



PENGEMBANGAN APLIKASI MONITORING DAN KONTROL UNTUK GREENHOUSE HIDROPONIK

Muhammad Aji Saputra¹⁾, Husni Thamrin²⁾

^{1,2} Program Studi Teknik Informatika, Universitas Muhammadiyah Surakarta

email: ¹muhammadaji1614@gmail.com, ² husni.thamrin@ums.ac.id

ARTICLE INFO

Article History:

Received : 18 Maret 2025

Accepted : 15 Juni 2025

Published : 18 Juni 2025

Keywords:

Android Application

Hydroponics

Internet of Things

IEEE style in citing this article:

M. A. Saputra, H. Thamrin, "pengembangan aplikasi monitoring dan kontrol untuk greenhouse hidropnik", *jurnal.ilmiah.informatika*, vol. 10, no. 1, pp. 1-12, Jul. 2025.

ABSTRACT

*This research aims to develop an Internet of Things (IoT) based monitoring and control application for hydroponic greenhouses using the Nutrient Film Technique (NFT) method for lettuce (*Lactuca sativa* L.). Hydroponics has emerged as an efficient solution for agricultural production in limited spaces, but environmental management challenges often remain a primary obstacle. The application is designed to automatically monitor and control environmental conditions, such as temperature, solution pH, and nutrient concentration, to reduce the risk of bolting that can diminish harvest quality. Software development was conducted using the Scrum methodology, which supports iterative and adaptive processes, employing Model-View-ViewModel (MVVM) architecture and clean architecture to create a modular and easily maintainable code structure. Jetpack Compose was utilized to generate a responsive and efficient user interface. Research findings demonstrate that the application offers key features including real-time monitoring through MQTT protocol, manual and automatic control, data history visualization in graphs, weather prediction, and hydroponic guidance. Application validation using the Aiken index showed a very high validity level (> 0.80), while black box testing ensured that all application functions operated by user requirements. This research is expected to improve hydroponic greenhouse management efficiency and significantly contribute to IoT technology development in the agricultural sector.*

Corresponding (optional)

Husni Thamrin

Universitas Muhammadiyah

Surakarta

Author:

1. PENDAHULUAN

Indonesia merupakan negara agraris dengan mayoritas penduduk bekerja di sektor pertanian. Sektor ini sangat penting karena menyediakan bahan pangan bagi masyarakat. Seiring dengan perkembangan ilmu pengetahuan dan teknologi, konsep pertanian hidroponik makin populer. Hidroponik adalah metode budidaya tanaman dengan menggunakan medium air yang kemudian diartikan sebagai cara budidaya bukan tanah [1]. Hidroponik menawarkan solusi untuk meningkatkan produktivitas pertanian di lahan terbatas. Teknik ini memiliki keunggulan seperti standarisasi metode kerja yang memudahkan proses budidaya dan kemampuannya untuk digunakan di lahan terbatas, terutama di daerah perkotaan [2].

Namun, meskipun hidroponik menawarkan banyak keuntungan, seperti standarisasi metode kerja dan efisiensi ruang, tantangan signifikan tetap ada. Kebutuhan pengetahuan yang mendalam tentang fisiologi tanaman dan kimia organik sangat penting untuk keberhasilan sistem ini. Selain itu, kontrol lingkungan yang presisi—termasuk pengelolaan suhu larutan dan konsentrasi nutrisi—merupakan tantangan utama yang harus dihadapi oleh para petani hidroponik [2]. *Monitoring* dan kontrol manual tidak hanya memakan waktu tetapi juga rentan terhadap kesalahan manusia, yang berpotensi mempengaruhi hasil panen.

Penelitian ini akan fokus pada tanaman selada (*Lactuca sativa* L.), yang sering mengalami masalah bolting yang dapat menurunkan kualitas hasil panen [3]. Salah satu metode hidroponik yang populer adalah *Nutrient Film Technique* (NFT). Sistem NFT menggunakan “film” atau lapisan tipis larutan nutrisi setebal 1-

3 mm yang dipompa untuk mengalir melewati akar tanaman dengan kecepatan aliran sekitar 1-2 liter per menit dan disirkulasikan secara terus-menerus [1]. Menjaga suhu optimal dalam sistem ini sangat penting karena suhu yang terlalu tinggi pada larutan nutrisi dapat menurunkan kadar *dissolved oxygen* (DO) dalam larutan nutrisi dan menghambat pertumbuhan tanaman. Pada umumnya tanaman membutuhkan suhu larutan nutrisi sekitar 18-28 derajat celsius [2].

Dengan teknologi *Internet of Things* (IoT), proses *monitoring* dan kontrol dapat dilakukan secara otomatis dan lebih efisien. IoT merupakan konsep menghubungkan perangkat-perangkat ke internet agar perangkat tersebut dapat mengumpulkan dan saling bertukar data [4]. Menurut Firmansyah, sistem IoT memiliki kemampuan untuk secara langsung mengawasi kondisi lingkungan dan mengatur parameter seperti nutrisi, pH, serta kelembaban sesuai dengan kebutuhan tanaman [5].

Teknologi IoT telah banyak digunakan untuk *monitoring* dan kontrol rumah kaca hidroponik guna meningkatkan hasil panen dan produksi tanaman. Sebagai contoh, penelitian oleh Supriyono, membahas tentang pengembangan sistem untuk memantau tingkat kekeruhan air tawar dengan memanfaatkan platform IoT [6]. Sistem ini memanfaatkan sensor kekeruhan yang mampu memantau kualitas air secara berkelanjutan dan dari jarak jauh, menunjukkan penerapan teknologi IoT dalam berbagai aspek pemantauan lingkungan. Selain itu, studi oleh Afwan menciptakan aplikasi Android yang memungkinkan petani untuk mengakses data sensor secara *real-time* serta melihat riwayat data dalam bentuk grafik [7]. Namun, aplikasi ini masih belum dilengkapi dengan fitur kontrol aktuator, autentikasi pengguna, serta tambahan seperti prediksi cuaca dan

konten edukatif. Penelitian terbaru oleh Rohmah juga menunjukkan bahwa sistem pengendalian hama menggunakan teknologi IoT dapat beroperasi secara otomatis dan dikendalikan dari jarak jauh, memberikan solusi yang ramah lingkungan dan efisien bagi petani dalam mengatasi masalah hama tanaman [8].

Aplikasi yang akan dikembangkan bertujuan untuk memonitor dan mengontrol rumah kaca hidroponik dengan memanfaatkan protokol MQTT sebagai media transmisi data pada perangkat IoT. Sistem ini terdiri atas dua komponen utama: backend dan frontend. Aplikasi *backend* dirancang menggunakan Express.js untuk mengelola penyimpanan data dan menjalankan fungsi terkait transfer data melalui protokol MQTT. Komponen *frontend* berupa aplikasi Android yang dikembangkan menggunakan bahasa Kotlin, memungkinkan petani untuk memantau dan mengontrol rumah kaca secara langsung melalui perangkat seluler.

Pendekatan ini menawarkan fleksibilitas tinggi, di mana aplikasi Android memberikan kemudahan akses kapan saja dan di mana saja tanpa memerlukan perangkat tambahan seperti komputer atau koneksi internet yang stabil sepanjang waktu. Sementara itu, aplikasi backend berjalan di server selama 24 jam *nonstop* untuk memastikan pengelolaan data tetap optimal tanpa membebani perangkat Android pengguna.

Dengan integrasi teknologi IoT, aplikasi ini dirancang untuk membantu petani hidroponik mengelola rumah kaca secara otomatis dan efisien. Fitur-fitur yang disediakan mencakup pemantauan sensor secara *real-time*, kontrol aktuator dalam mode manual dan otomatis, notifikasi status, prediksi cuaca, autentikasi pengguna, serta konten edukatif terkait hidroponik. Melalui fitur-

fitur ini, petani dapat melakukan pengaturan jarak jauh, meminimalkan risiko kesalahan, dan meningkatkan efisiensi dalam budidaya selada hidroponik.

Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi monitoring dan kontrol untuk *greenhouse* hidroponik NFT tanaman selada yang memenuhi kriteria fungsional dan teknis serta lolos dari pengujian seluruh fitur yang direncanakan. sehingga dapat meningkatkan hasil panen selada dengan mengurangi kejadian *bolting*. Selain itu, aplikasi ini dirancang agar mudah digunakan dan informatif bagi petani.

Melalui penelitian ini, diharapkan aplikasi dapat memberikan kontribusi pada pengembangan pertanian modern di Indonesia. Selain itu, aplikasi ini diharapkan dapat memberikan wawasan bagi masyarakat yang ingin memulai bertani hidroponik. Dengan aplikasi ini, petani dapat meminimalkan kunjungan ke *greenhouse*, menghemat waktu dan tenaga, serta memastikan kondisi *greenhouse* tetap optimal tanpa harus berada di Lokasi.

2. METODE PENELITIAN

2.1 Metode Scrum

Scrum adalah kerangka kerja yang mengimplementasikan prinsip-prinsip Agile melalui langkah-langkah praktis yang mendukung proses pengembangan perangkat lunak secara iteratif dan adaptif [9]. Metode agile dengan kerangka kerja scrum dipilih karena memiliki pendekatan yang lebih fleksibel dibandingkan dengan pengembangan perangkat lunak tradisional, seperti menggunakan Metode Waterfall. Metode Waterfall biasanya melibatkan beberapa langkah yang harus diikuti dalam urutan tertentu. Langkah-langkahnya mencakup analisis kebutuhan, perancangan sistem, implementasi, pengujian, serta pemeliharaan [10]. Metode tradisional

mengharuskan semua ide dikumpulkan di awal siklus pengembangan. Dalam praktiknya, ide-ide baru sering muncul selama proses pengembangan, sehingga pendekatan tradisional dapat menjadi kendala [11]. Berdasarkan penelitian dari Zulvi terhadap 34 literatur, 61,76% menggunakan model Scrum, 32,35% menggunakan *Extreme Programming* (XP), sementara sisanya merupakan model *Dynamic System Development Method* (DSDM) dan *Feature Driven Development* (FDD) [12]. Scrum memiliki fleksibilitasnya dalam menghadapi

perubahan yang dapat terjadi selama pengembangan, kemampuannya untuk menghasilkan hasil dalam waktu singkat, serta fokus pada kolaborasi tim dan peningkatan kualitas melalui umpan balik berkelanjutan.

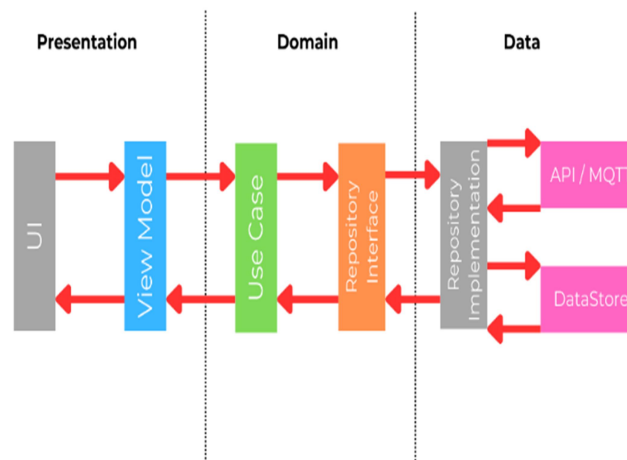
Aplikasi android ini dikembangkan dengan menerapkan arsitektur *Model-View-ViewModel* (MVVM). Arsitektur MVVM membantu memisahkan logika bisnis dari antarmuka pengguna. MVVM memiliki 3 komponen utama seperti pada gambar 1.



Gambar 1. Model-view-viewmodel (MVVM) patern

Model bertanggung jawab dalam mengakses berbagai sumber data, sedangkan view merupakan elemen antarmuka. Kemudian viewmodel merupakan komponen yang mewakili cara tampilan diharapkan terhadap interaksi pengguna dan bertugas menangani komunikasi antara View dan Model [13]. Meskipun MVVM efektif untuk proyek kecil, tetapi seiring bertambahnya ukuran proyek, ViewModel menjadi terlalu banyak

memiliki tanggung jawab dan memiliki banyak ketergantungan pada berbagai sumber data. Untuk mengatasi keterbatasan MVVM dalam proyek berskala besar, arsitektur aplikasi dibagi menjadi tiga lapisan: data, domain, dan presentasi, mengikuti prinsip Clean Architecture agar meningkatkan skalabilitas, modularitas, dan pemeliharaan kode seperti pada Gambar 2.



Gambar 2. Android clean architecture

Proses aliran data dimulai ketika pengguna berinteraksi dengan antarmuka pengguna (UI). Antarmuka ini berkomunikasi dengan ViewModel untuk mendapatkan data. Selanjutnya, ViewModel akan meminta *Use Case* untuk memproses data melalui *Repository Interfaces*, yang kemudian akan mengambil data dari implementasi *repository*, seperti API atau penyimpanan lokal.

Antarmuka pengguna pada aplikasi ini menggunakan *Jetpack Compose* karena menggunakan paradigma pemrograman baru, yakni deklaratif. Paradigma deklaratif yaitu berfokus pada menjelaskan yang harus dilakukan program, berbeda dengan *Extensible Markup Language* (XML) yang menggunakan paradigma imperatif yang berfokus bagaimana cara mencapai tujuan. Pendekatan berbasis XML untuk desain antarmuka bisa menjadi lebih rumit dan memakan waktu lebih dibandingkan dengan pendekatan *Jetpack Compose* [14].

2.1 Perencanaan Garis Besar

Perencanaan garis besar dalam penelitian ini dilakukan melalui pembuatan *product backlog*. *Product backlog* merupakan daftar prioritas yang diberi peringkat berdasarkan nilai bagi bisnis dan selalu berkembang selama masa proyek, seperti ditambahkan, dihapus, atau diprioritaskan kembali [11]. Penyusunan *product backlog* biasanya berbentuk user stories yang menjelaskan secara ringkas tentang fitur aplikasi yang diinginkan oleh pengguna. *User stories* ini memiliki format yang sederhana dan mudah dipahami, serta dapat menggambarkan nilai bisnis yang diharapkan dari setiap fitur yang dirancang. Format ini juga digunakan sebagai pengganti percakapan langsung

dengan pengguna dan dapat ditulis dengan berbagai tingkat detail [15].

Setiap item dalam *backlog* diberikan bobot kesulitan dan tingkat prioritas untuk membantu pengelolaan *backlog* selama pengembangan. Bobot kesulitan didefinisikan dengan angka 1, 2, 3, dan 4 yang melambangkan bobot pekerjaan kecil, sedang, besar, dan sangat besar. Sementara itu, tingkat prioritas dikelompokkan ke dalam skala *small*, *medium*, dan *large* untuk menunjukkan urgensi kebutuhan pengguna. *User stories* biasanya didefinisikan dalam format "Sebagai (siapa), saya ingin (apa) sehingga (mengapa)."

Sebagai bagian dari evaluasi aplikasi, uji validitas dilakukan untuk menilai kualitas media yang dikembangkan. Penilaian dilakukan oleh dua ahli media, dengan setiap butir soal dinilai menggunakan skala 1 hingga 5, di mana skor 1 menunjukkan kualitas yang kurang baik, dan skor 5 menunjukkan kualitas yang sangat baik. Penilaian oleh para ahli kemudian dihitung menggunakan teknik Aiken dengan persamaan 1 [16].

$$V = \frac{\sum S}{n(c-1)} \quad (1)$$

Dimana :

S : R – Lo

V : indeks Aiken

S : skor yang diberikan oleh penilai dikurangi dengan skor paling rendah

R : skor yang diberikan oleh penilai

Lo : skor penilaian terendah (1)

C : skor penilaian tertinggi (4)

n : jumlah penilai

Hasil perhitungan validitas kemudian diinterpretasikan berdasarkan kriteria tingkat validitas berikut [17]:

Tabel 1. Kriteria tingkat validitas butir soal

Nilai V	Tingkat Validitas
0,80 – 1,00	Sangat Tinggi
0,60 – 0,80	Tinggi
0,40 – 0,60	Sedang
0,20 – 0,40	Rendah
0,00 – 0,20	Sangat Rendah

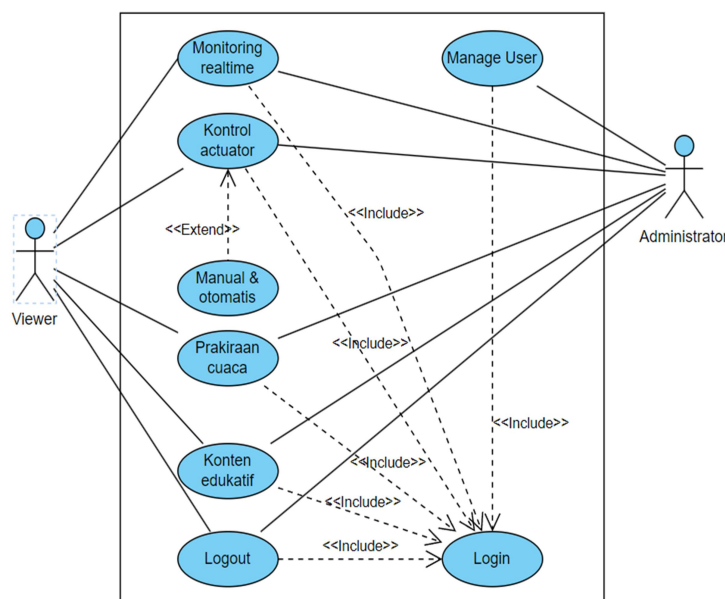
(Sumber : Arikunto, 2018)

Selain itu, *black box testing* diterapkan sebagai metode pengujian aplikasi untuk memverifikasi bahwa aplikasi memenuhi persyaratan fungsional, dengan memastikan fungsi, input, dan output telah sesuai [18]. Metode ini bertujuan memastikan aplikasi memberikan hasil yang konsisten, memenuhi kebutuhan pengguna, serta berfungsi dengan baik dalam berbagai kondisi operasional.

2.2 Desain Sistem

Setelah mengidentifikasi kebutuhan pengguna melalui *product backlog*, langkah

selanjutnya adalah membuat desain sistem. Pada tahap ini, pemodelan menggunakan UML (*Unified Modeling Language*) dilakukan. UML adalah standar bahasa yang umum digunakan dalam industri untuk mendefinisikan kebutuhan, analisis, desain, serta menggambarkan arsitektur sistem perangkat lunak berbasis objek [19]. UML yang digunakan dalam penelitian ini mencakup *use case diagram* yang dapat dilihat pada gambar 3.



Gambar 3. Use case diagram dengan 2 faktor pada AutoGreen aplikasi

2.4 Tahapan Sprint

Setiap siklus sprint pada penelitian ini berdurasi selama 1 minggu tanpa perpanjangan, meskipun *Sprint Goal* belum tercapai. *Sprint Goal* sendiri merupakan tujuan utama yang harus

dicapai dalam satu siklus sprint. Selama sprint berlangsung, perubahan besar harus ditunda hingga sprint berikutnya agar fokus tetap terjaga. Proyek ini terdiri dari 5 sprint, masing-masing dengan tujuan dan hasil yang berbeda.

3. Hasil Dan Pembahasan

3.1 User Stories

Berdasarkan hasil wawancara dengan calon *end user* didapatkan rincian *user stories* terlihat pada tabel 2.

Tabel 2. User Stories

No	Sebagai	Saya ingin	Sehingga saya dapat
1	Pengelola <i>Greenhouse</i>	mengetahui cuaca di masa depan	mengetahui apakah hujan atau tidak
2	Pekerja <i>Greenhouse</i>	memiliki pengetahuan lebih mendalam tentang pertanian hidroponik	meningkatkan pemahaman saya
3	Pengelola <i>Greenhouse</i>	memantau lingkungan di dalam <i>greenhouse</i> tanpa harus hadir di lokasi	memantau dari mana saja dan kapan saja
4	Pekerja <i>Greenhouse</i>	mengontrol lingkungan di dalam <i>greenhouse</i> secara jarak jauh dan mengotomatisasi <i>greenhouse</i>	mengoptimalkan waktu untuk pekerjaan lain
5	Pengelola <i>Greenhouse</i>	mengetahui riwayat data sebelumnya	menganalisis penyebab kegagalan panen

3.2 Product Backlog

Pembuatan *product backlog* dibuat oleh *scrum master* berdasarkan *user stories* dan analisis aplikasi yang sebelumnya sudah

ada. Sehingga didapatkan daftar *product backlog* seperti yang dijelaskan pada Tabel 3.

Tabel 3. Product Backlog

No	Product Backlog	Tingkat Kesulitan	Prioritas	Sprint
1	Pembuatan endpoint Login	1	Medium	1
2	Pembuatan endpoint CRUD user	1	Medium	1
3	Pembuatan subscribe MQTT & broadcast websocket (monitoring)	3	Large	1
4	Pembuatan Endpoint publish ke MQTT	3	Large	1
5	<i>Project android setup</i>	2	Large	1
6	Implementasi monitoring real-time	2	Large	1
7	Pembuatan endpoint riwayat	1	Small	2
8	Pembuatan subscribe from MQTT & broadcast websocket (actuator)	4	Large	2
9	Pembuatan subscribe from MQTT & broadcast websocket (setpoint)	4	Large	2
10	Implementasi kontrol aktuator manual	4	Large	2
11	Implementasi autentikasi	4	Medium	2
12	Pembuatan endpoint data rata rata harian, mingguan	1	Medium	3
13	Pembuatan endpoint CRUD kontrol otomatis	3	Large	3
14	Pembuatan endpoint panduan hidroponik	1	Small	3
15	Implementasi riwayat	3	Large	3
16	Implementasi grafik garis	4	Medium	3

17	Implementasi kontrol otomatis	4	Large	3
18	Pembuatan endpoint weather	2	Large	4
19	Implementasi panduan hidroponik	1	Large	4
20	Implementasi weather forecast	4	Medium	4
21	Implementasi weather real-time	4	Medium	4
22	Black box testing	3	Large	5
23	Validitas ahli media	2	Large	5
24	Finalisasi	2	Medium	5

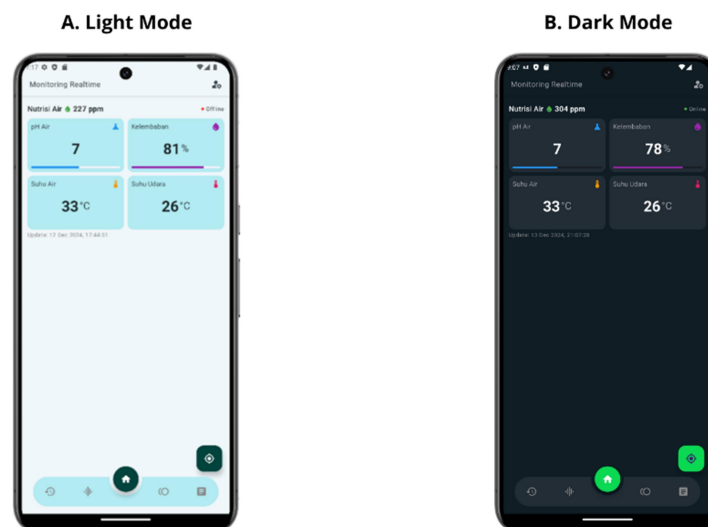
3.3 Sprint

Pengerjaan *product backlog* dilaksanakan dalam 5 sprint yang dimulai dari minggu ketiga bulan November hingga minggu kedua bulan Desember.

1) Sprint 1

Pada sprint pertama, dilakukan identifikasi kebutuhan pustaka (*library*) di Android Studio sebagai langkah awal untuk memulai proyek. Tahap ini meliputi pembuatan *setup* proyek Android, diikuti dengan pengembangan bagian *backend* untuk menyediakan *endpoint* API guna mendukung fitur *login*, pengelolaan akun, dan publikasi data ke

MQTT. Selain itu, dilakukan pengembangan fitur *broadcast* melalui *websocket* agar data yang di-*subscribe* dari topik MQTT herbalawu/monitoring dapat diteruskan (*broadcast*) ke klien, yaitu aplikasi Android *AutoGreen*. Data yang disiarkan melalui *websocket* merupakan data pemantauan (*monitoring*) secara *real-time*, yang kemudian diimplementasikan ke dalam aplikasi Android. Tahap implementasi dilakukan dengan cara mengambil data dari *websocket* dan menampilkannya pada aplikasi, seperti yang dapat dilihat pada Gambar 4.



Gambar 4. Monitoring real-time

2) Sprint 2

Sprint kedua fokus pengembangan diarahkan pada bagian fitur kontrol manual dengan dikembangkannya fitur *broadcast* untuk data yang diambil dari

topik MQTT herbalawu/state dan herbalawu/setpoint. Menggunakan *endpoint* publish yang telah dikembangkan pada sprint pertama, dikembangkan fitur kontrol manual untuk

aktuator. Data kontrol ini disinkronkan secara terus-menerus dengan mengambil data dari *websocket* yang berasal dari topik *herbalawu/state*. Selanjutnya, dilakukan pengembangan fitur autentikasi serta fitur manajemen pengguna, yang meliputi pembuatan, pengeditan, dan

penghapusan akun. Kemudian diakhiri dengan pengembangan *backend* untuk menciptakan *endpoint* API yang mendukung fitur riwayat. Tampilan dari halaman kontrol manual, login, manajemen pengguna, dan pembuatan akun dapat dilihat pada Gambar 5.



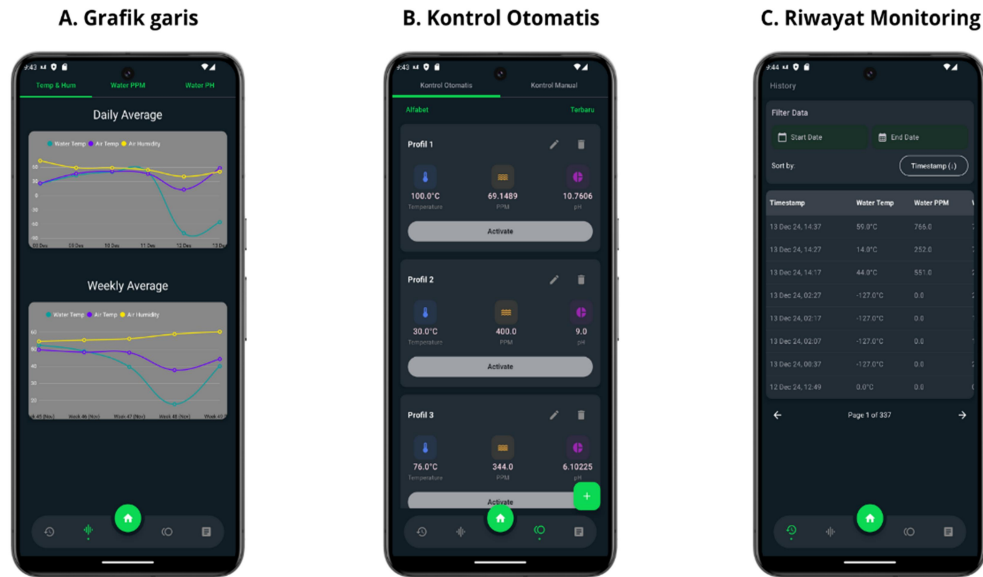
Gambar 5. Tampilan layar login, kontrol manual, manajemen akun, buat akun

3) Sprint 3

Sprint ketiga dimulai dengan penerapan fitur riwayat yang memanfaatkan data dari *endpoint* yang telah dikembangkan pada sprint sebelumnya. Fitur ini dilengkapi dengan kemampuan untuk penyaringan, penomoran halaman, dan pengurutan sehingga dapat meningkatkan pengalaman pengguna dalam menjelajahi data. Tahapan berikutnya adalah pengembangan *backend* untuk menyediakan *endpoint profil setpoint*. *Endpoint* ini memungkinkan pengguna

untuk membuat kontrol otomatis dengan menyusun sebuah profil yang terdiri dari tiga parameter lingkungan yang dapat diatur, yaitu suhu air, nutrisi air, dan pH air. Sprint ketiga diakhiri dengan penambahan *endpoint* untuk perhitungan rata-rata data pemantauan harian dan mingguan. Data ini kemudian dimanfaatkan untuk mengembangkan fitur grafik garis pada aplikasi Android. Tampilan dari halaman kontrol otomatis, grafik garis, dan riwayat dapat dilihat pada

Gambar 6



Gambar 6. Tampilan layar grafik garis, kontrol otomatis, riwayat monitoring

4) Sprint 4

Pada sprint 4 berfokus pada pengembangan fitur prediksi cuaca yang dimulai dengan membuat endpoint untuk mengirimkan koordinat lokasi *greenhouse* dan mengambil data cuaca pada lokasi tersebut. Selanjutnya dilakukan tahap penerapan fitur prediksi cuaca pada

aplikasi android yang terbagi menjadi 2 bagian yaitu data cuaca sekarang dan data prediksi cuaca tiap 3 jam. Sprint ini diakhiri dengan penerapan fitur panduan hidroponik dari tahap semai hingga panen. Tampilan dari halaman prediksi cuaca dan panduan hidroponik dapat dilihat pada gambar 7.



Gambar 7. Tampilan halaman cuaca dan halaman panduan dalam dark mode & light mode

5) Sprint 5

Sprint kelima berfokus pada pengujian aplikasi, meliputi validasi media aplikasi oleh ahli dan pengujian fungsionalitas menggunakan teknik *black box testing*.

Validasi media dilakukan dengan menilai

tiga aspek utama: Tampilan Antarmuka, Keterbacaan, dan Navigasi. Penilaian dari para ahli dihitung menggunakan rumus Aiken, dan hasilnya dirangkum dalam Tabel 4.

Tabel 4. Hasil uji validitas dan interpretasi media aplikasi oleh ahli

Aspek	Butir	Penilai		V	Keterangan
		I	II		
Tampilan Antarmuka	Butir 1	4	5	0,875	Sangat tinggi
	Butir 2	5	5	1	Sangat tinggi
	Butir 3	4	5	0,875	Sangat tinggi
	Butir 4	4	5	0,875	Sangat tinggi
	Butir 5	5	5	1	Sangat tinggi
	Butir 6	2	4	0,5	Cukup
Keterbacaan	Butir 7	5	5	1	Sangat tinggi
	Butir 8	4	4	0,75	Tinggi
	Butir 9	4	4	0,75	Tinggi
	Butir 10	4	4	0,75	Tinggi
	Butir 11	5	5	1	Sangat tinggi
	Butir 12	4	4	0,75	Tinggi
Navigasi	Butir 13	4	5	0,875	Sangat tinggi
	Butir 14	4	5	0,875	Sangat tinggi
	Butir 15	4	5	0,875	Sangat tinggi
	Butir 16	4	4	0,75	Tinggi
	Butir 17	4	4	0,75	Tinggi
	Butir 18	4	5	0,875	Sangat tinggi

Dari total 18 butir soal yang diuji, diperoleh 11 butir dengan validitas sangat tinggi, 6 butir dengan validitas tinggi, dan 1 butir dengan validitas cukup. Validitas sangat tinggi menunjukkan bahwa media aplikasi telah memenuhi kelayakan tanpa memerlukan perbaikan. Validitas tinggi mengindikasikan bahwa media layak digunakan tetapi masih bisa ditingkatkan untuk kenyamanan pengguna. Sementara

itu, validitas cukup pada Butir 6, menunjukkan adanya kekurangan dalam aplikasi, yaitu antarmuka yang kurang responsif pada ukuran layar tertentu, sehingga masih memerlukan perbaikan untuk memenuhi kelayakan.

Selanjutnya, *black box testing* dilakukan untuk memastikan setiap fitur aplikasi berfungsi sesuai kebutuhan pengguna. Tabel 5 merangkum hasil pengujian.

Tabel 5. Black box testing

Test Scenario	Test Case	Expected Result	Status
Login	Memasukkan username dan password dengan benar lalu klik login	Sistem menerima permintaan dan user dapat masuk aplikasi	Valid
	Memasukkan username saja tanpa password atau sebaliknya lalu klik login	Sistem menolak permintaan dan muncul alert bahwa username dan password harus diisi	Valid
	Memasukkan username atau password yang salah lalu klik login	Sistem menolak permintaan dan menampilkan alert bahwa username atau password salah	Valid
Logout	Klik tombol logout	Sistem menerima permintaan dan user dikembalikan ke halaman login	Valid
Tambah akun	Akun dengan role admin mengisi seluruh field	Sistem menerima permintaan dan akun baru ditambahkan	Valid
	Akun dengan role admin tidak mengisi keseluruhan field	Form tidak dapat disubmit dan ada pemberitahuan agar mengisi keseluruhan field	Valid
Delete Akun	Akun dengan role admin menghapus akun	Sistem menerima permintaan dan menghapus akun	Valid
Monitoring real-time	Menampilkan data realtime dari IoT	Data berhasil ditampilkan	Valid
Indikator status IoT	Perubahan status perangkat IoT tergantung konektivitas	Status online, offline, dan tidak terhubung	Valid
Grafik rata rata tiap hari dan tiap minggu	Menampilkan data rata rata tiap hari dan tiap minggu dalam grafik garis	Data berhasil ditampilkan	Valid
Ongoing notification	Status device online ketika data monitoring diterima	Status device connected	Valid
	Status offline ketika tidak ada jaringan internet	Status No internet connection	Valid
Kontrol manual	Mengaktifkan aktuator	Data berhasil dikirim dan diterima oleh perangkat IoT	Valid
	Menonaktifkan aktuator	Data berhasil dikirim dan diterima oleh perangkat IoT	Valid
Membuat profil setpoint	Mengisi keseluruhan field	Sistem menerima permintaan dan profil baru ditambahkan	Valid
	Mengosongi salah satu atau lebih field	Form tidak dapat disubmit dan ada pemberitahuan agar mengisi keseluruhan field	Valid
Mengaktifkan profil setpoint	Mengaktifkan salah satu profil setpoint	Sistem menonaktifkan semua profil lalu mengaktifkan profil yang dipilih	Valid
Delete Profil setpoint	Menghapus profil setpoint	Data profil berhasil dihapus	Valid
Kunci koordinat	Klik tombol kunci koordinat	Data lokasi ponsel dikirim ke server dan disimpan	Valid
Prediksi cuaca	Prediksi cuaca diperbarui secara otomatis	Data cuaca dapat ditampilkan diperbarui otomatis	Valid
Data Riwayat	Menyaring data berdasarkan tanggal	Data ditampilkan sesuai dengan tanggal yang dipilih	Valid

Hasil pengujian menunjukkan bahwa semua fitur berfungsi sesuai harapan. Namun, beberapa fitur seperti validitas pada aspek tertentu atau responsivitas antarmuka masih memerlukan pengoptimalan lebih lanjut untuk meningkatkan kualitas aplikasi.

4. Kesimpulan

Penelitian ini telah berhasil mengembangkan sebuah aplikasi Android untuk pengelolaan *greenhouse* dengan metode pengembangan perangkat lunak Scrum yang mendukung proses iteratif, adaptif, dan fleksibel, sehingga mampu merespons perubahan kebutuhan pengguna selama pengembangan.

Aplikasi ini menerapkan arsitektur *Model-View-ViewModel* (MVVM) yang dikombinasikan dengan *pendekatan clean architecture*, menghasilkan struktur kode yang modular, skalabel, dan mudah dipelihara. Fitur utama aplikasi meliputi pemantauan lingkungan secara *real-time* menggunakan protokol transfer data MQTT, kontrol manual dan otomatis, serta penyajian data riwayat dalam bentuk grafik. Validasi yang dilakukan oleh para ahli menunjukkan kualitas aplikasi yang sangat baik dengan indeks validitas Aiken di atas 0,80, sementara pengujian *black box* memastikan bahwa semua fungsi berjalan sesuai dengan kebutuhan pengguna.

REFERENSI

- [1] E. D. Purbajanti, W. Slamet, dan F. Kusmiyati, "Buku Hidroponic Florentina." EF Press Digimedia, hal. 1–84, 2017.
- [2] Susilawati, Dasar – Dasar Bertanam Secara Hidroponik. UNSRI PRESS, 2019.
- [3] Adlian, K. Lenci Patty, dan F. Kiriho, "EFEKTIVITAS PEMBERIAN AIR CUCIAN BERAS TERHADAP PERTUMBUHAN DAN HASIL TANAMAN SELADA (*Lactuca sativa* L.)," *Oryza J. Agribisnis dan Pertan. Berkelanjutan*, vol. 8, no. 2, hal. 1–10, 2023.
- [4] R. Nahari, E. D. Astuti, M. Pramudia, dan D. Rahmawati, "Fundamental Internet of Things (IOT) TEORI DAN APLIKASI," *Angew. Chemie Int. Ed.* 6(11), 951–952., hal. 82–95, 2023.
- [5] Firmansyah, B. Wibisana, I. M. Ziad, dan A. R. Sulthon, "Pertanian Cerdas Berbasis Internet of Things untuk Meningkatkan," *J. Pengabd. Masy. Nusantara*, vol. 4, no. 20, hal. 80–85, 2024.
- [6] H. Supriyono, A. Hibatullah, dan K. Harismah, "Turbidity Monitoring of Freshwater Using Internet of Things Platform Turbidity Monitoring of Freshwater Using Internet of Things Platform," 2021, doi: 10.1088/1742-6596/1858/1/012048.
- [7] M. Afwan, R. Septiana, R. Septiana, dan P. Aplikasi Pemantauan Rumah Kaca Pintar Berbasis, "Perancangan Aplikasi Pemantauan Rumah Kaca Pintar Berbasis Android," *J. Tek. Komput.*, vol. 1, no. 1, hal. 21–29, 2022, doi: 10.14710/jtk.v1i1.34573.
- [8] R. N. Rohmah, Y. Oktafianto, H. Supriyono, dan A. Supardi, "PEST CONTROL SYSTEM ON AGRICULTURAL LAND USING IOT ELECTRONIC CONTROLLER," *J. Appl. Eng. Technol. Sci.*, vol. 5, no. 2, hal. 1011–1019, 2024.
- [9] I. Pratiwi dan S. Y. Sari, "AMANDA (Your Plant Monitoring Application) Implementation Android Application Program Project Management Using Agile Scrum Method and Trello,"

- ICETIA 2023, vol. 02001, hal. 1–13, 2024.
- [10] D. Gunawan dkk., “WEB-BASED LIBRARY INFORMATION SYSTEM IN MADRASAH IBTIDAIYAH,” vol. 2, no. 1, hal. 33–41, 2021.
- [11] T. Poppendieck, H. Kniberg, R. Bunning, C. Thompson, dan J. Coplien, Scruminc. 2020.
- [12] M. sari Zulvi, “Systematic Literature Review Penerapan Metodologi Agile Dalam Berbagai Bidang,” J. Komput. Terap., vol. 7, no. 2, hal. 300–313, 2021, doi: 10.35143/jkt.v7i2.5116.
- [13] J. Kouraklis, MVVM in Delphi, no. October 2016. Berkeley, CA: Apress, 2016. doi: 10.1007/978-1-4842-2214-0_1.
- [14] E. Milla dan M. Radonjić, “Analysis of Developing Native Android Applications Using Xml and Jetpack Compose,” Balk. J. Appl. Math. Informatics, vol. 6, no. 2, hal. 167–178, 2023, [Daring]. Tersedia pada: <https://js.ugd.edu.mk/index.php/bjmi/article/view/6019>
- [15] K. Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process. Addison-Wesley Professional, 2019.
- [16] L. R. Aiken, “Three coefficients for analyzing the reliability and validity of ratings,” Educ. Psychol. Meas., vol. 45, no. 1, hal. 131–142, 1985.
- [17] S. Arikunto, Dasar dasar evaluasi pendidikan.pdf, Edisi 3. PT. Bumi Aksara, 2018.
- [18] E. H. Kusuma Dewi, I. S. Pratama, A. S. Putera, dan C. Carudin, “Black Box Testing pada Aplikasi Pencatatan Peminjaman Buku Menggunakan Boundary Value Analysis,” STRING (Satuan Tulisan Ris. dan Inov. Teknol., vol. 6, no. 3, hal. 315, 2022, doi: 10.30998/string.v6i3.11958.
- [19] F. N. Hasanah dan R. S. Untari, Rekayasa Perangkat Lunak. UMSIDA PRESS, 2020.