

SISTEM MONITORING IRIGASI OTOMATIS MENGUNAKAN APLIKASI MOBILE ANDROID BERBASIS KOTLIN

Deby Irawan¹, M. Taqijuddin Alawiy², Anang Habibi³

¹Jurusan Teknik Elektro, Fakultas Teknik, Universitas Islam Malang

^{2,3}Jurusan Teknik Elektro, Fakultas Teknik, Universitas Islam Malang

21901053007@unisma.ac.id¹, taqijuddin.alawiy@unisma.ac.id²,
ananghabibi@unisma.ac.id³

Abstract

Agriculture has exerted significant pressure on water resources among other sectors in arid and semi-arid regions. The recent adoption of drip irrigation systems has attracted attention due to its high potential in improving water use efficiency. The rapid pace of technological development leaves open the possibility for Indonesian farmers to renew agricultural systems, including the use of Internet of Things technology and mobile applications. Considering the frequent occurrence of water supply problems in agriculture, both in terms of location and time, and the fact that many people still use manual labor in the irrigation process, such as using a watering can or hose, these problems can be overcome by using drip irrigation methods. Drip irrigation is a method of watering at the root of the plant, with only a few parts of the plant usually being moistened, typically 10% to 15%. By utilizing mobile applications, especially in the Kotlin programming language, implementing an irrigation system is highly effective for monitoring and remotely controlling sensors in agricultural land that are integrated with API services. This study used the Rapid Application Development (RAD) method, which aims to minimize the duration needed to design and implement an information system, resulting in a very short system development cycle. The process of monitoring temperature, humidity, and water flow sensors through the REST API using the GET method, while the process of controlling water and fertilizer irrigation uses the POST method. Based on the results of the Black Box testing, 100% of the features worked well, and the results of the survey of 32 respondents, consisting of 2 IT experts, 1 lecturer, and 29 students, showed an 88.18% success rate.

Keywords— Kotlin, irrigation systems, REST APIs, monitoring, android.

Abstraksi

Pertanian telah memberikan tekanan besar pada sumber daya air di antara sektor-sektor lain di daerah kering dan semi-kering. Penerapan sistem irigasi tetes baru-baru ini menarik perhatian karena potensinya yang tinggi dalam meningkatkan efisiensi penggunaan air. Lajunya perkembangan teknologi yang semakin cepat, tidak menutup kemungkinan petani Indonesia melakukan pembaharuan sistem di sektor pertanian. Salah satunya adalah menggunakan teknologi *Internet of Things* serta aplikasi *Mobile*. Meninjau sering terjadinya masalah terhadap suplai air di bidang pertanian, baik dari sektor lokasi maupun waktu, serta masyarakat yang masih banyak menggunakan tenaga manual dalam proses irigasi, seperti menggunakan gembor dan selang pipa dapat diatasi dengan penggunaan metode penyiraman irigasi tetes. Irigasi tetes merupakan metode penyiraman pada titik akar tanaman, hanya beberapa bagian tanaman yang dibasahi biasanya 10% dan 15%[1]. Dengan memanfaatkan aplikasi *mobile* khususnya pada bahasa pemrograman kotlin, maka menerapkan sistem irigasi sangat efektif untuk memonitoring dan kontrol jarak jauh sensor yang ada di lahan pertanian dan terintegrasi dengan API service. Dalam penelitian ini menggunakan metode *Rapid Application Development* (RAD) yang bertujuan untuk menekan durasi waktu yang dibutuhkan atau diperlukan untuk mendesain serta mengimplementasikan sistem informasi sehingga dihasilkan siklus pembuatan sistem yang sangat pendek. Proses Monitoring suhu, kelembapan dan sensor waterflow melalui REST API dengan menggunakan metode GET, sedangkan proses kontrol penyiraman air dan pupuk menggunakan metode POST. Berdasarkan hasil pengujian *Black Box* menunjukkan 100 % fitur berjalan dengan baik, dan hasil dari pengujian responden sebanyak 32 orang yang terdiri dari 2 pakar IT, 1 dosen, 29 mahasiswa menghasilkan persentase 88,18 % berjalan sesuai yang diharapkan.

Kata Kunci— Kotlin, sistem irigasi, REST API, monitoring, android.

I. PENDAHULUAN

Pertanian telah memberikan tekanan besar pada sumber daya air di antara sektor-sektor lain di daerah kering dan semi-kering. Penerapan sistem irigasi tetes baru-baru ini menarik perhatian karena potensinya yang tinggi dalam meningkatkan efisiensi penggunaan air. Karena alokasi yang akurat dari air dan nutrisi, irigasi tetes memberikan manajemen yang tepat dan manipulasi kadar air tanah menjadi lebih optimal. Kebutuhan akan air sangat penting terutama untuk memenuhi kebutuhan kelestarian tanaman atau lahan pertanian. Pengaturan kontrol perlu diterapkan sistem saluran irigasi untuk mengoptimalkan penggunaan suplai air.

Indonesia adalah negara agraris yang mayoritas penduduk Indonesia bermata pencaharian sebagai petani. Salah satu kebutuhan pokok dalam bercocok tanam adalah air. Karena tumbuhan membutuhkan air untuk melakukan fotosintesis dan menjaga tumbuhan agar tetap segar, sehat dan tidak layu. Kebutuhan air sangat penting, terutama untuk memenuhi kebutuhan kelestarian tanaman atau lahan pertanian. Perlu dilakukan pengaturan pengendalian sistem saluran irigasi yang dapat mengoptimalkan pemanfaatan suplai air.

Lajunya perkembangan teknologi yang semakin cepat, tidak menutup kemungkinan petani Indonesia harus melakukan pembaharuan sistem di sektor pertanian. Salah satunya adalah menggunakan teknologi *Internet of Things* serta aplikasi *Mobile*. Perkembangan teknologi smartphone berbasis android yang begitu pesat khususnya di bidang agraris, akan membuka peluang untuk pengembangan sistem informasi di bidang agraris dalam bentuk aplikasi *mobile* android[2].

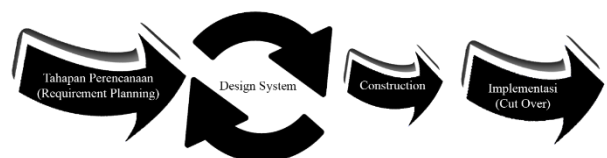
Meninjau sering terjadinya masalah terhadap suplai air di bidang pertanian, baik dari sektor lokasi maupun waktu, serta masyarakat yang masih banyak menggunakan tenaga manual dalam proses irigasi, seperti menggunakan gembor dan selang pipa dapat diatasi dengan penggunaan metode penyiraman irigasi tetes. Irigasi tetes merupakan metode penyiraman pada titik akar tanaman, hanya beberapa bagian tanaman yang dibasahi biasanya 10% dan 15%[1]. Suhu dan kelembaban merupakan variabel lingkungan yang sangat penting untuk diperhatikan dalam bercocok tanam tanaman selada hidroponik di dalam sebuah greenhouse. Pertumbuhan selada akan optimal pada kisaran suhu udara 25 °C sampai 28 °C dan kelembaban berkisar antara 65% sampai 78%[3]. Pada saat menyiram tanaman, sering kali mengalami naik turunnya debit air yang tidak terkontrol, sehingga dibutuhkan alat untuk mengatur volume air di sektor pertanian.

Pada penelitian sebelumnya telah mendeteksi air hujan dan sistem otomatis buka tutup pintu bendungan, monitoring aktivitas saluran irigasi seperti, memantau suhu, ketinggian air dengan mengintegrasikan aplikasi *mobile* android[4]. Untuk menghubungkan antara *mobile apps* dengan *web application* maka akan digunakan sebuah teknologi yang dinamakan Web Service yang akan menggunakan arsitektur HTTP RESTFul untuk menghasilkan sebuah API[5]. JetBrains adalah sebuah perusahaan *software development* yang telah mengembangkan bahasa pemrograman *Kotlin* dan juga perusahaan yang mengembangkan *software* Android Studio IDE. Berawal dari bahasa pemrograman *Java*, kini telah berkembang menjadi bahasa *Kotlin*, dan tentu saja *Kotlin* memiliki fitur-fitur yang lebih canggih, modern dan lebih unggul. Penggunaan aplikasi *mobile* berbahasa pemrograman *Kotlin* yang disinyalir lebih efektif daripada penggunaan bahasa *Java*[6]. Dengan memanfaatkan aplikasi *mobile* khususnya pada bahasa pemrograman *kotlin*, maka menerapkan sistem irigasi sangat efektif untuk memonitoring dan kontrol jarak jauh sensor yang ada di lahan pertanian dan terintegrasi dengan API service.

Fitur-fitur yang akan ditampilkan pada aplikasi sistem irigasi ini yaitu monitoring sensor kelembapan dan suhu, sensor flow, serta kontrol penyiraman air dan irigasi pupuk. Sistem irigasi berbasis IoT dan Aplikasi *Kotlin* ini sangat cocok digunakan karena memudahkan petani untuk monitoring maupun kontrol penyiraman lahan, dan juga menekan penggunaan *memory usage* pada *smartphone*, sehingga pekerjaan-pekerjaan di lahan pertanian dapat berjalan dengan efektif dan efisien.

II. METODE PENELITIAN

Penelitian ini menggunakan Metode (Rapid Application Development) yang bertujuan untuk menekan durasi waktu yang dibutuhkan atau diperlukan untuk mendesain serta mengimplementasikan sistem informasi sehingga dihasilkan siklus pembuatan sistem yang sangat pendek, tidak memerlukan banyak waktu yang terlalu panjang dan terbuang sia-sia.



Gambar 1. Metode RAD

2.1 Tahap Perencanaan (Planning)

Fokus dari tahap perencanaan adalah untuk mengidentifikasi semua kebutuhan yang diperlukan dalam pelaksanaan penelitian. Tahap awal ini sangat penting untuk meminimalkan potensi hambatan dalam jalannya penelitian. Tahapan ini terdiri dari tiga bagian, yaitu kebutuhan perangkat keras (Hardware Requirement), kebutuhan perangkat lunak (Software Requirement), dan kebutuhan data (Data Requirement).

2.2 Desain Sistem (System Design)

Desain Sistem adalah tahapan untuk membuat bagan-bagan atau diagram perancangan atau gambaran dari sistem program yang akan dibuat. Dalam desain ini menyangkut *use case diagram*, *activity diagram*, *data flow diagram*, *flowchart*, dan diagram konteks.

2.3 Pembuatan Cepat (Rapid Construction)

Tahap ini melibatkan pengujian sistem program yang telah dibuat dengan menggunakan berbagai skenario yang telah disiapkan. Pembangunan dengan bahasa pemrograman atau *coding* untuk membuat program aplikasi yang diperlukan. Pembangunan aplikasi menggunakan Android Studio dengan bahasa pemrograman Kotlin dan menggunakan API sebagai penyambung antara aplikasi dengan web. Untuk pengujian menggunakan sistem pengujian Black Box dan pengujian responden.

2.4 Implementasi (Cutover)

Setelah mendapat hasil yang diharapkan telah tercapai pada tahap Pembuatan Cepat, maka selanjutnya sistem program akan diimplementasikan.

III. HASIL DAN PEMBAHASAN

3.1. Tahapan Perancangan

Dalam tahapan pengembangan perangkat lunak ini menggunakan metode RAD (*Rapid Application Development*). Secara keseluruhan terdiri dari beberapa tahapan untuk membuat Sistem Irigasi Otomatis Menggunakan Aplikasi Mobile Android Berbasis Kotlin agar dapat diselesaikan dengan baik.

3.1.1. Kebutuhan Pengguna

3.1.1.1. Analisis Kebutuhan Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang bertumpu pada perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat lunak, perangkat keras, serta user sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.1.2. Kebutuhan Non-Fungsional

Penggambaran sistem dari analisis kebutuhan non fungsional bertumpu pada kebutuhan perangkat lunak, perangkat keras, serta *brainware* sebagai analisis kekurangan dan kebutuhan yang harus terpenuhi dalam merancang sistem informasi..

1. Perangkat Lunak (*software*)

Perangkat lunak digunakan dalam sebuah sistem yang merupakan sebuah himpunan perangkat yang mendukung atau dibutuhkan untuk membangun aplikasi ini. Perangkat lunak tersebut adalah sebagai berikut :

Adapun perangkat lunak yang digunakan dalam aplikasi ini adalah sebagai berikut:

- Bahasa Pemrograman Kotlin, dalam hal ini digunakan Java Development Kit (JDK) 1.6 dan Java Runtime Environment (JRE).
- Sistem Operasi Windows 8.1 Pro (64 bit).
- Android Software Development Kit (Android SDK).
- Android Studio.
- Adobe Photoshop CS6.
- Microsoft Office 2016.
- StarUML.
- MEMu Emulator.

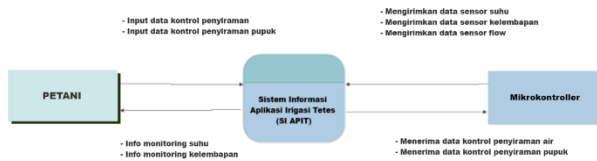
2. Perangkat Keras (*Hardware*)

- a. Laptop ASUS X450CA dengan spesifikasi sebagai berikut:
 - Processor Intel® Celeron® CPU 1007U 1.5GHz
 - Intel HD Graphics
 - Standard Memory 10 GB DDR3 2 Slot
 - Hard Drive Type 500 GB
 - Display Size 14" HD LED LCD
- b. Ponsel Vivo Y91C dengan spesifikasi sebagai berikut:
 - Android 8.1 Oreo dengan antarmuka Funtouch 4.5.
 - MediaTek Helio P22 octa-core processor.
 - Octa-core 2.0 GHz
 - Camera primer 13 MP, Camera Sekunder 5 MP
 - Memory Internal 32 GB, RAM 2 GB.
 - Layar 6.22-inch HD+ IPS LCD dengan resolusi 1520 x 720 pixels.

3.1.2. Data Flow Diagram

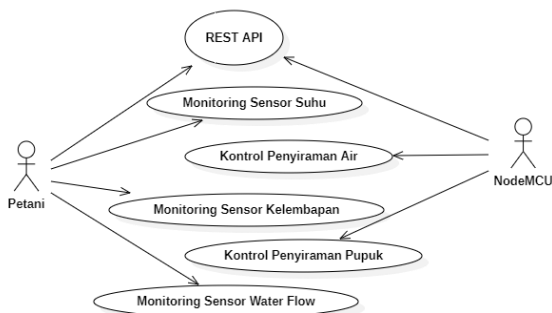
Pada setiap aplikasi memiliki data yang dikirimkan dan diterima oleh petani melalui API web service. Disini pertukaran data terjadi Ketika petani menginput kontrol penyiraman pada aplikasi lalu API mengupload kedalam web service dan di teruskan ke Node MCU, kemudian aplikasi akan melakukan penyiraman secara otomatis. Untuk monitoring sensor prosesnya adalah kebalikannya, dari mikrokontroler Node MCU akan mengirimkan data kedalam web

service, kemudian aplikasi akan menerima data dari web service melalui REST API, data yang dikirimkan meliputi sensor waterflow, sensor suhu dan kelembapan.



Gambar 2. Data Flow Diagram level 0

3.1.3. Use Case Diagram

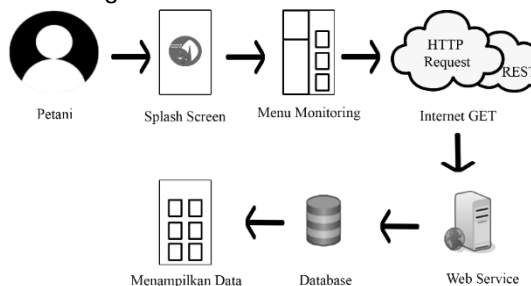


Gambar 3. Use Case Diagram

Pada Gambar 3 adalah merupakan alur penggunaan aplikasi irigasi tetes dimana aplikasi harus terhubung dengan internet terlebih dahulu sebelum digunakan, karena untuk terhubung dengan web service aplikasi harus diintegrasikan dengan REST API di android. Dalam REST API ada dua metode yang digunakan yaitu GET dan POST, metode GET digunakan untuk mendapatkan data dari hasil upload Node MCU ke web service, sedangkan metode POST digunakan untuk mengirim data ke web service dan diteruskan ke Node MCU. GET digunakan memonitoring suhu, kelembapan, dan *waterflow*, sedangkan POST digunakan untuk kontrol irigasi air dan kontrol irigasi pupuk.

3.1.4 Alur kerja Sistem

a. Monitoring

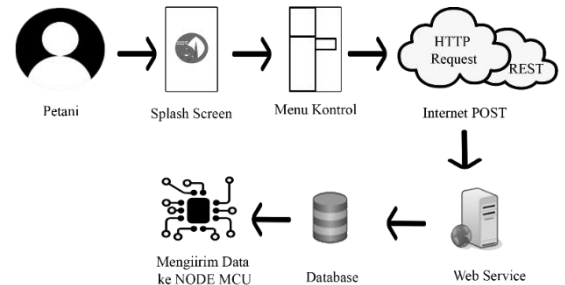


Gambar 4. Alur Kerja Sistem monitoring

Keterangan Gambar 4 alur kerja sistem monitoring yaitu : Petani membuka aplikasi akan ditampilkan *splash screen* dan akan langsung masuk ke halaman monitoring suhu dan

kelembapan. Setelah sampai di halaman monitoring, data yang didapatkan berasal dari web service dan harus terkoneksi dengan internet, kemudian data dari database akan ditampilkan di aplikasi android dengan menggunakan metode REST API GET.

b. Kontrol

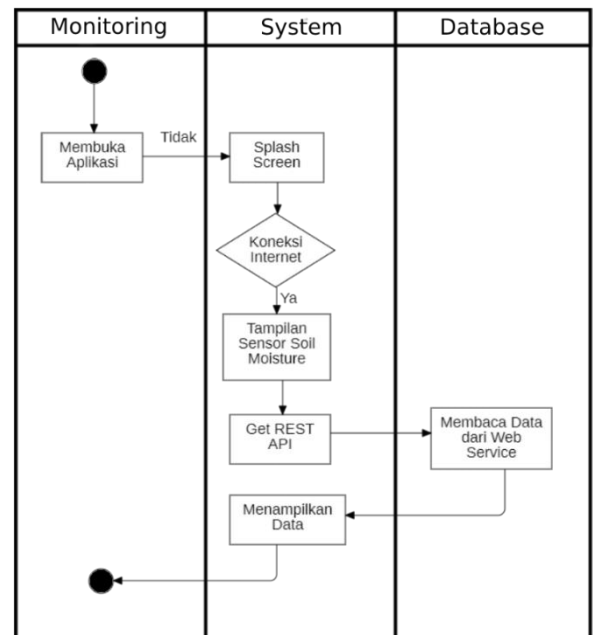


Gambar 5. Alur Kerja Sistem Kontrol

Keterangan Gambar 3.11 alur kerja sistem kontrol yaitu : Petani membuka aplikasi akan ditampilkan *splash screen* dan akan langsung masuk ke halaman monitoring suhu dan kelembapan. Disamping menu masuk ke halaman kontrol, data akan dikirimkan ke web service dengan menggunakan metode REST API POST dan harus terkoneksi dengan internet, kemudian data dari database akan dikirimkan ke *embedded system* atau ke NODE MCU dan akan menyiram tanaman secara otomatis.

3.1.5. Activity Diagram

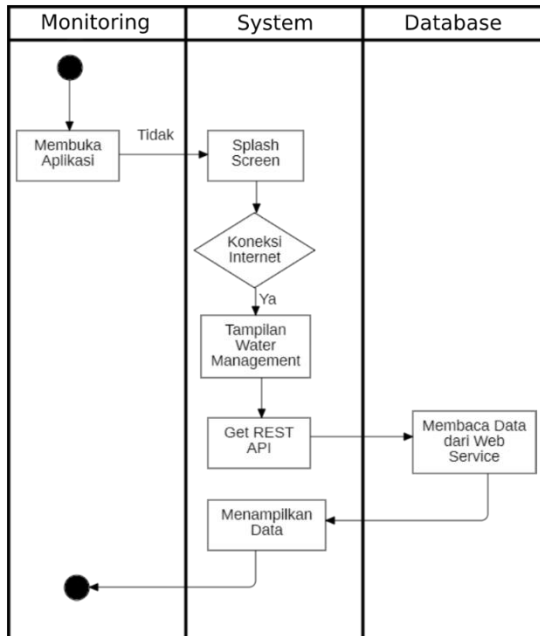
a. Activity Diagram Monitoring



Gambar 6. Activity Diagram Soil Moisture

Pada Gambar 6 merupakan *activity* untuk monitoring sensor *soil moisture*, dimana pada diagram tersebut menjelaskan alur apa saja untuk memonitoring sensor suhu dan kelembapan. Ketika petani membuka aplikasi

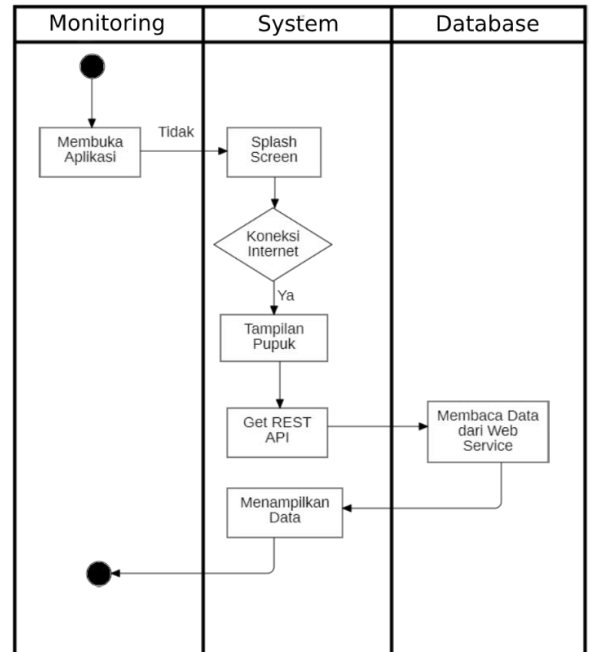
akan ditampilkan *splash screen* di awal dan otomatis masuk ke menu *Sensor Soil Moisture*. Data monitoring berasal dari Node MCU yang mengirim data ke *web service*, kemudian akan diterima di aplikasi sistem irigasi yang terintegrasi dengan *API service*.



Gambar 7. Activity Diagram Water Management

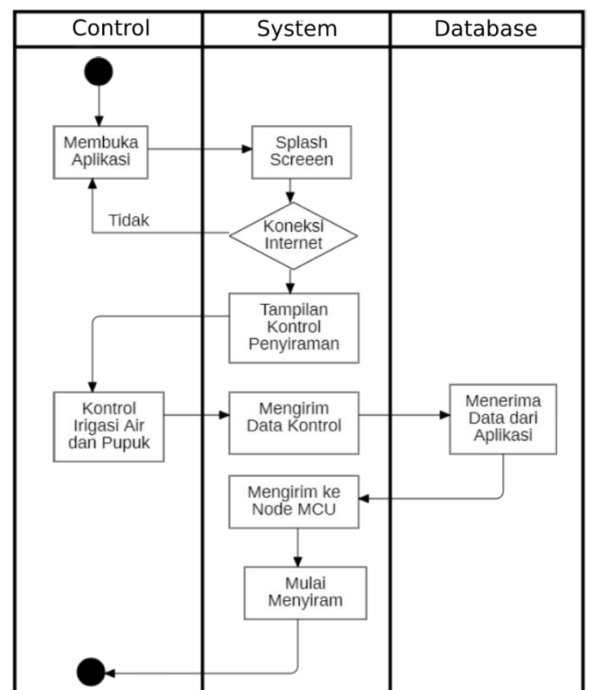
Pada Gambar 7 merupakan *activity* untuk monitoring *water management*, dimana pada diagram tersebut menjelaskan alur apa saja untuk memonitoring sensor *water flow*. Ketika petani membuka aplikasi akan ditampilkan *splash screen* di awal dan otomatis masuk ke menu *Sensor Soil Moisture* pilih menu *Water Management* pada menu samping. Data monitoring berasal dari Node MCU yang mengirim data ke *web service*, kemudian akan diterima di aplikasi sistem irigasi yang terintegrasi dengan *API service*.

Pada Gambar 8 merupakan *activity* untuk monitoring pupuk, dimana pada diagram tersebut menjelaskan alur apa saja untuk memonitoring sensor *water flow*. Ketika petani membuka aplikasi akan ditampilkan *splash screen* di awal dan otomatis masuk ke menu *Sensor Soil Moisture* pilih menu Pupuk pada menu samping. Data monitoring berasal dari Node MCU yang mengirim data ke *web service*, kemudian akan diterima di aplikasi sistem irigasi yang terintegrasi dengan *API service*.



Gambar 8. Activity Diagram Pupuk

b. Activity Diagram Kontrol



Gambar 9. Activity Diagram Kontrol

Pada Gambar 3.7 merupakan diagram *activity* Kontrol, dimana pada diagram tersebut menjelaskan tentang alur kontrol penyiraman yang meliputi irigasi air dan irigasi pupuk. Ketika petani membuka aplikasi akan ditampilkan *splash screen* di awal dan otomatis masuk ke aplikasi monitoring. Pada menu *control* petani dapat mengatur jumlah keluaran air dan dapat memilih pipa yang akan digunakan, data tersebut akan dikirim ke *web service* yang akan diteruskan ke Node MCU alat akan menyiram secara otomatis.

3.2 Tampilan Interface

1. Halaman Splash Screen



Gambar 10. Tampilan Splash Screen

Pada Gambar 10 menunjukan bahwa halaman pertama saat membuka aplikasi, pada halaman ini terdiri *ImageView* untuk menampilkan logo dari aplikasi SIAPIT dan *Progresbar* dengan durasi 3 detik.

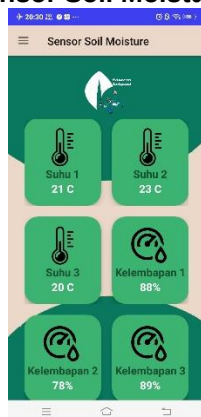
2. Halaman Menu Navigasi



Gambar 11. Tampilan Menu Navigasi

Pada Gambar 11 menunjukan bahwa setelah membuka ikon menu di samping kiri atas akan muncul beberapa menu pilihan. Pada halaman ini menggunakan layout *DrawerLayout* yang terdiri dari *ImageView* untuk menampilkan logo dari aplikasi SIAPIT, *TextView* untuk menampilkan teks Selamat Datang, *MenuItem* untuk menampilkan menu, dan *NavigationView* untuk menyatukan beberapa layout menjadi satu.

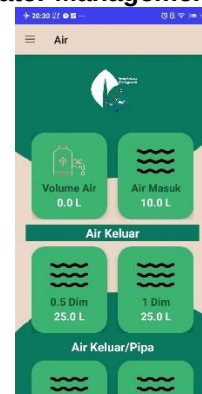
3. Halaman Sensor Soil Moisture



Gambar 12. Halaman Sensor Soil Moisture

Pada Gambar 12 menunjukan bahwa halaman fragment *Sensor Soil Moisture*, pada halaman ini terdiri dari *ImageView* pertama untuk menampilkan logo dari aplikasi SIAPIT, *CardView* sebagai wadah dari data yang akan ditampilkan, *ImageView* kedua untuk ikon sensor suhu dan kelembapan, *TextView* pertama untuk variabel nama sensor suhu 1, suhu 2, suhu 3, kelembapan 1, kelembapan 2, kelembapan 3, dan *TextView* kedua untuk memanggil data dari *Web Service* melalui REST API.

4. Halaman Water Management



Gambar 13. Halaman Water Management

Pada Gambar 13 menunjukan bahwa halaman fragment *Sensor Water Management*, pada halaman ini terdiri dari *ImageView* pertama untuk menampilkan logo dari aplikasi SIAPIT, *CardView* sebagai wadah dari data yang akan ditampilkan, *ImageView* kedua untuk ikon sensor suhu dan kelembapan, *TextView* pertama untuk variabel nama sensor Volume Air, Air Masuk, Air Keluar 0.5, Air Keluar 1, Air Keluar per pipa 0.5, Air Keluar per pipa 1, dan *TextView* kedua untuk memanggil data dari *Web Service* melalui REST API

5. Halaman Pupuk



Gambar 14. Halaman pupuk

Pada Gambar 14 menunjukan bahwa halaman fragment *Sensor Water Management*, pada halaman ini terdiri dari

ImageView pertama untuk menampilkan logo dari aplikasi SIAPIT, *CardView* sebagai wadah dari data yang akan ditampilkan, *ImageView* kedua untuk ikon sensor suhu dan kelembapan, *TextView* pertama untuk variabel nama sensor Volume Pupuk dan Debit, dan *TextView* kedua untuk memanggil data dari *Web Service* melalui REST API

6. Halaman Kontrol



Gambar 15. Halaman Kontrol

Pada Gambar 15 menunjukkan bahwa halaman fragment *control*, pada halaman ini terdiri dari *ImageView* pertama untuk menampilkan logo dari aplikasi SIAPIT, *CardView* yang pertama sebagai wadah dari data yang akan ditampilkan, *TextView* pertama untuk variabel nama kontrol air, dan *TextView* kedua untuk menunjukkan jam setempat, *EditText* pertama untuk menginputkan jumlah air dalam liter, *EditText* kedua menggunakan *dropdown* pemilihan pipa 0.5 dan pipa 1, dan *Button* sebagai aksi untuk mengirimkan data ke *Web Service* melalui REST API.

3.3 Pengujian

a. Pengujian Black Box

Tabel 1. Pengujian Black Box Fitur

No	Penilaian	Jumlah Hasil kriteria	Jumlah skor
1.	Ya	10	1000
2.	Tidak	0	0
Jumlah			1000

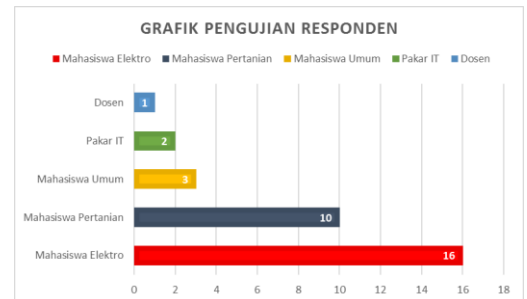
$$\begin{aligned}
 \text{Persentase} &= \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\% \\
 &= \frac{1000}{1000} \times 100\% \\
 &= 100\% \text{ Sangat layak}
 \end{aligned}$$

Keterangan :

Dari perhitungan menggunakan skala likert dengan tahap menghitung skor hasil observasi yaitu hasil perhitungan dari pengujian memperoleh skor 1000 di bagi dengan skor yang di harapkan yaitu skor dari perhitungan skor tertinggi ideal

menapatkan hasil 1000 di kali 100% maka memperoleh hasil 100% maka hasil pengujian ini mendapatkan kategori sangat layak karena memperoleh poin lebih besar dari perhitungan rentang skor.

b. Pengujian Responden



Gambar 16. Grafik Pengujian Responden

Keterangan pada Gambar 16 yaitu perolehan responden dari pengguna aplikasi SIAPIT berjumlah 32 orang dengan keterangan 16 Mahasiswa Elektro, 10 Mahasiswa Pertanian, 3 Mahasiswa Umum, 2 Pakar IT dari CV. Erdatamedia, dan 1 Dosen Teknik Elektro.

Tabel 2. Hasil Pengujian Responden

No	Penilaian	Poin	Jumlah skor	Total
1.	Sangat Kurang	1	0	0
2.	Kurang	2	0	0
3.	Cukup Baik	3	15	45
4.	Baik	4	178	712
5.	Sangat Baik	5	159	795
Jumlah				1552

$$\begin{aligned}
 \text{Persentase} &= \frac{\text{Skor yang diobservasi}}{\text{Skor yang diharapkan}} \times 100\% \\
 &= \frac{1552}{1760} \times 100\% \\
 &= 88,18 \%
 \end{aligned}$$

Dari perhitungan presentasi kelayakan diperoleh hasil 88,18% dan perhitungan menggunakan skala likert diperoleh hasil nilai X = 1552. Dengan demikian “**SISTEM INFORMASI APLIKASI IRIGASI TETES**” masuk ke dalam kategori sangat layak.

IV. KESIMPULAN

Dari hasil penelitian dan pembahasan yang dibuat oleh penulis, maka untuk membangun aplikasi monitoring sensor suhu, kelembapan, *water flow*, kontrol irigasi air dan pupuk secara otomatis di Laboratorium Enviro Hydro, ada beberapa tahap yang harus dilalui, seperti analisis kebutuhan, analisis software dan hardware, desain sistem dan antarmuka, metode *Rapid Application Development* (perancangan desain antarmuka dan pembuatan program),

dan pengujian (uji fitur menu). Aplikasi SIAPIT ini dibuat menggunakan perangkat lunak Android Studio dengan bahasa pemrograman Kotlin, yang mana aplikasi ini untuk memenuhi permintaan dari objek penelitian.

Berdasarkan hasil pengujian Black Box pada fitur, di mana aplikasi ini dapat berfungsi dengan baik dan efisien, dengan presentase 100% berjalan sesuai dengan yang diinginkan. Sementara hasil dari kuesioner menunjukkan bahwa responden menganggap aplikasi SIAPIT cukup penting untuk digunakan, dengan presentase 88,18 % untuk fitur yang di sediakan dan berjalan sesuai yang diharapkan.

DAFTAR PUSTAKA

- [1] F. L. Tenggara, B. H. Rinuastuti, Z. Handayani, and B. P. T. Anjani, "Pemanfaatan Metode Irigasi tetes Sederhana Untuk Budidaya Tanaman Hortikultura Di Desa Ungga, Kecamatan Praya Barat Daya, Kabupaten Lombok Tengah," *J. Pengabd. Magister Pendidik. IPA*, vol. 5, no. 2, pp. 267–271, 2022, doi: 10.29303/jpmipi.v5i2.1786.
- [2] S. Pebrianto, A. Andasuryani, and K. Fahmy, "Sistem Informasi Alat Dan Mesin Pertanian Berbasis Aplikasi Android Di Kecamatan Pariaman Utara Kota Pariaman," *J. Teknol. Pertan. Andalas*, vol. 24, no. 2, p. 98, 2020, doi: 10.25077/jtpa.24.2.98-108.2020.
- [3] A. A. Angga Dwipa, I. G. P. W. Wedashwara W, and A. Zubaidi, "Rancang Bangun Sistem Conditionning Udara Berbasis IoT pada Studi Kasus Tanaman Selada Hidroponik," *J. Comput. Sci. Informatics Eng.*, vol. 4, no. 1, pp. 16–25, 2020, doi: 10.29303/jcosine.v4i1.297.
- [4] D. Setiadi and M. N. Abdul Muhaemin, "PENERAPAN INTERNET OF THINGS (IoT) PADA SISTEM MONITORING IRIGASI (SMART IRIGASI)," *Infotronik J. Teknol. Inf. dan Elektron.*, vol. 3, no. 2, p. 95, 2018, doi: 10.32897/infotronik.2018.3.2.108.
- [5] P. B. Ramadhanu and A. T. Priandika, "Rancang Bangun Web Service Api Aplikasi Sentralisasi Produk Umkm Pada Uptd Plut Kumkm Provinsi Lampung," *J. Teknol. dan Sist. Inf.*, vol. 2, no. 1, pp. 59–64, 2021, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/JTSI>
- [6] N. S. Sibarani, G. Munawar, and B. Wisnuadhi, "Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin. In Prosiding Industrial Research Workshop and National Seminar," vol. 9, no. December, 2018.
- [7] R. C. Andrian, E. Noerhayati, and S. Bambang, "Pengoperasian pintu air irigasi otomatis berbasis iot pada perencanaan pola tata tanam di kecamatan tumpang - malang," *J. Rekayasa Sipil*, vol. 9, no. 1, pp. 1–14, 2021, [Online]. Available: <http://riset.unisma.ac.id/index.php/ft/article/view/10274>
- [8] A. Duana, P. M. Jasa, A. Bambang, and D. Sulo, "ALAT PENYIRAMAN OTOMATIS BERBASIS IoT (Internet Of Things) UNTUK BUDIDAYA PAKCOY," vol. 13, 2021.
- [9] A. Sapei, "Irigasi tetes," *Tek. Tanah Dan Air Dep. Tek. Pertan. Fateta IPB*, pp. 1–44, 2006.
- [10] Y. Yudhanto and A. Wijayanto, "Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio," *Kompas Gramedia*, p. 176, 2017.
- [11] I. Kurniawan, Humaira, and F. Rozi, "REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android," *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 1, no. 4, pp. 127–132, 2020, doi: 10.30630/jitsi.1.4.18.
- [12] A. Wantoro, S. Samsugi, and M. J. Suharyanto, "Sistem Monitoring Perawatan dan Perbaikan Fasilitas PT PLN (Studi Kasus : Kota Metro Lampung)," *J. TEKNO KOMPAK*, vol. 15, no. 1, pp. 116–130, 2021.
- [13] M. Nurdiansyah, E. C. Sinurat, M. Bakri, and I. Ahmad, "Sistem Kendali Rotasi Matahari Pada Panel Surya Berbasis Arduino UNO," *J. Tek. dan Sist. Komput.*, vol. 1, no. 2, pp. 7–12, 2020, doi: 10.33365/jtikom.v1i2.14.
- [14] U. Rahardja, Q. Aini, and N. P. L. Santoso, "Pengintegrasian YII Framework Berbasis API pada Sistem Penilaian Absensi," *Sisfotenika*, vol. 8, no. 2, p. 140, 2018, doi: 10.30700/jst.v8i2.403.
- [15] Syamsiah, "STRING (Satuan Tulisan Riset dan Inovasi Teknologi)," vol. 4, no. 1, pp. 86–93, 2019.