakukan pengujian API.
- Gitlab, sebagai layanan untuk menyimpan kode program dan juga version control.
- Figma, sebagai alat untuk mendesain UI dari sistem.

3. Kebutuhan Perangkat Lunak

- Laptop/PC dengan spesifikasi prosesor 11th Gen Intel(R) Core(TM) i7-11370H @ 3.30GHz, GPU NVIDIA GeForce MX450, RAM 16GB, dan storage 1TB.
- Smartphone android dengan spesifikasi Android 11, prosesor Mediatek MT6765 Helio P35 (12 nm), RAM 4GB, GPU PowerVR GE8320, dan storage 64GB.

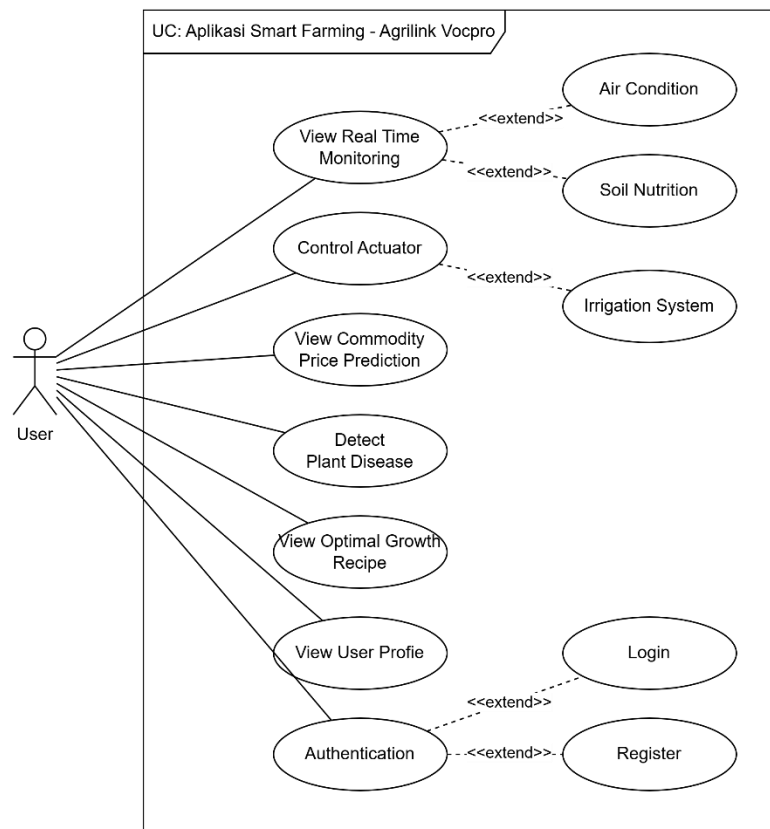
4.2 Perancangan Sistem

Perancangan sistem mencakup beberapa bagian yaitu mulai dari desain sistem hingga alur sistem.

4.2.1 Desain Sistem

Desain sistem pada proyek ini mencakup Use Case Diagram untuk menggambarkan kebutuhan fungsional, serta Activity Diagram untuk memodelkan alur proses utama dari sistem yang dikembangkan.

4.2.1.1. Use Case Diagram



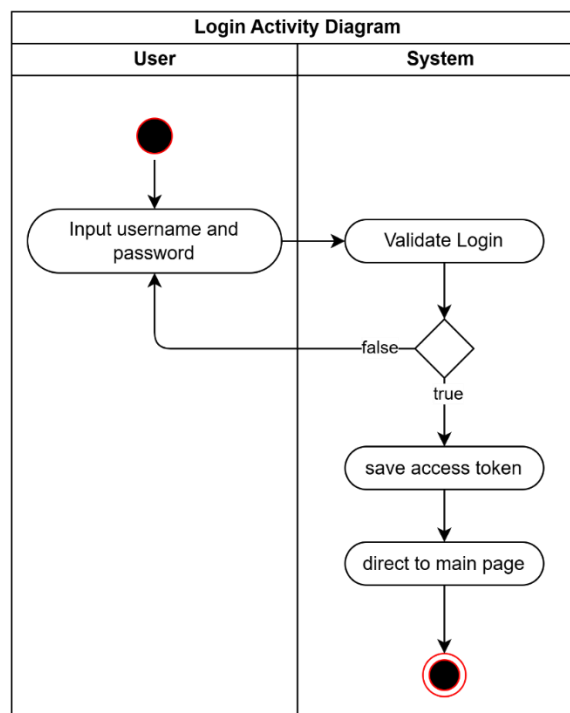
Gambar 4. 1 Use Case Diagram Aplikasi

Use Case Diagram aplikasi Smart Farming Agrilink Vocpro menggambarkan bagaimana pengguna berinteraksi dengan berbagai fitur utama dalam aplikasi. Sesuai dengan gambar 4.1 pengguna dapat melakukan pemantauan kondisi secara real-time seperti udara dan nutrisi tanah, mengendalikan aktuator seperti sistem irigasi, melihat prediksi harga komoditas, mendeteksi penyakit tanaman secara dini, serta mengakses formula pertumbuhan optimal tanaman dan profil pengguna. Proses autentikasi melalui login dan registrasi juga menjadi bagian penting dalam

menjaga keamanan akses. Hubungan extend menunjukkan bahwa fungsi-fungsi tambahan seperti kondisi udara, nutrisi tanah, dan sistem irigasi merupakan pengembangan dari fungsi utama, yang memberikan fleksibilitas dan modularitas dalam sistem.

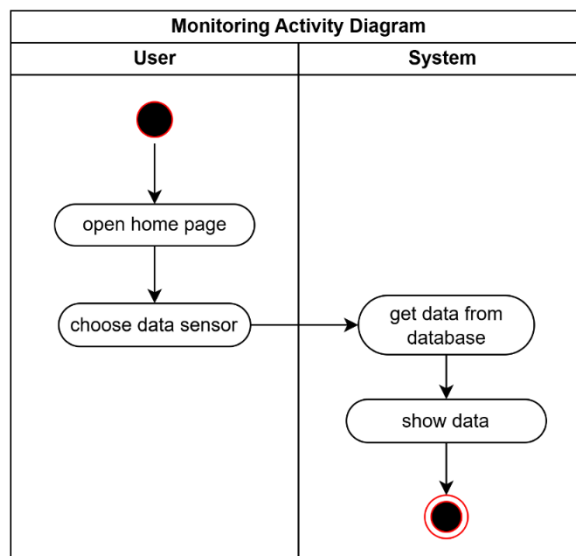
4.2.1.2. Activity Diagram

Activity Diagram merupakan diagram yang menggambarkan alur aktivitas yang dilakukan oleh pengguna dalam sebuah sistem. Diagram ini biasanya berupa penjabaran lebih rinci dari setiap *use case diagram*. Activity diagram menjelaskan proses bagaimana alur sebuah sistem berjalan dan digambarkan secara vertical.



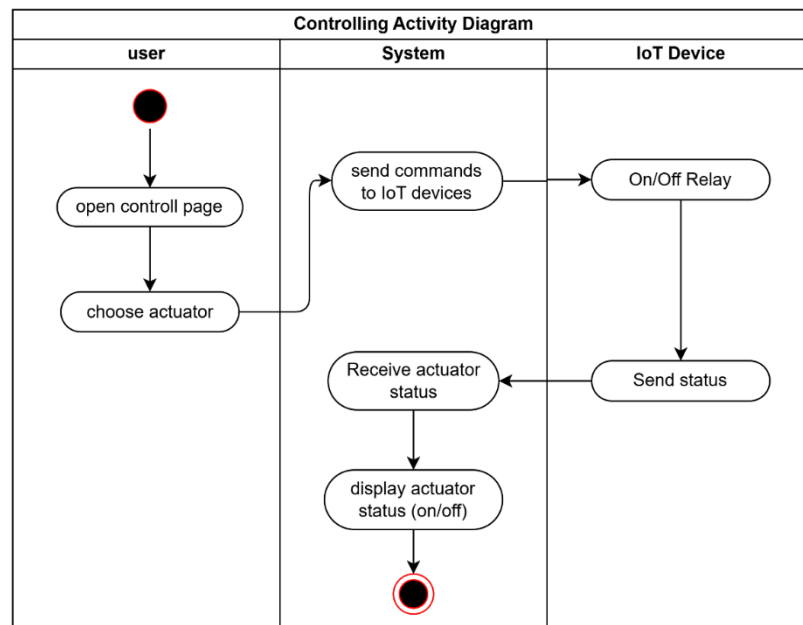
Gambar 4. 2 Login Activity Diagram

Pada aplikasi *mobile smart farming* "Agrilink Vocpro", diagram ini dimulai dengan langkah Login sesuai dengan gambar 4.2, di mana pengguna memasukkan kredensial mereka untuk mengakses aplikasi. Setelah berhasil login, sistem akan menyimpan token akses dan pengguna akan diarahkan ke halaman utama, yang berfungsi sebagai pusat navigasi untuk semua fitur yang tersedia.



Gambar 4. 3 Monitor Activity Diagram

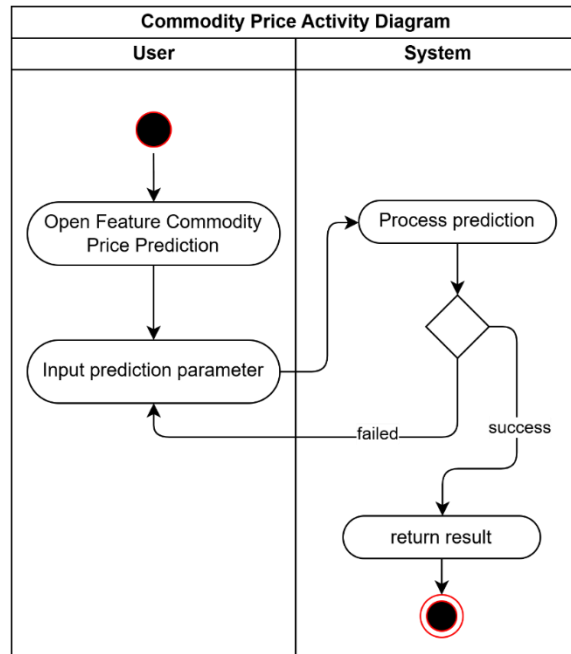
Pada gambar 4.3 merupakan alur dari aktivitas fitur monitoring. Alur ini dimulai ketika pengguna membuka halaman beranda dan pengguna memilih data sensor yang ingin dilihat. Setelah itu sistem akan mengambil data sesuai dengan data sensor yang diminta oleh user dan menampilkan data tersebut.



Gambar 4. 4 Controlling Activity Diagram

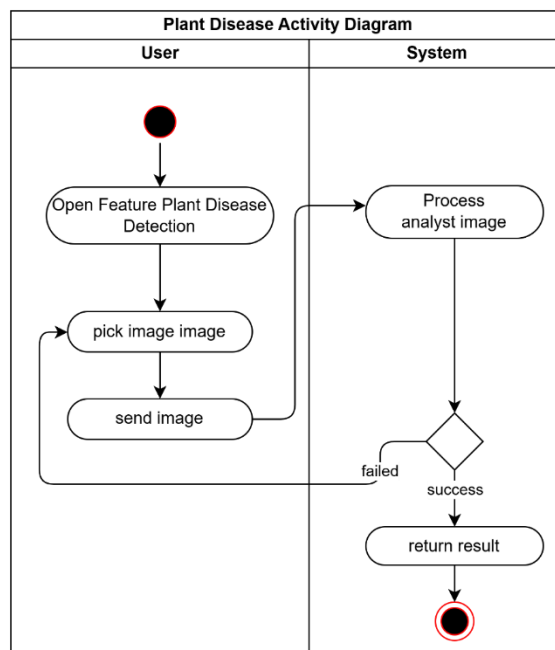
Gambar 4.4 merupakan alur dari aktivitas kontrol aktuator pada *green house*. Alur ini dimulai dengan pengguna membuka halaman kontrol. Lalu pengguna akan memilih jenis aktuator yang akan dikontrol, seperti katup nutrisi dan katup air.

Setelah itu sistem akan mengirim perintah ke perangkat IoT agar perangkat IoT dapat mengontrol aktuator sesuai dengan perintah. Kemudian perangkat IoT akan mengirim status dari aktuator tersebut untuk diterima oleh sistem. Terakhir sistem akan menampilkan status aktuator kepada pengguna.



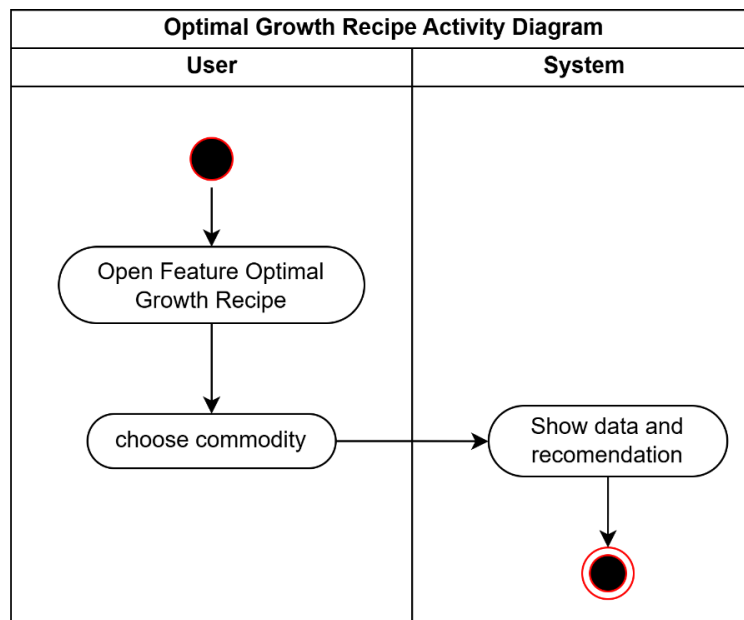
Gambar 4. 5 Commodity Price Activity Diagram

Gambar 4.5 merupakan alur aktivitas fitur prediksi harga komoditas. Alur dimulai dengan pengguna membuka fitur dan mengisi parameter yang dibutuhkan untuk prediksi harga pada form yang disediakan. Kemudian sistem akan melakukan proses prediksi dan menampilkan hasil prediksi kepada pengguna.



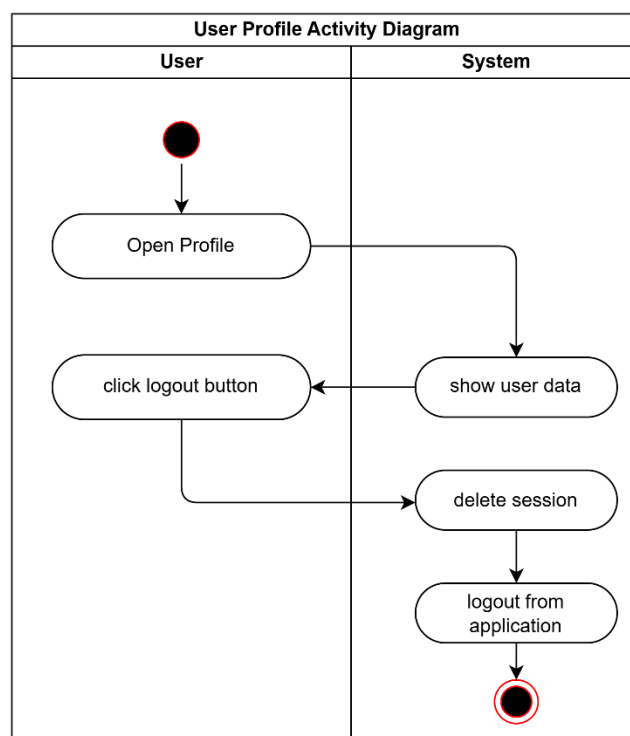
Gambar 4. 6 Plant Disease Detection Activity Diagram

Gambar 4.6 merupakan alur dari aktivitas fitur deteksi penyakit pada tanaman. Alur dimulai dengan pengguna membuka halaman fitur. Kemudian pengguna akan diarahkan untuk mengambil foto dari tanaman yang akan di deteksi. Kemudian pengguna dapat mengirim gambar tersebut. Sistem akan mulai menganalisis gambar dan mendeteksi penyakit pada tanaman. Setelah itu sistem akan menampilkan hasilnya ke pengguna.



Gambar 4. 7 Optimal Growth Formula

Gambar 4.7 adalah alur dari fitur formula pertumbuhan optimal dimulai ketika pengguna membuka halaman fitur *Growth Formula*. Kemudian pengguna dapat memilih jenis komoditas dan sistem akan menampilkan rekomendasi formula pertumbuhan dari komoditas tersebut.

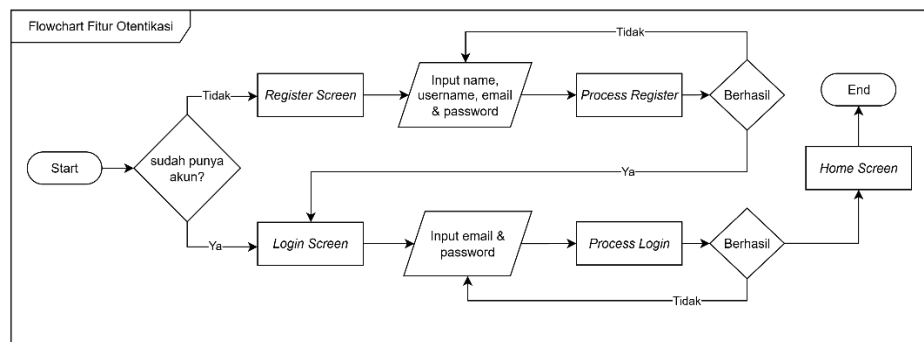


Gambar 4. 8 User Profile Activity Diagram

Gambar 4.8 merupakan alur dari aktivitas fitur profil pengguna. Pengguna bisa menekan ikon profil untuk melihat data pribadi pengguna. Pada halaman ini juga disediakan alur untuk *logout* dari aplikasi. Pengguna menekan tombol logout, sistem akan menghapus session dari aplikasi dan mengarahkan pengguna ke halaman login.

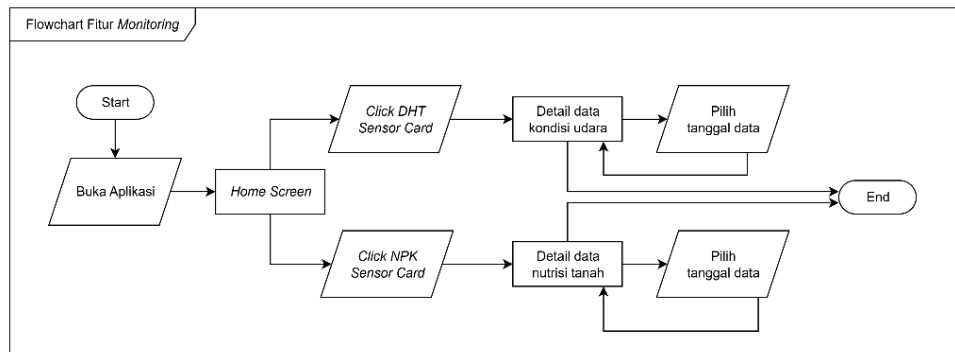
4.2.1.3. Flowchart

Flowchart disusun untuk mempermudah pemahaman alur kerja setiap fitur serta memastikan bahwa proses dalam sistem berjalan secara sistematis dan efisien sesuai dengan prinsip *user experience*.



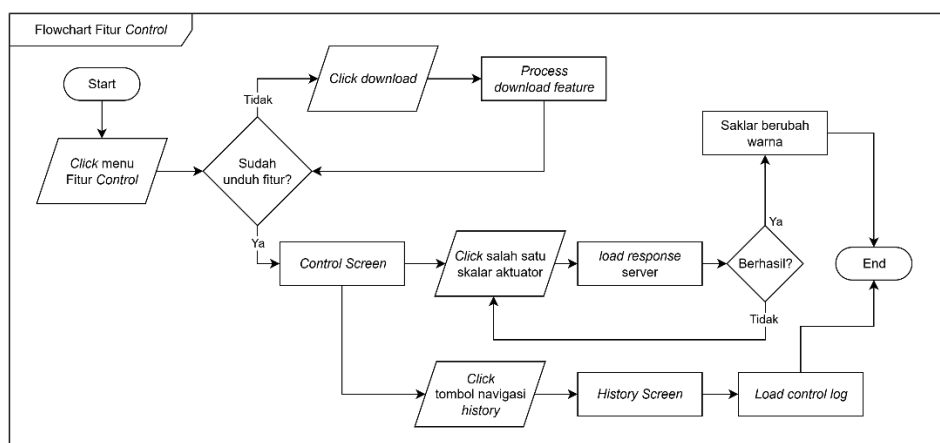
Gambar 4. 9 Flowchart fitur otentikasi

Flowchart pada Gambar 4.9 menunjukkan alur proses otentikasi dalam aplikasi Smart Farming Agrilink Vocpro. Proses dimulai dengan pengecekan apakah pengguna sudah memiliki akun. Jika belum, pengguna diarahkan ke halaman registrasi untuk mengisi nama, *username*, email, dan kata sandi. Setelah data dikirim dan proses registrasi berhasil, pengguna masuk ke halaman *login*. Jika gagal, pengguna diminta untuk mengulangi proses. Sebaliknya, jika pengguna sudah memiliki akun, sistem akan menampilkan halaman *login*. Pengguna cukup mengisi email dan kata sandi, lalu sistem memproses permintaan *login*. Jika data valid, pengguna diarahkan ke halaman utama (*Home Screen*). Namun jika gagal, sistem akan menampilkan pesan error dan tetap berada di halaman login.



Gambar 4. 10 *Flowchart fitur monitor*

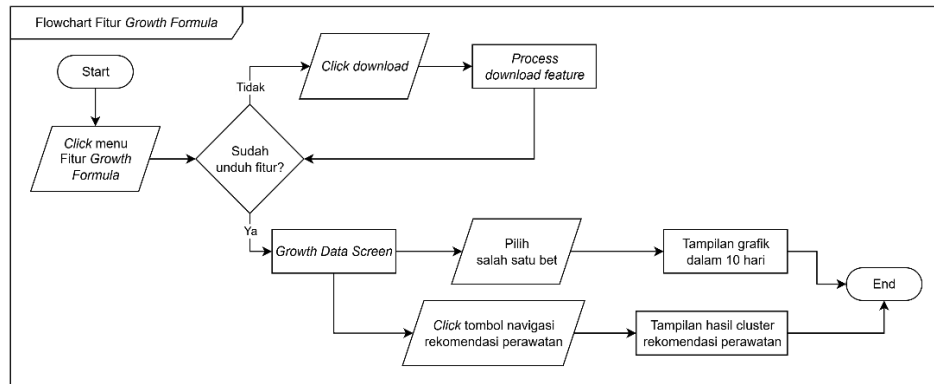
Flowchart gambar 4.10 di atas menjelaskan alur kerja fitur *Monitoring* pada aplikasi. Proses dimulai ketika pengguna membuka aplikasi dan diarahkan ke halaman utama (*Home Screen*). Di halaman ini, pengguna disajikan dengan dua pilihan utama berupa kartu sensor, yaitu *DHT Sensor Card* untuk memantau kondisi udara dan *NPK Sensor Card* untuk memantau kondisi nutrisi tanah. Jika pengguna mengklik salah satu kartu sensor, maka akan diarahkan ke halaman detail yang menampilkan data secara lengkap sesuai dengan jenis sensor yang dipilih. Selanjutnya, pengguna dapat memilih tanggal tertentu untuk melihat data historis pada hari tersebut.



Gambar 4. 11 *Flowchart fitur control*

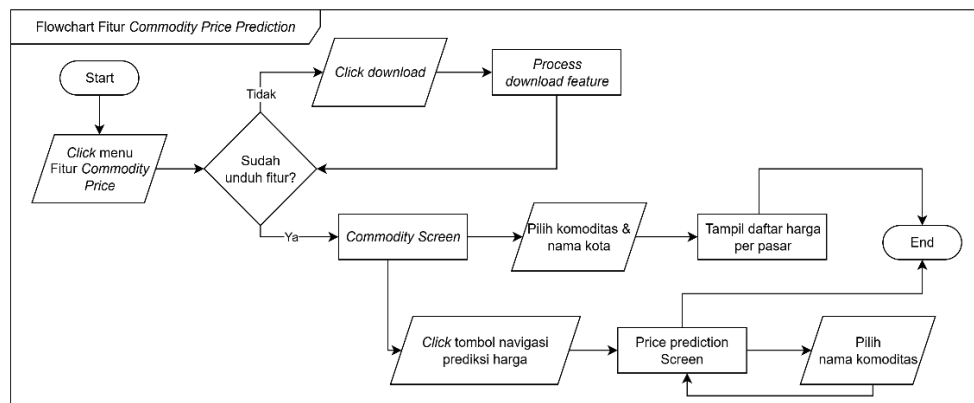
Flowchart gambar 4.11 di atas menggambarkan alur kerja fitur *control*, yang dapat diakses dari halaman utama aplikasi. Ketika pengguna menekan menu *control*, sistem akan terlebih dahulu memeriksa apakah fitur sudah diunduh (karena merupakan salah satu fitur *on-demand*). Jika belum, pengguna diarahkan untuk mengunduh terlebih dahulu sebelum masuk ke halaman kontrol (*Control Screen*).

Setelah berada di halaman *control*, pengguna dapat menyalakan atau mematikan aktuator (seperti pompa atau valve) dengan menekan saklar. Sistem kemudian akan mengirim permintaan ke server dan menunggu respons. Jika berhasil, saklar akan berubah warna sebagai indikasi visual. Pengguna juga dapat menavigasi ke halaman histori untuk melihat log aktivitas kontrol yang tersimpan sebelumnya.



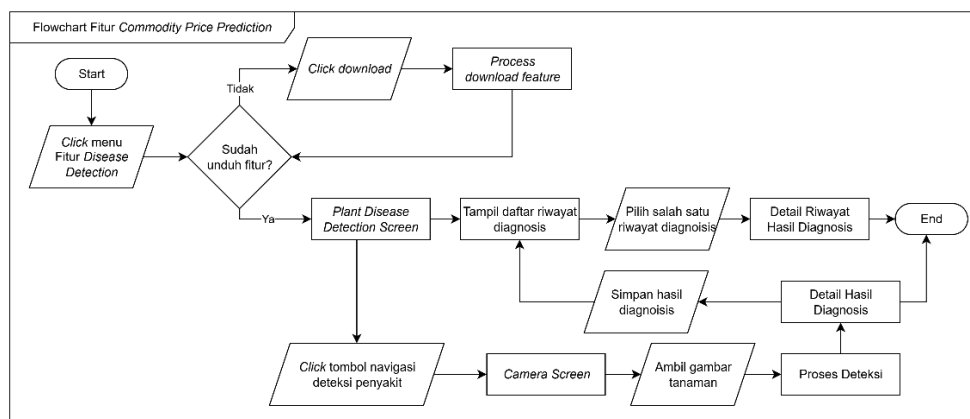
Gambar 4. 12 *Flowchart* fitur *growth formula*

Flowchart pada gambar 4.12 ini menjelaskan alur interaksi pengguna saat menggunakan fitur *Growth Formula*. Pengguna memulai dengan memilih menu *Growth Formula* dari halaman utama. Jika fitur belum diunduh (karena fitur *on-demand*), sistem akan meminta pengguna untuk mengunduh terlebih dahulu sebelum dapat melanjutkan. Setelah proses unduhan selesai, pengguna diarahkan ke halaman utama fitur yaitu *Growth Data Screen*. Di halaman tersebut, pengguna dapat memilih salah satu bet atau area tanam untuk ditinjau. Setelah dipilih, sistem akan menampilkan grafik perkembangan pertumbuhan selama 10 hari terakhir berdasarkan data sensor yang dikumpulkan. Selain itu, pengguna dapat menekan tombol navigasi untuk melihat hasil klasterisasi berupa rekomendasi perawatan tanaman berdasarkan kondisi aktual di bet tersebut.



Gambar 4. 13 *Flowchart* fitur prediksi harga komoditas

Flowchart pada gambar 4.13 ini menggambarkan alur kerja pengguna saat mengakses fitur *Commodity Price Prediction*. Pengguna memulai dengan memilih menu *Commodity Price* dari halaman utama aplikasi. Apabila fitur ini belum diunduh, maka pengguna akan diarahkan untuk mengunduh fitur terlebih dahulu. Setelah proses unduhan selesai, pengguna dapat mengakses halaman utama fitur yaitu *Commodity Screen*. Di halaman *Commodity Screen*, pengguna diminta untuk memilih nama komoditas dan kota yang diinginkan. Aplikasi akan menampilkan daftar harga komoditas tersebut di berbagai pasar di daerah yang dipilih. Selanjutnya, pengguna dapat menavigasi ke halaman prediksi harga dengan menekan tombol khusus. Di halaman ini, pengguna dapat memilih kembali nama komoditas untuk melihat prediksi harga.



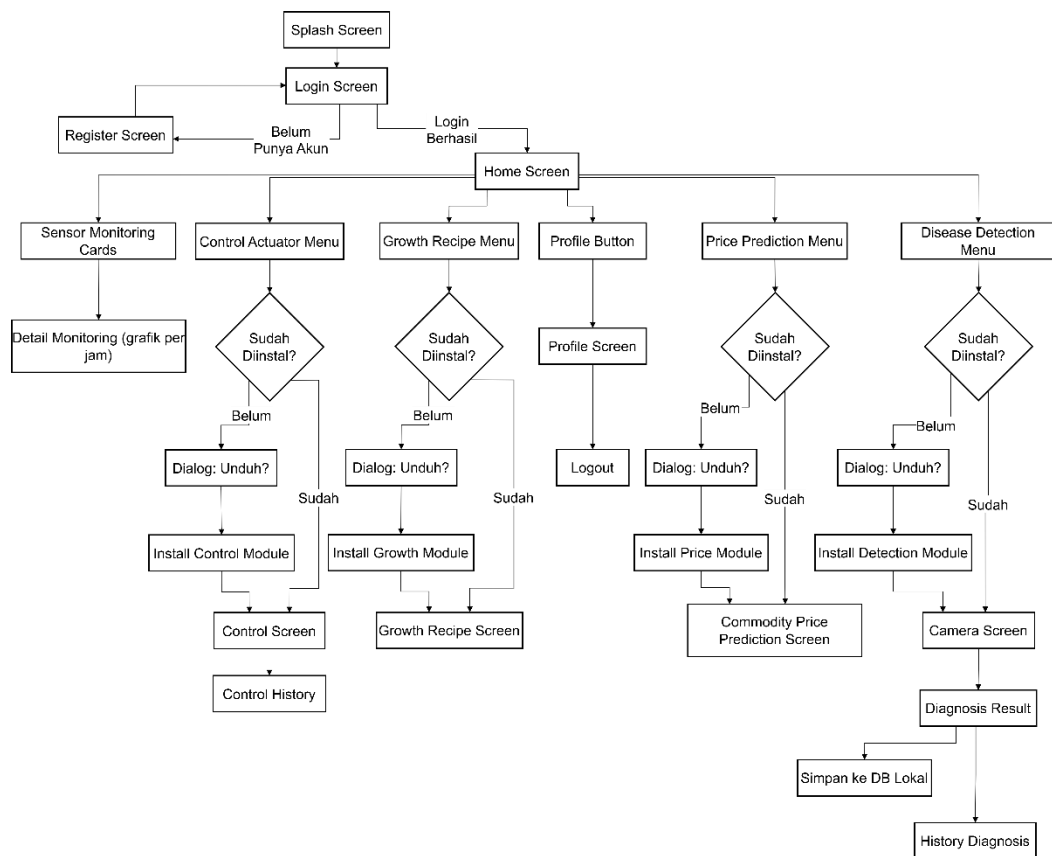
Gambar 4. 14 *Flowchart* fitur deteksi penyakit tanaman

Flowchart gambar 4.14 ini menggambarkan alur penggunaan fitur *Plant Disease Detection* dalam aplikasi Smart Farming Agrilink Vocpro. Pengguna memulai dengan memilih menu *Disease Detection* dari halaman utama. Jika fitur

belum diunduh, maka sistem akan mengarahkan pengguna untuk mengunduh fitur terlebih dahulu. Setelah berhasil diunduh, pengguna akan masuk ke halaman utama fitur dan dapat memilih untuk melakukan deteksi penyakit tanaman. Pengguna dapat menavigasi ke halaman kamera untuk mengambil gambar tanaman yang akan didiagnosis. Setelah gambar diambil, sistem akan memproses deteksi penyakit menggunakan model *AI* (Gemini API). Hasil diagnosis ditampilkan dan pengguna dapat memilih untuk menyimpan hasil tersebut. Selain itu, pengguna juga dapat melihat riwayat hasil diagnosis sebelumnya dan menampilkan detail diagnosis tertentu jika dibutuhkan.

4.2.2 Alur Sistem

Alur sistem ini adalah perancangan bagaimana struktur navigasi yang ada di dalam sistem. Gambaran alur aplikasi secara lengkap dapat dilihat pada gambar dibawah ini.

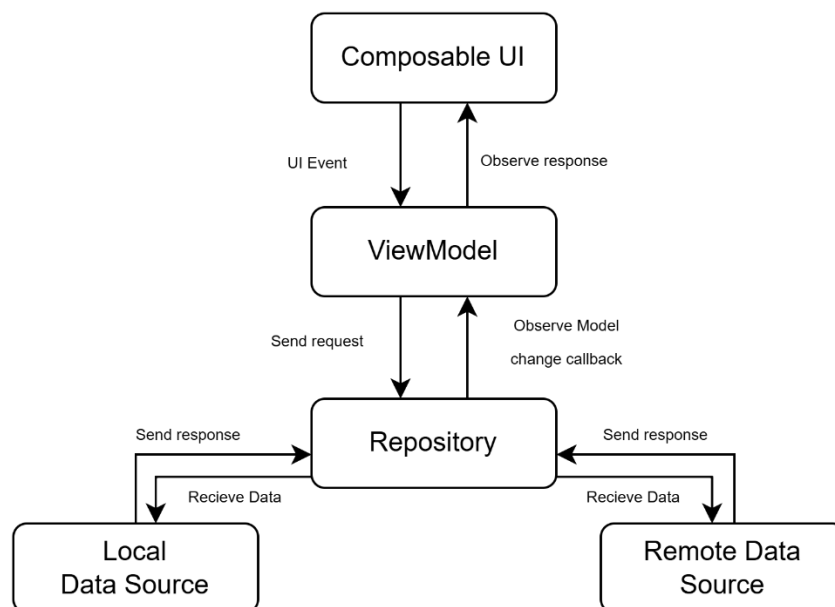


Gambar 4. 15 Gambar Alur Aplikasi secara umum

Gambar 4.15 menjelaskan tentang alur aplikasi. Alur sistem aplikasi Smart Farming Agrilink Vocpro dimulai dari *Splash Screen* yang kemudian mengarahkan

pengguna ke *Login Screen*, di mana pengguna yang belum memiliki akun dapat menuju Register Screen untuk membuat akun sebelum kembali ke *Login*. Setelah login berhasil, pengguna masuk ke *Home Screen* yang menampilkan fitur utama seperti pemantauan sensor melalui *Sensor Monitoring Cards* dengan detail grafik per jam. Pengguna juga dapat mengakses fitur on-demand seperti *Control Actuator*, *Growth Formula*, *Commodity Price Prediction*, dan *Plant Disease Detection*, di mana setiap fitur memeriksa instalasi modul terlebih dahulu dan memberikan opsi unduh serta instal apabila belum tersedia. *Control Actuator* memungkinkan kendali otomatisasi penyiraman dan pemupukan serta melihat riwayat kontrol, *Growth Formula* memberi rekomendasi perawatan dan grafik nutrisi tanaman, *Commodity Price Prediction* menampilkan prediksi harga komoditas di berbagai daerah, dan *Plant Disease Detection* menggunakan kamera untuk diagnosis penyakit tanaman dengan hasil yang dapat disimpan dan dilihat riwayatnya. Selain itu, pengguna dapat melihat data profil dan melakukan *logout* melalui halaman *Profile*. Alur ini dirancang agar penggunaan aplikasi efisien dan modul on-demand menjaga ruang penyimpanan tetap optimal.

4.2.3 Arsitektur Program



Gambar 4. 16 Arsitektur MVVM

Sistem aplikasi *smart farming* Agrilink Vocpro menggunakan arsitektur Model-View-ViewModel (MVVM) yang membagi aplikasi menjadi tiga

komponen utama untuk memisahkan logika bisnis, pengelolaan data, dan tampilan antarmuka pengguna secara jelas. Pendekatan ini memudahkan proses pengembangan, pengujian, dan pemeliharaan aplikasi. Proyek aplikasi memiliki struktur folder yang sistematis sesuai dengan tanggung jawab masing-masing bagian sesuai table 4.3. Package dan penyusunan folder ini juga berlaku untuk setiap modul fitur dinamis yang dibuat:

Tabel 4. 3 *Foldering* Arsitektur MVVM

Folder/Package	Fungsi
app/	Folder utama yang berisi keseluruhan modul dan komponen aplikasi.
core/	Menyimpan komponen inti dan utilitas yang digunakan di seluruh aplikasi.
core/component/	Berisi komponen-komponen reusable seperti custom UI components dan base classes.
core/utills/	Menyimpan berbagai fungsi utilitas dan helper yang membantu operasi aplikasi sehari-hari.
data/	Mengelola sumber data aplikasi baik dari lokal maupun jaringan.
data/local/	Implementasi database lokal aplikasi.
data/local/dao/	Mengatur query dan operasi CRUD pada database melalui Data Access Object (DAO).
data/local/databa se/	Konfigurasi dan pengaturan database lokal.
data/local/entity/	Definisi entitas atau skema tabel dalam bentuk kelas yang merepresentasikan data.
data/model/	Mendefinisikan model data bisnis yang digunakan sebagai representasi data dalam aplikasi.
data/network/	Bertanggung jawab atas komunikasi dengan server dan pengambilan data dari API jaringan.
data/repository/	Mediator antara sumber data (local/network) dengan ViewModel, menyederhanakan logika data.

Folder/Package	Fungsi
di/	Mengelola Dependency Injection untuk penyuntikan objek secara otomatis agar modularitas tinggi dan memudahkan pengujian.
navigation/	Mengoordinasikan perpindahan antar layar dalam aplikasi, mengatur rute navigasi secara sentral.
presentation/	Menangani tampilan dan interaksi pengguna.
presentation/screen/	Berisi kode UI setiap layar, misalnya composable functions pada Jetpack Compose.
presentation/viewmodel/	Berisi kelas ViewModel yang menghubungkan data dari repository dengan UI dan mengelola state aplikasi.

BAB V IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi Sistem

Sub bab ini menjelaskan proses implementasi sistem mulai dari pembuatan modul fitur dinamis, hingga implementasi pada semua fitur nya.

5.2 Implementasi *Dynamic Feature Module*

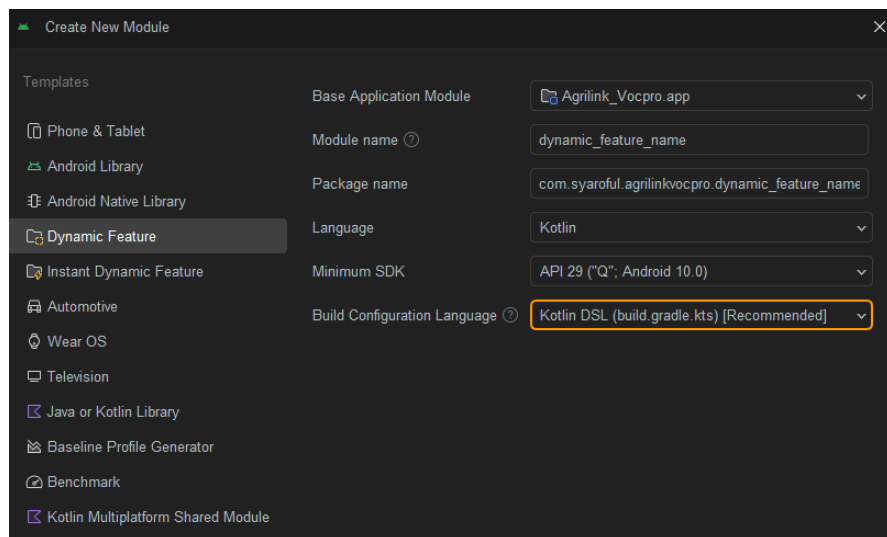
Proses implementasi modul fitur dinamis dimulai dari inialisasi *project* android, pembuatan fitur dinamis, dan imlementasi untuk pengunduhan fiturnya.

5.2.1 Integrasi Modul Dinamis

Pada tahap awal pengembangan aplikasi Smart Farming ini, dilakukan pembuatan modul baru yang mengimplementasikan konsep dynamic feature pada Android. Modul ini dibuat menggunakan template dynamic feature yang tersedia di Android Studio. Berikut ini adalah Langkah implementasinya:

- a. Penggunaan template dynamic feature dari Android Studio

Modul baru dibuat menggunakan template dari Android Studio, ini akan membuat sebuah root *project* folder baru dengan nama sesuai dynamic feature yang dibuat.



Gambar 5. 1 Pembuatan modul baru di Android Studio

Sesuai gambar 5.1 pembuatan modul baru dilakukan dengan mengisi beberapa form konfigurasi yang disediakan termasuk dengan nama package. Pembuatan nama package ini sangat penting karena berhubungan dengan proses navigasi. Dalam hal ini penulisan *package* sama dengan *root package* dari *base module* diikuti dengan nama fitur dinamis.

```
com.syaroful.agrilinkvocpro.control_feature
```

Kemudian pada bagian selanjutnya on-demand harus dipilih pada menu install-time inclusion.

b. Konfigurasi Gradle

Konfigurasi pertama sebelum *gradle* adalah memodifikasi *AndroidManifest* untuk *on-demand delivery*. Jika memakai template dari *Android Studio* biasanya akan langsung otomatis terkonfigurasi.

```
<dist:module
    dist:instant="false"
    dist:title="@string/title_growth_Formula_featur
e">
        <dist:delivery>
            <dist:on-demand />
        </dist:delivery>
        <dist:fusing dist:include="false" />
</dist:module>
```

Kemudian konfigurasi berikutnya adalah pada file *build.gradle.kts project* fitur dinamis harus mengimplementasikan base module (app).

```
dependencies {
    implementation(project(":app"))
    ...
}
```

Terakhir, konfigurasi pada file *setting.gradle.kts* dan *build.gradle.kts* (*base module*). module dari fitur dinamis harus diinisialisasikan di bagian ini. 'dynamicFeatures' di modul app digunakan untuk mendaftarkan modul dynamic feature yang dimiliki proyek.

```
rootProject.name = "Agrilink Vocpro"
include(":app")
include(":control_feature")
include(":plant_disease_detection_feature")
include(":growth_Formula_feature")
include(":commodity_price_prediction_feature")

android {
    . . .
    dynamicFeatures += setOf(
        ":control_feature",
```

```

        ":plant_disease_detection_feature",
        ":growth_Formula_feature",
        ":commodity_price_prediction_feature"
    )
}

```

5.2.2 Integrasi Penginstalan Fitur Secara *On-Demand*

Setelah setup module fitur dinamis selesai dilakukan Langkah berikutnya adalah melakukan konfigurasi kode program untuk mengunduh fitur secara on-demand. Langkah utama disini adalah bagaimana mengimplementasikan *library* utama dari Play Feature Delivery. Berikut adalah langkah implementasinya:

- a. Membuat *function* untuk cek modul yang di download

SplitInstallManager adalah komponen utama dari Play Feature Delivery untuk manajemen instalasi modul *dynamic feature* di Android. Semua function utama diinisialisasi pada class repository.

```

fun isModuleDownloaded(moduleName: String): Boolean {
    return
    splitInstallManager.installedModules.contains(moduleName)
}

```

Potongan kode di atas merupakan function pada repository untuk melakukan pengecekan apakah fitur atau modul sudah terinstal atau belum.

- b. Membuat implementasi pengunduhan dan penginstalan modul

```

fun startDownload(
    moduleName: String,
    listener: SplitInstallStateUpdatedListener,
): Task<Int> {
    val request = SplitInstallRequest.newBuilder()
        .addModule(moduleName)
        .build()
    splitInstallManager.registerListener(listener)
    return splitInstallManager.startInstall(request)
        .addOnSuccessListener { id -> sessionId =
            id }
}

```

Di atas ini adalah function untuk memulai pengunduhan fitur atau modul dari Google Play Store. Dengan memberikan nilai nama modul berupa

String maka Aplikasi akan mulai mengunduh fitur tersebut dan langsung melakukan proses instalasi jika pengunduhan selesai.

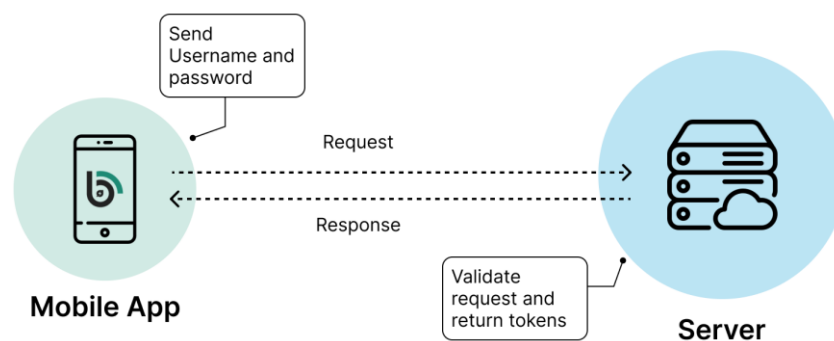
5.3 Implementasi Fitur

Pada bab ini akan dijelaskan implementasi berbagai fitur utama dalam aplikasi Smart Farming Agrilink Vocpro. Implementasi fitur dilakukan dengan memperhatikan kebutuhan pengguna dan integrasi yang optimal antar modul agar aplikasi berjalan dengan lancar dan memenuhi fungsi yang diharapkan.

5.3.1 Fitur Otentikasi

Fitur otentikasi merupakan aspek penting dalam sistem untuk menjaga keamanan dan memastikan hanya pengguna yang terverifikasi yang dapat mengakses aplikasi. Fitur ini meliputi proses registrasi, login, serta pengelolaan sesi pengguna. Implementasi autentikasi dilakukan dengan menggunakan mekanisme pengiriman kredensial (*username* dan *password*) yang kemudian divalidasi oleh server untuk menjaga integritas dan keamanan data pengguna.

5.3.1.1. Data Flow Model Fitur Autentikasi



Gambar 5. 2 Model Alur data Fitur Autentikasi

Gambar data flow model pada fitur autentikasi menjelaskan proses komunikasi antara aplikasi *mobile* dan server dalam autentikasi pengguna. Proses dimulai dari sisi aplikasi *mobile* yang mengirimkan data username dan password ke server melalui request. Server kemudian memvalidasi data tersebut dengan melakukan pengecekan terhadap database pengguna yang tersimpan. Jika validasi berhasil, server memberikan *return* berupa token autentikasi yang berfungsi sebagai bukti bahwa pengguna telah terotentikasi dan berhak mengakses layanan aplikasi.

Token ini kemudian digunakan oleh aplikasi untuk melakukan permintaan data selanjutnya tanpa harus mengirim ulang kredensial.

5.3.1.2. Implementasi Kode Fitur Autentikasi

Kode di bawah ini merupakan fungsi login yang dijalankan secara *suspend* (asinkron). Fungsi ini menerima dua parameter, yaitu *username* dan *password*. Kemudian, fungsi ini membuat header autentikasi dasar menggunakan *OkHttp* dan mengemas data login dalam objek *LoginRequest*. Selanjutnya, fungsi memanggil endpoint login dari API menggunakan *apiService* dengan mengirimkan *header* otentikasi dan *body request*.

```
suspend fun login(username: String, password: String):  
Response<LoginResponse> {  
    val credential = okhttp3.Credentials.basic(username,  
password)  
    val requestBody = LoginRequest(username, password)  
    return apiService.login(authHeader = credential, body =  
requestBody)  
}
```

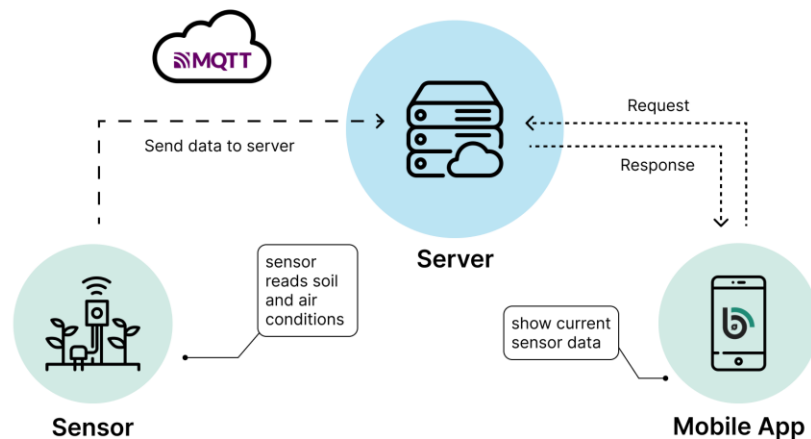
Bagian di bawah ini adalah rute navigasi *Compose* yang menampilkan layar *login* (*LoginScreen*). Saat login berhasil, navigasi diarahkan ke halaman utama aplikasi (*home*) dan halaman *login* dihapus dari tumpukan navigasi agar tidak bisa kembali. Terdapat juga aksi untuk navigasi ke halaman pendaftaran (*register*) dari layar *login*.

```
composable("login") {  
    LoginScreen(  
        loginViewModel = loginViewModel,  
        onLoginSuccess = {  
            navController.navigate("home") {  
                popUpTo("login") { inclusive = true }  
            }  
        },  
        ...  
    )  
}
```

5.3.2 Fitur *Monitoring*

Fitur ini memanfaatkan sensor yang terpasang di area *green house* untuk membaca berbagai parameter penting seperti kelembaban tanah, nutrisi tanah, dan kondisi udara di sekitar tanaman.

5.3.2.1. Data Flow Model Fitur *Monitoring*



Gambar 5. 3 Model Alur Data Fitur *Monitoring*

Model aliran data pada fitur monitoring menggambarkan proses komunikasi antara sensor, server, dan aplikasi *mobile* dalam sistem SmartFarming Agrilink Vocpro. Sensor yang dipasang di lapangan secara kontinu membaca kondisi tanah dan udara, seperti kelembaban dan nutrisi tanah serta kualitas udara sekitar tanaman. Data yang diperoleh oleh sensor ini kemudian dikirimkan secara teratur ke server menggunakan protokol MQTT, yang memungkinkan pengiriman data secara efisien dalam lingkungan IoT. Server bertindak sebagai pusat pengelolaan data, menerima informasi dari sensor, menyimpannya dalam database, dan memprosesnya. Ketika aplikasi *mobile* mengirimkan permintaan data saat pengguna ingin melihat kondisi terkini, server memberikan respon berupa data sensor terbaru yang kemudian ditampilkan dengan antarmuka yang mudah dipahami oleh pengguna.

5.3.2.2. Implementasi Kode Fitur *Monitoring*

```
private var _latestSensorData: SensorDataResponse? = null
val latestSensorData: SensorDataResponse?
    get() = _latestSensorData

suspend fun getLatestSensorData(authHeader: String):
Response<SensorDataResponse> {
    val response = apiService.getLatestSensorData(authHeader)
    if (response.isSuccessful) {
        _latestSensorData = response.body()
    }
    return response
}
```

Kode di atas merupakan fungsi *getLatestSensorData* yang dijalankan secara suspend (asinkron) untuk mengambil data sensor terkini dari server melalui API. Jika respons berhasil, data sensor terbaru disimpan dalam variabel privat *_latestSensorData* dan dapat diakses melalui properti *latestSensorData*.

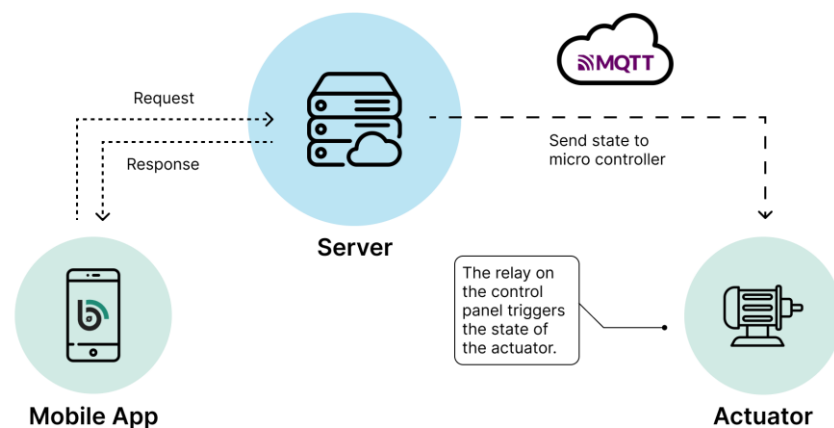
```
suspend fun getNpkDataSensor(
    authHeader: String,
    startDate: String,
    endDate: String,
    timeRange: String = "HOURLY",
    sensor: String
): Response<NpkGraphicDataResponse> {
    return apiService.getNpkDataSensor(authHeader, startDate,
    endDate, timeRange, sensor)
}
```

Fungsi *getNpkDataSensor* digunakan untuk mengambil data sensor NPK (*Nitrogen, Phosphorus, Potassium*) dalam rentang waktu tertentu dari server. Parameter yang digunakan mencakup header otentikasi, rentang tanggal mulai dan akhir, interval waktu (default "HOURLY"), serta tipe sensor. Fungsi mengembalikan respons dari API yang berisi data grafik sensor.

5.3.3 Fitur *Controlling*

Fitur controlling merupakan bagian penting dalam sistem *smart farming* Agrilink Vocpro yang memungkinkan pengguna untuk mengendalikan perangkat aktuator dari jarak jauh melalui aplikasi *mobile*. Melalui fitur ini, pengguna dapat mengatur berbagai perangkat seperti pompa air atau alat irigasi yang terhubung dengan sistem.

5.3.3.1. Data Flow Model Fitur *Controlling*



Gambar 5. 4 Model Alir Data Fitur *Controlling*

Model aliran data fitur controlling menjelaskan proses komunikasi antara aplikasi *mobile*, server, dan aktuator dalam sistem *smart farming*. Sesuai gambar 5.3 proses dimulai dengan aplikasi *mobile* yang mengirimkan permintaan kontrol ke server. Server kemudian menerima permintaan tersebut dan memprosesnya untuk menentukan status yang harus dikirimkan ke aktuator. Selanjutnya, server mengirimkan sinyal status tersebut ke mikrokontroler aktuator melalui protokol MQTT, yang bertindak sebagai perantara komunikasi dalam sistem IoT ini. Saat relay pada panel kontrol menerima sinyal, ia akan memicu perubahan status pada aktuator yang sesuai, seperti mengaktifkan atau mematikan perangkat yang terhubung. Server juga memberikan respons balik ke aplikasi *mobile* sebagai konfirmasi bahwa perintah telah diterima dan diproses dengan benar.

5.3.3.2. Implementasi Kode Fitur *Controlling*

```
suspend fun controlPump(authHeader: String, action: String):  
Response<ControlLogResponse> {  
    val actionPart =  
        action.toRequestBody("text/plain".toMediaTypeOrNull())  
    return controlService.controlPump(authHeader, actionPart)  
}
```

Kode di atas adalah fungsi `controlPump` yang dijalankan secara `suspend` (asinkron) untuk mengirim perintah kontrol pompa ke server. Fungsi ini menerima header autentikasi dan aksi berupa string, yang kemudian diubah menjadi `RequestBody` bertipe `text/plain` sebelum dikirim ke API melalui `controlService`.

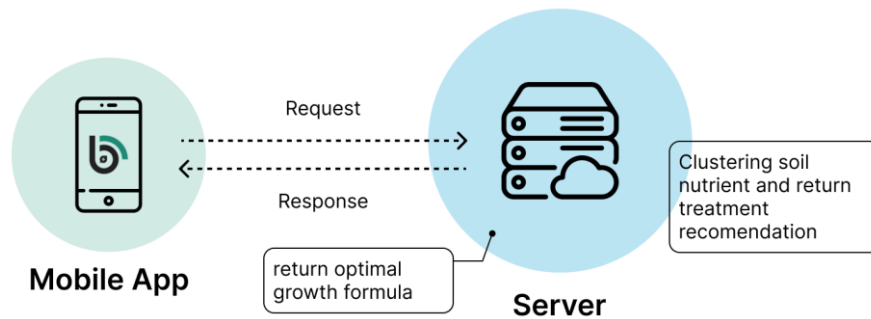
```
ControlCard(  
    iconRes = R.drawable.ic_valve,  
    label = "Pump",  
    isOn = pumpStatus,  
    isLoading = isLoading,  
    onToggle = { viewModel.controlActuator(ActuatorType.PUMP,  
        !pumpStatus) }  
)
```

Bagian ini adalah komponen UI berupa kartu kontrol (`ControlCard`) yang menampilkan status pompa (`pumpStatus`) dan ikon tertentu. Saat tombol toggle diaktifkan, fungsi `controlActuator` dipanggil di `ViewModel` untuk membalik status pompa, mengirimkan perintah menghidupkan atau mematikan pompa sesuai status baru.

5.3.4 Fitur *Growth Formula*

Fitur formula pertumbuhan optimal merupakan fitur untuk membantu petani dalam menganalisis data nutrisi pada media tanaman, sehingga dapat membantu pengguna dalam mengambil keputusan terkait dengan perawatan pada tanaman.

5.3.4.1. Data Flow Model Fitur *Optimal Growth Formula*



Gambar 5. 5 Model Alir Data Fitur Formula Pertumbuhan

Model alir data pada gambar 5.5 menunjukkan bahawa aplikasi akan meminta sebuah data dari server. Data ini berupa data nutrisi tanaman dalam rentang waktu tertentu dan juga data secara realtime yang telah di-*cluster* dengan teknologi *machine learning* yang dalam hal ini hasil *clustering* berupa rekomendasi tindakan perawatan sesuai dengan kondisi media tanam. Data tersebut kemudian akan dikirim ke aplikasi yang kemudian akan ditampilkan dalam grafik. Selain itu data hasil rekomendasi juga akan ditampilkan ke pengguna agar dapat mengambil keputusan terkait dengan perawatan tumbuhan.

5.3.4.2. Implementasi Kode Fitur *Optimal Growth Formula*

Fitur ini bekerja dengan melakukan *request* data ke server, kemudian menampilkannya dalam sebuah grafik yang dapat mempermudah pengguna dalam melakukan analisis data.

```
suspend fun getNpkDataSensor(  
    authHeader: String,  
    startDate: String,  
    endDate: String,  
    timeRange: String = "DAILY",  
    sensor: String  
) : Response<NpkGraphicDayResponse> {
```

```

        return      ApiService.getNpkDataSensorDaily(authHeader,
startDate, endDate, timeRange, sensor)
    }

```

Dalam melakukan request pada server `startDate` dan `endDate` ditentukan yaitu dalam kurun waktu 10 hari terakhir. Data yang diambil adalah data dari npk (nitrogen, pospor, kalium).

```

suspend fun getRecommendation(): List<ClusteringResponse> {
    return ApiService.getRecommendations()
}

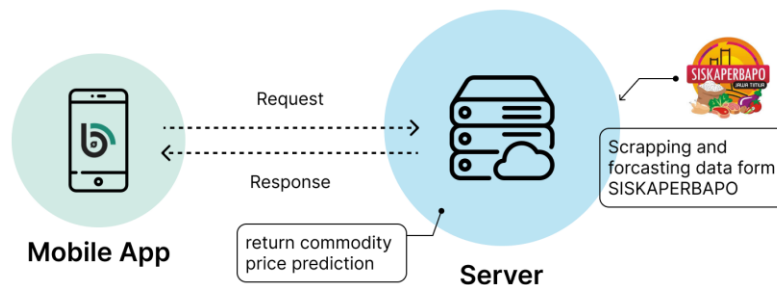
```

Kode di atas merupakan sebuah request ke *api* untuk mengambil data hasil clustering berupa rekomendasi perawatan berdasarkan kondisi green house.

5.3.5 Fitur *Commodity Price Prediction*

Fitur ini merupakan fitur untuk melihat harga komoditas sesuai dengan daerah dan komoditas yang dipilih. Cakupan data yang dilihat adalah dalam wilayah jawa timur karena data yang diambil server adalah dari Website SISKAPERBAPO. Selain itu fitur ini menyediakan prediksi terhadap harga dari beberapa komoditas dalam rentang waktu hingga 90 hari ke depan. Hal ini dapat menjadi pertimbangan pengguna dalam membuat keputusan komoditas apa yang ingin ditanam.

5.3.5.1. Data Flow Model Fitur *Commodity Price Prediction*



Gambar 5. 6 Model Alir Data Fitur Prediksi Harga

Model aliran data pada fitur ini digambarkan pada gambar di atas. Server akan melakukan *scrapping data* dari web SISKAPERBAPO yaitu website Pemerintahan Jawa Timur yang mencatat tiap harga komoditas di seluruh kabupaten dan kota di Jawa Timur. Data-data ini kemudian akan diproses oleh mesin learning yang ada di server sebagai proses *forecasting*. Kemudian hasil dari predisksi ini akan disimpan di database untuk dikirim ke aplikasi *mobile*.

5.3.5.2. Implementasi Kode Fitur *Commodity Price Prediction*

Ada dua function utama pada fitur ini, yaitu untuk mengambil data harga secara realtime dan juga mengambil data hasil prediksi oleh model *machine learning* yang ada di server.

```
suspend fun fetchCommodityPrice(commodityName: String,
marketName: String, date: String, city: String):
List<CommodityPriceResponseItem> {
    return apiService.getCommodityPrice(komoditas =
commodityName, pasar = marketName, tanggal = date, kabKota = city)
}
```

Kode di atas adalah untuk melakukan request data harga sesuai dengan tanggal yang ditentukan. Ada beberapa parameter seperti komoditas dan nama kota. Hasil *response* adalah berupa harga komoditas pada setiap pasar di kota atau kabupaten yang dipilih.

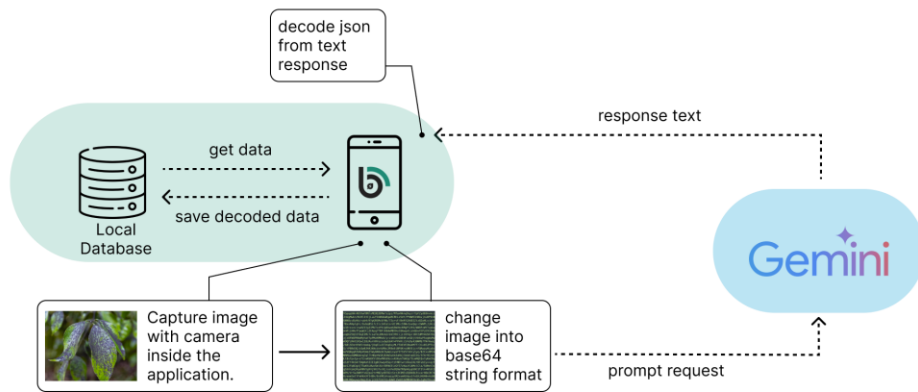
```
suspend fun fetchCommodityPredictions(date: String):
List<CommodityPredictionResponse> {
    return apiService.getCommodityPredictions(date)
}
```

Kode diatas adalah function utama untuk melakukan request pada data hasil prediksi, parameternya hanya ada tanggal dan akan diisi tanggal hari ini. *Response*-nya adalah hasil prediksi harga komoditas pada rentang waktu hingga 90 hari ke depan.

5.3.6 Fitur *Plant Disease Detection*

Fitur ini merupakan fitur yang dibangun dengan memanfaatkan model AI Gemini Flash 2.0. Fitur ini bekerja dengan mengirim sebuah gambar disertai dengan prompt yang telah ditentukan ke *Gemini API*.

5.3.6.1. Data Flow Model Fitur *Plant Disease Detection*



Gambar 5. 7 Model Alir Data Fitur Deteksi Penyakit Tanaman

Gambar 5.7 menjelaskan model aliran data fitur ini diawali dengan pengguna yang menangkap gambar tanaman menggunakan kamera di dalam aplikasi *mobile*. Gambar yang diperoleh kemudian dikonversi menjadi format string base64 untuk memudahkan pengiriman melalui jaringan. Selanjutnya, data gambar dalam format base64 dikirim sebagai permintaan (*prompt request*) ke layanan analisis berbasis AI Gemini. Gemini memproses data gambar tersebut dan mengirimkan respon berupa JSON dalam format JSON. Aplikasi *mobile* kemudian mendekode data JSON ini dan Data hasil deteksi tersebut kemudian ditampilkan kepada pengguna sebagai respons yang berisi informasi tentang kondisi dan penyakit tanaman yang terdeteksi. Aplikasi juga menyediakan *local database* untuk menyimpan hasil deteksi sebagai referensi atau riwayat deteksi.

5.3.6.2. Implementasi Kode Fitur *Plant Disease Detection*

Kode utama implementasi ini adalah *function* untuk mengirim *request* ke *Gemini API* dan meminta response dalam format JSON.

```
override suspend fun detectDisease(
    base64Image: String,
    prompt: String
): DiagnosisResult {
    return try {
        val request = GeminiRequest(
            contents = listOf(
                GeminiContent(
                    parts = listOf(
```

```

GeminiPart(text = prompt),
GeminiPart(inline_data
InlineData("image/jpeg", base64Image))
)
)
)
)
val response = api.detectDisease(
apiKey
"AIzaSyDYZHlMfOdFAcTFptWHqhPwIO734VdNUWE",
request = request
)

val rawJson
response.candidates.first().content.parts.first().text
val cleanedJson = rawJson
    .replace("`json", "")
    .replace("`", "")
    .trim()
try {
    val parsedSuccess =
json.decodeFromString<PlantDiseaseDetectionResponse>(cleanedJson
)
        DiagnosisResult.Success(parsedSuccess)
    } catch (e: Exception) {
        val parsedError =
json.decodeFromString<ErrorResponse>(cleanedJson)
        DiagnosisResult.Error(parsedError.error.toString())
    }
    } catch (e: Exception) {
        DiagnosisResult.Error("Terjadi kesalahan:
${e.localizedMessage ?: "tidak diketahui"}")
    }
}

```

Seperti yang terlihat pada kode di atas response dari Gemini akan di decode terlebih dahulu menjadi sebuah objek agar hasil nya bisa di tampilkan ke pengguna dengan lebih rapi. Selain itu agar hasilnya dapat disimpan dalam *database* lokal

menggunakan teknologi SQLite. Sedangkan prompt yang digunakan adalah sebagai berikut.

```
val prompt: String = """
Kamu adalah ahli tanaman dan agronomi profesional. Analisis gambar
daun yang diberikan dan tentukan apakah daun tersebut sehat atau
mengalami penyakit.
Jika daun sehat, balas dengan format JSON:
{"diagnosis":"Sehat"}
Jika daun sakit, balas dalam format JSON berikut:
{
  "diagnosis": "Nama penyakit tanaman",
  "cause": "Penyebab penyakit (misalnya jamur, bakteri, virus,
atau faktor lingkungan)",
  "description": "Deskripsi gejala pada daun secara visual",
  "treatment": [
    {
      "method": "Nama metode pengobatan",
      "description": "Penjelasan cara melakukannya"
    }
  ],
  "prevention": [
    "Langkah pencegahan 1",
    "Langkah pencegahan 2"
  ]
}

Jika gambar tidak menunjukkan daun tanaman, balas dalam format
JSON:
{"error":"Gambar tidak menunjukkan daun tanaman, mohon unggah
gambar daun yang jelas."}
Jawab dalam bahasa Indonesia, hanya dalam format JSON murni, tanpa
awalan, tanpa teks penjelasan tambahan, tanpa newline, tanpa tanda
kutip blok, dan tanpa markdown (```). Jawaban hanya boleh berupa
JSON valid.
""".trimIndent()
```

Prompt terdiri dari 3 hal utama yaitu perintah untuk analisis gambar, menentukan format *response* dan alternatif *response* jika analisis tidak sesuai.

5.4 Pengujian

Pada tahap ini, pengujian dilakukan untuk memastikan bahwa aplikasi yang telah dikembangkan dapat berjalan sesuai dengan fungsionalitas yang diharapkan dan dapat diterima dengan baik oleh target pengguna. Pengujian dibagi menjadi beberapa bagian, termasuk pengujian *black box* dan pengujian penerimaan pengguna (User Acceptance Testing).

5.4.1 Pengujian Black Box

Pengujian kotak hitam adalah teknik yang digunakan untuk mengidentifikasi kesalahan dalam fungsi, antarmuka, model informasi, dan akses ke sumber informasi luar dalam perangkat lunak (Mintarsih, 2023).

5.4.1.1. Pengujian *Dynamic Feature Module*

Tabel 5.1 dibawah ini memaparkan hasil uji coba aplikasi setelah diunggah ke Google Play Console. Pengujian ini dilakukan mulai dari mengunduh aplikasi melalui Google Play Store, kemudian mencoba mengunduh fitur-fitur yang sudah ditentukan untuk *on-demand*. berikut adalah hasilnya.

Tabel 5. 1 Tabel pengujian *dynamic feature module*

No	<i>Dynamic Feature</i>	<i>Download</i>	<i>Install</i>	<i>Feature Size (Mb)</i>
1	<i>Control Actuator feature</i>	<i>Download Success</i>	<i>Install Success</i>	1,46
2	<i>Growth Formula Feature</i>	<i>Download Success</i>	<i>Install Success</i>	1,22
3	<i>Commodity Price Prediction Feature</i>	<i>Download Success</i>	<i>Install Success</i>	3,31
4	<i>Plant Disease Detection Feature</i>	<i>Download Success</i>	<i>Install Success</i>	1,37

5.4.1.2. Pengujian Fitur

Pengujian kotak hitam pada fitur ini dilakukan dengan menentukan beberapa scenario penggunaan, kemudian akan dilakukan pengujian pada fitur sesuai dengan scenario yang ditentukan. Setelah pengujian dilakukan akan dicatat hasilnya sesuai

atau tidak sesuai harapan. Tabel 5.2 adalah paparan dari hasil uji coba fitur yang telah dilakukan.

Tabel 5. 2 Pengujian Fitur

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Status
1	Pengguna menginputkan Email dan Password yang telah terdaftar dan menekan tombol <i>login</i> .	Positif	Penguna berhasil masuk ke halaman utama	Berhasil
2	Pengguna menginputkan Email dan Password yang salah atau belum terdaftar dan menekan tombol <i>login</i> .	Negatif	Pengguna tidak dapat masuk dan muncul error di bawah tombol <i>login</i> .	Berhasil
3	Pengguna menekan ikon profil, menuju halaman profil dan menekan tombol <i>logout</i> untuk keluar.	Positif	Pengguna dapat keluar dari akunnya dan aplikasi menampilkan halaman <i>login</i>	Berhasil
4	Pengguna dapat melihat data sensor terkini dari halaman home screen.	Positif	Muncul <i>card</i> atau komponen UI yang menampilkan data NPK dan data DHT.	Berhasil
5	Pengguna menekan salah satu <i>card</i> data sensor dan diarahkan menuju ke halaman detail untuk melihat grafik.	Positif	Setelah <i>loading</i> selesai muncul data dalam bentuk grafik.	Berhasil
6	Pengguna memilih salah satu menu on-demand di	Positif	Muncul dialog progres	Berhasil

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Status
	halaman utama dan mengunduh fitur tersebut.		pengunduhan dan pemberitahuan jika selesai dinstal.	
7	Pengguna mengakses halaman kontrol aktuator dan menekan salah satu saklar yang ada.	Positif	Saklar berubah warna dan aktuator di <i>greenhouse</i> berhasil dinyalakan.	Berhasil
8	Pengguna mengakses halaman formula pertumbuhan dan melihat adanya pemberitahuan kondisi terkini <i>green house</i> .	Positif	Muncul grafik dari nutrisi tanah dan rekomendasi perawatan.	Berhasil
9	Pengguna menuju halaman harga komoditas dan memilih komoditas serta daerahnya untuk melihat harga saat ini.	Positif	Muncul daftar harga komoditas pada pasar-pasar di daerah yang dipilih.	Berhasil
10	Pengguna menuju halaman prediksi harga komoditas, dengan menekan tombol prediksi harga pada halaman harga komoditas.	Positif	Muncul prediksi harga komoditas yang dipilih pada kurun waktu besok, seminggu kemudian, 1 bulan kemudian, dan 3 bulan kemudian.	Berhasil
11.	Pengguna mengambil gambar tanaman melalui <i>camera screen</i> kemudian aplikasi akan mendeteksi	Positif	Halaman detail diagnosis menampilkan data penyakit tanaman.	Berhasil

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Status
	dan menampilkan hasil diagnosis pada halaman detail diagnosis.			
12	Pengguna menyimpan hasil deteksi penyakit tanaman dengan menekan tombol simpan dibagian paling bawah halaman hasil diagnosis.	Positif	Data tersimpan dan tampil di halaman riwayat deteksi.	Berhasil
13	Pengguna menekan ikon history di <i>camera screen</i> . Kemudian akan menuju ke halaman riwayat deteksi penyakit tanaman yang menampilkan daftar hasil diagnosis.	Positif	Halaman <i>history</i> menampilkan daftar diagnosis yang disimpan.	Berhasil

Berdasarkan hasil uji coba fitur ini, semua fitur dan skenario yang ditentukan berstatus berhasil. Hal ini menunjukkan bahwa sistem telah bekerja sesuai dengan kebutuhan fungsionalitasnya.

5.4.2 User Acceptance Testing

Pengujian Penerimaan Pengguna atau User Acceptance Testing (UAT) merupakan tahap pengujian akhir yang melibatkan pengguna langsung. Pengujian UAT dalam pengembangan perangkat lunak memastikan bahwa sistem memenuhi persyaratan fungsional yang ditentukan oleh pengguna (Aliyah Aliyah dkk., 2024).

5.4.2.1. Instrumen Penelitian

Instrumen utama yang digunakan untuk pengumpulan data UAT adalah kuesioner daring yang disusun menggunakan Google Forms. Kuesioner ini dirancang secara terstruktur berdasarkan kerangka yang telah divalidasi, mencakup 3 bagian utama seperti pada table 5.3 di bawah ini.

Tabel 5. 3 Tabel Pertanyaan UAT

Bagian	Pertanyaan	Jenis Jawaban
Bagian 1: Data diri dan pformulasi pengguna	Nama reponden?	Jawaban Singkat
	Usia Anda saat ini?	Pilihan Ganda
	Seberapa penting bagi anda ukuran awal aplikasi (size aplikasi) yang kecil saat mengunduh aplikasi baru?	Skala Linear (1-5)
	Apakah anda sering menunda atau membatalkan pengunduhan aplikasi karena ukuran aplikasi dirasa terlalu besar?	Skala Linear (1-5)
	Apakah Anda mempertimbangkan kapasitas penyimpanan (memori internal) perangkat Anda saat ingin mengunduh aplikasi?	Skala Linear (1-5)
	Seberapa familiar Anda dengan konsep atau penggunaan aplikasi smart farming?	Skala Linear (1-5)
	Fitur apa yang PALING Anda harapkan ada di sebuah aplikasi smart farming? (Anda bisa memilih maksimal 4)	Kotak centang
	Menurut Anda, apa tantangan paling mendesak dalam bidang pertanian saat ini di wilayah Anda? (Pilih maksimal 3 jawaban)	Kotak centang
Bagian 2:	Apakah Anda familiar dengan konsep aplikasi on-demand (Fitur diunduh sesuai kebutuhan)?	Pilihan Ganda

Bagian	Pertanyaan	Jenis Jawaban
Penggunaan Aplikasi Smartfarming Agrilink Vocpro	Seberapa mudah Anda menemukan dan mengunduh fitur tambahan (fitur on-demand) pada aplikasi ini?	Skala linear (1-5)
	Apakah fitur <i>Monitoring</i> berfungsi sesuai kebutuhan?	Skala linear (1-5)
	Apakah fitur <i>Controlling</i> berfungsi sesuai kebutuhan	Skala linear (1-5)
	Apakah fitur <i>Optimal Growth Formula</i> berfungsi sesuai kebutuhan	Skala linear (1-5)
	Apakah fitur <i>Price Prediction</i> berfungsi sesuai kebutuhan.	Skala linear (1-5)
	Apakah fitur <i>Plant Disease Detction</i> berfungsi sesuai kebutuhan	Skala linear (1-5)
	(Jika anda memilih skala 1 - 3, abaikan jika tidak) Bagian mana yang bermasalah? mohon deskripsikan!	Text jawaban panjang
Bagian 3: Evaluasi pengalaman pengguna (UI/UX}	Secara keseluruhan, apakah desain antarmuka (User Interface/UI) aplikasi ini menarik dan mudah dipahami?	Skala linear (1-5)
	Bagaimana kecepatan respons fitur-fitur yang Anda gunakan dalam aplikasi ini?	Skala linear (1-5)
	Apakah Anda menemukan bug (kesalahan/masalah teknis) selama menggunakan aplikasi ini?	Pilihan ganda
	Jika ya, mohon jelaskan bug yang Anda temukan (fitur apa, bagaimana terjadi, dll.	Text jawaban panjang

Bagian	Pertanyaan	Jenis Jawaban
	Apakah ada saran atau masukan untuk perbaikan fitur atau keseluruhan aplikasi ini	Text jawaban panjang

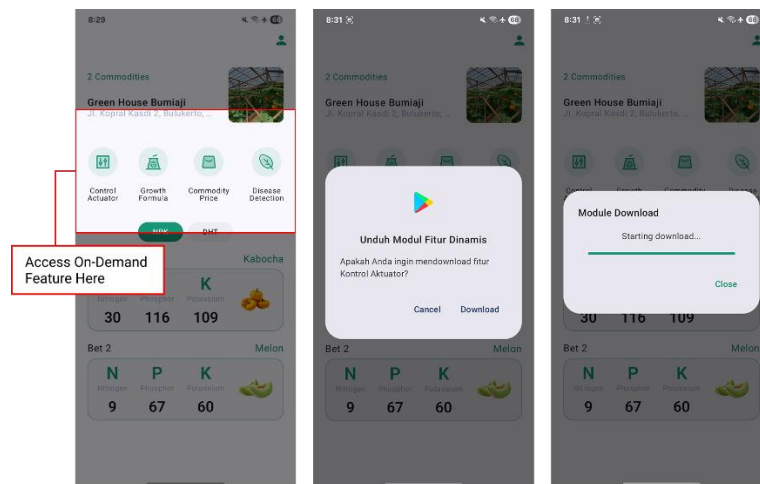
BAB VI HASIL DAN PEMBAHASAN

6.1 Hasil Implementasi Sistem

Pada bagian ini akan dijelaskan secara rinci mengenai implementasi fitur-fitur utama yang berhasil dikembangkan dalam aplikasi Smart Farming Agrilink Vocpro.

6.1.1 Download Fitur Secara On-Demand

Fitur *on-demand* dapat diakses melalui halaman utama. Ada 4 menu sebagai navigasi menuju ke fitur yang di unduh sesuai kebutuhan. Seperti yang terlihat pada gambar 6.1 jika sebuah fitur belum terunduh maka menekan menu fitur tersebut akan memunculkan konfirmasi pengunduhan fitur.

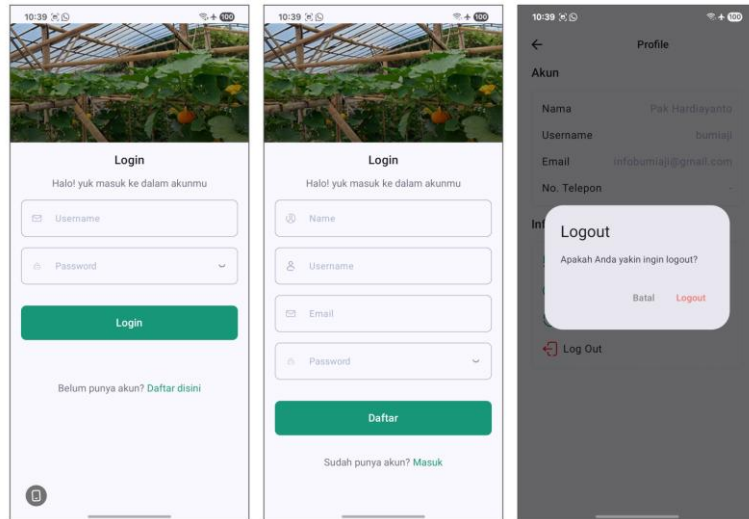


Gambar 6. 1 Screenshot pengunduhan fitur *on-demand*

Jika pengguna setuju untuk mengunduh fitur maka aplikasi akan menampilkan *progress bar* pengunduhan. Setelah selesai, maka ketika menu fitur yang diunduh di-*click* akan menuju ke halaman fitur tersebut.

6.1.2 Fitur Otentikasi

Fitur Otentikasi mencakup proses login menggunakan email dan *password*, pendaftaran akun (*register*) dengan mengisi nama, *username*, email, dan *password*, serta tampilan *profile screen* yang menampilkan data diri pengguna yang telah terdaftar, kecuali *password* demi keamanan sesuai dengan gambar 6.1. Pada layar profil juga tersedia tombol *logout* untuk keluar dari aplikasi.

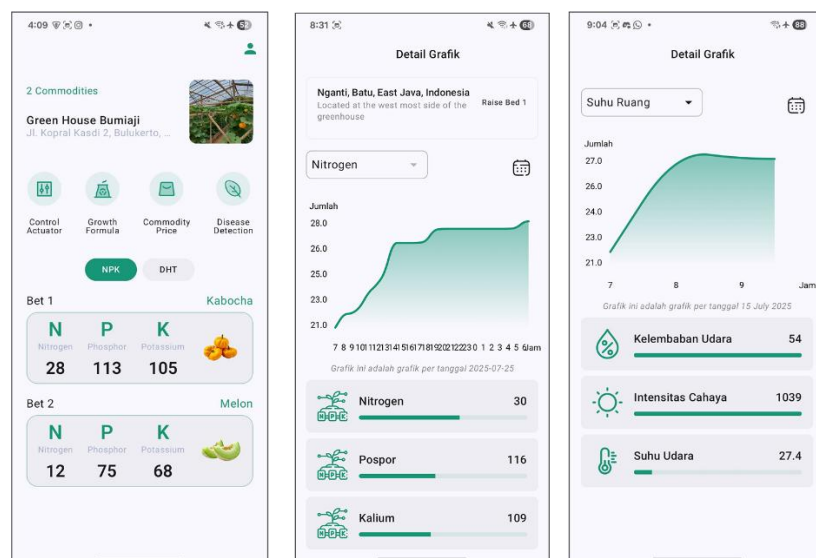


Gambar 6. 2 Screenshot Fitur Autentikasi

Fitur ini berhasil diimplementasikan dengan pengamanan yang cukup ketat pada proses verifikasi user dan pemberian sesi, sehingga pengguna hanya dapat mengakses data pribadi mereka serta fitur aplikasi dengan aman.

6.1.3 Fitur *Monitoring*

Fitur monitoring terletak pada halaman home yang menampilkan data NPK tanah dari dua *bed* (lahan tanam), dengan setiap *bed* memiliki satu sensor NPK. Selain itu juga terdapat satu sensor DHT yang digabung dengan sensor cahaya untuk mengukur kondisi udara.



Gambar 6. 3 Screenshot fitur Monitoring

Saat salah satu kartu sensor di-*click*, pengguna diarahkan ke halaman detail yang menampilkan grafik data rata-rata sensor per jam dalam kurun waktu satu hari.

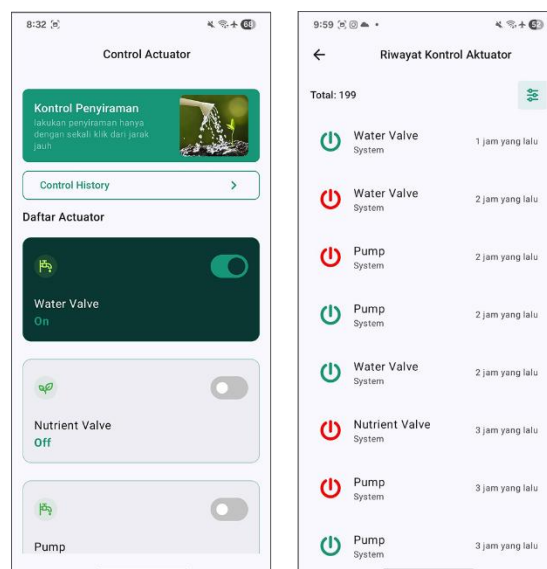
Fitur ini memberikan kemudahan bagi pengguna untuk memantau kondisi tanah dan lingkungan secara *real-time* dengan visualisasi data.

6.1.4 Fitur *Control Actuator*

Fitur pengendalian atau controlling merupakan fitur on-demand yang memungkinkan pengguna mengontrol tiga aktuator utama, yaitu:

1. Selenoid untuk penyiraman
2. Selenoid untuk pemupukan
3. Pompa air

Selain itu, terdapat riwayat kontrol aktuator yang memperlihatkan kapan suatu aktuator terakhir kali dinyalakan atau dimatikan.

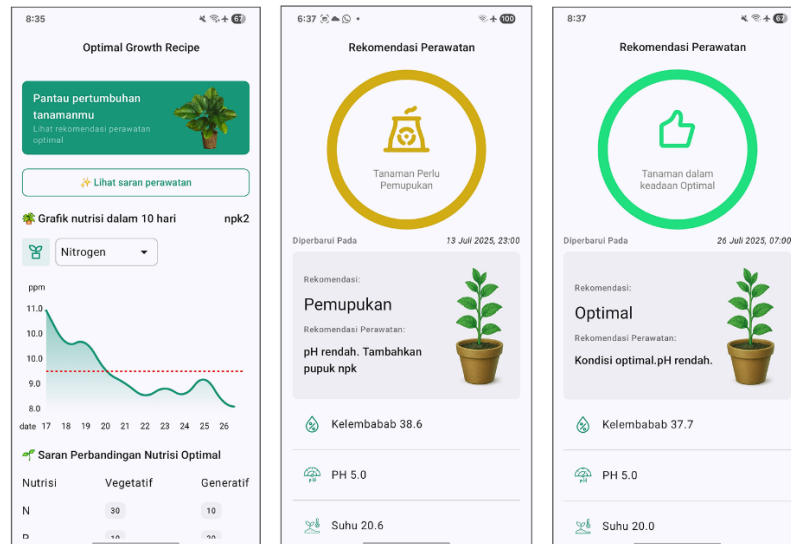


Gambar 6. 4 *Screenshot* Fitur *Control*

Pengguna dapat dengan mudah mengaktifkan atau menonaktifkan aktuator melalui aplikasi yang kemudian dikomunikasikan secara *real-time* ke perangkat IoT di *green house*.

6.1.5 Fitur *Optimal Growth Formula*

Fitur ini menampilkan grafik data sensor tanah (NPK) selama sepuluh hari terakhir, yang memberikan gambaran tren kondisi nutrisi tanah. Selanjutnya terdapat rekomendasi jumlah nutrisi ideal untuk tanah pada setiap fase pertumbuhan tanaman.

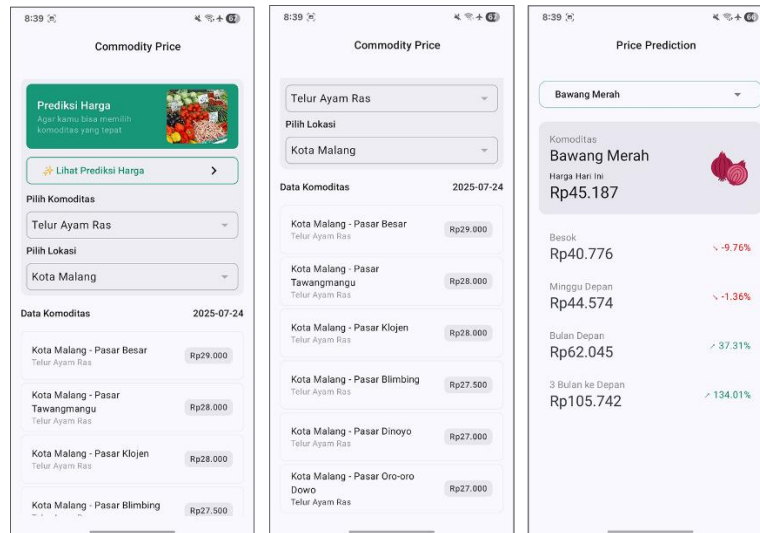


Gambar 6. 5 Screenshot fitur formula pertumbuhan optimal

Selain itu, fitur ini juga menampilkan hasil machine learning yang memberikan saran perawatan tanaman secara lebih optimal berdasarkan analisis data historis. Hal ini dapat mendukung keputusan petani dalam pemupukan dan perawatan tanah secara berbasis data. Namun, keputusan akhir terdapat pada petani itu sendiri.

6.1.6 Fitur *Commodity Price Prediction*

Fitur ini menyediakan prediksi harga beberapa komoditas yang berbeda untuk hari berikutnya, seminggu kemudian, 1 bulan kemudian dan 3 bulan kemudian. Tampilan menampilkan persentase kenaikan atau penurunan harga dari harga historis sebelumnya, serta opsi bagi pengguna untuk memilih komoditas yang ingin diprediksi.

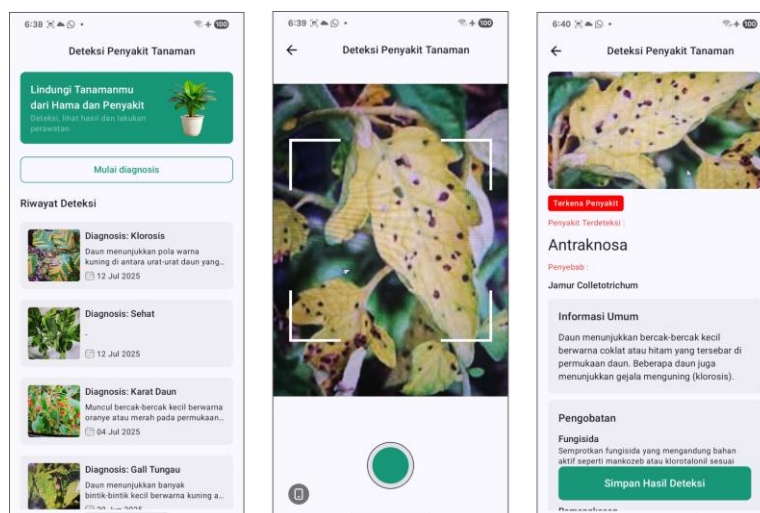


Gambar 6. 6 Screenshot fitur prediksi harga komoditas

Selain itu terdapat pemantauan harga komoditas secara realtime di seluruh pasar di daerah Jawa Timur. Dengan fitur ini, pengguna dapat terbantu dalam merencanakan penjualan produk secara lebih efisien sesuai dengan prediksi pasar.

6.1.7 Fitur *Plant Disease Detection*

Ketika fitur deteksi penyakit tanaman dibuka, secara langsung tampil data histori hasil diagnosis penyakit sebelumnya. Pengguna juga dapat mengambil gambar tanaman yang diduga sakit melalui tombol yang mengarah ke kamera aplikasi.



Gambar 6. 7 Screenshot fitur deteksi penyakit tanman

Setelah gambar diambil, aplikasi akan melakukan proses diagnosa dan menampilkan hasilnya berupa jenis penyakit tanaman. Pengguna dapat menyimpan

hasil diagnosis tersebut, yang nantinya akan tampil pada Riwayat diagnosis.

6.2 Hasil Pengujian Sistem

Pengujian telah dilakukan pada tahap sebelumnya. Menggunakan 2 metode pengujian yaitu Black Box testing untuk pengujian sistem dengan mengidentifikasi *bug* dengan lebih cepat. UAT digunakan untuk validasi sistem kepada pengguna secara langsung.

6.2.1 Hasil Black Box Testing

Black box testing yang dilakukan pada tahap sebelumnya menghasilkan persentase ukuran aplikasi pada saat peginstalan pertama dan setelah semua module diunduh. Aplikasi memiliki total ukuran sebesar 22,2 MB, yang terbagi menjadi beberapa modul, sebagai berikut:

Tabel 6. 1 Tabel ukuran AAB (Android App Bundle)

File	Size (Mb)	Download Size (Mb)	% Of Total Download Size
base	15,1	15,1	68%
commodity_price_prediction_feature	3,1	3,1	14,2%
control_feature	1,4	1,4	6,2%
plant_disease_detection_feature	1,3	1,3	6,1%
growth_recipe_feature	1,2	1,2	5,2%

Untuk menghitung efisiensi dari penggunaan *dynamic feature module* dalam konteks *initial download size*, rumus yang bisa digunakan:

Efisiensi Ukuran Awal (%)

$$= 1 - \left(\frac{\text{Ukuran Base}}{\text{Ukuran Total Semua Module Digabungkan}} \right) \times 100\%$$

Misalnya, jika seluruh fitur digabungkan dalam satu paket tanpa modularisasi, ukuran download awal adalah seluruh ukuran total, yakni 22,2 MB. Sementara pengguna baru hanya perlu mendownload modul dasar 15,1 MB. Maka efisiensi yang dihasilkan adalah:

$$= \left(1 - \frac{15,1}{22,2}\right) \times 100\% = 31,9\%$$

Artinya, dengan menggunakan *dynamic feature modules*, ukuran *download* awal aplikasi bisa dikurangi sekitar 31,9% dibandingkan jika seluruh fitur tersedia sekaligus dalam satu paket.

Berikut adalah rincian ukuran masing-masing fitur dalam arsitektur aplikasi setelah aplikasi diunduh dan diinstal pada perangkat.

Tabel 6. 2 Ukuran ril aplikasi setelah terinstal pada perangkat

Fitur	Jenis Modul	Ukuran Instalasi (MB)
<i>Monitoring & Autentikasi</i>	<i>Base Module</i>	35.59
Kontrol Aktuator	<i>Dynamic Feature</i>	1.58
Formula Pertumbuhan Tanaman	<i>Dynamic Feature</i>	1.32
Prediksi Harga Komoditas	<i>Dynamic Feature</i>	3.67
Deteksi Penyakit Tanaman	<i>Dynamic Feature</i>	3.04

Untuk mengilustrasikan efisiensi penggunaan sumber daya perangkat, dilakukan simulasi terhadap beberapa tipe pengguna (petani) berdasarkan kebutuhan fitur mereka:

Tabel 6. 3 Ilustrasi efisiensi ukuran berdasarkan kebutuhan pengguna

Tipe Petani	Fitur yang Digunakan	Total Ukuran	Persentase dari keseluruhan <i>size</i>
Petani dengan kebutuhan <i>monitor</i> nutrisi tanah	<i>Monitoring</i> (Base App)	35.59	78,7% (hemat 9.61 MB)
Pertani dengan kebutuhan monitor disertai dengan irigasi otomatis	<i>Monitoring, Actuator Control</i>	37.17	82,2% (hemat 8.03 MB)
Petani dengan kebutuhan	<i>Monitoring, Disease Detection</i>	38.63	91.5% (hemat 3.57 MB)

Tipe Petani	Fitur yang Digunakan	Total Ukuran	Persentase dari keseluruhan size
menaggulangi penyakit tanaman			
Petani dengan kebutuhan pemantauan harga pasar.	<i>Monitor, Commodity Price</i>	39.26	93% (hemat 2.94 MB)
Petani dengan kebutuhan menyeluruh	<i>Monitoring, Control, Growth Formula, Commodity Price, Disease Detection</i>	45.20	100% (full install)

Selain itu *black-box testing* juga dilakukan untuk menguji semua fitur dengan case yang sudah dilampirkan pada tabel 5.2. Hasilnya, semua scenario pengujian berhasil dilalui dengan tingkat keberhasilan 100%. Hal ini Mengindikasikan bahwa tidak ada bug atau error yang terjadi pada saat pengujian Black-box dan aplikasi yang dibangun telah memenuhi kebutuhan fungsionalitasnya.

6.2.2 Hasil UAT

UAT telah dilakukan dengan menyebarkan survei kepada responden dan meminta mereka mencoba menggunakan aplikasi yang telah dibuat. Survei tersebut dapat menjadi tolak ukur hasil dari penelitian ini. Dalam beberapa pertanyaan yang menggunakan skala likert akan diukur pada rumus berikut:

$$Mean = \frac{\sum (Skor \times Frekuensi)}{\text{Jumlah total responden}}$$

Skor adalah nilai skalanya sedangkan frekuensi adalah jumlah jawaban pada skala tersebut. Mean adalah nilai rata-rata skor yang diperoleh secara umum dari hasil survei dengan interpretasi nilai mean sebagai berikut:

Tabel 6. 4 Interpretasi nilai mean

Skala	Skor	Interpretasi
1	1.00 - 1.80	Sangat buruk
2	1.81 - 2.60	Buruk

Skala	Skor	Interpretasi
3	2.61 - 3.40	Sedang
4	3.41 - 4.20	Baik
5	4.21 - 5.00	Sangat bagus

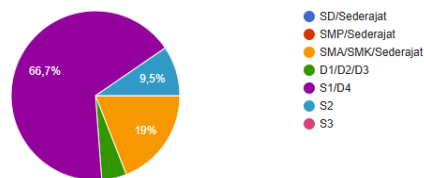
6.2.2.1. Data Responden

Berdasarkan data yang dikumpulkan dari 21 responden, dapat dilihat bahwa mayoritas responden berada pada rentang usia 18–24 tahun. Berikut adalah distribusi usia responden:

Tabel 6. 5 Data usia responden UAT

Rentang Usia	Jumlah Responden	Persentase
18–24 tahun	19 responden	90.5%
25–34 tahun	2 responden	9.5%

Pendidikan terakhir
21 jawaban



Gambar 6. 8 Data pendidikan responden UAT

Berdasarkan grafik pie yang ditampilkan pada gambar 6.8, latar belakang pendidikan terakhir responden terdiri dari beberapa jenjang. Adapun hasil distribusinya adalah sebagai berikut:

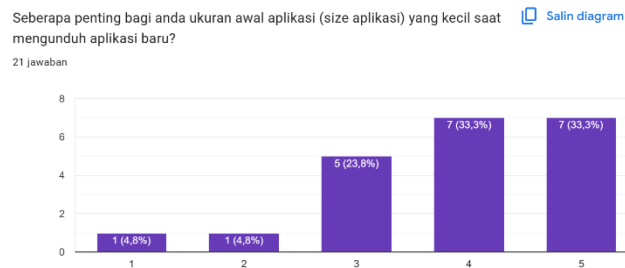
Tabel 6. 6 Distribusi pendidikan responden UAT

Pendidikan Terakhir	Persentase	Jumlah Responden (Estimasi)
SD/Sederajat	9.5%	0 orang
SMP/Sederajat	19%	0 orang
SMA/SMK/Sederajat	4.8%	4 orang
D1/D2/D3	0%	1 orang
S1/D4	66.7%	14 orang
S2	0%	2 orang

S3	0%	0 orang
----	----	---------

6.2.2.2. Persepsi Pentingnya Ukuran Awal Aplikasi

Mayoritas responden menganggap ukuran awal aplikasi yang kecil sangat penting saat mengunduh aplikasi baru.



Gambar 6. 9 Hasil survei pentingnya ukuran aplikasi

Dari 21 responden, 33,3% memberikan nilai tertinggi (skor 5) dan jumlah yang sama juga memilih skor 4, menunjukkan kepentingan tinggi terhadap ukuran aplikasi. Sekitar 23,8% menilai cukup penting (skor 3), sementara hanya sedikit yang menganggapnya kurang penting. Rata-rata skor yang diperoleh adalah 3,86, menunjukkan bahwa ukuran aplikasi merupakan faktor signifikan dalam keputusan pengguna mengunduh aplikasi.



Gambar 6. 10 Hasil survei seringnya pembatalan pengunduhan

Selain itu, sebanyak 76,2% responden mengaku pernah "kadang-kadang" menunda atau membatalkan pengunduhan aplikasi karena ukuran yang dirasa terlalu besar, sedangkan 14,3% menyatakan sering mengalami hal tersebut. Hanya 9,5% yang tidak pernah mengalami masalah ini, memperlihatkan bahwa ukuran aplikasi berpengaruh kuat terhadap perilaku pengguna.



Gambar 6. 11 Hasil survei pertimbangan penyimpanan perangkat

Survei juga menunjukkan bahwa pengguna mempertimbangkan kapasitas penyimpanan perangkat saat mengunduh aplikasi, dengan 33,3% selalu memperhatikan dan 61,9% kadang-kadang memperhitungkannya. Temuan ini menegaskan pentingnya pengoptimalan ukuran aplikasi agar dapat menarik lebih banyak pengguna dan mengurangi hambatan dalam proses instalasi.

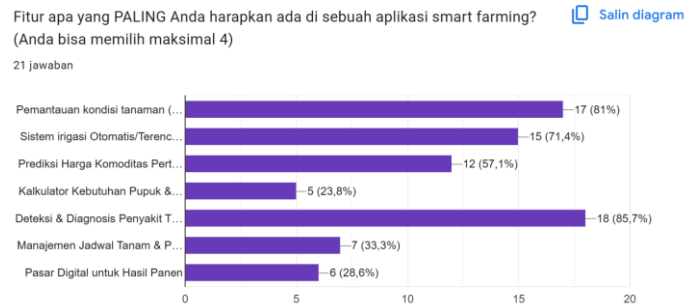
6.2.2.3. Persepsi Terhadap Aplikasi Smart Farming

Berdasarkan hasil survei tingkat familiaritas dengan konsep atau penggunaan aplikasi smart farming menunjukkan variasi, dengan mayoritas responden memilih skor tinggi.



Gambar 6. 12 Hasil survei familiaritas terhadap aplikasi *smart farming*

Sebanyak 28,6% responden menyatakan sangat familiar (skor 5) dan 28,6% lainnya cukup familiar (skor 4), sementara tingkat ketidaktahuan atau ketidakfamiliaran relatif rendah, menunjukkan pengetahuan yang cukup baik mengenai aplikasi smart farming di kalangan responden.



Gambar 6. 13 Hasil survei harapan fitur aplikasi *smart farming*

Terkait fitur yang paling diharapkan, mayoritas responden memilih fitur deteksi dan diagnosis penyakit tanaman (85,7%) dan pemantauan kondisi tanaman secara real-time (81%) sebagai fitur utama. Fitur lain yang juga banyak diminati adalah sistem irigasi otomatis atau terencana (71,4%) dan prediksi harga komoditas pertanian (57,1%). Fitur seperti kalkulator kebutuhan pupuk & pestisida serta manajemen jadwal tanam dan panen memiliki tingkat harapan yang lebih rendah.



Gambar 6. 14 Hasil survei tantangan dalam pertanian saat ini

Dalam hal tantangan di bidang pertanian, sebagian besar responden mengidentifikasi kurangnya akses informasi dan teknologi (81%) sebagai tantangan paling mendesak saat ini. Tantangan lain yang cukup signifikan meliputi harga jual komoditas yang tidak stabil (61,9%) dan serangan hama serta penyakit tanaman (61,9%). Perubahan iklim dan cuaca ekstrem juga menjadi perhatian penting (42,9%). Keterbatasan modal dan akses pembiayaan dianggap sebagai tantangan yang relatif kecil oleh sebagian besar responden (4,8%).

6.2.2.4. Penilaian Terhadap Aplikasi Agrilink Vocpro

Berdasarkan hasil survei, mayoritas responden menilai bahwa menemukan dan mengunduh fitur tambahan (on-demand) dalam aplikasi Agrilink Vocpro

sangat mudah, dengan 76,2% memberikan skor tertinggi dan sisanya memberikan skor yang juga tinggi.



Gambar 6. 15 Hasil survei kemudahan pengunduhan fitur *on-demand*

Pengguna juga memberikan penilaian positif terhadap kesesuaian fitur-fitur utama aplikasi berjalan sesuai dengan kebutuhan pertanian.

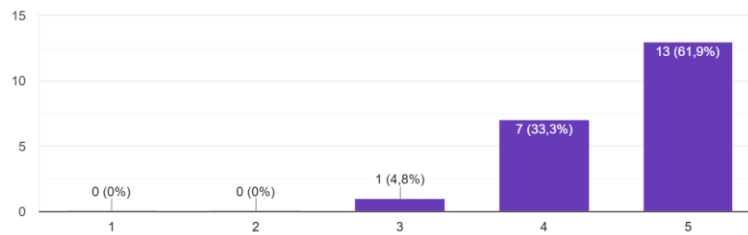
Tabel 6. 7 Hasil survei kesesuaian fitur

Fitur	Skala					Skor Mean
	5	4	3	2	1	
Monitoring	16	5	0	0	0	4,76
Controlling	14	7	0	0	0	4,66
Growth Formula	13	8	0	0	0	4,6
Commodity Price Prediction	14	2	3	2	0	4,3
Plant Disease Detection	18	3	0	0	0	4,85

Fitur Monitoring yang 76,2% menilainya sangat sesuai, serta fitur Controlling dan Growth Formula yang menerima nilai tinggi dari mayoritas responden. Fitur Prediksi Harga Komoditas menunjukkan variasi penilaian yang lebih beragam, namun mayoritas tetap merasa fitur ini memadai, sementara fitur Deteksi Penyakit Tanaman mendapat nilai sangat baik dari hampir semua responden.

Secara keseluruhan, apakah desain antarmuka (User Interface/UI) aplikasi ini menarik dan mudah dipahami?

21 jawaban



Gambar 6. 16 Hasil survei penilaian UI

Dari segi desain antarmuka pengguna, aplikasi ini dianggap menarik dan mudah dipahami oleh sebagian besar pengguna dengan 61,9% memberikan skor tertinggi. Kecepatan respons fitur juga dinilai cepat dan memuaskan, dengan mayoritas responden memberikan nilai 4 dan 5. Selain itu, seluruh responden menyatakan tidak menemukan bug atau masalah teknis selama penggunaan aplikasi, yang menunjukkan tingkat stabilitas dan kualitas aplikasi yang baik secara keseluruhan.

6.3 Pembahasan

Hasil survei dan evaluasi aplikasi Agrilink Vocpro menunjukkan bahwa penerimaan pengguna terhadap aplikasi ini secara umum sangat positif. Pengguna menilai kemudahan dalam menemukan dan mengunduh fitur tambahan on-demand sebagai kelebihan utama aplikasi. Hal ini penting mengingat fitur on-demand memungkinkan pengguna untuk menyesuaikan aplikasi sesuai dengan kebutuhan spesifik tanpa harus mengunduh seluruh modul, sehingga meningkatkan fleksibilitas penggunaan dan efisiensi penyimpanan pada perangkat. Selain itu, hampir semua fitur utama seperti monitoring, controlling, growth Formula, dan plant disease detection mendapat penilaian yang sangat baik dari responden, mencerminkan bahwa aplikasi sudah mampu memenuhi kebutuhan pengguna di bidang pertanian secara efektif.

Penggunaan dynamic feature modules memungkinkan aplikasi dikemas dalam satu base module kecil yang berisi fitur inti, sementara fitur tambahan yang tidak selalu diperlukan oleh semua pengguna ditempatkan dalam modul yang bisa diunduh secara on-demand. Strategi ini sangat berguna untuk mempercepat proses instalasi dan memperkecil penggunaan ruang penyimpanan awal. Berdasarkan hasil

penelitian, fitur on-demand berhasil menekan ukuran aplikasi pada *initial download size* sebesar 32%. Meskipun hasilnya tidak signifikan mengingat pengembangan ini masih dilakukan pada tahap awal, hal ini tidak menutup kemungkinan arsitektur ini berkerja signifikan jika aplikasi berkembang.

Desain antarmuka pengguna yang menarik dan mudah dipahami juga menjadi aspek yang sangat diapresiasi, yang berkontribusi pada pengalaman pengguna yang lebih baik dan meningkatkan kemungkinan penggunaan aplikasi secara rutin. Kecepatan respon fitur yang cepat tanpa ditemukannya bug juga menunjukkan kestabilan dan kualitas teknis aplikasi yang baik. Meski demikian, fitur prediksi harga komoditas mendapat penilaian beragam dari pengguna, menunjukkan masih adanya ruang untuk peningkatan agar fitur ini dapat memberikan nilai tambah yang lebih optimal. Secara keseluruhan, hasil ini mengindikasikan bahwa Agrilink Vocpro berhasil menghadirkan solusi aplikasi smart farming yang *user-friendly* dan sesuai kebutuhan praktis para petani di wilayah cakupan penelitian.

BAB VII KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan hasil penelitian dan survei yang dilakukan, dapat disimpulkan bahwa aplikasi Agrilink Vocpro berhasil diimplementasikan dengan baik dan diterima positif oleh pengguna. Aplikasi ini menyediakan fitur-fitur utama yang sesuai dengan kebutuhan smart farming, termasuk monitoring kondisi tanaman, sistem pengendalian irigasi, formula pertumbuhan, deteksi penyakit tanaman, serta evaluasi harga komoditas. Tingkat kepuasan terhadap fitur-fitur tersebut umumnya berada pada skala tinggi, mencerminkan keberhasilan aplikasi dalam memenuhi ekspektasi pengguna. Selain fitur yang memadai, aplikasi ini juga memiliki antarmuka yang menarik dan mudah digunakan, dengan kecepatan respon fitur yang cepat tanpa ditemukannya masalah serius selama pengujian pengguna. Kemudahan akses fitur tambahan berbasis on-demand menambah nilai fleksibilitas penggunaan aplikasi, sekaligus menjaga ukuran aplikasi agar tetap efisien di perangkat pengguna. Selain itu dari penerapan fitur on-demand dengan arsitektur DFM ini ada beberapa manfaat yang di peroleh antara lain:

- a. Memperkecil ukuran pengunduhan aplikasi di awal.
- b. Karena ukurannya lebih kecil maka secara langsung dapat mempersingkat waktu pengunduhan aplikasi di awal.
- c. Arsitektur DFM sangat membantu *developer* dalam mengembangkan aplikasi ini karena fitur dibangun secara independen, sehingga dapat mempermudah kolaborasi jika bekerja dalam sebuah tim.
- d. Kelebihan lain dari arsitektur DFM, yaitu setiap fitur on-demand dapat mengakses sumber daya dari *base module* atau package utama sehingga ketika ada komponen berulang *developer* tidak perlu membangun lagi komponen tersebut, hanya tinggal mengakses dari modul utama.

Keseluruhan hasil ini menegaskan bahwa Agrilink Vocpro merupakan aplikasi smart farming yang potensial untuk mendukung produktivitas pertanian melalui teknologi digital yang handal.

7.2 Saran

Berdasarkan temuan dan pengalaman selama pengembangan dan pengujian aplikasi, rekomendasi berikut diusulkan untuk peningkatan di masa mendatang:

a. Implementasi Mekanisme Penghapusan Fitur

Kembangkan mekanisme yang memungkinkan pengguna menghapus atau mencopot pemasangan modul sesuai permintaan yang telah mereka unduh. Fitur ini akan memungkinkan pengguna mengosongkan ruang penyimpanan di perangkat mereka, sehingga meningkatkan kegunaan dan manajemen perangkat.

b. Peningkatan Navigasi Tingkat Fitur

Peningkatkan sistem navigasi agar lebih dinamis dengan mengaktifkan komunikasi langsung antar-modul di antara fitur-fitur sesuai permintaan, alih-alih mengarahkan semua komunikasi melalui modul inti utama atau *base module*. Hal ini akan meningkatkan responsivitas dan pengalaman pengguna saat beralih atau mengintegrasikan fungsionalitas antar-modul.

c. Penyempurnaan dan Optimalisasi Fitur Berkelanjutan

Lakukan peningkatan dan optimalisasi berkelanjutan terhadap fitur-fitur yang ada berdasarkan masukan pengguna dan pengujian pengguna tambahan. Pembaruan rutin harus berfokus pada peningkatan kinerja, kegunaan, dan pengintegrasian teknologi baru agar aplikasi tetap relevan dan efektif dalam ranah pertanian cerdas.

DAFTAR PUSTAKA

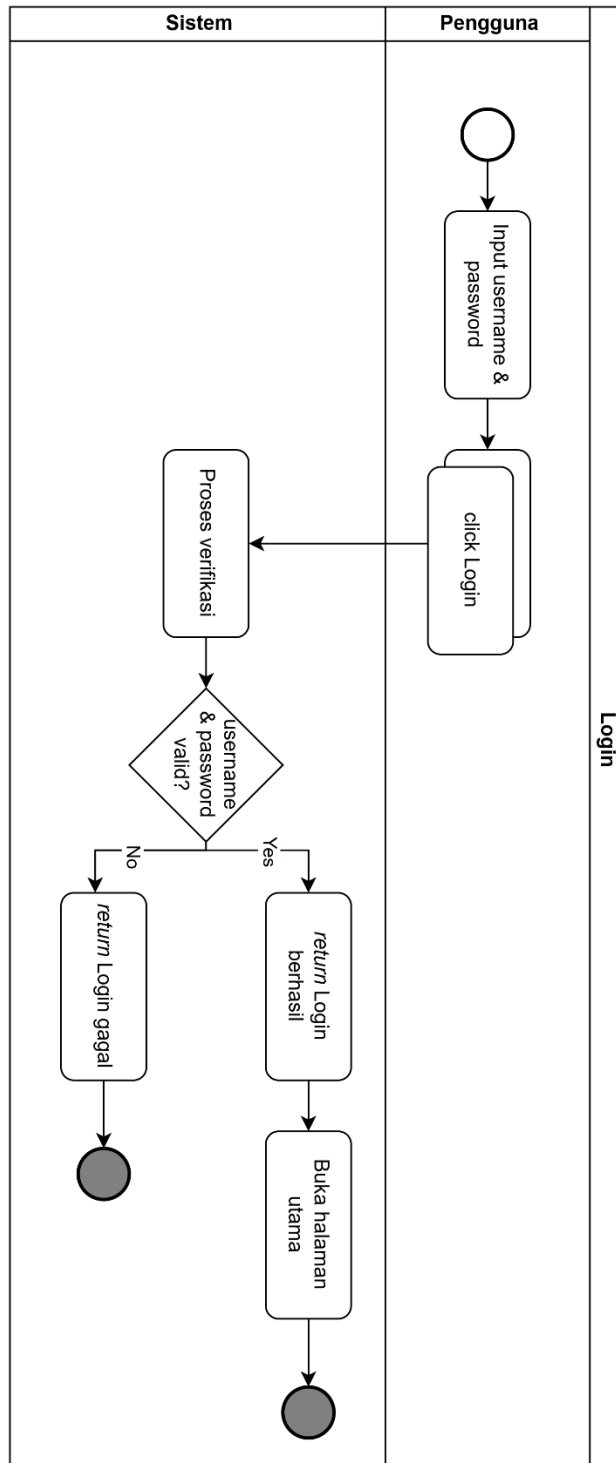
- Aliyah Aliyah, Nahrun Hartono, & Asrul Azhari Muin. (2024). Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. *Switch: Jurnal Sains dan Teknologi Informasi*, 3(1), 84–100. <https://doi.org/10.62951/switch.v3i1.330>
- B. Prashanth Kumar, & Y. Prashanth. (2014). Improving the Rapid Application Development Process Model. *2014 Conference on IT in Business, Industry and Government (CSIBIG)*, 170. 10.1109/CSIBIG.2014.7056962
- Darmawan LS. (2023, Maret 28). *Implementing Dynamic Feature Modules in our Android App*. <https://medium.com/julotech/implementing-dynamic-feature-modules-in-our-android-app-e9c7aa5db3e8>.
- Dewi, H. S., Kurniawan, A., Yoga Prakoso, B., Budilaksono, S., & Putri, M. A. (2022). SMART FARMING TEKNOLOGI MONITORING PRODUKSI DAN PEMASARAN KEBUN ORGANIK. *Jusikom : Jurnal Sistem Komputer Musirawas*, 7(1). <https://doi.org/10.32767/jusikom.v7i1.1587>
- Easttom, C. (2021). Android Operating System. Dalam *An In-Depth Guide to Mobile Device Forensics* (hlm. 23).
- Ghorbani, N., Jabbarvand, R., Salehnamadi, N., Garcia, J., & Malek, S. (2023). DeltaDroid: Dynamic Delivery Testing in Android. *ACM Transactions on Software Engineering and Methodology*, 32(4). <https://doi.org/10.1145/3563213>
- Inwood, S. E., & Dale, V. (2019). State of apps targeting management for sustainability of agricultural landscapes. A review. *Agronomy for Sustainable Development*, 39. <https://doi.org/10.1007/s13593-018-0549-8>
- Kautz, O., Rumpe, B., & Wachtmeister, L. (2022). Semantic Differencing of Use Case Diagrams. *Journal of Object Technology*, 21(3). <https://doi.org/10.5381/jot.2022.21.3.a5>
- Liu, Y., Ma, Y., Xiao, X., Xie, T., & Liu, X. (2023). LegoDroid: flexible Android app decomposition and instant installation. *Science China Information Sciences*, 66(4), 142103. <https://doi.org/10.1007/s11432-021-3528-7>

- Mintarsih, M. (2023). Pengujian Black Box Dengan Teknik Transition Pada Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Waterfall Pada SMC Foundation. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 5(1), 33–35. <https://doi.org/10.47233/jteksis.v5i1.727>
- Nalendra, A. K. (2021). Rapid Application Development (RAD) model method for creating an agricultural irrigation system based on internet of things. *IOP Conference Series: Materials Science and Engineering*, 1098(2), 022103. <https://doi.org/10.1088/1757-899x/1098/2/022103>
- Putu, I., Elang, G., Kurnia Sudaryana, E., Agung, A., & Ekayana, G. (2022). *RANCANG BANGUN SISTEM SMART FARMING BERBASIS IoT STUDI KASUS KEBUN NYOMAN GUMITIR*. <https://ejournal.catuspata.com/index.php/jkdn/index>
- Rachmawati, R. R. (2021). SMART FARMING 4.0 UNTUK MEWUJUDKAN PERTANIAN INDONESIA MAJU, MANDIRI, DAN MODERN. *Forum penelitian Agro Ekonomi*, 38(2), 137. <https://doi.org/10.21082/fae.v38n2.2020.137-154>
- Sheldon R. (2023, Juni). *What is Google Play?* <https://www.techtarget.com/searchmobilecomputing/definition/Google-Play-Android-Market>.
- Shubha Ramachandran, G. Sathish Kumar, Madhu K. Iyengar, K. Rangarajan, & Saravanan Sivagnanam. (2009). Modularization of a Large-Scale Business Application. *Santonu Sarkar, Accenture Technology Labs*, 28–25. <https://ieeexplore.ieee.org/document/4786949>
- Ulfada, E., Nurfiana, N., & Handayani, R. D. (2022). *Perancangan DesaiN UI/UX Pada Implementasi Sistem Kontrol Smart Farming Berbasis Internet of Things (IoT)*. <https://otomasi.sv.ugm.ac.id/2018/06/02>
- Uniyal, S. P., Joshi, K., Singh, V. K., Aggarwal, A., Chhabra, G., & Kumar, A. (2023). Comparative Analysis of App Size Variations between React Native and Apache Cordova Powered Android Applications. *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 1697–1702. <https://doi.org/10.1109/ICAISS58487.2023.10250551>

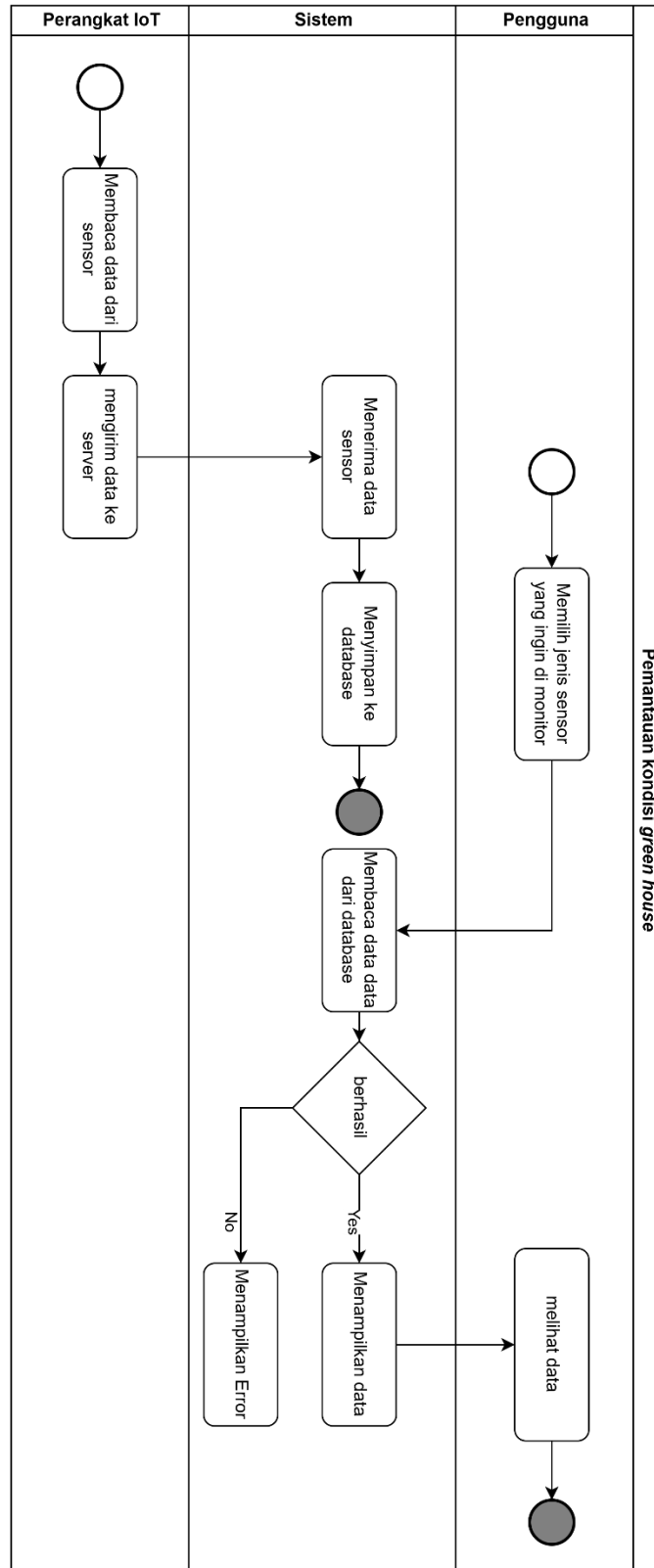
- .Venkadesh, Dr. P. (2025). Digital Agriculture: Leveraging Mobile Applications. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*. <https://doi.org/10.55041/ijsrem42704>
- Yuniarti, A., Siahaan, D., & Tendean, S. (2024, Oktober 9). Modularization in Object Oriented Software: A Comparative Study. *2024 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD)*.
- Zneika, M., Loulou, H., Houacine, F., & Bouzefrane, S. (2013). Towards a modular and lightweight model for android development platforms. *Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCOM* 2013, 2129–2132. <https://doi.org/10.1109/GreenCom-iThings-CPSCOM.2013.400>

LAMPIRAN-LAMPIRAN

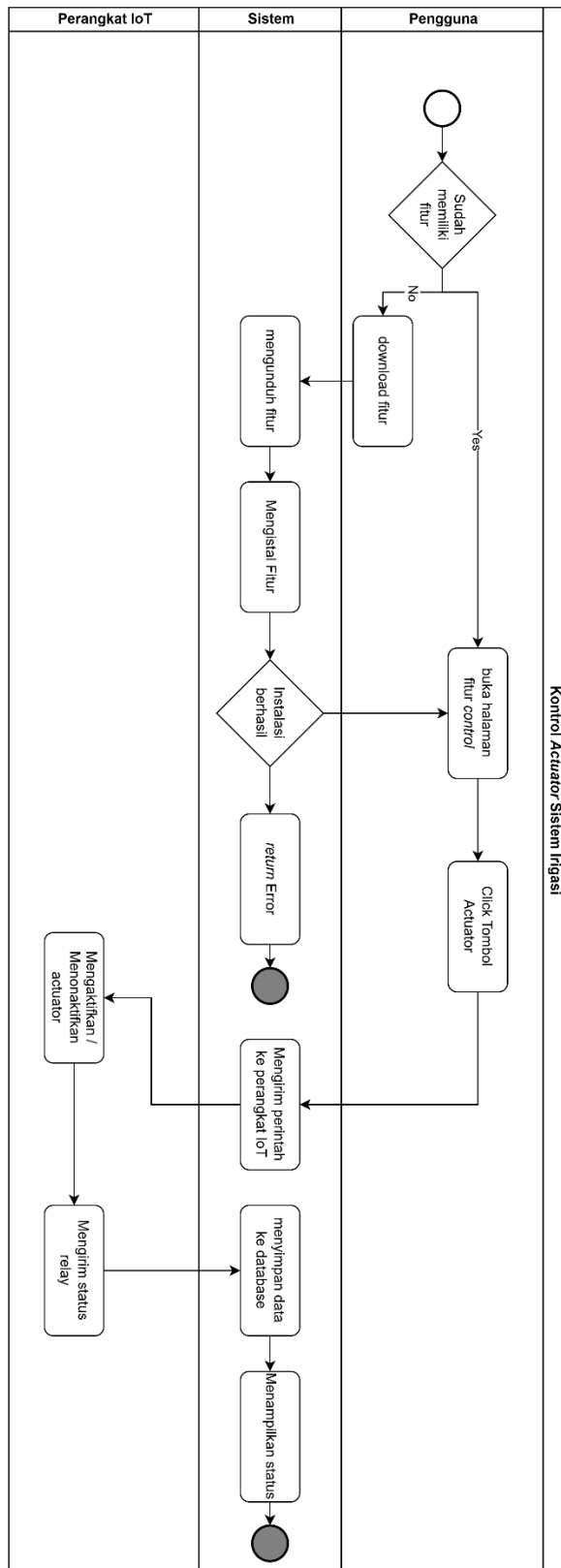
Lampiran 1. Business Flow Login



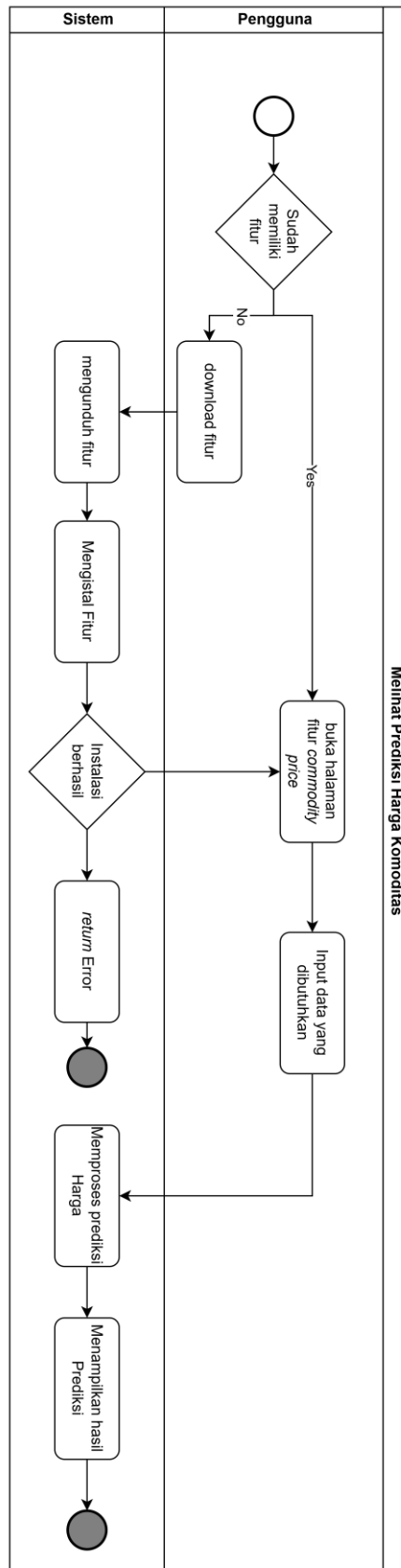
Lampiran 2. Business Flow Pemantauan Kondisi Green House



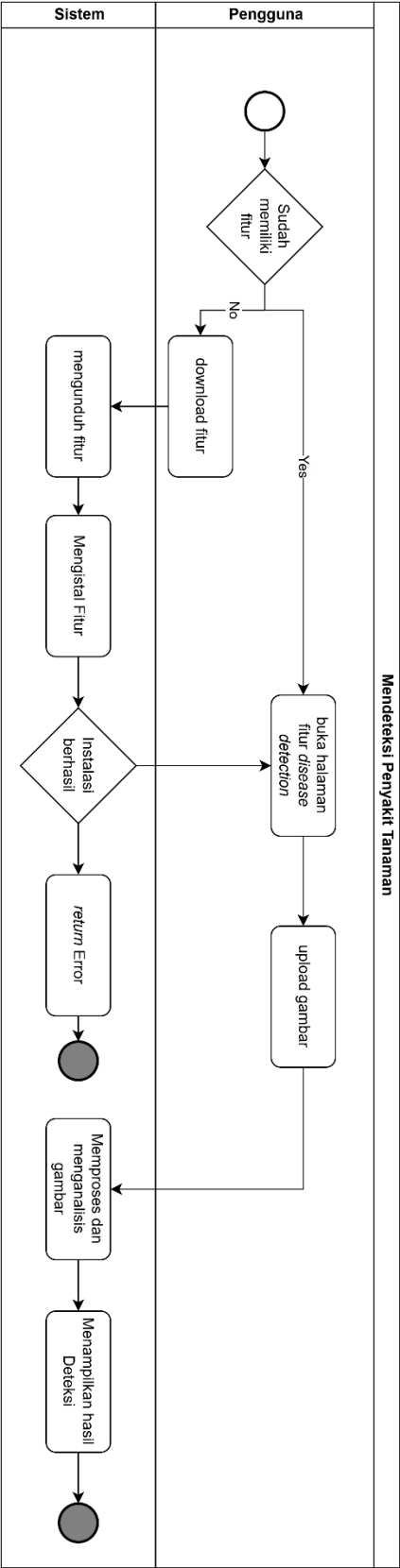
Lampiran 3. Business Flow Kontrol Sistem Irigasi Green House



Lampiran 4. Business Flow Prediksi Harga Komoditas



Lampiran 5. Business Flow Deteksi Penyakit Tanaman



Lampiran 6. Business Flow formula Pertumbuhan Optimal

