

**PENGEMBANGAN *DASHBOARD SMART FARMING*
GREENHOUSE KEBUN RAYA ITERA BERBASIS
PROGRESSIVE WEB APP DENGAN METODE AGILE SCRUM
(Studi Kasus Greenhouse ITERA)**

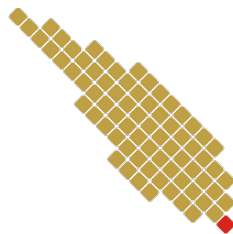
TUGAS AKHIR

Diajukan sebagai syarat menyelesaikan jenjang strata Satu (S-1) di
Program Studi Teknik Informatika, Fakultas Teknologi Industri, Institut
Teknologi Sumatera

Oleh:

Muhammad Yusuf

122140193



ITERA

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SUMATERA
LAMPUNG SELATAN
2025**

LEMBAR PENGESAHAN

Saya menyatakan bahwa Tugas Akhir berjudul “PENGEMBANGAN *DASHBOARD SMART FARMING GREENHOUSE* KEBUN RAYA ITERA BERBASIS *PROGRESSIVE WEB APP* DENGAN METODE AGILE SCRUM” merupakan hasil karya saya sendiri dan belum pernah diajukan, baik sebagian maupun seluruhnya, di Institut Teknologi Sumatera atau institusi pendidikan lain oleh saya maupun pihak lain.

Lampung Selatan, 20-09-2025
Penulis,

Muhammad Yusuf
NIM. 122140193

Foto 2x3

Diperiksa dan disetujui oleh,
Pembimbing

1. Muhammad Habib Algifari, S.Kom., M.T.I.
NIP. 19910525 2022 03 1 002

.....

2. Dosen Pembimbing I
NIP.19900000 2000 00 0 000

.....

Penguji

1. Dosen Penguji I
NIP.19900000 2000 00 0 000
2. Dosen Penguji II
NIP.19900000 2000 00 0 000

.....

.....

Disahkan oleh,
Koordinator Program Studi Teknik Informatika
Fakultas Teknologi Industri
Institut Teknologi Sumatera

Andika Setiawan, S.Kom., M.Cs.
NIP. 19911127 2022 03 1 008

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir dengan judul “**PENGEMBANGAN *DASHBOARD SMART FARMING GREENHOUSE* KEBUN RAYA ITERA BERBASIS *PROGRESSIVE WEB APP* DENGAN METODE AGILE SCRUM**” adalah karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan benar.

Nama : Muhammad Yusuf

NIM : 122140193

Tanda Tangan :

Tanggal :

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Institut Teknologi Sumatera, saya yang bertanda tangan di bawah ini:

Nama : Muhammad Yusuf

NIM : 122140193

Program Studi : Teknik Informatika

Fakultas : Teknologi Industri

Jenis Karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Sumatera **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

PENGEMBANGAN *DASHBOARD SMART FARMING GREENHOUSE* KEBUN RAYA ITERA BERBASIS *PROGRESSIVE WEB APP* DENGAN METODE AGILE SCRUM

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Sumatera berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Lampung Selatan

Pada tanggal : 20-09-2025

Yang menyatakan

Muhammad Yusuf

KATA PENGANTAR

Pada halaman ini mahasiswa berkesempatan untuk menyatakan terima kasih secara tertulis kepada pembimbing dan pihak lain yang telah memberi bimbingan, nasihat, saran dan kritik, kepada mereka yang telah membantu melakukan penelitian, kepada perorangan atau lembaga yang telah memberi bantuan keuangan, materi dan/atau sarana. Cara menulis kata pengantar beraneka ragam, tetapi hendaknya menggunakan kalimat yang baku. Ucapan terima kasih agar dibuat tidak berlebihan dan dibatasi pada pihak yang terkait secara ilmiah (berhubungan dengan subjek/materi penelitian).

Puji syukur kehadiran Allah SWT/Tuhan Yang Maha Esa atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga penyusunan tugas akhir ini telah terselesaikan dengan baik. Dalam penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. [Rektor ITERA] selaku Rektor Institut Teknologi Sumatera.
2. [Dekan FTI] selaku Dekan Fakultas Teknologi Industri.
3. [Koor Prodi IF] selaku Ketua Program Studi Teknik Informatika.
4. [Dosen Pembimbing] selaku Dosen Pembimbing atas ide, waktu, tenaga, perhatian, dan masukan yang telah disumbangsihkan kepada penulis.
5. [Isi nama lainnya]

Akhir kata penulis berharap semoga tugas akhir ini dapat memberikan manfaat bagi kita semua.

RINGKASAN

PENGEMBANGAN *DASHBOARD SMART FARMING GREENHOUSE* KEBUN RAYA ITERA BERBASIS *PROGRESSIVE WEB APP* DENGAN METODE AGILE SCRUM

Muhammad Yusuf

Halaman Ringkasan berisi uraian singkat tentang latar belakang masalah, rumusan masalah, tujuan, metodologi penelitian, hasil dan analisis data, serta kesimpulan dan saran. Isi ringkasan tidak lebih dari 1000 kata (sekitar maksimal 2 halaman).

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

ABSTRAK

PENGEMBANGAN *DASHBOARD SMART FARMING GREENHOUSE* KEBUN RAYA ITERA BERBASIS *PROGRESSIVE WEB APP* DENGAN METODE AGILE SCRUM

Muhammad Yusuf

Halaman ABSTRAK berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INDONESIA tidak lebih dari 250 kata, dengan jarak antar baris satu spasi. Pada akhir abstrak ditulis kata “Kata Kunci” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Kata kunci terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Kata kunci diketik pada jarak dua spasi dari baris akhir isi abstrak.

Kata Kunci: kunci1, kunci2

ABSTRACT

DEVELOPMENT OF A SMART FARMING GREENHOUSE DASHBOARD AT KEBUN RAYA ITERA GARDEN BASED ON PROGRESSIVE WEB APP USING AGILE SCRUM METHOD

Muhammad Yusuf

Halaman ABSTRACT berisi uraian tentang latar belakang, tujuan, metodologi penelitian, hasil / kesimpulan. Ditulis dalam BAHASA INGGRIS tidak lebih dari 250 kata, dengan jarak antar baris satu spasi. Secara khusus, kata dan kalimat pada halaman ini tidak perlu ditulis dengan huruf miring meskipun menggunakan Bahasa Inggris, kecuali terdapat huruf asing lain yang ditulis dengan huruf miring (misalnya huruf Latin atau Greek, dll). Pada akhir abstract ditulis kata “Keywords” yang dicetak tebal, diikuti tanda titik dua dan kata kunci yang tidak lebih dari 5 kata. Keywords terdiri dari kata-kata yang khusus menunjukkan dan berkaitan dengan bahan yang diteliti, metode/instrumen yang digunakan, topik penelitian. Keywords diketik pada jarak dua spasi dari baris akhir isi abstrak.

Keywords: keywords1, keywords2

DAFTAR ISI

LEMBAR PENGESAHAN	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PERSETUJUAN PUBLIKASI	iv
KATA PENGANTAR	v
RINGKASAN	vi
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR RUMUS	xiii
DAFTAR KODE	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Tujuan Penelitian	4
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	5
1.6 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA	8
2.1 Tinjauan Pustaka	8
2.2 Dasar Teori	13

BAB III METODE PENELITIAN	20
3.1 Alur Penelitian	20
3.2 Penjabaran Langkah Penelitian	21
3.2.1 Langkah 1: Mulai	21
3.2.2 Langkah 2: Studi Literatur	21
3.2.3 Langkah 3: Requirements	21
3.2.4 Langkah 4: Tahapan Metodologi Agile SCRUM.....	21
3.2.5 Langkah 5: Analisis dan Pembahasan	22
3.2.6 Langkah 6: Kesimpulan dan Saran.....	22
3.2.7 Langkah 7: Selesai	22
3.3 Alat dan Bahan Tugas Akhir.....	22
3.3.1 Alat.....	23
3.3.2 Bahan.....	23
3.4 Metode Pengembangan	24
3.4.1 Requirements	25
3.4.1.1 Rancangan Kebutuhan Fungsional.....	25
3.4.1.2 Rancangan Kebutuhan Non-Fungsional	26
3.4.1.3 Use Case Diagram	27
3.4.2 Pemodelan Sistem Menggunakan UML.....	30
3.4.2.1 Data Flow Diagram (DFD)	30
3.4.2.2 Activity Diagram	33
3.4.2.3 Sequence Diagram.....	34
3.4.2.4 Perancangan Basis Data (ERD)	35
3.4.3 Perencanaan Menggunakan Metode Agile Scrum	37
3.4.3.1 Planning	37
3.4.3.2 Design	38
3.4.3.3 Coding	43
3.4.3.4 Evaluate	43
3.4.3.5 Release Backlog.....	44

3.5	Ilustrasi Perhitungan Metode	44
3.6	Rancangan Pengujian	46
BAB IV HASIL DAN PEMBAHASAN		49
4.1	Hasil Penelitian	49
4.2	Hasil Pengujian	49
4.3	Analisis Hasil Penelitian	50
BAB V KESIMPULAN DAN SARAN		52
5.1	Kesimpulan	52
5.2	Saran	52
DAFTAR PUSTAKA		53
LAMPIRAN		55
A	Dataset	55
B	Hasil Wawancara	55
C	Rincian Kasus Uji	55

DAFTAR TABEL

Tabel 2.1	Literasi Penelitian Terdahulu	10
Tabel 3.1	Hasil Wawancara dengan Stakeholder Berdasarkan Peran <i>Agile Scrum</i>	25
Tabel 3.2	Daftar Kebutuhan Fungsional Sistem	26
Tabel 3.3	Daftar Kebutuhan Non-Fungsional Sistem	27
Tabel 3.4	Struktur Tabel Basis Data Utama Sistem	36
Tabel 3.5	Perencanaan Sprint Berdasarkan Fitur Pengembangan	38
Tabel 3.6	Rincian Coding Berdasarkan Sprint	43
Tabel 3.7	Release Backlog Berdasarkan Iterasi Sprint	44
Tabel 3.8	Skala Likert untuk Penilaian SUS	46
Tabel 3.9	Daftar 10 Pernyataan Umum SUS	46
Tabel 3.10	Rekap Skor Blackbox per Sprint	47
Tabel 3.11	Rekap Skor SUS per Sprint	48
Tabel 4.1	Data <i>dummy</i> Pengujian	50

DAFTAR GAMBAR

Gambar 3.1	Alur Penelitian	20
Gambar 3.2	<i>Agile SCRUM Methodology</i>	24
Gambar 3.3	Diagram Use Case untuk Aktor Admin	28
Gambar 3.4	Diagram Use Case untuk Aktor Pengelola	28
Gambar 3.5	Diagram Use Case untuk Aktor Staff	29
Gambar 3.6	Data Flow Diagram Level 0 Sistem Smart Farming	31
Gambar 3.7	Data Flow Diagram Level 1 Sistem Smart Farming	32
Gambar 3.8	Data Flow Diagram Level 2 – Proses Otomatisasi Kontrol Perangkat	33
Gambar 3.9	Activity Diagram Proses Login dan Akses Dashboard...	34
Gambar 3.10	Sequence Diagram untuk Proses Otomatisasi Aktuator Berdasarkan Sensor	35
Gambar 3.11	Diagram <i>Entity Relationship Diagram</i> (ERD) Sistem Dashboard <i>Smart Farming</i>	36
Gambar 3.12	Tampilan Halaman Login Sistem Dashboard <i>Smart Farming</i>	39
Gambar 3.13	Tampilan Awal Dashboard Sistem <i>Smart Farming</i>	39
Gambar 3.14	Tampilan Daftar <i>Greenhouse</i> dengan Informasi Lokasi ..	40
Gambar 3.15	Tampilan Daftar <i>Reservoir</i> dengan Informasi Volume ...	41
Gambar 3.16	Tampilan Daftar <i>Robot</i> dengan Informasi Status dan Baterai	41
Gambar 3.17	Tampilan Halaman Manajemen Pengguna Sistem <i>Smart Farming</i>	42
Gambar 4.1	Contoh Graf Pengujian	50

DAFTAR RUMUS

Rumus 3.1	Perhitungan Persentase Keberhasilan Blackbox Testing ..	44
Rumus 3.2	Contoh Hasil Perhitungan Blackbox Testing	45
Rumus 3.3	Perhitungan Skor Akhir System Usability Scale (SUS) ..	45
Rumus 3.4	Jumlah Total Skor Konversi SUS	45
Rumus 3.5	Skor Akhir SUS dari Total Skor Konversi	46
Rumus 3.6	Persentase Keberhasilan Pengujian Blackbox per Sprint .	47
Rumus 3.7	Rata-rata Tingkat Keberhasilan Blackbox Testing	47
Rumus 3.8	Perhitungan Skor SUS untuk Setiap Sprint	47
Rumus 3.9	Rata-rata Skor Usability dari Evaluasi SUS	48

DAFTAR KODE

Kode 4.1 Akuisisi Gambar	49
--------------------------------	----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sektor pertanian Indonesia berkontribusi signifikan terhadap Produk Domestik Bruto (PDB). Data PDB sektor pertanian Triwulan III 2024 menunjukkan pertumbuhan 1,69%, mencerminkan potensi sektor ini yang terus berkembang [1]. Meskipun demikian, sektor ini masih menghadapi berbagai tantangan, salah satunya adalah dalam budidaya tanaman buah dan sayuran yang memerlukan perlakuan khusus. Oleh karena itu, pengelolaan dan pemantauan yang efisien menjadi kunci untuk memaksimalkan potensi sektor ini, terutama dalam menghadapi kondisi lingkungan yang tidak selalu stabil.

Badan Pusat Statistik (BPS) mencatat peningkatan produksi melon nasional dari 117.793,81 ton pada tahun 2023 menjadi 125.383,77 ton pada tahun 2024 [2] [3]. Peningkatan ini mencerminkan adanya perkembangan dalam sektor budidaya melon, meskipun sektor ini juga dihadapkan pada tantangan dalam menjaga kestabilan produksi. Melon, sebagai tanaman dengan potensi ekonomi tinggi, memerlukan pengelolaan lingkungan yang ketat, termasuk kontrol suhu, kelembaban, dan sistem irigasi yang harus dijaga agar tetap konsisten. Tanaman melon yang memiliki sensitivitas tinggi terhadap kondisi lingkungan ini, menjadikannya salah satu komoditas pertanian yang paling menantang dalam hal pengelolaan dan pemantauan.

Sektor pertanian Indonesia berkontribusi signifikan terhadap Produk Domestik Bruto (PDB). Data PDB sektor pertanian Triwulan III 2024 menunjukkan pertumbuhan 1,69%, mencerminkan potensi sektor ini yang terus berkembang [1]. Meskipun demikian, sektor ini menghadapi tantangan, terutama dalam budidaya tanaman buah dan sayuran yang memerlukan perlakuan khusus, seperti melon. Melon memiliki potensi ekonomi tinggi namun membutuhkan

pengelolaan yang kompleks, dipengaruhi oleh faktor lingkungan seperti suhu, kelembaban, dan sistem irigasi yang harus dijaga konsisten.

Badan Pusat Statistik (BPS) mencatat fluktuasi produksi melon nasional, dari 117.793,81 ton pada tahun 2023 menjadi 125.383,77 ton pada tahun 2024 [2] [3]. Fluktuasi ini menunjukkan ketidakstabilan dalam sistem budidaya yang membutuhkan pengelolaan lingkungan yang teliti. Kontrol suhu dan kelembaban yang ketat menjadikan melon salah satu tanaman paling menantang dalam sektor pertanian semusim. Hal ini menuntut teknologi yang dapat memantau dan mengelola lingkungan tumbuh dengan efisien dan responsif, guna mencegah kerugian ekonomi akibat gagal panen atau penurunan kualitas buah.

Kebun Raya ITERA memiliki sistem pemantauan *real-time* untuk pengelolaan greenhouse melon, namun sistem tersebut masih menghadapi masalah efisiensi komunikasi data. Metode *polling* yang digunakan dengan loop berkelanjutan menyebabkan konsumsi bandwidth dan resource server yang tinggi. Selain itu, *polling* juga mengakibatkan delay dalam pengiriman data dan membebani sistem meskipun tidak ada perubahan data yang signifikan. Sistem ini juga belum memiliki *dashboard* penjadwalan otomatis atau otomatisasi aktuatur berdasarkan nilai sensor untuk kegiatan perawatan seperti penyiraman, pemberian nutrisi, dan pengaturan suhu serta kelembaban.

Solusi untuk masalah ini adalah implementasi teknologi *WebSocket* dan *Progressive Web App (PWA)*. *WebSocket* memungkinkan komunikasi dua arah secara *real-time* dengan konsumsi resource yang lebih rendah dibandingkan metode *polling*, mengurangi beban server dan delay pengiriman data. Penggunaan *PWA* untuk *dashboard* memberikan fleksibilitas akses melalui perangkat apapun, baik smartphone maupun desktop, dengan kemampuan offline yang mendukung operasional lapangan. Integrasi sistem penjadwalan otomatis dan otomatisasi aktuatur berbasis sensor akan mempermudah pengelolaan kegiatan greenhouse, meningkatkan efisiensi operasional.

Integrasi sensor *IoT* juga penting untuk pengambilan keputusan berbasis

data. Sensor suhu, kelembaban udara, pH tanah, dan kelembaban tanah akan mengumpulkan data secara kontinu, menyediakan informasi yang diperlukan untuk pengaturan lingkungan yang lebih responsif. Data ini digunakan untuk pengaturan otomatis suhu dan kelembaban, meminimalkan kesalahan manusia dan memastikan kondisi optimal bagi tanaman melon.

Penelitian ini menggunakan metodologi *Agile Scrum* untuk pengembangan sistem dalam siklus sprint pendek dengan evaluasi dan validasi berkelanjutan pada setiap iterasi. Pengujian dilakukan dengan *Black-box Testing*, *White-box Testing*, dan pengukuran *usability* menggunakan UMUX untuk memastikan sistem memenuhi kebutuhan pengguna. Tujuan utama pengembangan ini adalah meningkatkan *usability scale* dari 75.875 Nugraha2023 menjadi 90 atau lebih, untuk meningkatkan kepuasan pengguna dan efektivitas sistem di lapangan. Implementasi *WebSocket* dan sistem penjadwalan diharapkan dapat meningkatkan efisiensi operasional greenhouse hingga 40% dibandingkan sistem *polling* yang ada. Dampak yang diharapkan adalah peningkatan produksi melon yang lebih berkualitas, efisien, dan konsisten, serta kontribusi terhadap pengembangan teknologi pertanian modern di Indonesia.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang ada, maka rumusan masalah dalam penelitian ini dapat dijabarkan sebagai berikut:

1. Bagaimana merancang sistem *dashboard* berbasis *Progressive Web App (PWA)* yang dapat mendukung pemantauan kondisi lingkungan *greenhouse* secara *real-time* di ITERA?
2. Bagaimana mengukur tingkat fungsionalitas, *usability*, dan kepuasan pengguna terhadap *dashboard* berbasis *PWA* yang dikembangkan dalam konteks *smart farming greenhouse*?
3. Sejauh mana *dashboard* berbasis *PWA* yang dikembangkan dapat meningkatkan efisiensi operasional dan efektivitas pengelolaan lingkungan

greenhouse di ITERA?

1.3 Tujuan Penelitian

Berdasarkan latar belakang dan rumusan masalah, tujuan dari penelitian ini adalah sebagai berikut:

1. Mengembangkan sistem *dashboard* berbasis *Progressive Web App (PWA)* yang mendukung pemantauan *real-time* kondisi lingkungan *greenhouse* di ITERA.
2. Mengintegrasikan sensor *Internet of Things (IoT)* untuk mengumpulkan data lingkungan secara akurat dan berkelanjutan serta menyajikannya pada *dashboard* untuk mendukung pengambilan keputusan berbasis data.
3. Mengoptimalkan efisiensi operasional dan pengelolaan lingkungan *greenhouse* melalui sistem penjadwalan otomatis dan kontrol berbasis data sensor pada *dashboard*.
4. Menerapkan metodologi *Agile Scrum* untuk pengembangan sistem secara iteratif dan responsif terhadap kebutuhan pengguna.
5. Mengevaluasi sistem melalui pengujian fungsionalitas menggunakan *Black-box Testing* serta pengukuran *usability* menggunakan *Usability Metric for User Experience (UMUX)* untuk memastikan efektivitas dan kualitas sistem yang dikembangkan.
6. Mengembangkan sistem *smart farming* berbasis teknologi informasi yang dapat diintegrasikan dan digunakan di lingkungan pendidikan tinggi untuk mendukung kegiatan akademis dan penelitian.

1.4 Batasan Masalah

Agar ruang lingkup penelitian ini tetap terfokus dan dapat diselesaikan dalam periode kerja praktik dan tugas akhir, maka batasan-batasan dalam pengembangan sistem adalah sebagai berikut:

1. Fitur visualisasi citra dan pengolahan data gambar dari robot tidak

- diterapkan. Sistem hanya menerima dan menyimpan data berbasis *string*, angka, atau *JSON* dari robot melalui komunikasi *publish/subscribe*.
2. Kontrol perangkat otomatis dan penjadwalan alat (kipas, penyiram, pompa) hanya mengatur berdasarkan nilai ambang sensor dan waktu atau tombol *on/off* yang ditentukan pengguna, tanpa menggunakan *machine learning* atau *AI*.
 3. Pengembangan *dashboard* difokuskan pada platform *PWA* untuk website, serta perangkat *desktop* dan *mobile/ios*, tanpa distribusi melalui *Google Play Store* atau *Apple App Store*.
 4. Metodologi pengembangan menggunakan *Agile Scrum* dengan empat siklus *sprint*, dan dokumentasi *backlog* sesuai kebutuhan pengguna di *ITERA*.
 5. Integrasi sensor *IoT* dan robot terbatas pada lingkungan *greenhouse* *ITERA* dengan koneksi *WiFi*, tanpa mencakup produksi massal atau komersialisasi.

1.5 Manfaat Penelitian

Manfaat tugas akhir ini didefinisikan sebagai hasil yang diperoleh setelah sistem selesai dikembangkan dan diimplementasikan. Manfaat mencakup aspek teoritis dan praktis, serta berdampak pada mahasiswa, program studi Teknik Informatika, *ITERA*, masyarakat, dunia akademik, dan dunia penelitian.

1. Manfaat Teoritis

1. Menyediakan model penerapan *Internet of Things (IoT)* dan analitik data untuk pemantauan lingkungan *greenhouse* yang dapat dijadikan referensi penelitian lanjutan.
2. Memperkaya literatur akademik terkait transformasi digital sektor pertanian melalui pendekatan *Progressive Web App (PWA)*.
3. Menjadi acuan sistematis dalam penggunaan metode pengembangan

perangkat lunak berbasis *Agile Scrum* di bidang teknologi pertanian.

4. Mendorong integrasi multidisiplin antara informatika, pertanian, dan manajemen data lingkungan dalam pengembangan sistem cerdas berbasis data.

2. Manfaat Praktis

1. Menyediakan sarana pembelajaran dan praktikum berbasis data yang interaktif dan kontekstual untuk mahasiswa.
2. Menjadi portofolio pengembangan berbasis proyek yang relevan dengan kebutuhan nyata di lingkungan kampus untuk program studi Teknik Informatika ITERA.
3. Mendukung transformasi digital ITERA dalam pengelolaan *greenhouse* serta membuka jalan untuk pengembangan sistem otomatisasi lainnya.
4. Memberikan solusi *monitoring* pertanian yang murah, fleksibel, dan dapat dikembangkan secara lokal untuk masyarakat.
5. Menjadi contoh penerapan konsep *smart farming* yang dapat direplikasi oleh institusi pendidikan tinggi lain untuk dunia akademik.
6. Menjadi studi awal dalam pengembangan sistem pertanian berbasis data, integrasi *IoT*, dan visualisasi data dalam dunia penelitian.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun untuk memberikan gambaran umum mengenai struktur laporan tugas akhir. Sistematika penulisan dalam penelitian ini adalah sebagai berikut:

BAB I

Bab ini menjelaskan latar belakang penelitian, rumusan masalah, tujuan, batasan masalah, serta manfaat penelitian. Selain itu, bab ini juga menguraikan sistematika penulisan sebagai panduan dalam penyusunan laporan.

BAB II

Bab ini berisi tinjauan pustaka yang mencakup kajian teori dan penelitian terdahulu yang mendukung pengembangan *dashboard smart farming* berbasis *Progressive Web App (PWA)*. Pembahasan teori meliputi *Internet of Things (IoT)*, pengembangan *PWA*, metodologi *Agile Scrum*, serta metode pengujian *black-box testing*, *white-box testing* dan evaluasi *usability* menggunakan *Usability Metric for User Experience (UMUX)*.

BAB III

Bab ini menjelaskan metodologi penelitian yang digunakan, termasuk proses pengembangan sistem berbasis *Agile Scrum*, integrasi sensor *IoT*, serta strategi pengujian fungsionalitas, pengukuran *usability*, kepuasan pengguna, dan efektifitas sistem terhadap produksi melon.

BAB IV

Bab ini memaparkan hasil pengembangan sistem, mulai dari desain antarmuka hingga integrasi sensor *IoT* dan pengujian sistem. Pengujian sistem dilakukan melalui tiga proses utama, yaitu *black-box testing*, *white-box testing*, dan pengukuran *usability* menggunakan *UMUX*, yang dilakukan pada setiap iterasi pengembangan.

BAB V

Bab ini berisi kesimpulan dari penelitian yang dilakukan, serta saran untuk pengembangan sistem lebih lanjut dan implementasi di skala yang lebih luas.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Penelitian terkait pengembangan sistem informasi berbasis web dan *Internet of Things (IoT)* telah dilakukan dalam berbagai domain. Penelitian oleh Yaswanth et al. [4] mengembangkan sistem *smart greenhouse* berbasis IoT yang mampu memantau suhu dan kelembapan secara *real-time* serta dilengkapi dengan prediksi cuaca berbasis analitik data. Sistem tersebut memberikan kontribusi signifikan dalam penerapan teknologi pertanian presisi melalui integrasi perangkat sensor dan dashboard berbasis web. Namun, pendekatan pengembangan yang digunakan belum mengadopsi paradigma berorientasi pengguna seperti *usability testing*, sehingga menimbulkan keterbatasan dalam aspek pengalaman pengguna. Metodologi yang digunakan bersifat eksperimental dengan pemanfaatan perangkat *NodeMCU* dan sensor lingkungan sebagai komponen utama.

Penelitian dari Taufik et al. [5] menunjukkan integrasi metode *Agile* dalam pengembangan sistem berbasis web dan melakukan evaluasi kelayakan sistem menggunakan *System Usability Scale (SUS)*. Evaluasi tersebut memberikan data kuantitatif mengenai persepsi pengguna terhadap kemudahan penggunaan sistem, menjadi bukti bahwa metode iteratif dapat meningkatkan kualitas perangkat lunak. Penerapan *Agile* memungkinkan pengembangan dilakukan dalam siklus yang fleksibel dan adaptif terhadap kebutuhan pengguna. Penggunaan *SUS* sebagai alat ukur memastikan bahwa produk yang dihasilkan telah melewati proses validasi dari sisi fungsionalitas dan kepuasan pengguna. Studi ini menjadi landasan penting bagi penelitian yang mengedepankan aspek teknis dan non-teknis dalam pembangunan perangkat lunak berbasis web.

Kajian oleh Hegedüs et al. [6] memaparkan pengujian *usability* pada

aplikasi web kolaboratif dengan kombinasi metode *thinking aloud* dan SUS. Strategi ini memungkinkan periset memahami secara langsung kendala dan persepsi pengguna selama interaksi dengan sistem. Hasil pengujian memberikan informasi mendalam tentang bagaimana antarmuka dan alur sistem memengaruhi efisiensi pengguna. Dalam konteks pengembangan perangkat lunak berbasis web, pendekatan ini berkontribusi pada peningkatan desain interaktif dan keterlibatan pengguna. Model pengujian ini dapat direplikasi dalam pengembangan dashboard yang memiliki kompleksitas fitur serupa.

Penelitian oleh Rahman dan Yani [7] mengevaluasi aplikasi *e-learning* berbasis web menggunakan instrumen SUS. Instrumen tersebut digunakan untuk menilai aspek kemudahan penggunaan, efisiensi navigasi, dan kenyamanan visual dari sisi pengguna akhir. Evaluasi tersebut memberikan validasi kuantitatif yang berguna dalam proses penyempurnaan antarmuka pengguna. Penerapan metode ini dapat dijadikan acuan dalam mengembangkan sistem *Progressive Web App (PWA)* yang responsif dan ramah pengguna. Studi ini memperkuat pentingnya pengujian kelayakan sistem secara sistematis pada tahap akhir pengembangan.

Indrawan dan Wibisono [8] menjelaskan penerapan metode *Scrum* dalam pengembangan aplikasi administrasi kegiatan mahasiswa. Proses pengembangan dilakukan melalui serangkaian sprint yang melibatkan peran *Product Owner*, *Scrum Master*, dan tim pengembang. Metodologi ini memungkinkan manajemen proyek yang terstruktur serta penyesuaian fitur secara berkelanjutan berdasarkan masukan pengguna. Konsep ini sangat relevan dalam pembangunan sistem dashboard yang membutuhkan kecepatan iterasi serta validasi kontinu. Studi tersebut menunjukkan efektivitas *Scrum* dalam konteks pengembangan perangkat lunak skala menengah berbasis web.

Purnama et al. [9] mengembangkan dashboard pemantauan greenhouse berbasis web dengan konsep *smart farming*. Arsitektur sistem dan antarmuka pengguna yang dirancang menjadi referensi penting dalam merancang visualisasi data pertanian secara interaktif. Namun, penelitian tersebut belum mencakup

aspek evaluasi kelayakan pengguna secara eksplisit, sehingga tidak diketahui sejauh mana sistem memenuhi ekspektasi pengguna akhir. Keterbatasan ini menunjukkan pentingnya mengintegrasikan evaluasi usability sebagai bagian dari siklus pengembangan. Pendekatan tersebut akan memperkuat kualitas sistem dari sisi fungsional maupun pengalaman pengguna.

Penelitian terdahulu secara umum menunjukkan bahwa pengembangan sistem berbasis IoT dan web mulai menekankan aspek usability, fleksibilitas metodologi, serta penerapan teknologi modern seperti PWA. Keterbatasan muncul pada belum adanya penelitian yang secara terpadu menggabungkan seluruh pendekatan tersebut, mulai dari pengembangan sistem dengan *Agile Scrum*, integrasi PWA, hingga evaluasi sistem menggunakan *blackbox testing* dan SUS. Penelitian ini berupaya menjawab kekosongan tersebut melalui rancangan dan pembangunan sistem dashboard *smart farming* berbasis IoT yang adaptif, *usable*, dan teruji secara fungsional dan perseptual.

Tabel 2.1 Literasi Penelitian Terdahulu

No.	Judul, Tahun	Nama Peneliti	Masalah	Metode dan Hasil
1	<i>Smart Greenhouse: An IoT Based Monitoring and Control System with Predictive Analytics</i> (2021)[4]	P. Yaswanth, G. Swathi, C. Gokulnath, M. Vignesh, R. Nagarajan	Kontrol manual rumah kaca tidak efisien dalam pemantauan kondisi lingkungan tanaman.	Sistem IoT dengan analitik prediktif berbasis MQTT dan dashboard. Sistem mampu kontrol otomatis dan pantau kondisi lingkungan secara prediktif.

No.	Judul, Tahun	Nama Peneliti	Masalah	Metode dan Hasil
2	<i>Dashboard Pemantauan Greenhouse Berbasis Web dengan Konsep Smart Farming</i> (2023)[9]	D.C. Purnama, F.M. Nugraha, A.F. Hermawan, S. Sulistyo	Tidak tersedianya sistem pemantauan greenhouse berbasis smart farming secara aktual.	Pengembangan dashboard berbasis web. Sistem mampu menampilkan data kondisi lingkungan secara real-time.
3	<i>Implementation of System Usability Scale in the Development of Smart Classroom Web-based Application Using Agile Method</i> (2022)[5]	M. Taufik, R. Puspita, D. Kuswandi	Pengukuran usability untuk aplikasi berbasis web belum dilakukan secara sistematis.	Metode Agile dan evaluasi dengan System Usability Scale. Nilai SUS menunjukkan aplikasi diterima dengan baik.

No.	Judul, Tahun	Nama Peneliti	Masalah	Metode dan Hasil
4	<i>Usability testing of a collaborative programming web application using the System Usability Scale and thinking-aloud method</i> (2022)[6]	M. Hegedüs, Á. Nagy, B. Forstner, T. Szűts	Dibutuhkan pengujian kegunaan aplikasi kolaboratif secara menyeluruh.	Evaluasi menggunakan SUS dan metode thinking-aloud. Hasil menunjukkan usability tinggi dan rekomendasi peningkatan antarmuka.
5	<i>Usability Evaluation of E-Learning Application using System Usability Scale (SUS)</i> (2022)[7]	A. Rahman, E. Yani	Evaluasi antarmuka pengguna sistem e-learning diperlukan untuk menjamin kenyamanan penggunaan.	Evaluasi menggunakan SUS. Aplikasi memperoleh nilai yang baik dalam persepsi kegunaan oleh pengguna.

No.	Judul, Tahun	Nama Peneliti	Masalah	Metode dan Hasil
6	<i>Penerapan Metode Scrum pada Pengembangan Aplikasi Administrasi Kegiatan Mahasiswa Fakultas Teknik Universitas Muhammadiyah Jakarta (2021)[8]</i>	L.Y. Indrawan, A. Wibisono	Administrasi kegiatan mahasiswa belum terdigitalisasi secara efisien.	Pengembangan sistem menggunakan metodologi Scrum. Hasilnya aplikasi mampu mendukung kegiatan administrasi secara efektif.

2.2 Dasar Teori

Dasar teori dalam penelitian ini mencakup konsep dan teknologi utama yang digunakan dalam pengembangan sistem dashboard smart farming berbasis *Progressive Web App* (PWA). Pembahasan difokuskan pada teori mengenai pengembangan web modern, pemodelan sistem, komunikasi data dengan perangkat IoT, manajemen proyek perangkat lunak, serta metode evaluasi sistem. Seluruh teori yang disajikan berfungsi sebagai landasan konseptual untuk mendukung proses analisis, perancangan, implementasi, dan pengujian sistem yang dibangun pada *greenhouse* Kebun Raya ITERA.

2.2.1 Smart Farming

Smart farming adalah pendekatan pertanian modern berbasis *IoT*, *big data*, dan *cloud computing* untuk meningkatkan efisiensi serta keberlanjutan produksi pangan serta mengurangi dampak lingkungan. Integrasi *IoT* dan *AI*

memungkinkan pengumpulan data *real-time* dari sensor dan analisis otomatis untuk mengoptimalkan irigasi, pengendalian hama, dan aplikasi nutrisi yang presisi. Penerapan dalam *greenhouse* dikaitkan dengan pengendalian lingkungan secara otomatis dan efisiensi operasional yang lebih tinggi.

2.2.2 Sensor dan Pemantauan Lingkungan

Sensor modern di pertanian cerdas tidak hanya mencakup suhu dan kelembaban, tetapi berkembang menjadi sensor *embedded*, miniatur, dan terintegrasi dengan *IoT*. Tinjauan pustaka menunjukkan bahwa sebagian besar studi *smart farming* mengandalkan sensor suhu, kelembaban udara, dan kelembaban tanah sebagai parameter utama untuk kontrol lingkungan.

2.2.3 Website

Website adalah antarmuka dasar yang memungkinkan interaksi pengguna dengan sistem melalui peramban. Dalam pengembangan sistem informasi, *website* berfungsi sebagai media penyajian data secara dinamis dan interaktif dengan dukungan protokol HTTP dan format HTML. Kajian ini penting sebagai fondasi sebelum masuk ke ranah *dashboard* interaktif dan aplikasi progresif.

2.2.4 Progressive Web App (PWA)

PWA menggabungkan kekuatan *website* dengan fitur aplikasi *native* seperti *offline caching* menggunakan *service workers*, *push notification*, dan kemampuan instalasi. Strategi ini sangat relevan untuk operasional di lapangan dengan konektivitas terbatas, karena memberikan akses yang cepat, responsif, dan dapat berjalan secara persisten di berbagai perangkat.

2.2.5 UI/UX

UI/UX merupakan aspek penting dalam keberhasilan interaksi pengguna. Penelitian komparatif terhadap perangkat lunak desain *UI* menekankan hubungan langsung antara estetika antarmuka dan efektivitas penggunaan. Selain itu,

metode evaluasi *heuristic* dan *human-computer interaction* menjadi pendekatan penting untuk mengukur dan meningkatkan pengalaman pengguna pada aplikasi berbasis web.

2.2.6 React (Komponen Frontend)

React adalah pustaka *JavaScript* berbasis komponen yang menggunakan *Virtual DOM* dan pendekatan deklaratif untuk memperbarui *UI* secara efisien saat data berubah. Fitur ini sangat penting dalam *dashboard smart farming* yang menampilkan data *real-time* dari sensor. Arsitektur *React* memungkinkan pembuatan *widget* interaktif seperti grafik suhu dan kelembaban yang dinamis dan mudah diintegrasikan dengan pustaka grafik.

2.2.7 Laravel (Komponen Backend)

Laravel adalah *framework PHP* berbasis *MVC* yang menyediakan *routing* sederhana, *ORM* (Eloquent), dan *templating* menggunakan *Blade*. *Framework* ini umum digunakan dalam membangun *RESTful API* untuk sistem pemantauan *IoT* dengan *MySQL* sebagai basis data. *Laravel* mempermudah pengelolaan data sensor secara *real-time* dan menjamin keamanan *endpoint API*.

2.2.8 API dan Komunikasi Data

API RESTful digunakan untuk pertukaran data antara frontend *React*, backend *Laravel*, dan perangkat *IoT*. Format *JSON* digunakan dengan metode *HTTP* standar seperti *GET* untuk pengambilan data dan *POST* untuk pengiriman data sensor. Pendekatan ini menjamin interoperabilitas dan kemudahan integrasi antar komponen sistem.

2.2.9 MQTT

MQTT adalah protokol *publish-subscribe* ringan yang ideal untuk komunikasi *IoT* dalam jaringan dengan bandwidth terbatas. Evaluasi terhadap kualitas layanan *MQTT* dalam pemantauan *smart farming* menunjukkan bahwa

protokol ini memiliki *throughput* tinggi, keandalan pengiriman pesan, dan latensi rendah.

2.2.10 Docker (Untuk Deployment dan Isolasi Lingkungan)

Docker menyediakan *container* yang memungkinkan aplikasi (*React*, *Laravel*, *MQTT broker*, *MySQL*) berjalan secara terisolasi dalam lingkungan yang konsisten. Dalam pengembangan *agile*, *Docker* mempermudah *deployment* dan pengujian antar modul dalam tim, serta mendukung skalabilitas dan portabilitas sistem. Dengan *Docker Compose*, seluruh *stack* sistem dapat dijalankan secara terpadu untuk integrasi dan pengujian cepat.

2.2.11 Database MySQL

MySQL merupakan sistem manajemen basis data relasional (*RDBMS*) yang banyak digunakan dalam aplikasi *IoT* karena kemampuannya menyimpan dan mengolah data sensor secara *real-time*. Sistem ini sangat cocok untuk *dashboard smart farming* yang memerlukan integritas data tinggi, kinerja *query* yang cepat, serta penyimpanan yang stabil dalam jangka panjang.

2.2.12 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) digunakan sebagai teknik pemodelan konseptual yang memvisualisasikan entitas, atribut, dan relasi antar entitas sebelum pembangunan basis data fisik. *ERD* membantu merancang struktur tabel seperti *Sensor*, *DataLog*, dan *User*, serta relasi *one-to-many* yang menjaga konsistensi dan efisiensi *query*.

2.2.13 UML Diagram (Overview)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk mendesain sistem dari berbagai sudut pandang—baik struktur maupun perilaku. Diagram *UML* mencakup representasi struktural seperti *Class Diagram*, *Object Diagram*, dan *ERD*, serta perilaku sistem melalui *Use Case*

Diagram, Activity Diagram, dan Sequence Diagram.

2.2.13.1 Use Case Diagram

Use Case Diagram menggambarkan interaksi antara aktor eksternal seperti admin dan teknisi dengan sistem, termasuk fungsi-fungsi seperti memonitor data sensor dan mengatur ambang batas kelembaban.

2.2.13.2 Activity Diagram

Activity Diagram merepresentasikan alur kerja sistem secara grafis, termasuk kondisi logis dan proses paralel seperti pengiriman data oleh sensor dan pembaruan tampilan *dashboard* secara *real-time*.

2.2.13.3 Sequence Diagram

Sequence Diagram memodelkan urutan pertukaran pesan antar objek seperti frontend *React*, backend *Laravel*, *database*, dan *MQTT broker*, serta menunjukkan respons sistem terhadap interaksi pengguna.

2.2.13.4 Class Diagram

Class Diagram menampilkan struktur kelas sistem seperti *Sensor*, *User*, dan *DataLog* lengkap dengan atribut, metode, serta hubungan seperti *inheritance* dan *association*.

2.2.13.5 Object Diagram

Object Diagram menunjukkan instansiasi objek pada saat sistem berjalan, misalnya objek *Sensor1* dengan atribut suhu dan *timestamp* tertentu.

2.2.14 Agile Scrum

Agile Scrum adalah metodologi pengembangan perangkat lunak yang bersifat iteratif dan inkremental, sangat cocok untuk proyek *IoT* karena menekankan adaptasi cepat dan umpan balik berkelanjutan. Metodologi ini membagi proses kerja ke dalam *sprint* pendek dan melibatkan peran seperti

Product Owner, *Scrum Master*, dan tim pengembang untuk kolaborasi intensif.

2.2.15 Blackbox Testing

Blackbox Testing adalah metode pengujian fungsional yang menilai perilaku sistem berdasarkan masukan dan keluaran tanpa melihat kode sumber. Teknik seperti *Boundary Value Analysis* dan *Equivalence Partitioning* sering digunakan untuk memvalidasi sistem sesuai spesifikasi.

2.2.16 System Usability Scale (SUS)

System Usability Scale (SUS) adalah instrumen evaluasi *usability* yang terdiri dari 10 pernyataan dalam skala *Likert*. Alat ini mengukur persepsi pengguna terhadap kemudahan dan kenyamanan penggunaan sistem, serta sering digunakan pada sistem berbasis web.

2.2.17 Teori Jumlah Responden SUS

Menentukan jumlah responden yang tepat dalam penerapan *SUS* penting untuk memperoleh hasil evaluasi yang valid. Panduan umum menyebutkan bahwa untuk margin kesalahan $\pm 5\%$, diperlukan setidaknya 36 responden, sedangkan margin $\pm 10\%$ dapat dicapai dengan 6–10 responden saja.

2.2.18 Dashboard dalam Sistem Informasi

Dashboard merupakan antarmuka visual yang menyajikan data secara ringkas dan terstruktur untuk mendukung pengambilan keputusan. Dalam sistem *IoT*, *dashboard* berfungsi sebagai jembatan antara data mentah dari sensor dengan interpretasi pengguna. Prinsip desain *dashboard* mencakup keterbacaan, visualisasi yang informatif (grafik garis, *pie chart*, indikator), dan kemampuan *real-time update*.

2.2.19 Pengujian Non-Fungsional Sistem

Pengujian non-fungsional mengevaluasi atribut kualitas sistem seperti waktu respon, skalabilitas, dan ketersediaan. Sistem *IoT* seperti *smart farming*

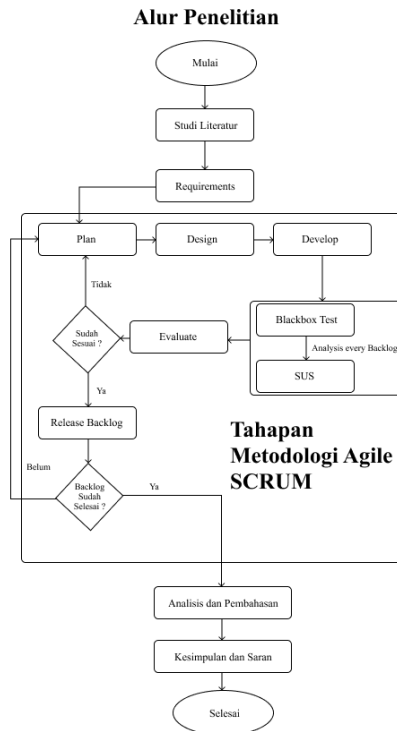
dashboard perlu mampu menangani beban tinggi, mempertahankan waktu respon rendah, dan tahan terhadap kegagalan perangkat. Evaluasi dapat dilakukan melalui simulasi beban dan pengukuran *latency* dan *throughput* sistem.

BAB III

METODE PENELITIAN

3.1 Alur Penelitian

Alur penelitian ini dirancang untuk menggambarkan tahapan-tahapan utama yang dilakukan secara sistematis, dimulai dari identifikasi masalah hingga evaluasi hasil. Setiap proses dijalankan secara berurutan untuk memastikan efisiensi dan kejelasan langkah-langkah pelaksanaan. Diagram alur penelitian yang ditampilkan pada Gambar 3.1 memberikan representasi visual dari proses tersebut dan menjadi pedoman dalam pelaksanaan penelitian.



Gambar 3.1 Alur Penelitian

3.2 Penjabaran Langkah Penelitian

Penjabaran langkah penelitian ini merujuk pada alur sistematis yang tergambar dalam subbab 3.1. Setiap tahapan dilaksanakan secara berurutan dan iteratif dalam proses pengembangan sistem dashboard *smart farming* berbasis *Progressive Web App (PWA)*. Berikut rincian tiap langkah yang dijalankan selama penelitian.

3.2.1 Langkah 1: Mulai

Peneliti menentukan ruang lingkup permasalahan, objek kajian, serta batasan sistem yang akan dikembangkan. Fokus diarahkan pada rancangan sistem pemantauan lingkungan greenhouse dengan integrasi perangkat IoT di Kebun Raya ITERA.

3.2.2 Langkah 2: Studi Literatur

Kajian pustaka dilakukan untuk memperkuat landasan teoritis terkait *smart farming*, teknologi IoT, pengembangan antarmuka web modern, serta metodologi pengembangan perangkat lunak. Referensi diambil dari jurnal ilmiah, buku, dan publikasi terpercaya.

3.2.3 Langkah 3: Requirements

Kebutuhan sistem dirumuskan berdasarkan hasil observasi, studi pustaka, dan analisis kebutuhan pengguna. Kebutuhan fungsional mencakup fitur utama, sedangkan kebutuhan non-fungsional mencakup keandalan, performa, dan skalabilitas sistem.

3.2.4 Langkah 4: Tahapan Metodologi Agile SCRUM

Pengembangan sistem mengikuti pendekatan iteratif melalui siklus *sprint*. Setiap sprint terdiri dari:

- **Plan:** Penyusunan backlog dan penentuan prioritas pengembangan.
- **Design:** Perancangan antarmuka pengguna, struktur basis data, dan

arsitektur sistem.

- **Develop:** Implementasi komponen frontend, backend, dan integrasi data dari perangkat IoT.
- **Evaluate:** Pengujian fungsional menggunakan *Blackbox Testing* dan pengukuran usability melalui *System Usability Scale (SUS)*.
- **Release Backlog:** Backlog dirilis setelah lolos uji evaluasi dan dianggap layak digunakan.

Proses diulang hingga seluruh backlog terselesaikan dan sistem mencapai versi final yang stabil.

3.2.5 Langkah 5: Analisis dan Pembahasan

Hasil pengembangan dan evaluasi sistem dianalisis secara kuantitatif dan kualitatif. Fokus analisis meliputi performa fungsional sistem dan interpretasi skor usability berdasarkan hasil SUS.

3.2.6 Langkah 6: Kesimpulan dan Saran

Peneliti menyimpulkan hasil capaian sistem serta efektivitas metodologi yang digunakan. Saran disusun sebagai bentuk kontribusi terhadap pengembangan sistem lebih lanjut maupun arah penelitian selanjutnya.

3.2.7 Langkah 7: Selesai

Seluruh proses perancangan, implementasi, dan evaluasi telah diselesaikan. Sistem siap digunakan dan dilaporkan dalam bentuk dokumen penelitian final.

3.3 Alat dan Bahan Tugas Akhir

Bagian ini menjelaskan alat dan bahan yang digunakan dalam proses pengembangan sistem dashboard *smart farming* berbasis *Progressive Web App (PWA)* untuk greenhouse di Kebun Raya ITERA. Penggunaan alat dan bahan disesuaikan dengan kebutuhan teknis pengembangan dan evaluasi sistem.

3.3.1 Alat

Pengembangan sistem dilakukan menggunakan perangkat keras dan perangkat lunak berikut:

1. *Laptop* dengan sistem operasi Windows 11, prosesor AMD Ryzen 5 4600H 6-core 3.0 GHz, RAM 16 GB DDR4, dan SSD 512 GB. Digunakan untuk pengembangan sistem frontend dan backend, konfigurasi Docker, serta pengujian sistem secara lokal.
2. *Smartphone* dengan sistem operasi minimal Android 10 atau iOS 13. Perangkat ini digunakan untuk menguji responsivitas antarmuka pengguna dan fitur instalasi *Progressive Web App (PWA)* di perangkat bergerak.
3. Perangkat mikrokontroler berbasis WiFi (misalnya ESP32 atau Raspberry Pi) dengan kemampuan membaca berbagai jenis sensor lingkungan seperti suhu, kelembapan udara, dan kelembapan tanah. Digunakan sebagai pengirim data ke server melalui pola *endpoint* URL yang telah ditentukan.
4. Broker MQTT Mosquitto yang berfungsi sebagai perantara komunikasi berbasis publish–subscribe antara perangkat IoT dan backend sistem.
5. *Code editor* Visual Studio Code, digunakan untuk menulis dan mengelola kode sumber berbasis React, Laravel, dan konfigurasi sistem Docker.
6. Docker Desktop sebagai alat kontainerisasi yang digunakan untuk membangun, menjalankan, dan mengelola lingkungan sistem terintegrasi (frontend, backend, database, MQTT broker).
7. GitHub sebagai platform manajemen versi dan repositori pengembangan sistem.

3.3.2 Bahan

Bahan penelitian mencakup sumber informasi, data lingkungan, dan dokumen pendukung yang digunakan selama proses pengembangan, yaitu:

1. Dataset sensor IoT greenhouse berisi suhu, kelembapan udara dan tanah, serta waktu pencatatan.

2. Dokumen kebutuhan sistem berdasarkan studi literatur, observasi lapangan, dan wawancara dengan pengelola greenhouse Kebun Raya ITERA.
3. Panduan teknis pengembangan *Progressive Web App (PWA)* menggunakan React, Laravel, dan layanan berbasis kontainer.
4. Kuesioner *System Usability Scale (SUS)* untuk evaluasi pengalaman dan kepuasan pengguna.

3.4 Metode Pengembangan

Metode pengembangan penelitian yang digunakan dalam sistem ini adalah *Agile Scrum*. *Agile Scrum* merupakan metodologi pengembangan perangkat lunak yang bersifat iteratif dan inkremental. Pendekatan ini memungkinkan pengembangan dilakukan dalam siklus-siklus pendek yang disebut *sprint*. Setiap *sprint* berfokus pada perencanaan, pembangunan, pengujian, dan evaluasi yang menghasilkan produk yang dapat diuji dan dievaluasi pada setiap iterasi. Dengan demikian, *Agile Scrum* memberikan fleksibilitas untuk beradaptasi dengan perubahan dan umpan balik dari pengguna secara terus-menerus. Ilustrasi dari alur kerja *Agile Scrum* dalam pengembangan sistem ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 *Agile SCRUM Methodology*

3.4.1 Requirements

Perancangan sistem dashboard smart farming diawali dengan identifikasi kebutuhan yang dikumpulkan dari hasil observasi dan wawancara terbatas bersama pengelola greenhouse Kebun Raya ITERA. Teknik pengumpulan data menggunakan pendekatan semi-terstruktur agar respons informan tetap sesuai kerangka kerja *Agile Scrum* yang berfokus pada pengguna. Tujuan utama tahap ini adalah merumuskan kebutuhan fungsional dan non-fungsional sebagai dasar backlog dalam sprint pengembangan. Setiap masukan pengguna diklasifikasikan berdasarkan peran mereka dalam sistem serta urgensi dari fitur yang dibutuhkan. Hasil wawancara peran stakeholder dirangkum pada Tabel 3.1, sedangkan daftar kebutuhan fungsional dan non-fungsional dijabarkan secara rinci pada Tabel 3.2 dan Tabel 3.3.

Tabel 3.1 Hasil Wawancara dengan Stakeholder Berdasarkan Peran *Agile Scrum*

No	Peran (<i>Scrum Role</i>)	Masukan dan Kebutuhan Utama
1	<i>Product Owner</i> (Koordinator Greenhouse)	Sistem harus menampilkan kondisi suhu, kelembapan udara, dan tanah secara real-time; data disimpan dan bisa diakses untuk analisis bulanan.
2	<i>Scrum Master</i> (Penanggung Jawab IT Lab)	Proses pengembangan harus terdokumentasi, iteratif, dan dapat diuji setiap sprint; komunikasi antar tim wajib terkoordinasi melalui backlog.
3	<i>User Representative</i> (Mahasiswa Praktikum)	Dashboard harus responsif di mobile dan desktop, memiliki tampilan antarmuka yang intuitif, serta bisa digunakan untuk melihat histori data harian.
4	<i>Developer Team</i> (Tim Pengembang)	Backend dan frontend harus modular, integrasi IoT melalui protokol <i>MQTT</i> , dan sistem siap dikemas dalam <i>Docker</i> .

3.4.1.1 Rancangan Kebutuhan Fungsional

Kebutuhan fungsional merupakan spesifikasi sistem yang wajib diimplementasikan agar sistem dapat beroperasi sesuai dengan harapan pemangku kepentingan. Kebutuhan ini disusun berdasarkan hasil analisis

dari wawancara, observasi, dan studi dokumentasi yang dilakukan bersama pengguna utama. Terdapat total 14 kebutuhan fungsional yang telah dirancang untuk mencerminkan alur operasional greenhouse ITERA secara digital dan otomatis. Seluruh kebutuhan ini juga menjadi acuan dalam pembuatan *backlog* pengembangan sistem berbasis *Agile Scrum*, sebagaimana dirangkum dalam Tabel 3.2.

Tabel 3.2 Daftar Kebutuhan Fungsional Sistem

No	Kebutuhan Fungsional	Deskripsi
1	Autentikasi Pengguna	Pengguna dapat melakukan login dan logout untuk mengakses sistem sesuai peran.
2	Manajemen Pengguna	Admin dapat menambahkan, mengedit, menghapus, dan melihat daftar pengguna serta peranannya.
3	Akses Dashboard	Pengguna dapat melihat ringkasan data dan status sistem melalui tampilan dashboard.
4	Manajemen Notifikasi	Pengguna dapat menandai notifikasi sebagai dibaca dan melihat notifikasi sebelumnya.
5	Manajemen Greenhouse	Admin dapat mengelola data greenhouse termasuk penambahan, penghapusan, serta unggahan gambar.
6	Manajemen Reservoir	Sistem mendukung pengelolaan reservoir, formula nutrisi, dan aktivasi berdasarkan kebutuhan.
7	Manajemen Sensor	Admin dapat menambah, mengedit, menghapus sensor serta memantau status dan log sensor.
8	Manajemen Aktuator	Pengguna dapat mengelola aktuator dan mengatur kontrol manual atau otomatis.
9	Kontrol Robot	Sistem mendukung input data robot serta kontrol arah dan statusnya.
10	Otomatisasi Berdasarkan Jadwal	Pengguna dapat menjadwalkan automasi aktuator berdasarkan hari dan waktu tertentu.
11	Otomatisasi Berdasarkan Sensor	Aktuator akan aktif berdasarkan pembacaan nilai sensor lingkungan.
12	API RESTful	Seluruh interaksi frontend-backend dilakukan melalui REST API modular.
13	Dokumentasi Endpoint	Sistem menyediakan dokumentasi teknis API dan cara penggunaannya.
14	Feedback Error Handling	Sistem memberikan pesan kesalahan yang informatif dan sesuai konteks input.

3.4.1.2 Rancangan Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan karakteristik kualitas yang menentukan performa dan reliabilitas sistem secara keseluruhan. Aspek-aspek ini tidak berkaitan langsung dengan fitur utama, namun berperan penting dalam menjamin pengalaman pengguna, keamanan, serta keberlanjutan sistem dalam

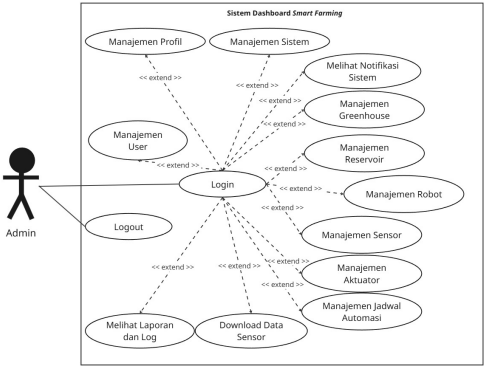
jangka panjang. Terdapat 10 kebutuhan non-fungsional yang telah diidentifikasi dan wajib dipenuhi dalam proses implementasi sistem dashboard smart farming ini, sebagaimana dirangkum dalam Tabel 3.3.

Tabel 3.3 Daftar Kebutuhan Non-Fungsional Sistem

No	Kebutuhan Non-Fungsional	Deskripsi
1	Responsivitas UI	Antarmuka harus mendukung berbagai ukuran layar dan perangkat (mobile, tablet, desktop).
2	<i>Progressive Web App (PWA)</i>	Sistem dapat diinstal, berjalan offline, dan memberikan performa menyerupai aplikasi native.
3	Skalabilitas Backend	Backend dirancang modular untuk mendukung ekspansi fitur dan skala pengguna di masa depan.
4	Real-Time Communication	Sistem mendukung komunikasi data real-time antara IoT dan server melalui protokol MQTT.
5	Keamanan Akses	Sistem menerapkan otorisasi berbasis peran untuk membatasi akses sesuai wewenang pengguna.
6	Konsistensi Data	Validasi input dan sinkronisasi data antar frontend dan backend dijaga agar tetap akurat.
7	Interoperabilitas	Sistem mampu terintegrasi dengan berbagai perangkat IoT dan robot yang menggunakan protokol terbuka.
8	Usability	Evaluasi pengalaman pengguna dilakukan melalui SUS untuk menjamin kenyamanan penggunaan.
9	Deployment Portabel	Sistem dapat dijalankan secara lokal menggunakan Docker dan siap untuk implementasi di cloud.
10	Maintainability	Kode dan dokumentasi disusun agar mudah dipelihara dan dipahami oleh pengembang lain.

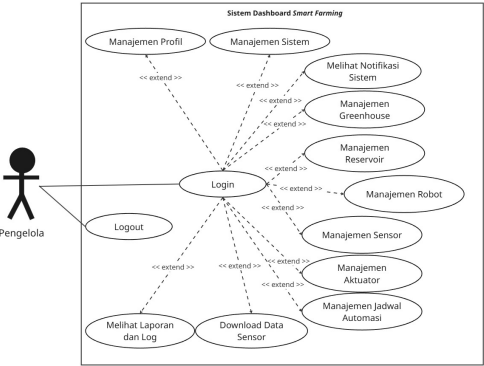
3.4.1.3 Use Case Diagram

Perumusan kebutuhan sistem dashboard *smart farming* telah diawali melalui tahap observasi dan wawancara terbatas dengan berbagai pemangku kepentingan di lingkungan greenhouse Kebun Raya ITERA. Wawancara dilakukan secara semi-terstruktur untuk menggali kebutuhan aktual berdasarkan peran masing-masing pengguna, sesuai dengan prinsip pendekatan *Agile Scrum*. Hasil wawancara diklasifikasikan menjadi kebutuhan fungsional dan non-fungsional, sebagaimana ditunjukkan pada Tabel 3.1, Tabel 3.2, dan Tabel 3.3. Seluruh kebutuhan tersebut kemudian diterjemahkan ke dalam model visual berupa diagram *use case* guna memetakan interaksi antara sistem dan aktor yang terlibat secara sistematis dan komprehensif.



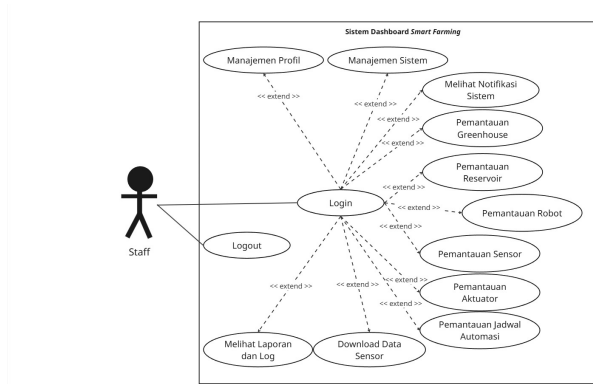
Gambar 3.3 Diagram Use Case untuk Aktor Admin

Gambar 3.3 menampilkan peran *Admin* yang memiliki otorisasi penuh dalam pengelolaan sistem. Admin dapat mengakses seluruh fitur manajemen seperti pengguna, greenhouse, sensor, reservoir, aktuator, dan robot. Seluruh fungsi utama sistem dikendalikan melalui proses login yang menjadi titik awal validasi identitas, sedangkan pengelolaan data dan kontrol sistem diperluas melalui fitur-fitur yang dihubungkan dengan relasi *« extend »*. Admin juga memiliki akses terhadap laporan sistem dan log aktivitas sebagai bagian dari mekanisme audit internal.



Gambar 3.4 Diagram Use Case untuk Aktor Pengeloa

Gambar 3.4 menggambarkan aktor *Pengelola* yang berperan sebagai *Product Owner* dalam kerangka kerja *Agile Scrum*. Pengelola memiliki tanggung jawab dalam pemantauan sistem, termasuk data dari greenhouse, reservoir, sensor, dan jadwal automasi. Walaupun tidak memiliki kewenangan penuh dalam mengubah konfigurasi sistem, pengelola memainkan peran penting dalam memberikan umpan balik terhadap performa sistem yang akan dikomunikasikan kepada tim pengembang untuk pengambilan keputusan pengembangan lanjutan.



Gambar 3.5 Diagram Use Case untuk Aktor Staff

Gambar 3.5 menunjukkan aktor *Staff* yang memiliki akses terbatas dan difokuskan pada fungsi pemantauan. Staff hanya diperbolehkan melihat data dari sensor, aktuator, robot, dan greenhouse tanpa memiliki otorisasi untuk melakukan konfigurasi sistem. Akses ini ditujukan untuk mendukung pelaksanaan operasional harian dan memastikan proses pemantauan berlangsung sesuai prosedur standar, tanpa risiko perubahan yang tidak sah terhadap sistem.

Dengan adanya representasi diagram *use case* dari ketiga aktor utama, yaitu Admin, Pengelola, dan Staff, maka interaksi pengguna dengan sistem telah terpetakan secara menyeluruh. Hal ini memberikan dasar yang kuat bagi pengembangan sistem berbasis kebutuhan aktual yang ditemukan di lapangan.

3.4.2 Pemodelan Sistem Menggunakan UML

Setelah tahap perumusan kebutuhan selesai, proses perancangan dilanjutkan dengan pemodelan sistem menggunakan pendekatan *Unified Modeling Language* (UML). UML merupakan standar pemodelan visual yang digunakan secara luas untuk menggambarkan struktur, perilaku, serta alur komunikasi dalam suatu sistem perangkat lunak. Pemodelan ini berfungsi untuk mengilustrasikan arsitektur sistem berdasarkan hasil wawancara dan analisis kebutuhan pada tahap sebelumnya (Tabel 3.1, Tabel 3.2, dan Tabel 3.3). Diagram yang digunakan dalam perancangan ini meliputi Data Flow Diagram (DFD), *Activity Diagram*, *Sequence Diagram*, serta *Entity Relationship Diagram* (ERD) untuk basis data.

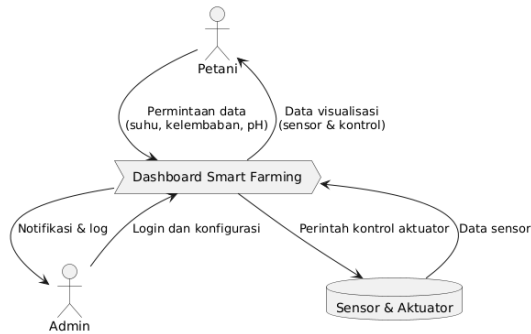
3.4.2.1 Data Flow Diagram (DFD)

Diagram Alir Data atau Data Flow Diagram (DFD) digunakan untuk menggambarkan aliran data dan proses utama dalam sistem. DFD membantu memetakan hubungan antara entitas eksternal, proses internal, serta penyimpanan data dalam sistem yang dirancang. Pada penelitian ini, DFD disusun secara bertingkat mulai dari level 0 (konteks umum), level 1 (proses utama), hingga level 2 (rincian proses spesifik).

1. DFD Level 0

DFD Level 0 menggambarkan konteks sistem secara menyeluruh sebagai satu kesatuan proses utama. Diagram ini menunjukkan bagaimana entitas eksternal seperti *Admin* dan *Petani* berinteraksi dengan sistem *Dashboard Smart Farming* dalam bentuk input dan output.

Data Flow Diagram Level 0 - Smart Farming Dashboard

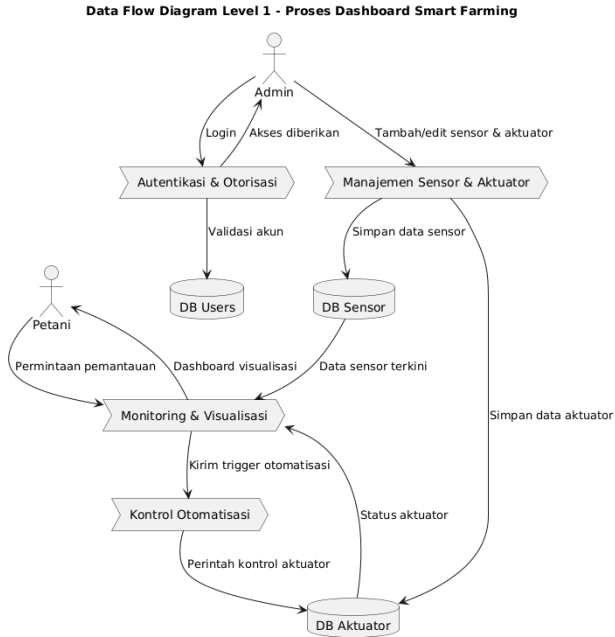


Gambar 3.6 Data Flow Diagram Level 0 Sistem Smart Farming

Gambar 3.6 memperlihatkan bahwa *Admin* bertugas mengelola konfigurasi sistem dan data perangkat, sedangkan *Petani* menerima informasi hasil pemantauan dari sistem. Sistem ini mengelola data sensor, jadwal otomatisasi, dan proses kontrol perangkat berbasis ambang batas.

2. DFD Level 1

DFD Level 1 menjabarkan proses utama yang ada dalam sistem menjadi sub-proses seperti autentikasi pengguna, pemantauan sensor, pengaturan jadwal, dan kontrol otomatis perangkat. Diagram ini memberikan gambaran lebih rinci tentang aliran data antar proses utama dan data store.

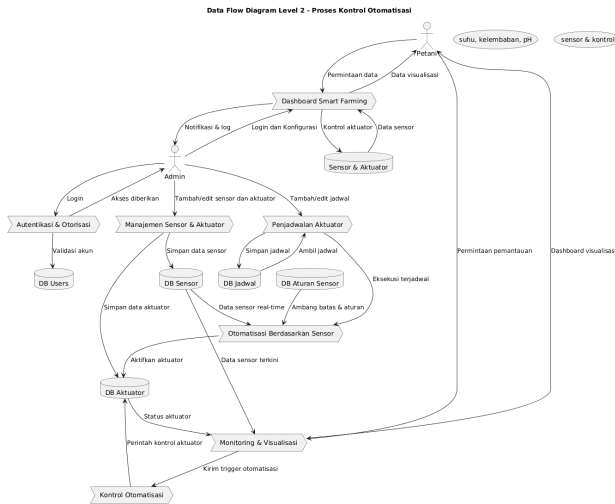


Gambar 3.7 Data Flow Diagram Level 1 Sistem Smart Farming

Gambar 3.7 menjelaskan bahwa sistem terdiri dari beberapa proses utama, antara lain proses login, pengaturan jadwal, manajemen sensor dan aktuator, serta pemantauan kondisi lingkungan. Setiap proses berhubungan dengan data store seperti basis data pengguna, perangkat, jadwal, dan data historis sensor.

3. DFD Level 2

DFD Level 2 memfokuskan pada rincian dari salah satu proses pada Level 1, yaitu otomatisasi kontrol perangkat berdasarkan sensor. Diagram ini menjelaskan detail bagaimana input dari sensor diproses dan bagaimana sistem memutuskan tindakan otomatis.

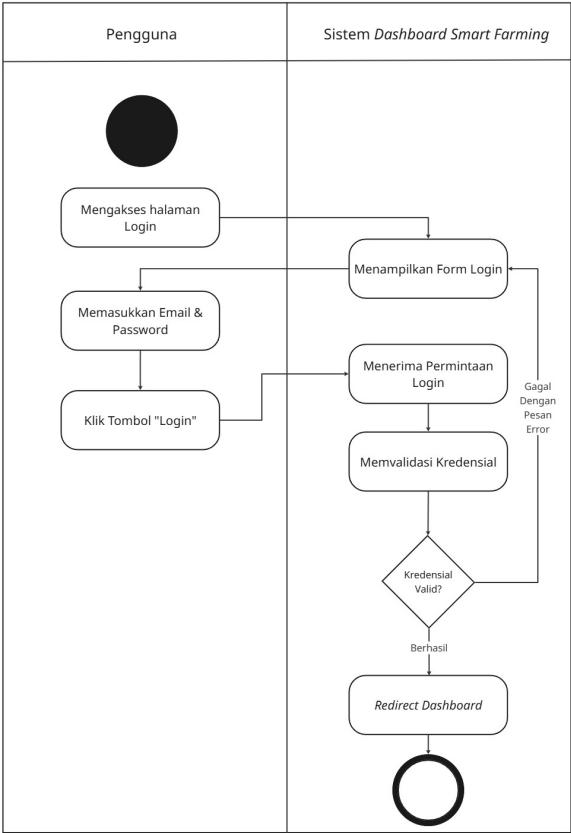


Gambar 3.8 Data Flow Diagram Level 2 – Proses Otomatisasi Kontrol Perangkat

Gambar 3.8 menggambarkan bahwa proses kontrol otomatis dimulai dari pembacaan nilai sensor, dibandingkan dengan ambang batas, dan jika kondisi terpenuhi maka sistem akan mengirimkan perintah ke aktuator untuk menyalakan atau mematikan perangkat. Selain itu, terdapat input tambahan dari jadwal harian yang mempengaruhi keputusan kontrol perangkat.

3.4.2.2 Activity Diagram

Diagram aktivitas digunakan untuk menggambarkan urutan langkah-langkah yang dilakukan pengguna dalam menjalankan suatu proses di dalam sistem. Diagram ini bermanfaat dalam merinci logika alur kerja serta percabangan keputusan yang mungkin terjadi selama interaksi pengguna dengan sistem.



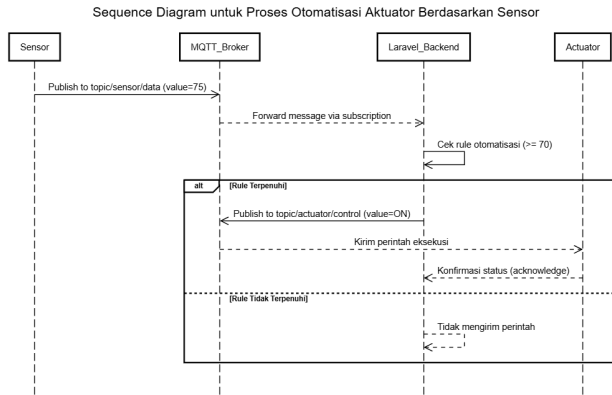
Gambar 3.9 Activity Diagram Proses Login dan Akses Dashboard

Gambar 3.9 menunjukkan alur proses login yang dilakukan oleh pengguna hingga berhasil mengakses fitur dashboard sesuai hak akses masing-masing. Alur ini mencakup validasi kredensial, pengalihan halaman, serta inisialisasi data pengguna.

3.4.2.3 Sequence Diagram

Diagram sekuensial menampilkan interaksi antar objek dalam sistem berdasarkan urutan waktu. Diagram ini menunjukkan bagaimana suatu

permintaan dikirim dari satu aktor ke komponen sistem dan bagaimana respon dikembalikan, cocok untuk menggambarkan komunikasi antara frontend, backend, dan IoT device.

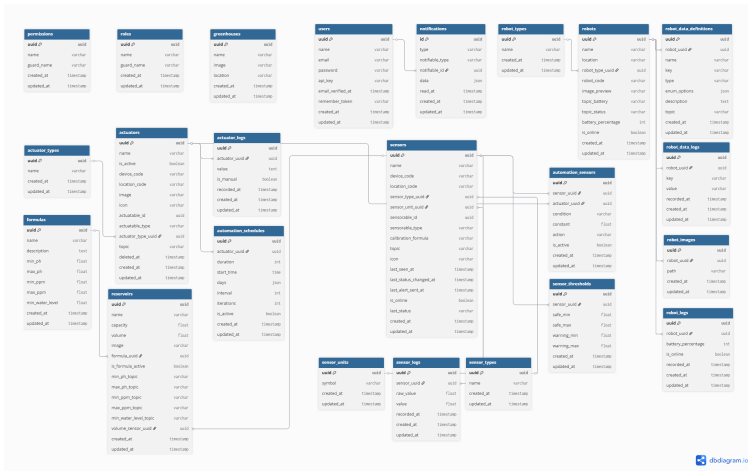


Gambar 3.10 Sequence Diagram untuk Proses Otomatisasi Aktuator Berdasarkan Sensor

Gambar 3.10 memperlihatkan urutan komunikasi ketika sensor mendeteksi kondisi tertentu, sistem memproses logika, dan aktuator diaktifkan secara otomatis berdasarkan aturan yang telah ditetapkan oleh pengguna.

3.4.2.4 Perancangan Basis Data (ERD)

Untuk mendukung implementasi sistem secara teknis, dirancang struktur basis data menggunakan pendekatan *Entity Relationship Diagram (ERD)*. Diagram ini menggambarkan entitas utama dalam sistem, atributnya, serta hubungan antar entitas tersebut. Model basis data dirancang untuk memenuhi kebutuhan sistem *smart farming* berbasis automasi, dengan mempertimbangkan efisiensi relasional dan skalabilitas data.



Gambar 3.11 Diagram *Entity Relationship Diagram* (ERD) Sistem Dashboard *Smart Farming*

Gambar 3.11 menggambarkan hubungan antar entitas inti dalam sistem seperti *users*, *greenhouses*, *actuators*, *robots*, *sensors*, *reservoirs*, hingga *notifications*. Setiap entitas memiliki atribut penting yang direpresentasikan dalam bentuk tabel, sebagaimana dijelaskan secara rinci pada Tabel 3.4.

Tabel 3.4 Struktur Tabel Basis Data Utama Sistem

Tabel	Kolom Utama	Deskripsi
users	uuid, email, api_key	Menyimpan data pengguna dan autentikasi berbasis peran.
actuators	uuid, device_code, actuator_type_uuid	Menyimpan data perangkat aktuator dan hubungan polimorfik ke greenhouse, reservoir, dan robot.
actuator_logs	uuid, actuator_uuid, value	Log histori eksekusi aktuator baik manual maupun otomatis.
actuator_types	uuid, name	Master data jenis aktuator.
automation_schedules	uuid, actuator_uuid, days	Menyimpan jadwal otomatisasi aktuator.
automation_sensors	uuid, sensor_uuid, condition, constant	Aturan otomatisasi berdasarkan pembacaan sensor.
formulas	uuid, min_ph, max_ppm	Batasan standar parameter larutan untuk reservoir.
greenhouses	uuid, name, location	Entitas rumah kaca sebagai lokasi utama pemantauan dan kontrol.
reservoirs	uuid, capacity, formula_uuid	Data tangki cairan dan referensi ke rumus kualitas larutan.
robots	uuid, robot_code, robot_type_uuid	Menyimpan data perangkat robotik, termasuk status dan lokasi.
robot_data_definitions	uuid, robot_uuid, key	Struktur parameter data yang dikirimkan/dibaca dari robot.
robot_data_logs	uuid, key, value	Log histori nilai-nilai parameter robot.
robot_images	uuid, robot_uuid, path	Penyimpanan gambar robot.
robot_logs	uuid, battery_percentage, is_online	Riwayat status baterai dan konektivitas robot.
robot_types	uuid, name	Master data jenis robot.
sensors	uuid, device_code, sensor_type_uuid	Sensor lingkungan, memiliki relasi polimorfik ke greenhouse, reservoir, dan robot.
sensor_logs	uuid, sensor_uuid, value	Histori pembacaan data sensor.
sensor_thresholds	uuid, safe_min, warning_max	Ambang batas nilai aman dan peringatan sensor.
sensor_types	uuid, name	Jenis sensor (misalnya suhu, pH).
sensors_units	uuid, symbol	Satuan sensor (e.g. °C, ppm).
permissions	uuid, name, guard_name	Hak akses individual pengguna.
roles	uuid, name	Peran pengguna dalam sistem (admin, pengelola, staff).
notifications	uuid, type, data	Notifikasi sistem yang ditujukan ke entitas pengguna.

Struktur tabel di atas dirancang agar fleksibel terhadap perubahan topologi dan fungsionalitas sistem, termasuk pemantauan multi lokasi, otomatisasi hybrid (berbasis waktu dan sensor), serta integrasi antar komponen berbasis *Internet of Things (IoT)*.

3.4.3 Perencanaan Menggunakan Metode Agile Scrum

Pengembangan sistem dashboard *smart farming* dalam penelitian ini menggunakan metodologi *Agile Scrum*. Metode ini dipilih karena mampu mendukung proses pengembangan yang iteratif dan adaptif terhadap perubahan kebutuhan pengguna. Scrum membagi proses pengembangan menjadi beberapa *sprint* yang memiliki fokus dan keluaran spesifik. Setiap *sprint* bertujuan untuk menghasilkan peningkatan fungsi sistem secara bertahap, terukur, dan dapat dievaluasi langsung. Kerangka kerja ini juga memungkinkan pelibatan pemangku kepentingan secara langsung melalui peran-peran terstruktur, sehingga kebutuhan aktual dapat ditangkap secara cepat dan akurat. Penerapan pendekatan ini memastikan bahwa pengembangan sistem dilakukan dengan efisien, terorganisir, dan selaras dengan tujuan proyek.

3.4.3.1 Planning

Tahapan perencanaan dilakukan melalui pembentukan *Product Backlog* yang kemudian dipilah menjadi daftar *Sprint Backlog* untuk masing-masing iterasi. Fokus pengembangan pada setiap *sprint* telah ditentukan sejak awal, sebagaimana ditampilkan pada Tabel 3.5.

Tabel 3.5 Perencanaan Sprint Berdasarkan Fitur Pengembangan

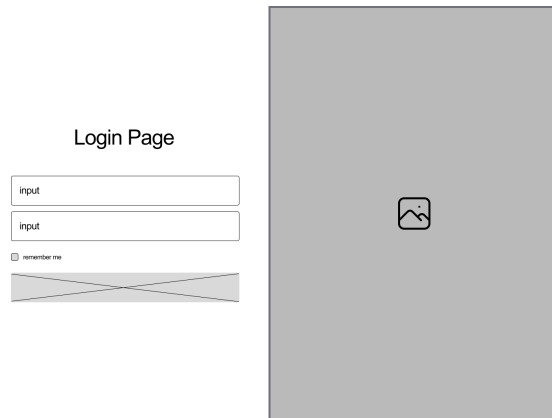
Sprint	Fokus Pengembangan	Fitur yang Dikembangkan	Tujuan Sprint
Sprint 1	Fondasi dan Manajemen Inti	<ul style="list-style-type: none"> Manajemen Pengguna Manajemen Greenhouse Manajemen Reservoir Manajemen Robot 	Menyediakan sistem yang mampu mengelola entitas inti dan menangani proses autentikasi serta input data secara akurat.
Sprint 2	Monitoring dan Kontrol Dasar	<ul style="list-style-type: none"> Monitoring Data Sensor Kontrol Manual Aktuator 	Sistem mampu membaca data lingkungan dan memberikan kontrol manual bagi pengguna terhadap perangkat di greenhouse.
Sprint 3	Visualisasi dan Notifikasi	<ul style="list-style-type: none"> Dashboard Visual Sistem Notifikasi Unduhan Data Sensor 	Menyajikan informasi lingkungan secara visual, memberikan peringatan sistem, dan mendukung kegiatan analisis data.
Sprint 4	Otomatisasi dan Penyempurnaan	<ul style="list-style-type: none"> Otomasi Berbasis Sensor Otomasi Berbasis Jadwal 	Sistem mendukung kontrol perangkat otomatis berbasis kondisi lingkungan dan pengaturan waktu.

3.4.3.2 Design

Desain fitur pada setiap iterasi disusun untuk mengakomodasi kebutuhan pengguna secara progresif. Rancangan antarmuka dan alur kerja sistem dijabarkan sebagai berikut:

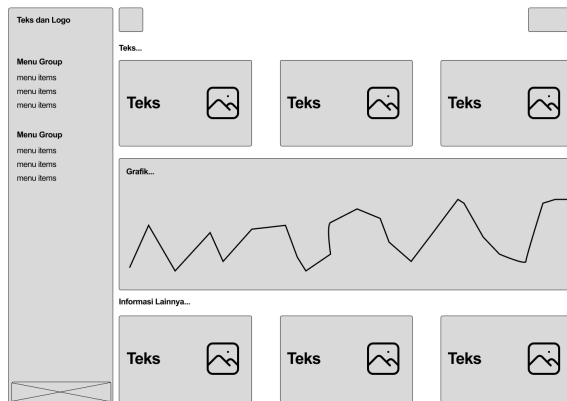
1. Sprint 1

Tahapan awal dalam proses pengembangan difokuskan pada perancangan dan implementasi antarmuka dasar sistem, mencakup halaman login, dashboard utama, serta halaman manajemen entitas seperti *greenhouse*, *reservoir*, *robot*, dan pengguna. Seluruh komponen ini dikembangkan untuk membentuk fondasi sistem yang fungsional, dapat diuji, serta mendukung prinsip *progressive enhancement* dalam pengembangan berbasis *web*.



Gambar 3.12 Tampilan Halaman Login Sistem Dashboard *Smart Farming*

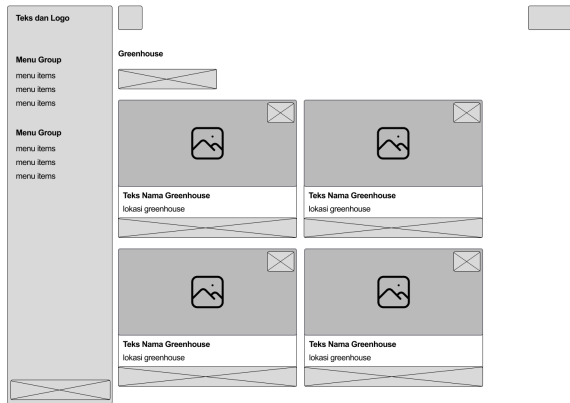
Gambar 3.12 menampilkan antarmuka halaman login dengan dua *input field* untuk autentikasi pengguna serta tombol aksi masuk ke dalam sistem. Desain halaman login menerapkan prinsip minimalis untuk memastikan proses masuk berlangsung cepat dan tidak membingungkan pengguna. Letak komponen ditata secara responsif agar dapat digunakan secara optimal di berbagai ukuran perangkat, termasuk *smartphone*.



Gambar 3.13 Tampilan Awal Dashboard Sistem *Smart Farming*

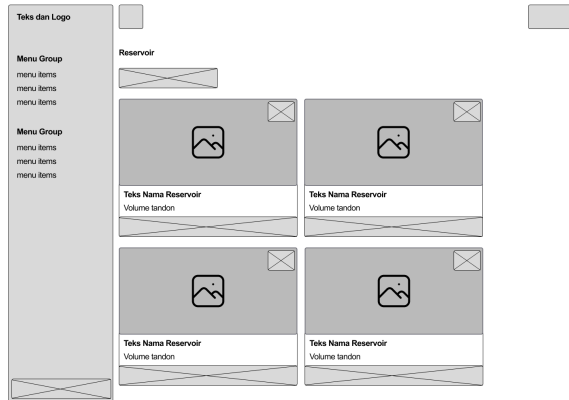
Gambar 3.13 memperlihatkan struktur dashboard utama yang menyajikan

informasi status sistem secara *real-time* dalam bentuk kartu dan grafik. Terdapat panel navigasi di sisi kiri layar yang berfungsi sebagai pusat akses ke berbagai fitur utama seperti *greenhouse*, *reservoir*, dan *robot*. Visualisasi informasi dilakukan secara hierarkis untuk meningkatkan efektivitas pemantauan serta kecepatan pengambilan keputusan.



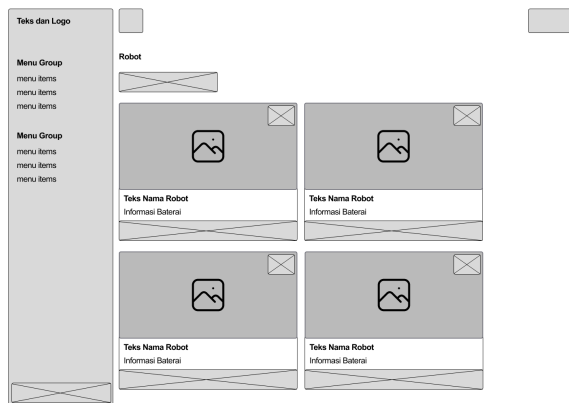
Gambar 3.14 Tampilan Daftar *Greenhouse* dengan Informasi Lokasi

Gambar 3.14 memperlihatkan halaman daftar *greenhouse* yang menampilkan nama unit serta lokasi operasionalnya. Setiap elemen ditampilkan dalam bentuk kartu bergambar untuk memudahkan identifikasi visual. Informasi ini merupakan fondasi navigasi untuk masuk ke detail tiap *greenhouse* serta menjadi pintu masuk utama untuk pemantauan lingkungan pertanian berbasis digital.



Gambar 3.15 Tampilan Daftar *Reservoir* dengan Informasi Volume

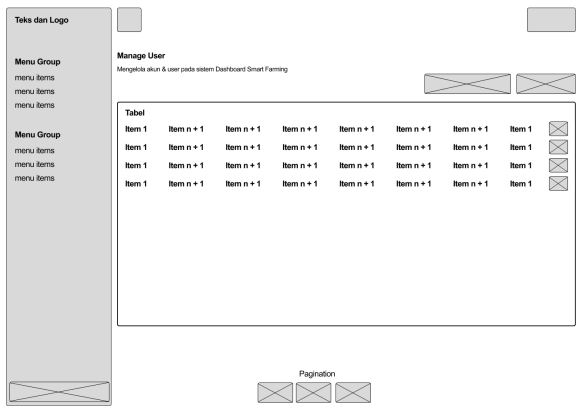
Gambar 3.15 menampilkan daftar *reservoir* aktif yang menyajikan informasi volume air pada setiap unit tandon. Komponen kartu pada halaman ini dirancang untuk memperjelas kapasitas air yang tersedia serta mendukung perencanaan otomatisasi irigasi. Penataan antarmuka ini dirancang untuk memberikan visibilitas penuh terhadap kondisi sistem irigasi secara kontekstual.



Gambar 3.16 Tampilan Daftar *Robot* dengan Informasi Status dan Baterai

Gambar 3.16 menyajikan tampilan daftar *robot* yang digunakan dalam sistem otomasi pertanian. Informasi utama yang disampaikan mencakup status

koneksi, kapasitas baterai, serta pengendalian perangkat melalui *toggle*. Desain ini mendukung kontrol langsung dari dashboard serta menampilkan status operasional secara *real-time* untuk memastikan kinerja perangkat berjalan optimal.



Gambar 3.17 Tampilan Halaman Manajemen Pengguna Sistem *Smart Farming*

Gambar 3.17 menggambarkan antarmuka untuk mengelola pengguna sistem. Informasi akun disusun dalam format tabel dengan fungsi pencarian, pengurutan, serta aksi *edit* dan *delete*. Halaman ini memegang peran penting dalam pengaturan hak akses dan kontrol otorisasi, sehingga sistem dapat digunakan secara aman oleh pengguna yang berwenang.

Setiap antarmuka pada tahap awal ini dikembangkan dengan pendekatan modular dan *responsive*, memungkinkan sistem untuk berevolusi pada iterasi selanjutnya. Strategi perancangan berbasis *user-centered design* menjamin bahwa seluruh fitur benar-benar menjawab kebutuhan aktual pengguna akhir dalam konteks pemantauan dan pengendalian *smart farming* modern.

2. Sprint 2

Antarmuka sensor monitoring menyajikan grafik waktu nyata untuk suhu, kelembaban, dan pH tanah. Desain kontrol manual memungkinkan pengguna

menyalakan dan mematikan pompa, penyiram, atau kipas secara langsung.

3. Sprint 3

Desain dashboard utama menampilkan grafik data historis, pemberitahuan otomatis pada ambang batas ekstrem, dan tombol ekspor data dalam format *.csv*. Panel notifikasi didesain muncul kontekstual berdasarkan kondisi sistem.

4. Sprint 4

Fitur otomasi memiliki form untuk konfigurasi aturan berdasarkan nilai sensor dan jadwal mingguan. Desain mencakup kalender interaktif dan log aktivitas sistem otomatis.

3.4.3.3 Coding

Proses pengkodean dilakukan berdasarkan pembagian *sprint*. Setiap sprint memiliki alur pengembangan fitur yang dirinci dalam Tabel 3.6.

Tabel 3.6 Rincian Coding Berdasarkan Sprint

Sprint	Flow Coding dan Komponen
Sprint 1	<ul style="list-style-type: none">• Pembuatan model dan skema database untuk entitas pengguna, greenhouse, reservoir, robot• Pengembangan <i>REST API</i> untuk CRUD data inti menggunakan Express.js• Implementasi halaman login dan autentikasi JWT• Pembuatan UI dengan React dan Tailwind untuk dashboard admin
Sprint 2	<ul style="list-style-type: none">• Integrasi data sensor melalui MQTT atau REST polling• Komponen React untuk menampilkan grafik real-time (misal: Chart.js)• Kontrol manual dengan tombol aksi terhubung ke endpoint kontrol aktuator
Sprint 3	<ul style="list-style-type: none">• Pengembangan modul notifikasi berbasis threshold• Penambahan fitur ekspor data CSV• Implementasi dashboard visual dengan layout grid responsif
Sprint 4	<ul style="list-style-type: none">• Penambahan fitur rule engine berbasis nilai sensor• Penjadwalan kontrol dengan cron-task dan timepicker• Logging otomatis ke basis data dan halaman riwayat

3.4.3.4 Evaluate

Evaluasi terhadap sistem dilakukan pada setiap akhir *sprint* menggunakan dua pendekatan: *System Usability Scale* (SUS) untuk menilai kualitas pengalaman pengguna dan metode *Blackbox Testing* untuk menguji fungsionalitas tiap fitur. Pendekatan ini memungkinkan evaluasi sistem secara menyeluruh baik dari sisi kegunaan maupun ketahanan fitur terhadap berbagai

skenario input.

3.4.3.5 Release Backlog

Setelah menyelesaikan setiap *sprint*, dilakukan rilis hasil pengembangan yang terdokumentasi dalam tabel backlog. Evaluasi setiap iterasi dilakukan menggunakan metode *Blackbox Testing* untuk menguji fungsionalitas sistem dan *System Usability Scale* (SUS) untuk mengukur persepsi pengguna terhadap kemudahan penggunaan sistem. Tabel 3.7 merangkum kode sprint, fitur yang dirilis, skor evaluasi, dan status rilis berdasarkan hasil pengujian.

Tabel 3.7 Release Backlog Berdasarkan Iterasi Sprint

Code Sprint	Sprint	Fitur Dirilis	Score Evaluasi (Blackbox/SUS)	Status
SP1	Sprint 1	Login, CRUD Pengguna, Greenhouse, Reservoir, Robot	Contoh: 100% / 82.5	Done
SP2	Sprint 2	Monitoring Real-time Sensor, Kontrol Manual Aktuator	Contoh: 95% / 80.0	Done
SP3	Sprint 3	Dashboard Visual, Notifikasi, Ekspor Data	Contoh: 90% / 77.5	Done
SP4	Sprint 4	Otomasi Sensor dan Jadwal, Logging Otomatis	Contoh: 85% / 72.5	Progress

3.5 Ilustrasi Perhitungan Metode

Penelitian ini menerapkan dua metode evaluasi untuk menilai kelayakan sistem, yaitu pengujian *Blackbox* dan pengukuran usability menggunakan *System Usability Scale* (SUS). Kedua metode ini digunakan untuk mengevaluasi baik dari sisi fungsionalitas sistem maupun persepsi pengguna terhadap kemudahan dan kenyamanan penggunaan. Penjabaran perhitungan dari masing-masing metode disampaikan pada subbagian berikut.

1. Perhitungan Metode Blackbox Testing

Metode *Blackbox Testing* digunakan untuk menguji apakah setiap fitur yang telah dikembangkan dapat berjalan sesuai fungsi yang dirancang tanpa memperhatikan struktur internal kode. Evaluasi dilakukan berdasarkan skenario uji yang telah ditentukan.

$$P = \frac{T_b}{T_s} \times 100\%$$

(Rumus 3.1)

- P = Persentase keberhasilan pengujian (%)
- T_b = Jumlah skenario pengujian yang berhasil
- T_s = Jumlah total skenario pengujian

Sebagai ilustrasi, jika dari 20 skenario pengujian terdapat 19 skenario yang berhasil, maka:

$$P = \frac{19}{20} \times 100\% = 95\%$$

(Rumus 3.2)

2. Perhitungan Metode System Usability Scale (SUS)

Pengukuran usability dilakukan dengan menggunakan instrumen SUS yang terdiri dari 10 pernyataan yang dijawab oleh pengguna menggunakan skala *Likert* 1–5. Skor untuk setiap pernyataan dihitung berdasarkan kriteria berikut:

- Untuk pernyataan ganjil (positif): Skor = Nilai jawaban – 1
- Untuk pernyataan genap (negatif): Skor = 5 – Nilai jawaban

Setelah dikonversi, seluruh skor dijumlahkan dan dikalikan dengan faktor 2.5 untuk mendapatkan skor akhir pada rentang 0 hingga 100. Rumusnya adalah:

$$SUS = \left(\sum_{i=1}^{10} S_i \right) \times 2.5$$

(Rumus 3.3)

- S_i = Skor hasil konversi dari pernyataan ke- i
- SUS = Skor usability akhir (skala 0–100)

Sebagai contoh, jika responden memberikan jawaban berikut:

[4, 2, 5, 2, 4, 2, 5, 1, 4, 1]

Maka konversinya adalah:

Pernyataan ganjil: 3, 4, 3, 4, 3 (dari 4–1, 5–1, *dst.*)

Pernyataan genap: 3, 3, 4, 4, 4 (dari 5–2, 5–1, *dst.*)

$$\sum_{i=1}^{10} S_i = 3 + 3 + 4 + 3 + 3 + 3 + 4 + 4 + 3 + 4 = 34$$

(Rumus 3.4)

$$SUS = 34 \times 2.5 = 85$$

(Rumus 3.5)

Dengan demikian, skor usability sistem adalah **85** yang termasuk dalam kategori sangat baik atau *acceptable*.

3. Tabel Skala Likert dan Pernyataan SUS

Tabel 3.8 Skala Likert untuk Penilaian SUS

Skor	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Netral
4	Setuju
5	Sangat Setuju

Tabel 3.9 Daftar 10 Pernyataan Umum SUS

No.	Pernyataan
1	Saya akan sering menggunakan sistem ini
2	Sistem ini terlalu rumit
3	Sistem ini mudah digunakan
4	Saya merasa perlu bantuan teknis untuk dapat menggunakan sistem ini
5	Fitur-fitur dalam sistem ini terintegrasi dengan baik
6	Sistem ini memiliki terlalu banyak inkonsistensi
7	Sebagian besar orang akan dapat belajar menggunakan sistem ini dengan cepat
8	Sistem ini terasa rumit ketika digunakan
9	Saya merasa percaya diri ketika menggunakan sistem ini
10	Saya harus banyak belajar sebelum dapat menggunakan sistem ini secara efektif

3.6 Rancangan Pengujian

Rancangan pengujian pada penelitian ini disusun untuk mengevaluasi sistem secara menyeluruh baik dari aspek teknis maupun pengalaman pengguna. Evaluasi dilakukan terhadap komponen perangkat keras, fungsionalitas sistem, serta usability antarmuka pengguna berdasarkan setiap iterasi pengembangan

(*sprint*). Setiap hasil pengujian pada tiap sprint akan dirata-rata untuk mendapatkan kesimpulan akhir mengenai kelayakan sistem.

1. Pengujian Perangkat Keras

Pengujian perangkat keras difokuskan pada dua aspek utama:

- **Delay Sensor ke Sistem:** Mengukur waktu yang dibutuhkan oleh data sensor untuk sampai ke dashboard.
- **Delay Perintah ke Aktuator:** Mengukur waktu respon aktuator terhadap perintah dari dashboard.

Nilai toleransi keterlambatan sistem yang dapat diterima adalah ; 1000 milidetik. Hasil pengujian disajikan dalam bentuk tabel dan grafik komparatif.

2. Pengujian Perangkat Lunak

Evaluasi perangkat lunak dilakukan dari dua aspek:

a. Pengujian Fungsional (Blackbox Testing)

$$P = \frac{T_b}{T_s} \times 100\%$$

(Rumus 3.6)

Tabel 3.10 Rekap Skor Blackbox per Sprint

Sprint	Jumlah Skenario	Tingkat Keberhasilan (%)
Sprint 1	10	100%
Sprint 2	8	95%
Sprint 3	9	90%
Sprint 4	10	85%

$$P_{rata2} = \frac{100 + 95 + 90 + 85}{4} = 92.5\%$$

(Rumus 3.7)

b. Pengujian Non-Fungsional (System Usability Scale)

$$SUS = \left(\sum_{i=1}^{10} S_i \right) \times 2.5$$

(Rumus 3.8)

Tabel 3.11 Rekap Skor SUS per Sprint

Sprint	Nilai SUS
Sprint 1	82.5
Sprint 2	80.0
Sprint 3	77.5
Sprint 4	72.5

$$SUS_{rata2} = \frac{82.5 + 80.0 + 77.5 + 72.5}{4} = 78.125$$

(Rumus 3.9)

3. Hipotesis dan Evaluasi Kelayakan

Berdasarkan hasil pengujian:

- Rata-rata keberhasilan pengujian fungsional adalah **92.5%**, melebihi ambang 90%.
- Skor SUS rata-rata adalah **78.125**, berada di atas batas *acceptable* yaitu 70.
- Rata-rata delay komunikasi sistem berada di bawah 1000 ms, masih dalam batas sistem real-time.

Sistem dashboard *smart farming* diperkirakan memenuhi kriteria kelayakan secara teknis, fungsional, dan usability berdasarkan hipotesis awal. Validasi terhadap asumsi ini disajikan pada Bab 4.1 melalui analisis hasil pengujian empiris.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Berisi hasil penelitian berdasarkan rancangan yang sudah dijelaskan pada Bab III, terutama dari Subbab 3.4. Bagi yang membuat alat, jelaskan alat yang jadi dalam bentuk apa. Bagi yang membuat aplikasi, jelaskan aplikasi yang jadi dalam bentuk seperti apa. Jabarkan dalam bentuk pseudocode dan dijelaskan per bagian kodenya. Gunakan gambar dan tabel sebagai alat bantu menjelaskan hasil.

Contoh implementasi kode dapat ditulis menggunakan `\begin{lstlisting}`. Contoh kode dapat dilihat pada Kode 4.1.

Kode 4.1 Akuisisi Gambar

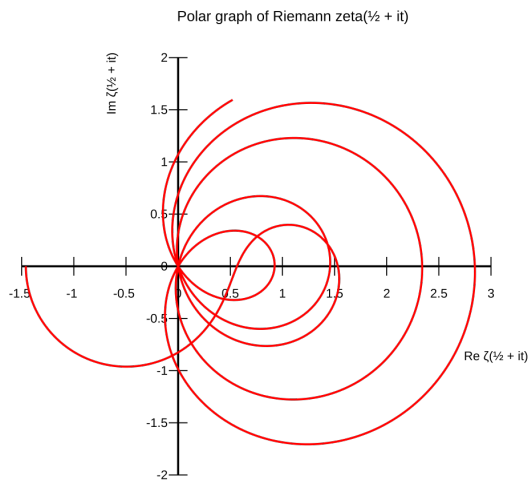
```
1 def process_dataset(dataset_path):
2     image_files = glob(os.path.join(dataset_path, '*.png'))
3     image_files.sort()
4     for image_file in image_files:
5         frame = cv2.imread(image_file)
6         if frame is None:
7             continue
8         frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
9         cv2.imshow('Frame', frame)
10        if cv2.waitKey(1) & 0xFF == ord('q'):
11            break
12    cv2.destroyAllWindows()
13 def main():
14     datasets = get_all_dataset_folders(DATASET_ROOT)
15     for dataset in datasets:
16         process_dataset(dataset)
17         print("print string")
```

4.2 Hasil Pengujian

Berikan hasil pengujian berdasarkan rancangan & skenario yang sudah direncanakan sebelumnya pada Subbab 3.6.

Tabel 4.1 Data *dummy* Pengujian

Subjek	Hasil Prediksi (BPM)							GT
	F	NA	NO	RC	LC	M	C	
1	68	69	68	70	68	71	69	68
2	69	69	68	70	68	71	69	69
3	70	70	69	71	68	73	69	70
4	71	70	70	72	69	73	70	71
5	72	72	70	72	70	74	70	72



Gambar 4.1 Contoh Graf Pengujian

4.3 Analisis Hasil Penelitian

Berikan analisis hasil penelitian & pengujian, berupa data yang didapatkan dari penelitian & pengujian Tugas Akhir yang sudah anda kerjakan. Gunakan gambar dan tabel sebagai alat bantu menjelaskan analisis hasil. Data luaran penelitian yang dapat dianalisis berupa:

- 1. Hasil pengujian

2. Hasil kuesioner

3. Aplikasi yang dikembangkan

Analisis dapat membandingkan dengan hasil penelitian sebelumnya yang memiliki kemiripan topik.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berisi kesimpulan dari hasil dan pembahasan terkait penelitian yang dilakukan, dapat juga berupa temuan yang Anda dapatkan setelah melakukan penelitian atau analisis terhadap tugas akhir Anda. Memberikan jawaban dari poin pada subbab Rumusan Masalah dan Tujuan Penelitian.

5.2 Saran

Berisi saran mengenai aspek tugas akhir atau temuan yang dapat dikembangkan dan diperkaya di tugas akhir selanjutnya. Saran dapat berkaitan erat pada subbab Analisis Hasil Penelitian.

DAFTAR PUSTAKA

- [1] AA Dani Kadir Wahyudi Endro Gunawan. “Analisis Produk Domestik Bruto (PDB) pada Sektor Pertanian Triwulan III 2024”. *Policy Brief Perencanaan Pembangunan* (2024).
- [2] Badan Pusat Statistik. *Produksi Tanaman Sayuran dan Buah–Buahan Semusim Menurut Jenis Tanaman*. Available at: <https://www.bps.go.id/>. Accessed: 2024-09-17. 2023.
- [3] Badan Pusat Statistik. *Produksi Tanaman Sayuran dan Buah–Buahan Semusim Menurut Jenis Tanaman*. Available at: <https://www.bps.go.id/>. Accessed: 2024-09-17. 2024.
- [4] P Yaswanth et al. “Smart Greenhouse: An IoT Based Monitoring and Control System with Predictive Analytics”. *International Journal of Engineering Research Technology (IJERT)* 9 (13 2021).
- [5] M Taufik, R Puspita, and D Kuswandi. “Implementation of System Usability Scale in the Development of Smart Classroom Web-based Application Using Agile Method”. *International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, 2022.
- [6] Márton Hegedüs et al. “Usability testing of a collaborative programming web application using the System Usability Scale and thinking-aloud method”. *Human Factors and Ergonomics in Manufacturing Service Industries* 32 (1 2022), pp. 5–20.
- [7] A Rahman and E Yani. “Usability Evaluation of E-Learning Application using System Usability Scale (SUS)”. *International Conference on Engineering, Technology and Education (ICETE)*. 2022.
- [8] Lalu Yusron Indrawan and Arief Wibisono. “Penerapan Metode Scrum pada Pengembangan Aplikasi Administrasi Kegiatan Mahasiswa

Fakultas Teknik Universitas Muhammadiyah Jakarta”. *Jurnal Teknologi dan Sistem Informasi* 2 (2 2021).

- [9] Dwi Cahyo Purnama et al. “Dashboard Pemantauan Greenhouse Berbasis Web dengan Konsep Smart Farming”. *Seminar Nasional Teknologi dan Rekayasa (SENTRA)*. 2023.

LAMPIRAN

A Dataset

B Hasil Wawancara

C Rincian Kasus Uji