

PENGEMBANGAN *DASHBOARD MOBILE APP* UNTUK *MONITORING INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE* MENGGUNAKAN *PLATFORM ANDROID STUDIO* DAN PEMOGRAMAN KOTLIN

(Skripsi)

Oleh

KIAGUS MUHAMMAD ADE IQBAL RAMADHAN

NPM 2015031085



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
2024**

PENGEMBANGAN *DASHBOARD MOBILE APP* UNTUK *MONITORING INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE* MENGGUNAKAN *PLATFORM ANDROID STUDIO* DAN PEMOGRAMAN KOTLIN

Oleh

KIAGUS MUHAMMAD ADE IQBAL RAMADHAN

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024**

ABSTRAK

PENGEMBANGAN *DASHBOARD MOBILE APP* UNTUK *MONITORING INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE* MENGGUNAKAN *PLATFORM ANDROID STUDIO* DAN PEMROGRAMAN KOTLIN

Oleh

Kiagus Muhammad Ade Iqbal Ramadhan

Teknologi pertanian memainkan peran krusial dalam memenuhi kebutuhan pangan nasional dan mendukung perekonomian Indonesia, namun sektor ini menghadapi tantangan signifikan seperti rendahnya produktivitas, konversi lahan, dan perubahan iklim. Untuk mengatasi tantangan ini, penelitian ini mengembangkan *dashboard mobile app* untuk memantau sistem kontrol cerdas (*Intelligence Control System*) di *smart greenhouse* menggunakan *platform android studio* dan pemrograman kotlin, serta *library Retrofit* untuk mengambil data *API* dari *backend*. Proyek ini dikembangkan dengan menggunakan metode scrum, yang memungkinkan pengembangan aplikasi dilakukan secara *iteratif* dan *incremental*, memastikan kualitas dan fungsionalitas yang optimal pada setiap tahap pengembangan. Aplikasi ini memungkinkan pemantauan *real-time* kondisi tanaman dan lingkungan secara akurat yang dapat diakses kapan saja dan di mana saja. Pengujian aplikasi dilakukan dengan metode *blac-kbox testing* menggunakan *google forms* untuk menyusun skenario dan skala penilaian dari 1 hingga 10, serta *android profiler* dan *app inspection* untuk memeriksa kinerja aplikasi, termasuk konsumsi *CPU*, *RAM*, dan penggunaan kuota internet. Hasil pengujian menunjukkan aplikasi berjalan sesuai yang diharapkan dan menggunakan konsumsi sumber daya yang cukup rendah. Kesimpulannya, aplikasi *dashboard mobile* ini dapat membantu petani dalam mengambil keputusan yang tepat, meningkatkan produktivitas pertanian, dan menyediakan data yang bermanfaat bagi peneliti untuk analisis lebih lanjut, menunjukkan potensi besar dalam mendukung keberlanjutan pangan dan ekonomi melalui inovasi teknologi di sektor pertanian.

Kata kunci : Android Studio, *dashboard mobile app*, *Intelligence Control System*, Kotlin, *smart greenhouse*.

ABSTRACT

PENGEMBANGAN *DASHBOARD MOBILE APP* UNTUK *MONITORING INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE* MENGGUNAKAN *PLATFORM ANDROID STUDIO* DAN *PEMOGRAMAN KOTLIN*

Oleh

Kiagus Muhammad Ade Iqbal Ramadhan

Agricultural technology plays a crucial role in meeting national food needs and supporting Indonesia's economy. However, this sector faces significant challenges such as low productivity, land conversion, and climate change. To address these challenges, this research develops a mobile dashboard app for monitoring an Intelligent Control System in a smart greenhouse using the Android Studio platform and Kotlin programming language, along with the Retrofit library for retrieving API data from the backend. This project is developed using the Scrum method, which allows for iterative and incremental application development, ensuring optimal quality and functionality at each stage. The application enables real-time monitoring of plant and environmental conditions, accessible anytime and anywhere. The application was tested using black-box testing methods with Google Forms to create scenarios and a rating scale from 1 to 10, as well as Android Profiler and App Inspection to assess application performance, including CPU consumption, RAM usage, and internet quota usage. Test results show that the application performs as expected with relatively low resource consumption. In conclusion, this mobile dashboard application can assist farmers in making informed decisions, enhance agricultural productivity, and provide valuable data for researchers for further analysis, demonstrating significant potential in supporting food sustainability and economic growth through technological innovation in the agricultural sector.

keyword : Android Studio, *dashboard mobile app*, *Intelligence Control System*, Kotlin, *smart greenhouse*.

**PENGEMBANGAN *DASHBOARD MOBILE APP* UNTUK *MONITORING INTELLIGENCE CONTROL SYSTEM (ICS) SMART GREENHOUSE*
MENGUNAKAN *PLATFORM ANDROID STUDIO* DAN
PEMOGRAMAN KOTLIN**

Oleh :

KIAGUS MUHAMMAD ADE IQBAL RAMADHAN

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar

SARJANA TEKNIK

Pada

Jurusan Teknik Elektro

Fakultas Teknik Universitas Lampung



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2024

Judul Skripsi

**PENGEMBANGAN DASHBOARD MOBILE
APP UNTUK MONITORING INTELLIGENCE
CONTROL SYSTEM (ICS) SMART
GREENHOUSE MENGGUNAKAN
PLATFORM ANDROID STUDIO DAN
PEMOGRAMAN KOTLIN**

Nama Mahasiswa

Kiagus Muhammad Ade Iqbal Ramadhan

Nomor Pokok Mahasiswa

2015031085

Jurusan

Teknik Elektro

Fakultas

Teknik



1. Komisi Pembimbing

Yetti Yuniati, S.T., M.T.

NIP. 198001132009122002

Dr. Ing Ardian Ulvan, S.T., M.Sc.

NIP. 197311281999031005

2. Mengetahui

Ketua Jurusan Teknik Elektro

Ketua Program Studi Teknik Elektro

Herlinawati, S.T., M.T.

NIP. 197103141999032001

Sumadi, S.T., M.T.

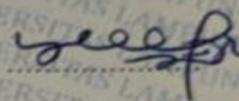
NIP. 197311042000031001

MENGESAHKAN

1. **Tim Penguji**

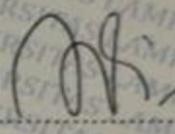
Ketua

Yetti Yuniati, S.T., M.T.



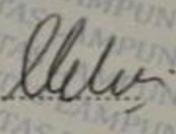
Sekretaris

Dr. Ing Ardian Ulvan, S.T., M.Sc.



Penguji

: Dr. Ing Melvi, S.T., M.T.



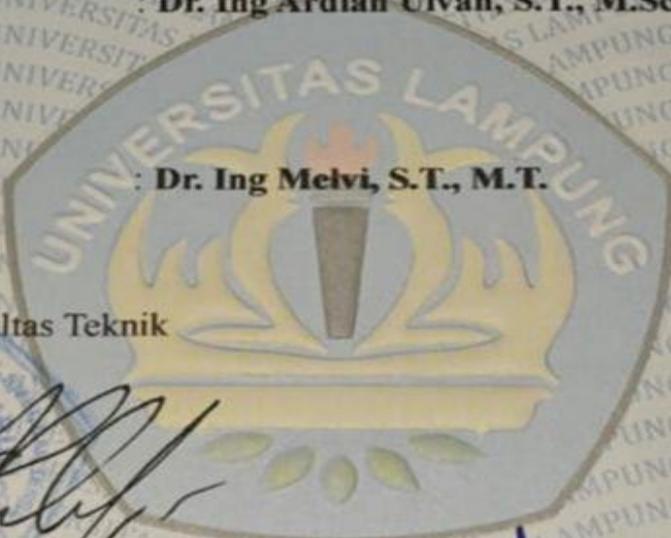
2. **Dekan Fakultas Teknik**

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP. 197509282001121002



Tanggal Lulus Ujian Skripsi : 9 Agustus 2024



SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa dalam skripsi ini tidak terdapat karya yang pernah dilakukan orang lain dan sepanjang pengetahuan saya tidak terdapat atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan dalam daftar pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila pernyataan saya tidak benar, maka saya bersedia dikenai sanksi akademik sesuai dengan hukum yang berlaku.

Bandar Lampung, 26 September 2024



Kiagus Muhammad Ade Iqbal Ramadhan
NPM. 2015031085

RIWAYAT HIDUP



Saya lahir di Palembang, pada tanggal 9 November 2002 sebagai anak bungsu dari 2 bersaudara, anak dari bapak Zainal Abidin dan ibu Taslamiah. Pendidikan Sekolah Dasar diselesaikan di SD IT Kamiliyah Palembang pada tahun 2014, Sekolah Menengah Pertama di SMP Adabiyah Palembang diselesaikan pada tahun 2017, dan Sekolah Menengah Atas di SMA Negeri 5 Palembang diselesaikan pada tahun 2020. Pada tahun 2020 saya terdaftar sebagai mahasiswa di Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung melalui jalur SBMPTN (tes tertulis). Keputusan memilih Teknik Elektro sebagai jurusan studi saya didasarkan pada ketertarikan yang mendalam terhadap dunia teknologi yang terus berkembang pesat. Saya percaya bahwa menjadi bagian dari perkembangan ini adalah langkah yang tepat untuk mengikuti arus zaman yang terus berubah dan menuntut keahlian di bidang teknologi.

Selama menempuh pendidikan di Jurusan Teknik Elektro, saya telah mengalami berbagai mata kuliah yang mendalam mengenai prinsip dasar teknik elektro, pemrograman, dan aplikasi teknologi terkini. Saya juga aktif terlibat dalam kegiatan organisasi kemahasiswaan yang membantu memperluas wawasan saya di luar kurikulum akademik. Salah satu pengalaman paling berharga saya adalah menjadi bagian dari Himpunan Mahasiswa Teknik Elektro (Himatro Unila). Selama dua periode kepengurusan, saya bergabung dengan Departemen Komunikasi dan Informasi (Kominfo), dimana saya ikut bertanggung jawab dalam bagian hubungan masyarakat dan berbagai program kerja yang bertujuan untuk membangun relasi dan menjalin hubungan positif dengan berbagai perusahaan dan lembaga kemahasiswaan.

Selain itu, saya juga aktif mengikuti berbagai kegiatan dan penelitian di luar kelas sebagai bagian dari komitmen saya untuk terus mengembangkan diri. Pada semester enam, saya mengambil bagian dalam program Magang dan Studi Independen Bersertifikat (MSIB) Kampus Merdeka yang diselenggarakan oleh Kemendikbudristek. Program ini memberikan kesempatan kepada mahasiswa untuk memperluas

pengetahuan dan pengalaman mereka di bidang yang mereka minati. Saya memilih untuk fokus pada bidang *Android Engineering* dan mengambil program studi independen di *Binar Academy*. Sebagai bagian dari program ini, saya diberikan tugas-tugas yang relevan dengan bidang *android engineer*, seperti mengimplementasikan bahasa pemrograman dan *tools* untuk membuat aplikasi android, melakukan *testing* pada aplikasi *android*, dan mempelajari sejarah dan perkembangan mengenai *android*. Selain itu, saya juga diajak untuk terlibat dalam proyek-proyek sederhana yang memungkinkan saya untuk mengaplikasikan pengetahuan yang telah saya pelajari.

Selanjutnya pada semester tujuh, saya bergabung dalam penelitian dosen yang berjudul *AGRI-ICS : Produk Intelligence Control System untuk Smart Greenhouse*. Dalam pelaksanaan program ini, saya bertujuan untuk mengembangkan sebuah sistem yang dapat *monitoring* pertanian menggunakan sebuah *dashboard mobile* dengan menggunakan teknologi *Intelligence Control System (ICS)*. *Dashboard mobile* tersebut akan memberikan informasi akurat dan secara *real-time* mengenai informasi data sensor ICS yang telah di pasang di *smart greenhouse*. Dalam proyek ini saya memegang tanggung jawab dalam mengembangkan sistem untuk *dashboard mobile app* berbasis android. Partisipasi saya dalam tim ini tidak hanya memberikan saya pengalaman dalam melakukan riset ilmiah, tetapi juga mengasah kemampuan kolaborasi tim, pemecahan masalah, dan menerapkan pengetahuan saya yang telah saya dapat dari studi independen yang saya ikuti sebelumnya.

Semua pengalaman ini telah membentuk saya menjadi individu yang memiliki komitmen, semangat, dan siap untuk menghadapi tantangan di jalur karir teknik saya yang akan datang. Saya bertekad untuk terus memberikan kontribusi positif dalam bidang teknik elektro, serta berdampak baik untuk masyarakat.



PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dengan Ridho Allah SWT

Teriring shalawat kepada Nabi Muhammad SAW kupersembahkan Skripsi ini sebagai tanda terimakasih kepada:

Papa dan Mama Tercinta

Zainal Abidin Dan Taslamiah

Dan Kakak Tersayang

Cici Meliana Zaita

Serta Keluarga besarku yang banyak berjasa

Selalu memberikan dukungan, kasih sayang, dan dorongan moral yang tak terhingga selama perjalanan studi saya. Atas doa dan semangat yang selalu mengiringi setiap langkah saya dalam mengejar mimpi terima kasih untuk do'a dan dukungan yang telah diberikan.



MOTTO HIDUP

“Khoirunnas Anfa’uhum Linnas, Sebaik-baik manusia adalah yang bermanfaat bagi manusia lainnya.”

(HR. Ahmad)

“Hatiku tenang setelah mengetahui apa yang melewatkanmu tidak akan pernah menjadi takdirku, dan apa yang ditakdirkan untukmu tidak akan pernah melewatkanmu”

(Umar bin Khatab)

“Cukuplah Allah menjadi penolong kami dan sebaik-baik pelindung”

(Q.S Al Imran:173)

“Jangan biarkan rencana yang kita ekspektasikan, merusak rencana indah yang Tuhan siapkan.”

(Penulis)

SANWACANA

Puji syukur kehadirat Allah SWT atas segala karunia, hidayah, serta inayah-Nya kepada penulis, sehingga laporan skripsi ini yang berjudul “**Pengembangan Dashboard Mobile App Untuk Monitoring Intelligence Control System (ICS) Smart Greenhouse Menggunakan Platform Android Studio dan Pemrograman Kotlin**” dapat diselesaikan tepat waktu. Skripsi ini merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Fakultas Teknik Universitas Lampung.

Shalawat serta salam selalu terecurah kepada junjungan seluruh alam, Nabi Muhammad SAW. Sahabatnya, serta para pengikutnya yang selalu istiqomah diatas jalan agama islam hingga akhir zaman. Selama menjalani pengerjaan Skripsi ini, penulis mendapatkan bantuan pemikiran maupun dorongan moril dari berbagai pihak. Oleh karena itu dalam kesempatan kali ini, penulis ingin mengucapkan terimakasih kepada:

1. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M., selaku Rektor Universitas Lampung.
2. Bapak Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlinawati, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
4. Ibu Yetti Yuniati, S.T., M.T. selaku Dosen Pembimbing Utama yang telah memberikan ilmu, bimbingan, bantuan, masukan, dan pandangan serta berbagai pola pikir kepada penulis disetiap kesempatan.
5. Bapak Dr. Ing. Ardian Ulvan, S.T., M.Sc. selaku Dosen Pembimbing Pendamping yang telah memberikan ilmu, bimbingan, masukan, motivasi, dukungan, dan membantu penyelesaian masalah kepada penulis.
6. Ibu Dr. Ing. Melvi, S.T., M.T. selaku Dosen Penguji yang telah memberikan kritik, saran, solusi, dan memberikan sudut pandang lain serta berbagai pola pikir kepada penulis.

7. Bapak Dr. Herman Halomoan Sinaga S.T., M.T. selaku Dosen Pembimbing Akademik yang telah memberikan arahan, nasehat, dan bantuan kepada penulis selama menempuh perkuliahan.
8. Segenap Dosen di Jurusan Teknik Elektro yang telah memberikan ilmu yang bermanfaat kepada penulis selama menempuh pendidikan perkuliahan.
9. Gisting Team selaku tim dari Penelitian AGRI-ICS yang telah memberikan bantuan dan dukungan serta waktu dan kerjasama yang telah diluangkan dalam penelitian ini.
10. Segenap staff di Jurusan Teknik Elektro yang telah membantu penulis baik dalam hal administrasi dan lain-lain.
11. Papa, Mama, dan Mba tersayang yang telah memberikan dukungan moril dan materil selama menjalani pendidikan di Universitas Lampung.
12. Keluarga Besar Telti 2020 yang memberikan banyak ilmu, masukan, dukungan, dan saran kepada penulis yang tidak ternilai.
13. Keluarga besar HELLIOS Angkatan 2020, yang telah memberikan banyak bantuan dan dukungan dalam berbagai hal
14. Keluarga besar HIMATRO UNILA, yang telah menjadi wadah dalam mengembangkan nilai-nilai organisasi bagi penulis.

Semua pihak yang tidak dapat disebutkan satu persatu dan terlibat langsung maupun tidak langsung yang telah membantu penulis dalam pembuatan skripsi. Semoga Allah SWT membalas semua perbuatan dan kebaikan yang telah diberikan kepada penulis sampai dengan terselesaikannya skripsi ini. Penulis menyadari bahwa laporan skripsi ini masih memiliki banyak kekurangan, baik dari segi penyusunan maupun pemilihan kata. Maka dari itu penulis terbuka untuk menerima kritikan dan saran yang dapat membangun penulis kedepannya. Semoga penulisan skripsi ini dapat bermanfaat bagi pembaca.

Bandar Lampung 26 September 2024

Kiagus Muhammad Ade Iqbal Ramadhan
NPM. 2015031085

DAFTAR ISI

	Halaman
ABSTRAK	II
ABSTRACT	III
SURAT PERNYATAAN	V
RIWAYAT HIDUP	VI
PERSEMBAHAN.....	VIII
MOTTO HIDUP	IX
SANWACANA	X
DAFTAR ISI.....	XI
DAFTAR GAMBAR.....	XIII
DAFTAR TABEL	XV
BAB I. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
BAB II. TINJAUAN PUSTAKA.....	5
2.1 Penelitian Terkait	5
2.2 Application Programming Interface (API).....	8
2.3 Github.....	8
2.4 Dashboard	9
2.5 Android Studio.....	9
2.6 Kotlin	10
2.7 <i>Testing Method</i>	10
2.7.1 <i>Black-box Testing</i>	11
2.7.2 <i>Android Profiler</i>	11
2.7.3 <i>App Inspection</i>	11
2.8 RAM, CPU, dan Internet	12
2.9 <i>Scrum Method</i>	12

BAB III. METODOLOGI PENELITIAN	13
3.1 Waktu dan Tempat Penelitian	13
3.2 <i>Capstone Project</i>	13
3.3 Alat dan Bahan	15
3.3.1 Alat	15
3.3.2 Bahan	17
3.4 Tahapan Penelitian	18
3.4.1 Studi Literatur	19
3.4.2 Analisa Kebutuhan	19
3.4.3 Pengembangan Sistem	20
3.4.4 Analisa Keberhasilan	22
BAB IV. HASIL DAN PEMBAHASAN	24
4.1 Gambaran Umum Aplikasi	24
4.2 Implementasi Antarmuka Aplikasi <i>Smart Greenhouse</i>	26
4.3 <i>Testing Mobile App</i>	35
4.3.1 <i>Black-box Testing Method</i>	35
4.3.2 Pemantauan Kinerja Aplikasi Menggunakan <i>Android Profiler</i>	36
4.3.3 Analisis Konsumsi Kuota Internet Menggunakan <i>App Inspection</i>	63
BAB V. KESIMPULAN DAN SARAN	65
5.1 Kesimpulan	65
5.2 Saran	66
DAFTAR PUSTAKA	67

DAFTAR GAMBAR

	Halaman
Gambar 2.1 <i>Icon Platform Github</i>	8
Gambar 2.2 <i>Icon Platform Android Studio</i>	10
Gambar 3.1 Diagram Keseluruhan Pengembangan System	14
Gambar 3.2 Gambar Diagram Alur Alat dan Bahan.....	15
Gambar 3.3 Diagram Alir Tahapan Penelitian.....	18
Gambar 3.4 Tahapan Pengembangan Sistem dengan <i>Scrum Method</i>	20
Gambar 3.5 Indikator Pengujian	22
Gambar 4.1 <i>Flowchart</i> Pembuatan Aplikasi <i>Monitoring Smart Greenhouse</i>	24
Gambar 4.2 <i>Flowchart</i> Sistem yang Dikembangkan	25
Gambar 4.3 <i>Flowchart Layout Mobile App</i>	26
Gambar 4.4 Antarmuka <i>Home Screen</i> Aplikasi <i>Dashboard Monitoring Smart Greenhouse</i>	27
Gambar 4.5 Antarmuka <i>Nutrient Tank Screen</i> Aplikasi <i>Smart Greenhouse</i>	29
Gambar 4.6 Antarmuka <i>Menu Microclimate</i> pada Aplikasi <i>Smart Greenhouse</i> ..	30
Gambar 4.7 Antarmuka <i>Menu Soil</i> dalam Aplikasi <i>Smart Greenhouse</i>	31
Gambar 4.8 Antarmuka Tabel pada Aplikasi <i>Smart Greenhouse</i>	32
Gambar 4.9 Antarmuka Grafik pada Aplikasi <i>Smart Greenhouse</i>	33
Gambar 4.10 Halaman <i>Info</i> pada Aplikasi <i>Smart Greenhouse</i>	34
Gambar 4.11 Analisa Konsumsi CPU Dan RAM pada <i>Home Layout</i> Aplikasi ...	38
Gambar 4.12 Analisa Konsumsi CPU Dan RAM pada <i>Nutrient Layout</i>	40
Gambar 4.13 Analisa Konsumsi CPU Dan RAM pada <i>Microclimate Layout</i>	42
Gambar 4.14 Analisa Konsumsi CPU Dan RAM pada <i>Soil Layout</i>	44
Gambar 4.15 Analisa Konsumsi CPU Dan RAM pada <i>Info Layout</i>	46

Gambar 4.16 Analisa Konsumsi CPU Dan RAM pada <i>Home Layout</i> Aplikasi ...	49
Gambar 4.17 Analisa Konsumsi CPU Dan RAM pada <i>Nutrient Layout</i>	51
Gambar 4.18 Analisa Konsumsi CPU Dan RAM pada <i>Microclimate Layout</i>	53
Gambar 4.19 Analisa Konsumsi CPU Dan RAM pada <i>Soil Layout</i>	56
Gambar 4.20 Analisa Konsumsi CPU dan RAM pada <i>Table Layout</i>	58
Gambar 4.21 Analisa Konsumsi CPU dan RAM pada <i>Graph Layout</i>	59
Gambar 4.22 Analisa Konsumsi CPU dan RAM pada <i>Info Layout</i>	60
Gambar 4.23 Analisa Konsumsi Kuota Internet pada <i>Mobile App</i>	63

DAFTAR TABEL

	Halaman
Tabel 3.1 Alat Berupa <i>Hardware</i> dan <i>Software</i> yang digunakan.	16
Tabel 3.2 Bahan Berupa Sensor ICS.....	17
Tabel 4.3.1 Hasil Penilaian Pengujian <i>Black-box Method</i>	35

BAB I. PENDAHULUAN

1.1 Latar Belakang

Pertanian di Indonesia memiliki peran yang sangat penting dalam memenuhi kebutuhan pangan nasional dan mendukung perekonomian negara [1]. Sektor pertanian di Indonesia memiliki peluang yang besar dalam pentas ekonomi dunia, dan ini nantinya akan menunjang peningkatan perekonomian Indonesia jika benar-benar dimanfaatkan dengan baik. Pembangunan Pertanian di Indonesia tetap dianggap terpenting dari keseluruhan pembangunan ekonomi, apalagi semenjak sektor pertanian ini menjadi penyelamat perekonomian nasional karena pertumbuhannya yang meningkat, sementara sektor lain pertumbuhannya *negative* [2]. Meskipun Indonesia kaya akan sumber daya alam, namun rendahnya produktivitas, konversi lahan pertanian, dan perubahan iklim menjadi hambatan utama dalam mengoptimalkan sektor pertanian [3]. Peran teknologi dibutuhkan dalam pertanian di Indonesia yang memainkan peran sangat penting dalam memenuhi kebutuhan pangan nasional dan mendukung perekonomian negara, di era teknologi yang pesat ini. Untuk memastikan keberlanjutan pangan dan ekonomi, pertanian Indonesia membutuhkan pendekatan yang terus berkembang dan berinovasi [4]. Implementasi *Intelligence Control System (ICS)* telah menjadi pendorong utama perubahan dalam industri pertanian di tengah perkembangan teknologi yang pesat [5].

Penggunaan teknologi yang lebih canggih dan peningkatan konektivitas memungkinkan pemantauan pertanian yang lebih akurat dan efisien. Pertanian yang efektif membutuhkan pemantauan yang akurat tentang kondisi tanah, suhu, kelembapan, dan faktor lingkungan lainnya [6]. Perubahan iklim dapat menyebabkan variasi suhu dan kelembapan yang signifikan, yang dapat berdampak negatif pada pertumbuhan dan kesehatan tanaman [7]. Untuk memahami dan merespons kondisi pertanian secara tepat waktu, petani membutuhkan solusi yang mudah digunakan dan modern [8]. Akibatnya, diperlukan alat yang dapat membantu petani untuk memantau pertanian dengan teknologi *Intelligence Control*

System (ICS) yang berbasis *Internet of Thing* (IoT) dengan kemudahan penggunaan melalui aplikasi *mobile* [9].

Berdasarkan hal tersebut, penelitian ini merancang sebuah sistem untuk mengolah data dan menampilkannya kedalam *dashboard* yang informatif di *mobile app*. Sehingga pengguna mudah untuk memahami dan *monitoring* kondisi tanaman dan lingkungan di area *greenhouse* secara *real-time*. Dengan data secara *real-time* dan akurat yang dapat diakses dimana saja dan kapan saja. Implementasi *dashboard mobile app* ini menggunakan *platform* Android Studio dan pemrograman Kotlin, dengan menggunakan *library* Retrofit untuk mengambil data API dari *back-end*.

Melalui integrasi teknologi ini, diharapkan penelitian ini dapat membantu petani untuk mengambil keputusan yang tepat sehingga memungkinkan untuk meningkatkan produktivitas pertanian dan membantu peneliti terkait untuk menganalisa data yang telah disajikan.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Bagaimana data yang dikirim oleh sensor-sensor yang digunakan pada ICS dapat divisualisasikan secara *real-time* pada sebuah *dashboard mobile*.
2. Bagaimana *dashboard mobile app* dapat meningkatkan efisiensi *monitoring* pada *Intelligence Control System* (ICS) di *smart greenhouse*.
3. Bagaimana ketersediaan data melalui *mobile app* dapat mendukung pengambilan keputusan yang lebih baik dalam manajemen *smart greenhouse*.
4. Bagaimana implementasi menggunakan *platform* Android Studio dengan *library* Retrofit dalam pemrograman kotlin untuk integrasi antara *dashboard mobile app* dan API dari *backend*.

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini adalah:

1. Penelitian ini berfokus pada mengimplementasikan *platform* Android Studio dengan pemrograman kotlin menggunakan *library* Retrofit yang terintegrasi dengan API yang didapat dari *backend* dapat memvisualisasikan data secara

real-time pada *dashboard mobile app*.

2. Penelitian ini tidak mencakup bagaimana proses pembuatan API dan alat sensor *Intelligence Control System (ICS)* pada *smart greenhouse*.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Mengembangkan *dashboard* yang bertujuan untuk melakukan visualisasi data *real-time* yang berasal dari sensor ICS.
2. Mengimplementasikan *dashboard mobile app* menggunakan *platform Android Studio* dengan bahasa pemrograman Kotlin dan *library Retrofit* untuk komunikasi antara *dashboard mobile app* dan API yang diperoleh dari *backend*.

1.5 Manfaat Penelitian

Beberapa manfaat dari penelitian ini adalah :

1. Membantu petani dan peneliti memantau data secara *real-time* dari mana saja menggunakan data sensor yang dikirimkan ICS melalui *dashboard mobile app*.
2. Memudahkan para peneliti dan petani untuk menganalisa data yang telah disajikan.
3. Dengan data *real-time* dan akurat, dapat membantu pengambilan keputusan yang lebih tepat yang memungkinkan meningkatkan produktivitas tanaman.

1.6 Sistematika Penulisan

Sistematika penulisan yang digunakan adalah sebagai berikut :

Bab I : Pendahuluan

Pada bab ini menjelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian serta sistematika penulisan laporan mengenai penelitian pengembangan *dashboard mobile app* untuk *monitoring intelligence control system (ICS) smart greenhouse* menggunakan *platform Android Studio* dan pemrograman kotlin.

Bab II : Tinjauan Pustaka dan Landasan Teori

Membahas tentang penelitian-penelitian sebelumnya pada tinjauan pustaka, dan dasar-dasar teori dari penelitian pengembangan *dashboard mobile app* untuk *monitoring intelligence control system (ICS) smart greenhouse* menggunakan *platform* Android Studio dan pemrograman Kotlin.

Bab III : Metode Penelitian

Menjelaskan waktu, tempat, alat dan bahan, dan metode penelitian beserta tahapannya mengenai penelitian pengembangan *dashboard mobile app* untuk *monitoring intelligence control system (ICS) smart greenhouse* menggunakan *platform* Android Studio dan pemrograman kotlin.

Bab IV : Hasil dan Pembahasan

Menjelaskan hasil dan pembahasan penelitian mengenai penelitian pengembangan *dashboard mobile app* untuk *monitoring intelligence control system (ICS) smart greenhouse* menggunakan *platform* Android Studio dan pemrograman kotlin.

Bab V : Simpulan dan Saran

Menjelaskan kesimpulan dan saran mengenai penelitian pengembangan *dashboard mobile app* untuk *monitoring intelligence control system (ICS) smart greenhouse* menggunakan *platform* Android Studio dan pemrograman kotlin.

Daftar Pustaka

Lampiran

BAB II. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian sebelumnya dilakukan oleh Fenty Ariani, Robby Yuli Endra, Erlangga Erlangga, Yuthsi Aprlinda, dan Ananta Reza Bahar dari Fakultas Ilmu Komputer, Universitas Bandar Lampung, pada tahun 2020 dengan Judul “Sistem *Monitoring* Suhu dan Pencahayaan Berbasis *Internet of Thing (IoT)* untuk Penetasan Telur Ayam”. penelitian ini membuat pemantauan terhadap suhu dan pencahayaan sehingga tidak perlu lagi datang langsung ke kandang. Penelitian ini bertujuan yaitu untuk meningkatkan *quantity* pada penetasan telur ayam serta mempercepat waktu panen telur ayam dengan penerapan teknologi *IoT (Internet of Thing)* sebagai alternatif untuk mengatasi permasalahan tersebut melalui aplikasi berbasis *mobile*.

Perbedaan penelitian yang dilakukan oleh Fenty Ariani, Robby Yuli Endra, Erlangga Erlangga, Yuthsi Aprlinda, dan Ananta Reza Bahar dengan penelitian yang dilakukan oleh penulis ialah pada penelitian ini hanya *monitoring* suhu dan pencahayaan pada *incubator* untuk penetasan telur ayam dan ketika suhu berada pada nilai tertentu maka lampu dan kipas akan aktif otomatis. Sedangkan pada penelitian penulis ini, *monitoring* berbagai sensor seperti suhu ruang, kelembapan, suhu air, ph, kecepatan angin, arah angin, dan jika suhu berada pada kondisi tertentu akan mengaktifkan *cooling system* secara otomatis. Sedangkan persamaan penelitian ini yaitu, sama sama menggunakan aplikasi *monitoring* yang berbasis *mobile*.

Kesimpulan dari penelitian yang dilakukan oleh Fenty Ariani, Robby Yuli Endra, Erlangga Erlangga, Yuthsi Aprlinda, dan Ananta Reza Bahar adalah telah berhasil membuat sebuah sistem *monitoring* suhu dan pencahayaan berbasis *IoT (Internet of Thing)* melalui aplikasi *mobile* untuk penetasan telur ayam yang dapat meningkatnya *quantity* pada penetasan telur ayam serta mempercepat masa panen telur [10].

Selanjutnya terdapat penelitian sebelumnya yang dilakukan oleh Uray Ristian, Ikhwan Ruslianto, dan Kartika Sari pada tahun 2022. Penelitian yang dilakukan ini adalah membangun suatu sistem *Smart Greenhouse* berbasis *IoT (Internet of Things)*, yang mana sistem ini dapat melakukan manipulasi dan rekayasa iklim sesuai dengan kebutuhan yang diperlukan tanaman agar bisa hidup dan tumbuh dengan subur di dalam *smart Greenhouse*. Adapun kondisi yang dimanipulasi dan dikendalikan adalah suhu, air, kelembapan tanah, kelembapan udara, dan cahaya [11].

Ada juga penelitian terkait yang menggunakan *platform* serta bahasa dan *tools* yang sama pada penelitian saat ini yang dilakukan oleh Rangga Satria Wirabangsa, Dwi Ratnasari, Giri Wahyu Wiriasto pada tahun 2022 yang membuat aplikasi peminjaman barang berbasis android. Aplikasi android ini juga menggunakan *library* Retrofit untuk mengakses API. Retrofit ini melakukan proses pertukaran data dari format json atau xml, Adapun *method* yang digunakan yaitu *get* untuk mendapatkan data dari server, *put* untuk memperbarui data pada server, *post* untuk mengirimkan data pada server, dan *delete* untuk menghapus data. Berdasarkan hasil analisis dan pengujian sistem dihasilkan kesimpulan bahwa metode *REST API* telah dapat mengintegrasikan data antara aplikasi dengan berbasis *mobile*. Pengintegrasian ini juga menggunakan *library* Retrofit dan bahasa pemrograman kotlin. Dari hasil yang didapat keseluruhan pemanggilan API berjalan dengan baik dan didapat data mengenai peminjaman barang dan *user* yang melakukan peminjaman [12].

Selanjutnya, pada penelitian yang dilakukan Nirwana Hendrastuty, Yusril Ihza pada tahun 2021 yaitu penelitian yang berupa rancang bangun aplikasi *monitoring* santri berbasis android yang berguna untuk memudahkan para orang tua untuk mengamati kegiatan yang sedang dilakukan anaknya di pesantren, tetapi menggunakan bahasa java untuk membuat aplikasi tersebut yang berbeda dengan penelitian yang dilakukan saat ini yaitu menggunakan bahasa pemrograman kotlin. Bahasa java dan kotlin dikompilasi ke dalam *bytecode* yang sama dan mesin virtual yang sama yaitu *Java Virtual Machine (JVM)* [13].

Berdasarkan penelitian – penelitan tersebut, dapat diketahui apa saja perbedaan dengan penelitian yang sedang dikerjakan ini, yaitu pada penelitian yang dilakukan pada tahun 2020 dan 2021 masih menggunakan bahasa java sedangkan penelitian ini menggunakan bahasa kotlin untuk membuat *mobile app*. Pada penelitian 2022 menggunakan API *firebase* sedangkan penelitian ini menggunakan retrofit, dan penelitian 2023 menggunakan pemrograman kotlin dan *library* retrofit tetapi membuat aplikasi yang berbeda. Keunggulan menggunakan pemrograman Kotlin dan retrofit yaitu mempunya *code* yang lebih sedikit dan punya kontrol penuh atas *back-end*.

2020	2021	2022	2023	2024
Fenty Ariani, Robby Yuli Endra, Robby Yuli Endra, Yuthsi Aprlinda and Ananta Reza Bahar, "Sistem <i>Monitoring</i> Suhu dan Pencahayaan Berbasis <i>Internet of Thing (IoT)</i> untuk Penetasan Telur Ayam".	Nirwana Hendrastuty and Yusril Ihza, "Rancang Bangun Aplikasi <i>Monitoring</i> Santri Berbasis Android".	Uray Ristian, Ikhwan Ruslianto and Kartika Sari, "Sistem <i>Monitoring Smart Greenhouse</i> pada Lahan Terbatas Berbasis <i>Internet of Things (IoT)</i> ".	Rangga Satria Wirabangsa, Giri W and Dwi Ratnasari, "Aplikasi Sistem Administrasi Inventarisasi dan Peminjaman Barang Berbasis Android".	Kgs M. Ade Iqbal Ramadhan, "Pengembangan <i>Dashboard Mobile App</i> Untuk <i>Monitoring Intelligence Control System (ICS) Smart Greenhouse</i> Menggunakan <i>Platform Android Studio</i> Dan Pemograman Kotlin".
Membuat sebuah sistem <i>monitoring</i> suhu dan pencahayaan berbasis <i>IoT (Internet of Thing)</i> melalui aplikasi mobile untuk penetasan telur ayam yang dapat meningkatnya <i>quantity</i> pada penetasan telur ayam serta mempercepat masa panen telur	Perancangan <i>system</i> aplikasi <i>monitoring</i> santri berbasis OOP yaitu menggunakan UML (<i>Unified Modelling language</i>) dan menggunakan <i>tools platform</i> android studio dan bahasa pemrograman java untuk membuat aplikasi berbasis android	Sistem <i>monitoring Smart greenhouse</i> berhasil diimplementasikan dengan membaca kondisi kelembaban tanah, kelembaban udara, suhu dan pH Tanah dan data tersebut dikirimkan ke server untuk ditampilkan ke pengguna sistem yang dapat diakses menggunakan perangkat mobile	Sistem ini terdiri dari <i>develop tools</i> dan <i>web server</i> . <i>Develop tools</i> merupakan tahap pengembangan sistem menggunakan bahasa pemrograman PHP untuk sisi <i>backend web</i> , bootstrap untuk sisi <i>frontend web</i> dan Kotlin <i>Object Oriented Programming</i> dari sisi android.	Membuat sebuah aplikasi mobile menggunakan bahasa pemrograman kotlin dan yang berfungsi untuk memvisualisasikan data yang didapatkan dari ICS yang dikirimkan oleh <i>backend</i> dan diintegrasikan dengan menggunakan <i>library retrofit</i>

Gambar 2.1 *State of the Art* Penelitian Terkait

2.2 Application Programming Interface (API)

Application ProgRAMming Interface (API) adalah sebuah antarmuka yang memungkinkan aplikasi untuk berinteraksi dengan aplikasi atau layanan lainnya secara programatik. API menyediakan kumpulan instruksi dan protokol yang dapat digunakan oleh pengembang untuk membangun aplikasi yang dapat berkomunikasi dengan sistem atau layanan yang telah ada [14].

API memungkinkan pengembang untuk mengakses fungsi dan layanan yang disediakan oleh sistem atau layanan lainnya tanpa harus mengetahui seluruh detail implementasi di belakangnya. Dengan menggunakan API, pengembang dapat mempercepat proses pengembangan, mengurangi biaya, dan meningkatkan efisiensi dalam pengembangan aplikasi karena tidak perlu membangun seluruh fitur dari awal. Contoh penggunaan API adalah ketika sebuah aplikasi mengakses data dari layanan seperti *Facebook*, *X*, atau *Google Maps*. API yang disediakan oleh layanan tersebut memungkinkan aplikasi untuk mengambil data dari layanan tersebut dan menggunakannya dalam aplikasi mereka sendiri [15].

2.3 Github

GitHub adalah *platform hosting* kode sumber berbasis *web* yang sangat populer di kalangan pengembang perangkat lunak. *Platform* ini menyediakan repositori *git* publik dan pribadi yang memungkinkan pengguna untuk menyimpan, mengelola, dan berkolaborasi dalam pengembangan perangkat lunak secara terdistribusi [16]. *GitHub* memiliki fitur-fitur kolaborasi yang kuat, yang memungkinkan tim pengembang bekerja bersama dalam proyek perangkat lunak seperti pelacakan perubahan kode, serta memungkinkan manajemen versi yang efisien untuk meningkatkan kerjasama dan pengembangan aplikasi yang konsisten dan menjamin penyimpanan yang aman [17].



Gambar 2.1 Icon Platform Github

2.4 Dashboard

Dashboard merupakan antarmuka pengguna (*user interface*) dari sistem *monitoring* dan *data logging* ICS serta kondisi *data logger* yang dirancang berbasis *Internet of Things (IoT)*. Pada penelitian ini, *dashboard* dibangun menggunakan *platform* Android Studio dan bahasa pemrograman Kotlin [18].

2.5 Android Studio

Android studio adalah *IDE* pemrograman *android* resmi dari *google* yang dikembangkan oleh *IntelliJ*. Sebelumnya, *IDE* resmi pemrograman *android* adalah *Eclipse*. Tetapi sejak kemunculan *android studio*, *google* telah 'pindah ke lain hati dan menjadikan *android studio* sebagai *IDE* resminya. Dikarenakan sudah meresmikan *android studio*, *google* menghentikan *support ADT* ke *Eclipse* dan *ADT* resmi hanya didapatkan oleh *android studio* [19]. *Android studio* dipilih karena memiliki banyak fitur yang memudahkan para pembuat program terutama *programmer* level dasar. Selain memiliki banyak fitur, *android studio* juga memiliki banyak *library* yang sudah siap untuk digunakan. Walaupun *android studio* lebih banyak menghabiskan *memory*, tetapi hal ini dapat ditutupi dengan kelebihan - kelebihan yang dimiliki oleh *android studio* itu sendiri [20]. *Android studio* adalah sistem operasi operasi seluler yang dikembangkan oleh *google*, berdasarkan kernel *linux* dan dirancang terutama untuk perangkat seluler layar sentuh seperti *smart phone* dan tablet. Mulai Mei 2017, *Android* memiliki dua miliar pengguna aktif setiap bulannya, dan memiliki basis terinstal terbesar di sistem operasi apapun. Aplikasi *android* dikembangkan berdasarkan pada kerangka dan selalu berbasis pola [21]. Untuk mengembangkan *android*, dapat difasilitasi oleh rekomendasi kode untuk memastikan efisiensi dan kualitas pengembangan yang tinggi. *Android studio* adalah lingkungan pengembangan terintegrasi resmi untuk sistem operasi *android google*, yang dibangun di perangkat lunak *JetBrains IntelliJ IDEA* dan dirancang khusus untuk pengembangan *android* [22]. Adapun pada penelitian ini, *android studio* digunakan sebagai *IDEA* untuk menulis *source code* untuk pembuatan aplikasi *android*. *Android studio* mempunyai semua *tools* yang diperlukan untuk membuat aplikasi

android dan mendukung banyak fitur seperti *debugging*, pemecahan kesalahan (*error checking*), dan integrasi dengan sistem kontrol versi seperti Git [23].



Gambar 2.2 Icon Platform Android Studio

2.6 Kotlin

Kotlin adalah bahasa pemrograman yang dikembangkan oleh JetBrains, perusahaan yang juga mengembangkan IDEA Android Studio. Bahasa Kotlin adalah pengembangan dari bahasa Java yang sudah populer sebelumnya. Bahasa Kotlin memiliki fitur-fitur bahasa modern yang lebih dibandingkan bahasa Java [24]. Kotlin adalah bahasa pemrograman berbasis *Java Virtual Machine* (JVM). Kotlin merupakan bahasa pemrograman yang pragmatis untuk android yang mengkombinasikan *object oriented* (OO) dan bahasa fungsional. Kotlin juga bahasa pemrograman yang interoperabilitas yang membuat bahasa ini dapat digabungkan dalam satu *project* dengan bahasa pemrograman Java [25]. Bahasa pemrograman ini juga dapat digunakan untuk pengembangan aplikasi berbasis *desktop*, *web* dan *back-end*. Kotlin awalnya dikembangkan oleh JetBrains, perusahaan dibalik IntelliJ IDEA. Setelah melalui banyak perkembangan, JetBrains merilis kotlin secara *open source* dan kini perkembangannya semakin maju. Google bahkan mendukung penuh kotlin untuk pengembang aplikasi *android* [26]. Adapun pada penelitian ini, kotlin digunakan sebagai bahasa pemrograman untuk membuat aplikasi *android*.

2.7 Testing Method

Metode *testing* yang digunakan adalah *black-kbox testing* dengan menyusun skenario dan skala penilaian dari 1 hingga 10 menggunakan *Google Forms* yang

mencakup pengujian fungsionalitas aplikasi, kompatibilitas perangkat, dan pengalaman pengguna. Pengujian juga dilakukan dengan menggunakan *android profiler* untuk memeriksa kinerja aplikasi dan konsumsi sumber daya untuk aplikasi seperti konsumsi CPU dan RAM. Pengujian *app inspection* digunakan untuk menganalisa konsumsi kuota internet saat menjalankan aplikasi. Pengujian ini memungkinkan analisis mendalam terhadap penggunaan sumber daya untuk memastikan kinerja optimal.

2.7.1 Black-box Testing

Black-Box Testing merupakan untuk menguji suatu perangkat lunak dari segi spesifikasi fungsi tanpa harus menguji desain dan kode program. Pengujian dimaksudkan agar mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang diharapkan. Pengujian *black-box* merupakan suatu teknik pengujian yang memungkinkan anda untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Jadi pengujian *black-box testing* itu sendiri melakukan pengujian yang tidak diketahui kinerja internalnya hanya menguji dari tampilan luarnya (*inteface*), fungsionalitasnya [27] .

2.7.2 Android Profiler

Pengujian aspek *performance efficiency* dilakukan dengan menggunakan *tools* yang disediakan oleh aplikasi *android studio* yakni *android profiler*. *android profiler* merupakan *tools* yang berfungsi untuk melakukan tes performansi dan kinerja aplikasi pada suatu perangkat *mobile*. *android profiler* menyediakan dua *test* pengujian yakni CPU dan *MEMORY*. Secara spesifik, sistem *android profiler* akan mengukur dan menunjukkan persentase waktu *CPU* yang tersedia secara keseluruhan. Pada pengukuran penggunaan memori, *android profiler* akan menunjukkan jumlah memori yang digunakan oleh setiap kategori memori [28].

2.7.3 App Inspection

App Inspection merupakan *tool* dalam Android Studio yang digunakan untuk memantau dan menganalisis aktivitas jaringan aplikasi Android secara *real-time*. Ketika aplikasi dijalankan, *App Inspection* menampilkan semua permintaan

jaringan yang dibuat oleh aplikasi. Sehingga, membantu dalam proses *debugging* dan optimalisasi performa jaringan aplikasi [29].

2.8 RAM, CPU, dan Internet

RAM (Random Access Memory) adalah jenis memori komputer yang digunakan untuk menyimpan data dan instruksi yang sedang aktif digunakan oleh *CPU* saat menjalankan program [30]. *RAM* berfungsi untuk memberikan akses cepat dan sementara kepada data yang dibutuhkan oleh *CPU*, memungkinkan kinerja komputer yang lancar dan responsif. *CPU (Central Processing Unit)* merupakan otak dari komputer yang bertanggung jawab untuk mengeksekusi instruksi-instruksi dari program yang berjalan. *CPU* mengkoordinasikan semua operasi komputasi, termasuk pengolahan data, pengambilan keputusan, dan menjalankan instruksi-instruksi dasar [31]. Internet adalah jaringan *global* yang menghubungkan jutaan perangkat komputer di seluruh dunia, memungkinkan mereka untuk berkomunikasi dan bertukar informasi. Melalui infrastruktur jaringan yang kompleks, internet mendukung *transfer* data, komunikasi email, *browsing web*, *streaming media*, dan aplikasi *online* lainnya, memfasilitasi konektivitas *global* dan akses informasi secara *instan* [32].

2.9 Scrum Method

Scrum adalah metode pengembangan perangkat lunak yang bersifat iteratif dan inkremental, dirancang untuk memberikan fleksibilitas dan respons cepat terhadap perubahan kebutuhan. Dalam Scrum, tim bekerja dalam siklus waktu tetap yang disebut *sprint*, biasanya berlangsung selama dua sampai empat minggu. Setiap *sprint* dimulai dengan *sprint planning*, di mana tim memilih pekerjaan dari daftar *product backlog* untuk diselesaikan dalam *sprint* tersebut. Selama *sprint*, tim mengadakan pertemuan harian singkat (*daily scrum*) untuk memastikan semua anggota tim terkoordinasi. *Sprint* berakhir dengan *review* dan *retrospektif* untuk mengevaluasi hasil kerja yang berupa *increment* dan proses kerja, guna terus meningkatkan efisiensi dan efektivitas tim [33].

BAB III. METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Pelaksanaan penelitian ini dilakukan pada :

Waktu : Desember 2023 – Juli 2024

Tempat : Laboratorium Lapang Terpadu, Fakultas Pertanian, Universitas Lampung.

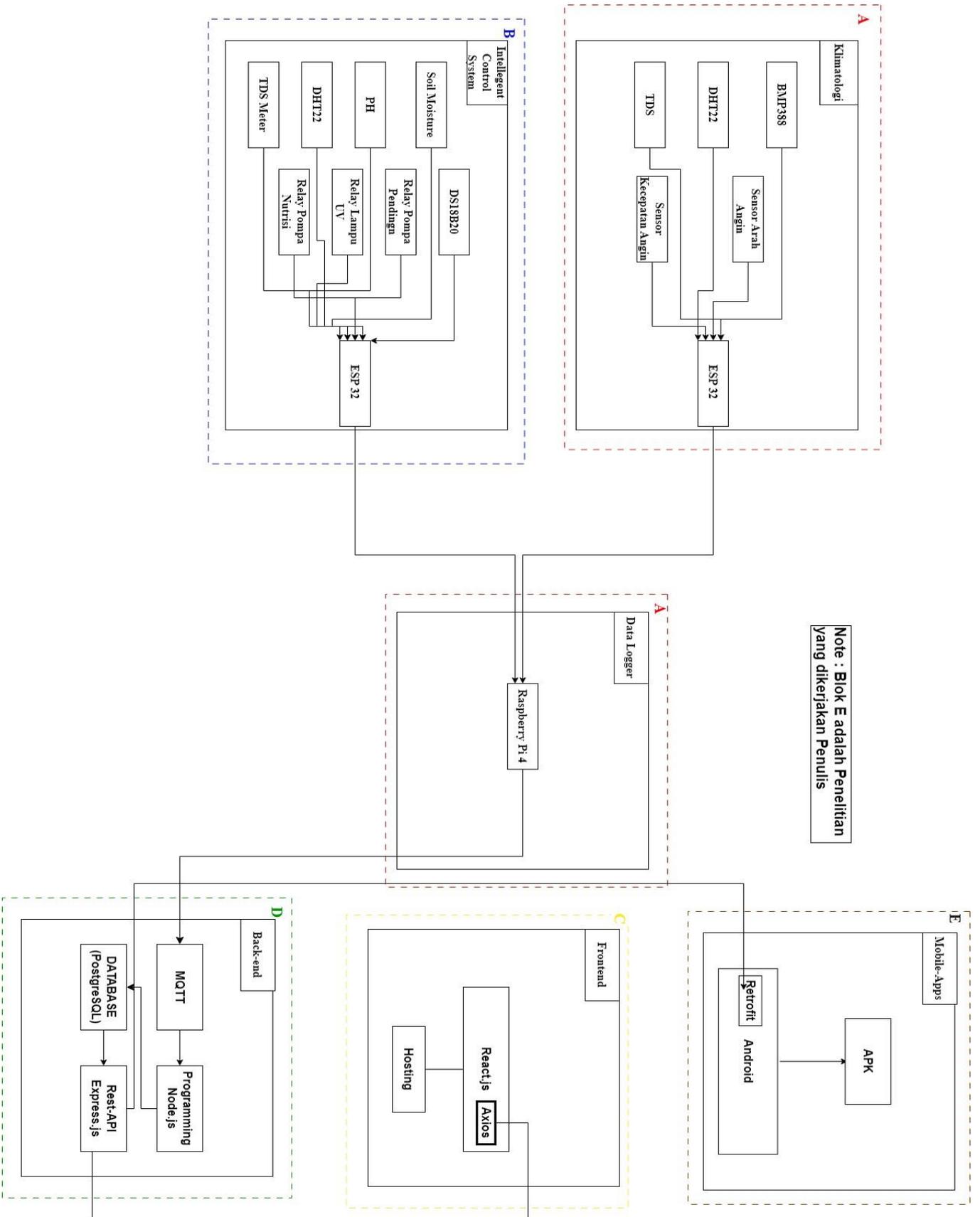
3.2 Capstone Project

Project yang dikerjakan tersebut terdapat beberapa bagian yang akan dikerjakan secara terpisah. Pada diagram keseluruhan *project*, terdapat lima bagian yang dikerjakan secara terpisah. Seperti terlihat pada Gambar 3.1, bagian A merupakan pengerjaan sensor *Intelligence Control System* yaitu *hardware* yang dipasang pada *smart greenhouse* untuk mengetahui kondisi tanaman, lalu bagian B merupakan pengerjaan sensor *microclimate* yaitu sensor untuk mendeteksi kondisi iklim di sekitar *smart greenhouse*, bagian D merupakan pengerjaan *backend* yang mengurus server untuk penghubung *hardware* dan *software* yang dikembangkan, bagian C merupakan pengerjaan *front-end* yang menampilkan data dari sensor yang terpasang di *smart greenhouse* menggunakan *website*, dan terakhir bagian E merupakan pengerjaan *mobile app* yang memvisualisasikan data sensor pada *smart greenhouse* menggunakan aplikasi *mobile*.

Penelitian ini, hanya akan berfokus pada bagian *Mobile App Developer*, sementara bagian lain seperti *Hardware* dan *web* dikerjakan oleh anggota yang lain.

Mobile App developer akan membuat sebuah *dashboard* aplikasi *mobile* yang menampilkan data tiap parameter sensor dari *hardware* yang telah dirancang.

Diagram keseluruhan *project* penelitian dapat dilihat pada Gambar 3.1



Gambar 3.1 Diagram Keseluruhan Pengembangan Sistem

Pada Gambar 3.1 Diagram keseluruhan Pengembangan *System*, pada bagian A dan B merupakan pengerjaan *hardware* yang dirancang pada *greenhouse* untuk mendeteksi kondisi tanaman dan lingkungan. Semua sensor tersebut dihubungkan ke *ESP 32* dan data sensor tersebut *logger* ke *raspberry pi4*. Selanjutnya data sensor – sensor tersebut dikirimkan ke *MQTT* yang berfungsi sebagai fasilitas untuk *publish* data kedalam *topic* yang sudah ditentukan, lalu *topic* tersebut di *subscribe* oleh *node.js* yang berfungsi sebagai tempat untuk mengontrol data yang akan disimpan ke dalam *database*. *Database* berfungsi untuk menyimpan data dan menyediakan data kembali apabila diperlukan. *Rest api* sebagai API yang mengirimkan sejumlah data berdasarkan *request* yang diambil dari *database*. API tersebut diintegrasikan ke *front-end web* menggunakan *axios* dan ke *mobile app* menggunakan *library Retrofit* berbasis android yang menghasilkan *file apk* yaitu *file* untuk *install* dan *mobile app* sehingga *front-end* dan *mobile app* dapat memvisualisasikan data dari sensor – sensor *ICS* dan *microclimate* secara *real-time* dengan tampilan menarik.

3.3 Alat dan Bahan

Alat dan bahan yang digunakan dalam pengerjaan penelitian dan pengembangan sistem yang dibuat adalah sebagai berikut:



Gambar 3.2 Alat dan Bahan yang digunakan

3.3.1 Alat

Adapun alat yang digunakan pada penelitian dapat dilihat pada Tabel 3.2 Alat berupa *Hardware* dan *Software* yang digunakan.

Tabel 3.2 Alat berupa *Hardware* dan *Software* yang digunakan.

No	Perangkat	Spesifikasi	Kegunaan	Keterangan
1	Laptop	<i>Processor Intel Core i3 11400H, RAM 12GB, SSD 512GB, Sistem Operasi Windows</i>	Perangkat pembuatan dan pengujian sistem.	Perangkat keras yang digunakan sebagai <i>compiler</i> dalam pemrograman
2	<i>Virtual Private Server (VPS)</i>	Penyedia: Niagahoster.co.id, Lokasi: Singapore, 3 CPU <i>Core</i> , RAM 3GB, <i>Storage</i> 60GB.	<i>Virtual private server</i> yang digunakan sebagai pusat layanan sistem yang akan dibuat.	Diakses melalui <i>Secure Shell (SSH)</i>
3	Android Studio	Android Studio Koala 2024.1.1	<i>Software</i> yang digunakan untuk membuat aplikasi tersebut baik pada bagian <i>layout</i> ataupun untuk mengintegrasikan API dari <i>back-end</i>	Terpasang di Laptop
4	<i>Git</i>	<i>Git</i> versi 2.39.1	<i>Software</i> yang digunakan untuk menyimpan <i>source code</i> dan untuk <i>update</i> atau pengembangan lebih lanjut pada aplikasi	Terpasang di Laptop

5	Retrofit	Retrofit versi 2.9.0	<i>Library</i> pada Android Studio yang digunakan sebagai penghubung antara aplikasi Android yang dikembangkan menggunakan Android Studio dengan server <i>backend</i> .	Diakses melalui Android Studio
6	<i>Android Profiler</i>	Android Studio Koala 2024.1.1	<i>Tools</i> pada Android Studio yang digunakan untuk menganalisa penggunaan CPU dan RAM pada aplikasi	Diakses melalui Android Studio
7	<i>App Inspection</i>	Android Studio Koala 2024.1.1	<i>Tools</i> pada Android Studio yang digunakan untuk menganalisa penggunaan kuota internet pada aplikasi	Diakses melalui Android Studio

3.3.2 Bahan

Adapun Bahan yang digunakan pada penelitian ini yaitu sensor dari ICS dapat dilihat pada Tabel 3.3 Bahan Berupa Sensor ICS

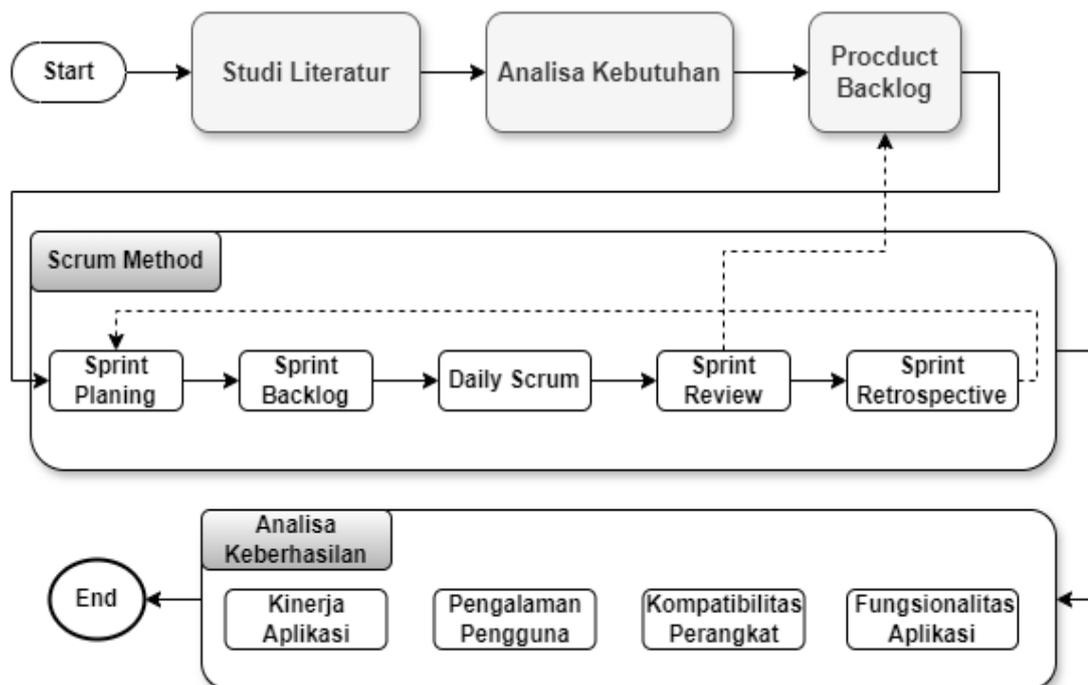
Tabel 3.3 Bahan Berupa Sensor ICS

No	Sensor	Fungsi
1	<i>Timestamp</i>	Sensor yang menunjukkan waktu
2	Ph	Sensor yang menunjukkan Tingkat keasaman dan kebasaan air
3	Suhu air	Sensor yang menunjukkan suhu air
4	<i>Total Dissolve Solid (TDS)</i>	Sensor yang menunjukkan kualitas air
5	Anemometer	Sensor yang menunjukkan arah mata angin

6	<i>Wind speed</i>	Sensor yang menunjukkan kecepatan angin
7	<i>Soil Moisture</i>	Sensor yang menunjukkan kelembapan tanah pada tumbuhan di meja
8	Suhu ruangan	Sensor yang menunjukkan suhu ruangan <i>greenhouse</i>
9	Barometric pressure	Sensor yang mengukur suhu dan tekanan udara pada atmosfer
10	Pompa air	Sensor yang menampilkan apakah pompa air menyala atau mati
11	Pompa nutrisi	Sensor yang menunjukkan apakah pompa nutrisi menyala atau mati
12	Kelembapan	Sensor yang menunjukkan kelembapan di <i>greenhouse</i>

3.4 Tahapan Penelitian

Penelitian ini, pengerjaan dilakukan bertahap mulai dari studi literatur, analisa kebutuhan, lalu dilanjutkan dengan pengembangan sistem dengan metode scrum dan diakhiri dengan Analisa hasil. Diagram alir tahapan penelitian dapat dilihat pada Gambar 3.2



Gambar 3.3 Diagram Alir Tahapan Penelitian

3.4.1 Studi Literatur

Tahapan studi literatur merupakan langkah untuk mencari dan mempelajari berbagai referensi yang diperlukan dengan tujuan memahami konteks dan metode yang sesuai dengan penelitian hingga hasil yang mungkin pada penelitian. Studi literatur dilakukan dengan cara mempelajari terlebih dahulu latar belakang yang menjadi dasar penelitian dan melakukan evaluasi.

3.4.2 Analisa Kebutuhan

Langkah selanjutnya dalam penelitian ini adalah analisa kebutuhan. Analisa kebutuhan diperlukan setelah mempelajari latar belakang yang menjadi dasar penelitian serta literasi yang didapat. Berdasarkan studi pustaka yang telah dilakukan dan melihat kondisi pertanian saat ini, maka diperlukan adanya sensor-sensor *agriculture ICS (Intelligence Control System)* untuk mengoptimalkan berbagai aspek operasional dalam kegiatan pertanian dan *microclimate* untuk mengetahui kondisi iklim dan lingkungan yang akan mempengaruhi kondisi tanaman, serta *dashboard* untuk menampilkan parameter sensor tersebut dan membantu menganalisis data.

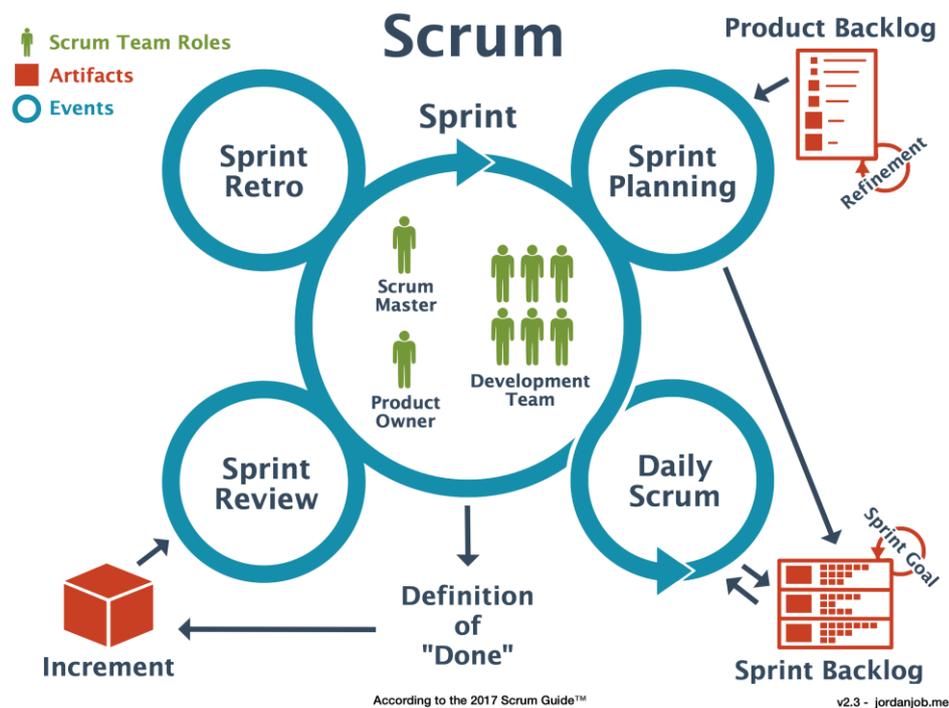
Setelah dilakukan Analisa terhadap pengembangan aplikasi serta terhadap tujuan dari penelitian ini, didapat beberapa kebutuhan yang diperlukan untuk penelitian ini antara lain:

1. Sensor *Agriculture Intelligence Control System (ICS)* dan *microclimate* sebagai perangkat keras yang digunakan untuk mengetahui kondisi tanaman serta lingkungan.
2. *Virtual Private Server (VPS)* yang berfungsi untuk broker MQTT, menyimpan data pada *database*, serta membuat *endpoint* pada API.
3. Data yang didapat berdasarkan sensor *ICS* dan *microclimate* harus dapat di simpan dalam *database* dalam waktu yang lama untuk keperluan penelitian lebih lanjut.
4. Data yang tersimpan dalam database dapat ditampilkan dalam format *Javascript Object Notation (JSON)* untuk keperluan visualisasi data pada *dashboard mobile app*

5. Pembuatan *dashboard* untuk menampilkan data yang telah disimpan dalam *database* sehingga data tersebut dapat ditampilkan dalam sebuah *mobile app*.
6. File apk yang berfungsi untuk membuka *app dashboard* yang dibuat agar dapat diakses.
7. Terakhir, integrasi dan pengujian menyeluruh dari semua komponen sistem harus dilakukan untuk memastikan bahwa sistem bekerja secara harmonis dan dapat mengatasi berbagai skenario operasional. Implementasi yang hati-hati dari kebutuhan-kebutuhan ini akan memastikan keberhasilan penelitian, meningkatkan performa aplikasi, dan memberikan pengalaman pengguna yang optimal.

3.4.3 Pengembangan Sistem

Penelitian ini memiliki tiga metode yang bisa diterapkan yaitu, metode scrum, metode kanban dan *SDLC (Software Development Life Cycle)*. Metode yang digunakan pada penelitian ini adalah metode scrum (*scrum method*). Berikut ini merupakan tahapan yang akan dilaksanakan dalam metode scrum.



Gambar 3.4 Tahapan pengembangan sistem dengan *Scrum method*

Pada tahap-tahap dalam metode scrum, pengembangan perangkat lunak dipecah menjadi serangkaian iterasi terjadwal yang disebut *sprint*. Setiap *sprint* biasanya berlangsung antara satu hingga empat minggu, tergantung pada kebutuhan proyek. Selama setiap *sprint*, tim bekerja untuk menghasilkan *output* produk yang dapat digunakan. *Output product* tersebut disebut juga dengan *Increment* yang berupa hasil kerja yang telah selesai pada akhir setiap *sprint* yang memenuhi kriteria *Definition of "Done"* dan *product backlog* yang merupakan tugas - tugas apa saja yang perlu diselesaikan dalam pengerjaan sampai batas waktu pengerjaan. Terdapat juga *scrum team roles* yang terdiri tiga peran utama yaitu: *Scrum Master*, *Product Owner*, dan *Development Team*. *Scrum master* adalah orang yang bertanggung jawab dan membantu tim bekerja dengan baik, memastikan semua aturan scrum diikuti dan mengatasi masalah yang muncul. *Product owner* adalah orang yang menentukan apa yang perlu dikerjakan oleh tim, membuat daftar tugas serta memastikan tugas – tugas dijalankan dengan baik, dan mengatur prioritasnya. *Development team* adalah kelompok orang yang mengerjakan tugas-tugas tersebut, mengembangkan produk atau fitur yang diperlukan. Ketiga peran ini bekerja sama dan bertanggung jawab untuk memastikan proyek berjalan lancar dan menghasilkan produk berkualitas tinggi.

Penelitian ini menggunakan metode scrum karena implementasi metode scrum lebih cocok untuk proyek yang memiliki tujuan dan *deliverables* yang jelas dan memerlukan iterasi reguler dengan hasil yang dapat dievaluasi. Lebih terstruktur dengan peran, acara, dan *output* yang ditentukan dengan jelas. Ini termasuk *sprint*, *daily scrum*, *sprint planning*, *sprint review*, dan *sprint retrospective*. Struktur ini membantu tim baru untuk lebih mudah beradaptasi dan memahami proses kerja. Dibanding metode kanban yang lebih fleksibel dan kurang terstruktur. Metode scrum yang lebih dinamis dan responsif dibandingkan dengan metode *SDLC* (*Software Development Life Cycle*), terutama dalam konteks proyek yang kompleks dan cepat berubah. Scrum mendorong iterasi cepat, umpan balik berkelanjutan, dan peningkatan konstan, yang semuanya penting untuk keberhasilan proyek dalam lingkungan yang tidak pasti dan selalu berubah. Berdasarkan Gambar 3.4 yang merupakan tahapan pengembangan sistem dengan metode scrum. Berikut adalah kegiatan atau pertemuan yang dilakukan pada metode scrum :

1. *Sprint planning*

Pertemuan ini diadakan di awal setiap *Sprint*. Tim scrum (termasuk *product owner*, *scrum master*, dan *development team*) berdiskusi dan menentukan tujuan *sprint* dan *item backlog* yang akan dikerjakan selama *sprint* tersebut.

2. *Sprint backlog*

Pertemuan untuk memilih daftar tugas dari *product backlog* untuk dikerjakan selama satu *sprint*. Ini berisi pekerjaan yang harus diselesaikan untuk mencapai tujuan *sprint*, seperti pengembangan fitur dan pengujian yang spesifik.

3. *Daily scrum*

Pertemuan singkat yang diadakan setiap hari selama *sprint*. Anggota tim membahas tiga hal yaitu: apa yang telah dikerjakan sejak pertemuan terakhir, apa yang akan dikerjakan hari ini, dan apakah ada kendala yang dihadapi.

4. *Sprint review*

Pertemuan yang diadakan di akhir setiap *sprint*. Anggota mempresentasikan hasil pekerjaan mereka kepada pemangku kepentingan dan mendapatkan umpan balik. Tujuannya adalah untuk menunjukkan apa yang telah diselesaikan dan mendapatkan *input* untuk perbaikan di *sprint* berikutnya.

5. *Sprint retrospective*

Pertemuan ini juga diadakan di akhir setiap *sprint*, setelah *sprint review*. Tim merefleksikan proses kerja mereka, mengidentifikasi apa yang berjalan baik dan apa yang perlu diperbaiki dan membuat rencana tindakan untuk peningkatan proses di *sprint* berikutnya.

3.4.4 Analisa Keberhasilan

Tahap analisa hasil dilakukan untuk pengujian mengenai hasil dari pengembangan aplikasi *mobile* sebagai *dashboard monitoring* ICS yang dilakukan setelah proses pengembangan telah selesai. Analisa keberhasilan terdiri dari beberapa indikator pengujian seperti berikut.



Gambar 3.5 Indikator Pengujian

Adapun tahap pengujian pada aplikasi ini, meliputi:

1. Fungsionalitas Aplikasi

Uji Fungsionalitas aplikasi adalah menguji apakah aplikasi berjalan sesuai dengan persyaratan dan spesifikasi yang ditentukan, memastikan bahwa aplikasi bekerja sebagaimana mestinya dari perspektif pengguna. Dilakukan dengan cara menguji setiap fitur dan memastikan tidak ada *bug* atau *error* saat penggunaan aplikasi. Pengujian ini dilakukan dengan menggunakan metode *black-box testing* menggunakan skenario yang telah dirancang dan skala penilaian.

2. Kompatibilitas Perangkat

Uji Kompatibilitas Perangkat adalah menguji apakah aplikasi berjalan dengan baik di berbagai perangkat android dengan berbagai versi sistem operasi yang berbeda dan pastikan antarmuka pengguna terlihat dan berfungsi dengan baik pada berbagai resolusi layar. Pengujian dilakukan dengan menjalankan aplikasi di berbagai perangkat android untuk memastikan tampilan dan fungsi aplikasi tetap konsisten. Pengujian ini dilakukan dengan menggunakan *black-box testing* pada beberapa perangkat yang berbeda.

3. Pengalaman Pengguna

Uji Pengalaman Pengguna adalah menilai sejauh mana aplikasi memberikan pengalaman yang nyaman dan intuitif kepada *user*. Pengujian dilakukan dengan menguji apakah desain tampilan aplikasi sudah sesuai pedoman desain UI yang relevan, serta memastikan teks mudah dibaca dan tombol mudah diakses. Pengujian dilakukan dengan metode *black-box testing* dengan cara mengirim borang pada *user* yang berisi skenario, kriteria, dan skala penilaian.

4. Kinerja Aplikasi

Uji Kinerja Aplikasi adalah menguji kecepatan *respons* aplikasi, termasuk waktu buka dan waktu *respons* terhadap *input* pengguna, serta seberapa efisien aplikasi menggunakan memori dan baterai perangkat. Pengujian dilakukan dengan mengukur *respons* aplikasi dan memastikan tidak ada penurunan kinerja yang signifikan. Pengujian dilakukan dengan menggunakan *tools* Android Studio yaitu *android profiler* dan *app inspection* untuk menganalisa konsumsi sumber daya seperti CPU, RAM, dan Internet

BAB V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan tujuan penelitian yang disajikan, dapat disimpulkan bahwa penelitian ini bertujuan untuk mengembangkan sebuah aplikasi *dashboard mobile* yang mampu melakukan visualisasi data *real-time* yang berasal dari sensor ICS. Aplikasi tersebut akan diimplementasikan menggunakan *platform* Android Studio dan bahasa pemrograman Kotlin, dengan mengintegrasikan Retrofit sebagai *library* untuk komunikasi antara *dashboard mobile app* dan API yang diperoleh dari *back-end*. Dengan demikian, berikut adalah kesimpulan dan saran berdasarkan tujuan penelitian tersebut:

1. Berhasil mengimplementasikan sebuah *dashboard mobile app* yang memungkinkan visualisasi data *real-time* dari sensor ICS, memberikan kemampuan pemantauan yang lebih efektif dan efisien.
2. Penggunaan *platform* Android Studio dan bahasa pemrograman kotlin merupakan pilihan tepat untuk mengembangkan aplikasi *mobile*, karena kemudahan dalam penggunaan dan tersedia berbagai macam *tools* yang sangat membantu dalam pengembangan aplikasi android.
3. Integrasi *Retrofit* sebagai *library* untuk komunikasi antara aplikasi *mobile* dan API menonjol dalam pengembangan aplikasi android karena menyediakan cara yang deklaratif dan mudah digunakan untuk melakukan komunikasi dengan API. Retrofit membuat kode menjadi lebih terstruktur dan mudah dipahami, integrasinya yang langsung dengan GSON memungkinkan konversi JSON ke objek java atau kotlin secara otomatis memastikan pertukaran data yang lancar dan cepat.
4. Hasil dari pengujian *dashboard mobile app* menggunakan metode *testing* berupa *black-box testing*, *android profiler*, dan *app inspection* menunjukkan bahwa aplikasi berhasil berfungsi sesuai yang diharapkan dan menunjukkan konsumsi sumber daya yang cukup rendah dan stabil.

5.2 Saran

Mobile app yang dikembangkan berbasis *android*, perlu dikembangkan juga aplikasi berbasis iOS serta optimisasi kinerja untuk berbagai model *iPhone* guna meningkatkan pengalaman pengguna dan menjangkau lebih banyak pengguna *Apple*.

DAFTAR PUSTAKA

- [1] V. Sidharta, R. M. Tambunan, Azwar, and A. Ghaniyyu, "Suatu Kajian : Pembangunan Pertanian Indonesia," *Kajian Ilmu Sosial*, vol. 2, no. 2, pp. 229-232, 2021.
- [2] R. R. Rachmawati, "Smart Farming 4.0 untuk Mewujudkan Pertanian Indonesia," *Forum Penelitian Agro Ekonomi*, vol. 38, no. 2, pp. 137-154, 2020.
- [3] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, "Enhancing Smart farming through the applications of Agriculture 4.0 technologies," *International Journal of Intelligent Networks*, vol. 3, pp. 150–164, 2022.
- [4] S. Hemming, F. de Zwart, A. Elings, A. Petropoulou, and I. Righini, "Cherry Tomato Production in Intelligent Greenhouses—Sensors and AI for Control of Climate, Irrigation, Crop Yield, and Quality," *Sensors*, vol. 20, no. 22, pp. 1-30, 2020.
- [5] S. Pamungkas, "Smart Greenhouse System On Paprican Plants Based On Internet of Things," *Telekontran : Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 7, no. 2, pp. 197–207, 2020
- [6] Muhammad Tayyab Sohail, S. Mustafa, M. Mohd Ali, and S. Riaz, "Agricultural Communities' Risk Assessment and the Effects of Climate Change: A Pathway Toward Green Productivity and Sustainable Development," *Frontiers in Environmental Science*, vol. 10, 2022.
- [7] A. I. Rokade, A. D. Kadu, and K. S. Belsare, "An Autonomous Smart farming System for Computational Data Analyt ICS using IoT," *In Journal of PhysICS: Conference Series*, vol. 2327, no. 1, 2022.
- [8] H. X. Huynh, L. N. Tran, and N. Duong-Trung, "Smart Greenhouse construction and irrigation control system for optimal Brassica Juncea development," *Plos one*, vol. 18, no. 10, 2023.
- [9] F. Ariani, R. Y. Endra, Y. Aprlinda and A. R. Bahar, "Sistem Monitoring Suhu dan Pencahayaan Berbasis Internet of Thing (IoT) untuk Penetasan Telur Ayam," *Jurnal Manajemen Sistem Informasi dan Teknologi*, pp. 36-41, 2020.
- [10] R. S. Wirabangsa, W. Giri and D. Ratnasari, "Aplikasi Sistem Administrasi Inventarisasi dan Peminjaman Barang Berbasis Android," *Journal of Electrical Engineering and Information technology*, pp. 1-13, 2023.
- [11] N. Hendrastuty and Y. Ihza, "Rancang Bangun Aplikasi Monitoring Santri Berbasis Android," *Jurnal Data Mining dan Sistem Informasi*, pp. 21-34, 2021.
- [12] U. Ristian, I. Ruslianto and K. Sari, "Sistem Monitoring Smart Greenhouse pada Lahan Terbatas Berbasis Internet of Things (IoT)," *Jurnal Edukasi dan Penelitian Informatika*, pp. 87-94, 2022.

- [13] W. Aliman, "Perancangan Perangkat Lunak Untuk Menggambar Diagram Berbasis Android," *Jurnal Ilmiah Indonesia*, vol. 6, no. 6, pp. 3091-3098, 2021.
- [14] M. Kim, S. Sinha, Q. Xin and A. Orso, "Automated Test Generation for REST APIs: No Time to Rest Yet," *ISSTA*, pp. 289-301, 2022.
- [15] T. Kinsman, M. Wessel, M. A. Gerosa and C. Treude, "How Do Software Developers Use GitHub Actions to Automate Their Workflows?," 2021.
- [16] B. Yetistiren, I. Ozsoy and E. Tuzun, "Assessing the Quality of GitHub Copilot's Code Generation," *Promise*, vol. 978, pp. 62-71, 2022.
- [17] A. Wijaya, "Evaluasi Sistem Dashboard Monitoring Presensi Akademik Mahasiswa," *Jurnal Riset Teknologi dan Inovasi Pendidikan*, pp. 410-421, 2020.
- [18] F A. Febriandirza, "Perancangan Aplikasi Absensi Online Dengan Menggunakan Bahasa Pemrograman Kotlin," *Pseudocode*, vol. VII, no. 2, pp. 123-133, 2020.
- [19] E. M. Rianof, B. P. Adhi, and F. Putra, "Pengembangan Aplikasi M-Commerce pada Toko Optik Menggunakan Android Studio," *Jurnal Pinter*, vol. 4, no. 2, 2020.
- [20] R. S Siregar, L. Koryanto and N. M. Faizah, "Aplikasi Pencarian Hotel di Kota Jakarta Berbasis Android dengan Metode Location Based Service (LBS) Menggunakan Android Studio," *Computer Jurnal*, vol. 1, no. 1, 2023.
- [21] A. F. Fadillah and H. Zakaria, "Rancang Bangun Sistem Informasi Aplikasi Jadwal Pemberangkatan Bus Menggunakan Kotlin dengan Model Perancangan Agile berbasis Android," *Jurnal Ilmu Komputer dan Pendidikan*, vol. 1, no. 2, pp. 310-333, 2023.
- [22] Fatmawati, Yusrizal and A. M. Hasibuan, "Pengembangan Media Pembelajaran Berbasis Aplikasi Android untuk Meningkatkan Hasil Belajar Siswa," *Elementary School Journal*, pp. 134-143, 2021.
- [23] T. Iqbal and M. Wali, "IDOL: Retrofit Kotlin Service Based Online Digital Library Application and College Open Data Repository," *International Journal Software Engineering and Computer Science*, vol. 2, no. 1, pp. 1-8, 2022.
- [24] I. O. Suzanti, N. Fitriani, A. Jauhari and A. Khozaimi, "REST API Implementation on Android Based Monitoring Application," *Journal of PhysICS: Conference Series*, 2020.
- [25] N. Laranjeiro, J. Agnelo, and J. Bernardino, "A black box tool for robustness testing of REST services," *IEEE Access*, vol. 9, no. 24, pp 738-754, 2021.
- [26] S. Ohk, YS. Kim and YJ. Kim, "Phase-Based Low Power Management Combining CPU and GPU for Android Smartphones," *ElectronICS*, vol. 11, no. 2480, pp. 1-19, 2022.

- [27] P. Fitriani, "SISTEM PENDUKUNG KEPUTUSAN PEMBELIAN SMARTPHONE ANDROID DENGAN METODE MULTI ATTRIBUTE UTILITY THEORY (MAUT)," *Jurnal Mantik Penusa*, vol. 4, no. 1, pp. 6-11, 2020.
- [28] R. S. Sasmita, "Pemanfaatan Internet Sebagai Sumber Belajar," *Jurnal Pendidikan dan Konseling*, vol. 2, no. 1, pp. 99-103, 2020.
- [29] H. A. Epiloksa, D. S. Kusumo and M. Adrian, "Effect Of MVVM Architecture Pattern on Android Based Application Performance," *Jurnal Media Informatika Budidarma*, vol. 6, no. 4, pp. 1949-1955, 2022.
- [30] N. Anwar, A. Marwanto and Muhamad Haddin, "Android-Based Inspection and Testing Analytical System of Electrical Safety," *Journal of TelematICS and InformatICS (JTI)*, vol. 9, no. 1, pp. p. 22-37, 2021.