

BAB I

PENDAHULUAN

A. Latar Belakang

Irigasi merupakan upaya yang dilakukan manusia untuk mengairi lahan pertanian. Dalam dunia modern, saat ini sudah banyak model irigasi yang dapat dilakukan manusia (Najimuddin, 2019).

Sistem irigasi yang terdapat di Indonesia kebanyakan masih memakai sistem manual, salah satunya di Kec. Mattiro Bulu, Kabupaten Pinrang, yaitu dimana membuka dan menutup saluran irigasi masih secara tradisional. Dimana apabila terjadi peluapan air di akibatkan hujan deras, petugas merasa kesulitan untuk mengatur irigasi dikarenakan jarak yang jauh. Jadi masih sering kali petani ataupun petugas pulang pergi untuk mengecek saluran irigasi.

Untuk mengatasi masalah tersebut dibutuhkan sistem baru dan otomatis untuk membantu meringankan pekerjaan petani terkhususnya penjaga pintu air. Dengan memanfaatkan teknologi Internet dan Smartphone sebagai sarana kendali jarak jauh, Sistem ini mempunyai akses untuk membuka dan menutup pintu irigasi, teknologi ini dinamakan *Internet of Things* (IOT), sehingga memudahkan petani maupun petugas dalam mengontrol sistem irigasi.

Internet of Things atau disingkat dengan istilah IoT adalah sebuah istilah dengan pengertian sebuah akses perangkat elektronik melalui media internet. (Wasista et al., 2019). IoT menjadi suatu bidang penelitian tersendiri semenjak internet mulai berkembang dengan pesat, karena semakin berkembangnya

kebutuhan manusia terhadap sebuah teknologi, maka semakin banyak pula penelitian mengenai IoT diantaranya yaitu pengembangan sistem monitoring irigasi.

Pengembangan sistem monitoring irigasi otomatis berbasis Android memiliki potensi untuk memberikan manfaat signifikan, seperti kemampuan mengetahui ketinggian air serta debit air secara *real-time*. Dengan menggabungkan pengetahuan dalam irigasi dan teknologi informasi dan otomatisasi, pengembangan sistem ini bertujuan untuk menyediakan alat yang lebih efisien dan efektif bagi petani dalam mengelola irigasi dan pertumbuhan tanaman. Melalui latar belakang ini, diharapkan dapat menemukan solusi dan kontribusi dalam meningkatkan produktivitas pertanian secara berkelanjutan.

B. Rumusan Masalah

Adapun rumusan masalah yang akan dibahas adalah bagaimana mengendalikan dan memonitoring Pintu Irigasi Otomatis berbasis *Android*.

C. Tujuan Penelitian

Adapun tujuan penelitian dari proposal ini yakni, sebagai berikut :

1. Mengembangkan cara operasi sensor air (*water flow*) dan sensor level air (*water level*) menggunakan NodeMCU dengan bantuan Arduino IDE.
2. Menyusun langkah-langkah yang diperlukan untuk guna menghubungkan NodeMCU dengan *firebase*, sehingga data yang diperoleh dapat dari sensor dapat disimpan dan diakses secara online.
3. Mengintegrasikan layanan *firebase* ke dalam aplikasi android, memungkinkan pengguna untuk memantau dan mengelola data sensor air berbasis android.

4. Mengoptimalkan penggunaan NodeMCU dalam sistem monitoring sehingga mampu menampilkan data sensor air dan mengontrol irigasi secara otomatis berbasis android.

D. Batasan Masalah

Penelitian ini akan memfokuskan pada pengembangan aplikasi *Android* untuk memonitor dan mengendalikan system irigasi otomatis. Penelitian ini akan berfokus pada skala *prototype* dan tidak akan membahas secara mendalam tentang desain perangkat keras, platform *Non-Android*, atau aspek keamanan data yang rumit.

E. Manfaat Penelitian

Adapun manfaat penelitian dari proposal ini, yaitu :

1. Manfaat bagi penulis

Mendapatkan kesempatan untuk mengembangkan keterampilan dalam pengembangan system teknologi, pemrograman aplikasi *Android*, dan pemahaman lebih mendalam tentang irigasi otomatis. Penelitian ini juga dapat meningkatkan rekam jejak profesional penulis dalam bidang teknologi irigasi dan pengembangan aplikasi.

2. Manfaat bagi pengguna

Memberikan solusi yang efisien dan mudah digunakan untuk mengontrol irigasi secara otomatis. Pengguna dapat lebih mengoptimalkan penggunaan air, menghemat biaya operasional, serta meningkatkan produktivitas pertanian dengan menghindari keukrangan atau kelebihan air yang berdampak negative pada pertumbuhan tanaman.

3. Manfaat bagi pembaca

Memberi wawasan tentang bagaimana teknologi modern, seperti aplikasi Android dan IoT, dapat diaplikasikan dalam irigasi. Hal ini dapat mendorong pembaca untuk lebih terbuka terhadap inovasi dalam ilmu irigasi serta memahami manfaat penggunaan teknologi untuk meningkatkan efisiensi dan keberlanjutan.

4. Manfaat bagi Instansi

Untuk instansi yang bergerak di bidang Irigasi alat ini dapat memantau ketinggian air dan debit air secara *real-time*. Hal ini memungkinkan pengambilan keputusan yang lebih tepat dalam pengaturan aliran air. Mengoptimalkan penggunaan sumber daya, dan mengurangi pemborosan air. Instansi irigasi akan dapat meningkatkan efisiensi pengelolaan air dan memberikan layanan yang lebih baik kepada para petani.

F. Sistematika Penulisan

Adapun sistem penulisan pada proposal ini sebagai berikut :

1. Bab I : Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, dan sistematika penulisan.

2. Bab II : Tinjauan Pustaka

Bab ini berisi tentang kajian teori yang memuat refensi penunjang yang menjelaskan tentang definisi dan fungsi dari perangkat-perangkat yang digunakan dalam pembuatan tugas akhir, bab ini juga memuat kajian penelitian

terdahulu yang beresensikan poin-poin kajian sebelumnya sebagai referensi dan pembanding dalam pembuatan tugas akhir ini.

3. Bab III : Metode Penelitian

Bab ini memuat poin-poin berupa lokasi dan waktu, jenis penelitian, alat dan bahan, rancangan sistem, dan metode pengumpulan data.

4. Bab IV : Hasil dan Pembahasan

Bab ini menjelaskan mengenai analisis sistem, implementasi perangkat, rancangan aplikasi android, dan mengenai metode pengujian dan teknik pengujian.

4. Bab V : Penutup

Bab ini menjelaskan mengenai kesimpulan dan saran

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. Internet of Things

Internet of Things didasarkan pada perangkat yang menyediakan aktivitas control, penginderaan, aktuasi, dan pemantauan. Perangkat IoT dapat berkomunikasi dengan perangkat dan aplikasi lain yang terhubung, atau mengumpulkan dan memproses data secara lokal, mengirimkannya ke server pusat di aplikasi berbasis cloud, atau melakukan beberapa tugas lokal serta tugas lain dalam infrastruktur IoT, tergantung pada batasan temporal dan spasial seperti memori, kemampuan pemrosesan, latensi komunikasi, kecepatan, dan tenggat waktu. Perangkat IoT sering kali dilengkapi dengan berbagai alat untuk berkomunikasi dengan perangkat lain, baik melalui kabel maupun nirkabel. Alat ini meliputi I/O untuk sensor, konektivitas internet, penyimpanan memori, serta audio/video. Berbagai jenis perangkat IoT, termasuk sensor, smartwatch, lampu LED, perangkat mobile, dan mesin industri, hampir selalu menghasilkan data dalam berbagai bentuk. Ketika data ini diproses oleh sistem, informasi yang berguna dapat dihasilkan untuk membantu pengguna dalam melakukan interaksi, baik secara lokal maupun jarak jauh. Misalnya, data dari sensor kelembaban tanah yang dipantau dapat digunakan untuk menentukan jadwal penyiraman yang optimal (Ray, 2018).

2. Irigasi

Sistem irigasi adalah sebuah jaringan air dengan tujuan untuk menyediakan serta mengatur aliran air agar terus dapat tersedia untuk pertanian. Pemberian air irigasi yang efisien dapat meningkatkan produktivitas dalam pertanian.

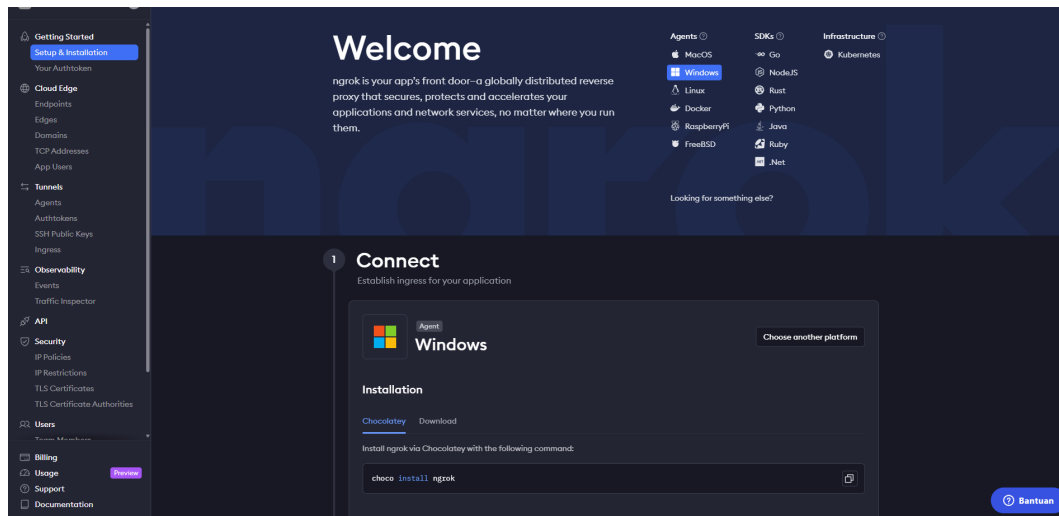
3. Sistem Monitoring

Monitoring merupakan kegiatan rutin pengumpulan data untuk mendeteksi dan memperbaiki kesalahan dalam suatu program atau objek. Aktivitas ini menitikberatkan pada proses serta hasilnya. Monitoring menyediakan laporan mengenai status dan perkembangan perubahan yang dilakukan secara berkala (Hutagaol et al., 2022).

Sistem monitoring adalah sebuah sistem yang menjalankan proses pengawasan secara berkelanjutan. Sistem ini diperlukan untuk memantau kondisi objek yang diamati agar dapat memperoleh informasi tepat waktu. Monitoring sistem juga dapat digunakan untuk memantau ketinggian air dan banjir di berbagai lokasi serta menyajikan data yang akurat dengan cepat (Hasiholan et al., 2018).

4. Ngrok

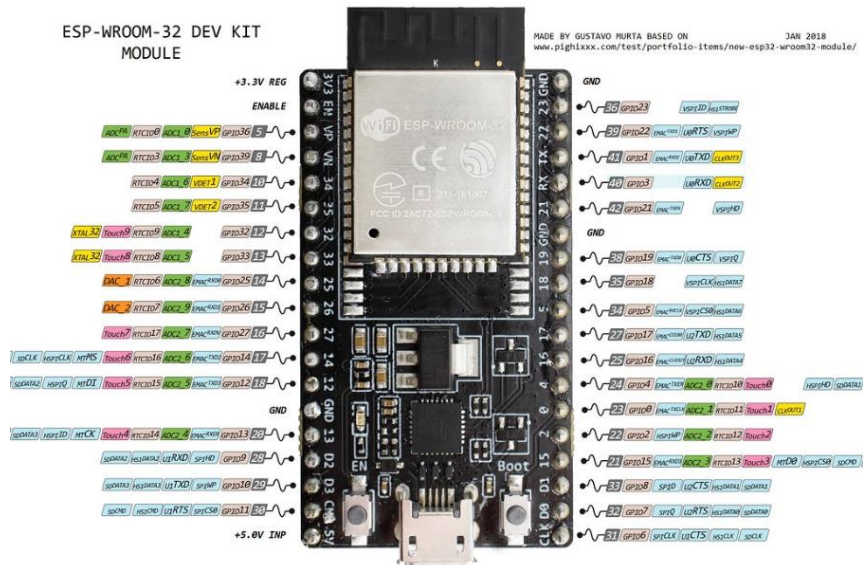
Penggunaan local hosting atau server yang sering digunakan saat ini adalah XAMPP dan Ngrok. Ngrok adalah aplikasi gratis yang memungkinkan kita untuk membuat tunnel dari URL publik ke aplikasi yang berjalan secara lokal (Bariyah & Imania, 2022). Dengan Ngrok, kita dapat mempublikasikan server lokal ke internet. Selain itu, Ngrok menyediakan antarmuka pengguna yang memungkinkan pemantauan semua lalu lintas yang melewati server tersebut dengan efektif (Parlika et al., 2021).



Gambar 2. 1 Dashboard NGROK

5. Mikrokontroler (Node MCU)

NodeMCU adalah sebuah firmware open-source dan kit pengembangan yang memudahkan pembuatan prototipe untuk produk IoT (Internet of Things) menggunakan beberapa baris skrip Lua. Sebagai platform open-source untuk IoT, NodeMCU memanfaatkan Lua sebagai bahasa pemrogramannya. Platform ini berbasiskan pada proyek Elua dan dibangun di atas ESP8266 SDK 1.4. NodeMCU mengintegrasikan berbagai proyek open-source, seperti lua-cjson, dan mencakup firmware yang berjalan pada SoC Wi-Fi ESP8266 serta perangkat keras yang berbasis modul ESP-12. NodeMCU menawarkan spesifikasi seperti open-source, interaktif, mudah diprogram, biaya rendah, sederhana, pintar, dan terhubung dengan Wi-Fi (Hakim et al., 2018)



Gambar 2. 2 Node MCU ESP32

Tabel 2. 1 Spesifikasi Node MCU ESP32

Nama	Spesifikasi
Microcontroller	ESP32-D0WDQ6
Core	Dual-core Tensilica LX6 microprocessor
Clock Frequency	Hingga 240 MHz
RAM	520 KB SRAM
ROM	448 KB ROM
Wi-Fi	802.11 b/g/n (hingga 150 Mbps)
Bluetooth	Bluetooth v4.2 BR/EDR dan BLE
GPIO	34 GPIO pins
ADC	18 channels of 12-bit SAR ADC
DAC	2 channels of 8-bit DAC
Touch Sensors	10 capacitive touch sensing GPIOs
Flash Memory	4 MB of SPI Flash
Operating Voltage	2.3V to 3.6V
Power Supply Options	USB, 5V via VIN pin, or 3.3V via 3V3 pin

Nama	Spesifikasi
<i>Dimensions</i>	<i>Sekitar 51mm x 25.4mm</i>
<i>USB-to-UART Bridge</i>	<i>Integrated CP2102 USB-to-UART Bridge</i>
<i>Programming Languages</i>	<i>C, C++, MicroPython, Lua, JavaScript (NodeMCU)</i>
<i>Supported IDEs</i>	<i>Arduino IDE, Espressif IoT Development Framework (ESP-IDF), PlatformIO</i>

6. Motor Servo

Motor servo adalah perangkat listrik yang digunakan dalam mesin-mesin industri untuk mendorong atau memutar objek dengan kontrol presisi tinggi, baik dalam hal sudut posisi, akselerasi, maupun kecepatan.



Gambar 2. 3 Motor Servo SG90

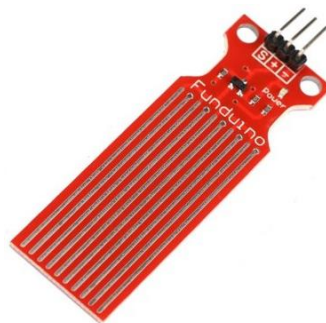
Tabel 2. 2 Spesifikasi Motor Servo SG90

<i>Parameter</i>	<i>Spesification</i>
<i>Operating Speed</i>	<i>0.1sec/60degree(4.8v)</i>
<i>Operating Voltage</i>	<i>4.8v</i>

<i>Parameter</i>	<i>Spesification</i>
<i>Dimension</i>	<i>22mm x 11.5mm x 22.5mm</i>
<i>Gear type</i>	<i>POM gear set</i>
<i>Power Supply</i>	<i>Through External Adapter</i>
<i>Servo wire length</i>	<i>25 cm</i>
<i>Weight</i>	<i>9g</i>

7. Sensor Water Level

Sensor Water Flow adalah perangkat yang digunakan untuk mengukur aliran air dalam pipa, termasuk kecepatan dan volume air yang mengalir. Alat ini sering diterapkan dalam sistem irigasi dan proses industri untuk memastikan pengaturan aliran air yang akurat dan efisien, serta untuk memantau penggunaan air secara real-time.



Gambar 2. 4 Sensor Water Level

Tabel 2.3 Spesifikasi Sensor Water Level

<i>Parameter</i>	<i>Spesifikasi</i>
<i>Operating Voltage</i>	<i>3V – 5V</i>
<i>Operating Current</i>	<i>Less than 20mA</i>

<i>Parameter</i>	<i>Spesifikasi</i>
<i>Sensor type</i>	<i>Analog</i>
<i>Detection area</i>	<i>40 x 60mm</i>
<i>Production process</i>	<i>FR4 double-sided HASL</i>
<i>Operating Temperature</i>	<i>10C – 30C</i>
<i>Product Dimensions</i>	<i>62 x 20 x 8mm</i>

8. Sensor Water Flow

Sensor Water Flow adalah alat yang berfungsi untuk mengukur aliran air dalam pipa, memantau kecepatan dan volume air yang mengalir. Perangkat ini sering digunakan dalam sistem irigasi dan aplikasi industri untuk memastikan pengaturan aliran air yang tepat.



Gambar 2. 5 Sensor Water Flow

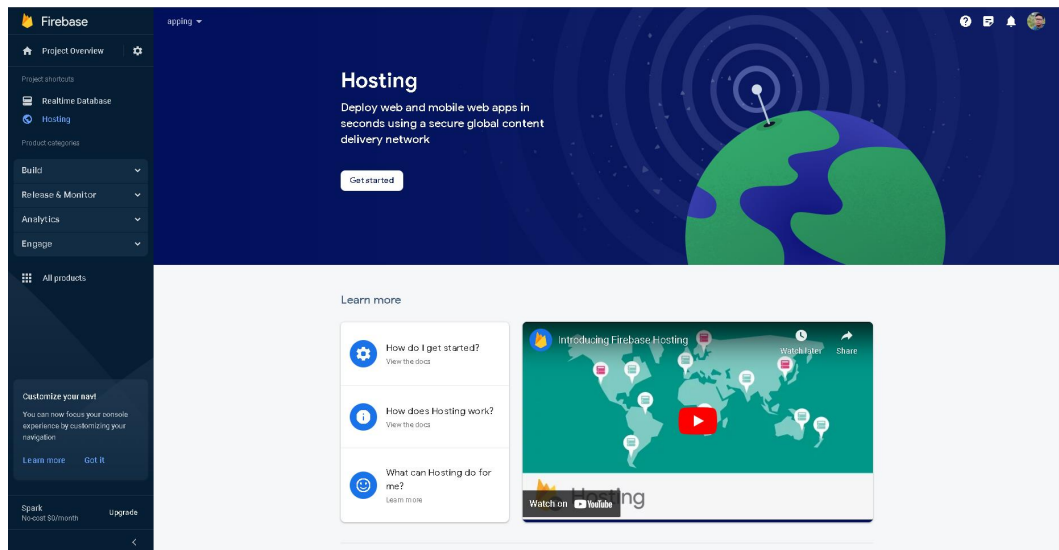
Tabel 2.4 Spesifikasi Sensor Water Flow

<i>Parameter</i>	<i>Spesifikasi</i>
<i>Tegangan</i>	<i>5 – 18V</i>
<i>Output Type</i>	<i>5V TTL</i>

<i>Parameter</i>	<i>Spesifikasi</i>
<i>Working Flow Rate</i>	<i>1 – 30 Liters/Minute</i>
<i>Working Temperature range</i>	<i>-25 to +80</i>
<i>Working Humidity Range</i>	<i>35%-80% RH</i>
<i>Accuracy</i>	<i>10%</i>
<i>Maximum Water Pressure</i>	<i>2.0 MPa</i>
<i>Durabilty</i>	<i>Min 300.000 cycles</i>
<i>Size</i>	<i>2.5" x 1.4" x 1.4"</i>

9. Firebase

Firebase adalah platform untuk aplikasi *real-time* yang memungkinkan perubahan data langsung diperbarui di semua perangkat yang terhubung, baik itu di website maupun mobile. *Firebase* menyediakan pustaka lengkap untuk berbagai platform web dan mobile, dan dapat diintegrasikan dengan berbagai framework seperti *Node.js*, *Java*, *JavaScript*, *Kotlin*, dan lainnya. *Application Programming Interface (API)* *Firebase* menyimpan dan menyinkronkan data dalam bentuk *JSON* (*JavaScript Object Notation*) di *cloud*, yang kemudian disinkronkan *secara real-time* (Sanad, 2019).



Gambar 2. 6 Halaman *Firebase*

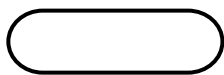
10. **Android**

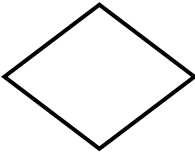
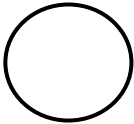
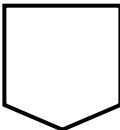
Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat mobile. Sebagai sistem operasi gratis dan open source, Android menawarkan platform terbuka bagi pengembang untuk membuat aplikasi yang dapat berjalan di perangkat Android. Keleluasaan ini memungkinkan Android untuk bersaing dengan iPhone, yang sudah lebih dulu meramaikan pasar (Masruri & Creativity, 2015).

11. **Flowchart**

Flowchart adalah diagram yang menggunakan simbol-simbol khusus untuk menggambarkan proses secara terurut dan mendetail, serta menunjukkan hubungan antar proses dalam sebuah program.

Tabel 2.5 Simbol *flowchart*

Simbol	Nama	Fungsi
	Terminator	Permulaan/akhir program

Simbol	Nama	Fungsi
	Garis alur	Arah aliran program
	Preparation	Proses Inisialisasi/pemberian harga awal
	Proses	Proses perhitungan/proses pengolahan data
	Input/Output Data	Proses Input/Output data, parameter dan informasi
	<i>Predefined Process</i> (Sub Program)	Pemutaran sub program/proses menjalankan sub program
	<i>Decision</i>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<i>On Page Connection</i>	Penghubung bagian- bagian flowchart yang berada pada satu halaman
	<i>Off Page Connection</i>	Penghubung bagian- bagian flowchart yang berada pada halaman berbeda

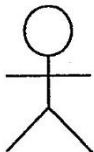
12. UML (*Unified Modelling Language*)

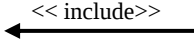
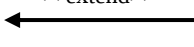
Unified Modeling Language (UML) adalah bahasa pemodelan standar yang memiliki sintaks dan semantik khusus. Menurut *Chonoles* (2003) sebagai bahasa, UML memerlukan aturan-aturan tertentu dalam pembuatan model, termasuk bagaimana elemen-elemen dalam model saling berhubungan, dan harus mematuhi standar yang berlaku. UML tidak hanya berupa diagram, tetapi juga menggambarkan konteks dari model tersebut (Muslihudin, 2016). Terdapat 14 jenis diagram dalam UML yang digunakan untuk pemodelan perangkat lunak, namun dalam praktiknya, empat diagram yang sering digunakan adalah: *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

a. *Use Case Diagram*

Use case diagram adalah salah satu jenis diagram dalam *Unified Modeling Language (UML)* yang menggambarkan interaksi antara sistem dan aktor. Diagram ini menunjukkan aktor-aktor yang terlibat dan aktivitas yang dilakukan terhadap sistem, serta bagaimana aktor berinteraksi dengan berbagai fitur atau fungsi dalam sistem (Voutama, 2022).

Tabel 2.6 Simbol pada use case diagram

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>

Simbol	Keterangan
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara actor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi actor untuk berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

b. Activity Diagram

Activity Diagram admin menggambarkan bagaimana aktivitas yang terjadi pada admin dan sistem. Admin mengakses website dan melakukan login, kemudian admin akan diarahkan oleh sistem ke menu utama website setelah itu admin memilih aktivitas apa yang akan dilakukan (Voutama, 2022).

Tabel 2.7 Simbol activity diagram

Simbol	Nama	Fungsi
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
	<i>Initial Node</i>	Bagaimana objek dibentuk atau diamati
	<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
	<i>Fork Node</i>	Proses Input/Output data, parameter dan informasi

c. Class Diagram

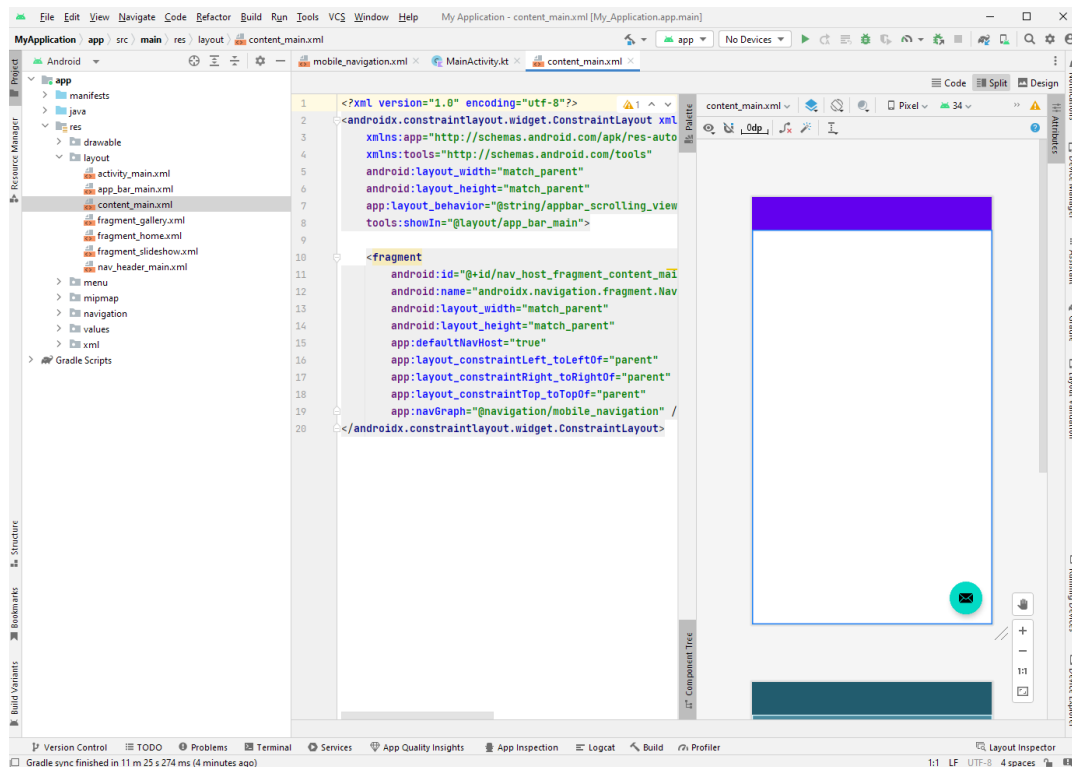
Class Diagram memberikan gambaran hubungan antara tabel-tabel yang ada dalam database. Masing-masing *class* memiliki attribute dan fungsi sesuai dengan proses yang terjadi (Voutama, 2022).

d. Sequence Diagram

Sequence Diagram merupakan penjelasan dari aturan *use case* yang dijelaskan berdasarkan urutan waktu kejadian atau secara sistematis.

13. *Android Studio*

Android Studio adalah *Integrated Development Environment (IDE)* untuk pengembangan aplikasi *Android* yang berbasis *IntelliJ IDEA*. Selain berfungsi sebagai editor kode dan alat pengembangan yang kuat, *Android Studio* menyediakan berbagai fitur tambahan untuk meningkatkan produktivitas dalam membuat aplikasi *Android*. Fitur-fitur tersebut termasuk sistem versi berbasis *Gradle* yang fleksibel, emulator cepat dan kaya fitur, lingkungan pengembangan terpadu untuk semua perangkat *Android*, *Instant Run* yang memungkinkan penerapan perubahan tanpa membuat APK baru, template kode dan integrasi *GitHub* untuk berbagi dan mengimpor kode contoh, alat pengujian dan kerangka kerja yang luas, alat *Lint* untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, serta dukungan untuk *C++* dan *NDK*. Selain itu, *Android Studio* juga mendukung *Google Cloud Platform*, mempermudah integrasi *Google Cloud Messaging* dan *App Engine* (Herlinah & Musliadi, 2019).

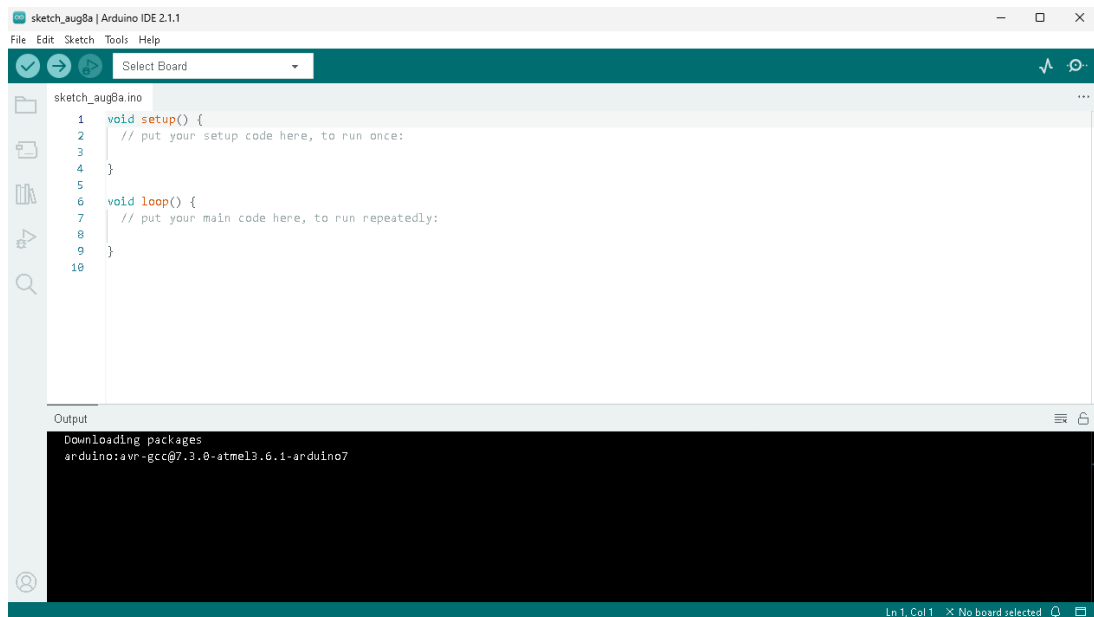


Gambar 2. 7 Tampilan dalam Android Studio

14. Arduino IDE

Arduino IDE (Integrated Development Environment) adalah perangkat lunak yang digunakan untuk memprogram board Arduino. Dengan kata lain, Arduino IDE berfungsi sebagai media untuk menulis dan mengelola kode program untuk Arduino.

IDE ini termasuk text editor untuk membuat, mengedit, dan memvalidasi kode, serta untuk meng-upload data ke board Arduino. Selain itu, Arduino IDE dilengkapi dengan library C/C++ yang dikenal sebagai Wiring, yang mempermudah operasi input dan output (Fauzan & Adiputri, 2020).



Gambar 2. 8 Arduino IDE

15. Kotlin

Kotlin adalah bahasa pemrograman statically typed yang dapat berjalan di *Java Virtual Machine* (JVM) atau dikompilasi menggunakan *Compiler LLVM* untuk menghasilkan kode *JavaScript*. Dikembangkan oleh tim programmer *JetBrains* yang berbasis di Rusia, Kotlin dirancang untuk berintegrasi dengan kode *Java*, serta memanfaatkan berbagai library dan *framework Java* yang ada. (Aljundi & Akbar, 2018).

16. C++

C++ adalah bahasa pemrograman yang ditemukan pada tahun 1970-an oleh Bjarne Stroustrup di Bell Labs. Bahasa ini sangat populer dalam pengembangan perangkat lunak, khususnya untuk aplikasi desktop dan game. C++ merupakan pengembangan dari bahasa C, dengan fitur-fitur tambahan yang lebih kompleks yang dapat meningkatkan produktivitas pemrogram dibandingkan dengan bahasa C.

17. Evapotranspirasi

Evapotranspirasi adalah proses yang menggabungkan evaporasi dan transpirasi dalam siklus hidrologi. Evapotranspirasi potensial mengacu pada laju penguapan yang terjadi dengan asumsi bahwa ketersediaan air dan kelembapan tanah tetap mencukupi sepanjang waktu (Wilnaldo et al., 2020).

B. Kajian Penelitian Terdahulu

Penelitian terdahulu bertujuan untuk mendapatkan bahan referensi serta bahan perbandingan dan acuan. Maka peneliti akan mencantumkan beberapa hasil penelitian terdahulu sebagai berikut :

Penelitian terdahulu dilakukan oleh Muh Amin (2021) dengan judul “Kontrol Jarak Jauh Sistem Irigasi Sawah Berbasis Internet Of Things”. Pada penelitian ini Hasil penelitian jika tinggi air pembagian < 1 cm maka servo akan berputar 15° , dan jika tinggi air pembagian $> 4-10$ cm maka motor servo akan berputar 180° , kemudian jika tinggi air pada sawah mencapai 3,47 cm maka pintu akan tertutup otomatis, jika tinggi air pada sawah < 1 cm maka tombol pintu sawah akan di kontrol secara manual di aplikasi *web*. Persamaan pada penelitian ini dengan penelitian yang akan di bawahkan pada proposal ini adalah kesamaan penggunaan IoT dalam memonitoring serta menentukan tinggi air sehingga dapat membuka pintu air secara otomatis maupun secara manual melalui aplikasi, tentu terdapat juga perbedaan diantaranya yaitu adanya penambahan fitur menghitung debit air yang keluar pada pintu air dan memonitoring melalui Aplikasi Mobile atau berbasis Android.

BAB III

METODE PENELITIAN

A. Lokasi dan Waktu

Lokasi penelitian ini dilakukan di Dinas Pengairan Mannarang, Kec. Mattiro Bulu, Kabupaten Pinrang dan waktu pelaksanaan penelitian dilakukan selama 3 bulan pada bulan Agustus-September tahun 2023.

B. Jenis Penelitian

Pada penelitian ini menggunakan metode penelitian lapangan dan pengembangan dengan menguji rangkaian yang dibuat dengan mengacu pada referensi yang telah ada.

C. Alat dan Bahan

Dalam pelaksanaan perancangan alat pada penelitian yang akan dilakukan, dibutuhkan beberapa alat dan bahan . Adapun alat dan bahan yang dibutuhkan dapat dilihat pada tabel dibawah ini.

Tabel 3.1 Alat

No.	Alat	Spesifikasi
1	Laptop	Asus X550V Inter i7-7700HQ 2.8HZ (8 CPUs) up to 3,8Ghz
2	Ponsel Android	INFINIX HOT 40 PRO

Tabel 3.2 Bahan

No.	Bahan	Spesifikasi
1	Node MCU	ESP 32, ESP32-CAM
2	Bahasa Program	C++ & Kotlin
3	Sensor	Water Flow YF-S201, Water Level
No.	Alat	Spesifikasi
5	Kabel	Type C , Kabel <i>Jumper Male, Female</i> & Micro USB
6	Motor Servo	Servo SG90
7	Perangkat Tambahan	Air, Pipa, Selang, <i>Breadboard</i> & <i>WiFi Router</i>

D. Metode Pengumpulan Data

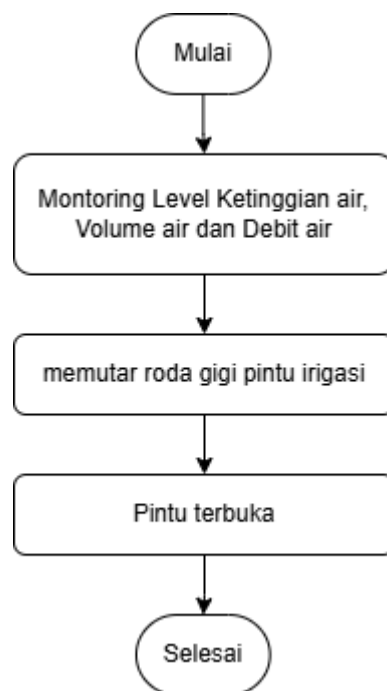
Pengumpulan data yang dilakukan pada penelitian ini terdiri dari beberapa tahapan. Tahapan pertama dilakukan studi literatur dengan mengumpulkan data serta informasi menggunakan bahan perpustakaan serta referensi sumber-sumber menjadi landasan teori yang mengenai perancangan sistem monitoring irigasi otomatis berbasis android .Tahapan kedua yaitu Desain dan perancangan sistem dilakukan dengan menggambarkan rencana serta membuat sketsa serta pengaturan beberapa bagian terpisah kedalam satu kesatuan yang utuh yang saling berkesinambungan.

BAB IV

HASIL DAN PEMBAHASAN

Pada Pengembangan Sistem Monitoring Irigasi Otomatis berbasis Android ini perlu diperhatikan dengan baik pada saat pembuatan dan perancangan setiap komponen untuk memastikan integrasi yang optimal dan kinerja yang maksimal. Sistem ini mencakup beberapa komponen utama yang saling terintegrasi untuk mencapai kinerja optimal.

A. Rancangan Sistem Berjalan

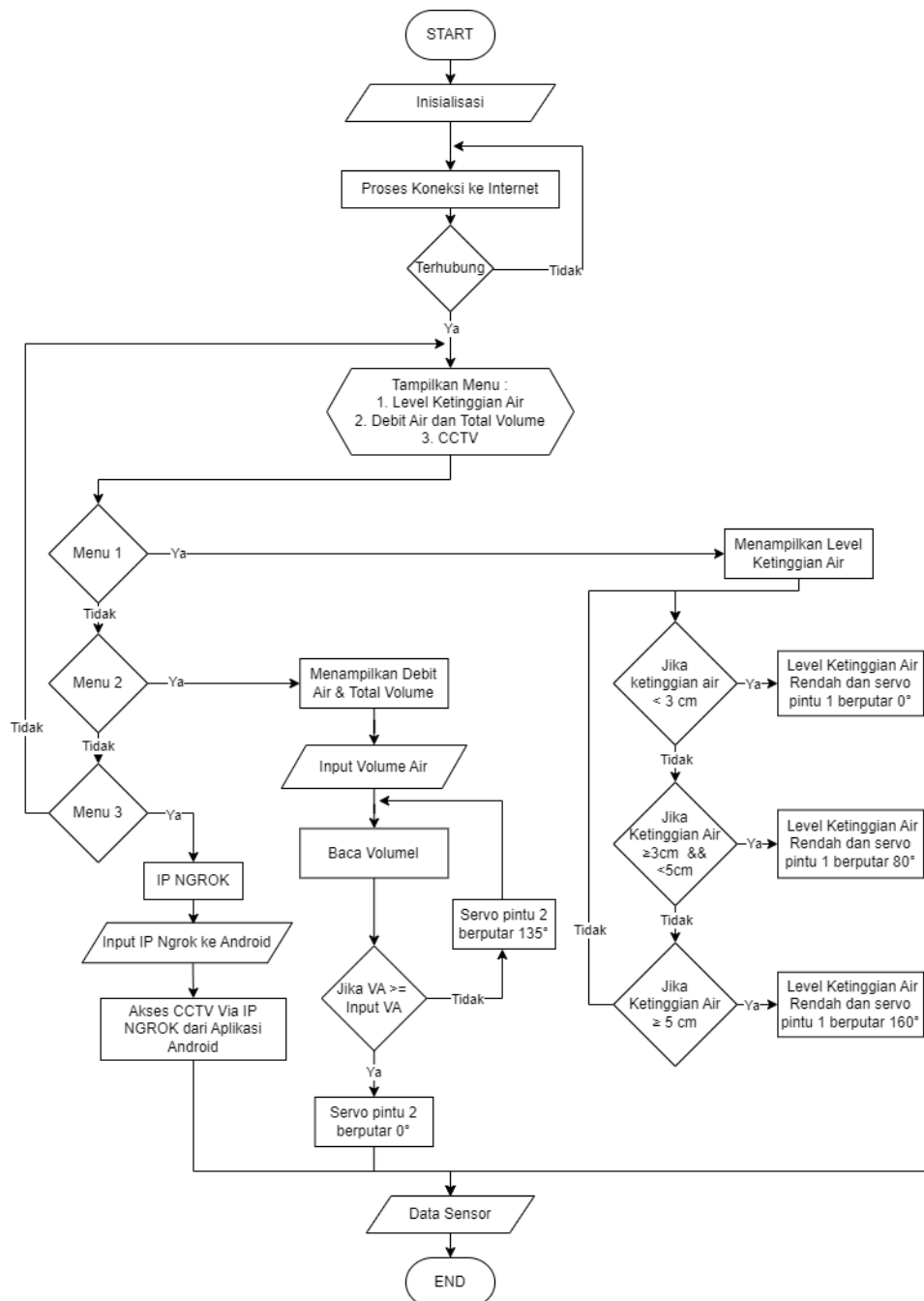


Gambar 4. 1 Rancangan Sistem Berjalan

Pada gambar di atas menjelaskan bagaimana memonitoring secara dan mengontrol pintu secara manual maupun otomatis.

B. Rancangan Sistem yang Diusulkan

Sistem dimulai dengan mengoperasikan komponen alat yang terintegrasi pada prototype sistem monitoring berbasis *android* dengan menerapkan *Internet of Things* dan menghubungkan semua perangkat.



Gambar 4. 2 Diagram Alir Flowchart

Pada diagram alir atau *flowchart* diatas diawali dengan start atau mulai seperti syarat pembuatan *flowchat* pada umumnya. Kemudian melakukan inisialisasi atau login ke aplikasi. Setelah inisialisasi selesai, sistem mencoba melakukan konektivitas ke Aplikasi. Jika berhasil, sistem akan menampilkan tiga menu utama yaitu :

- a. Level Ketinggian Air
- b. Debit Air dan Total Volume
- c. CCTV

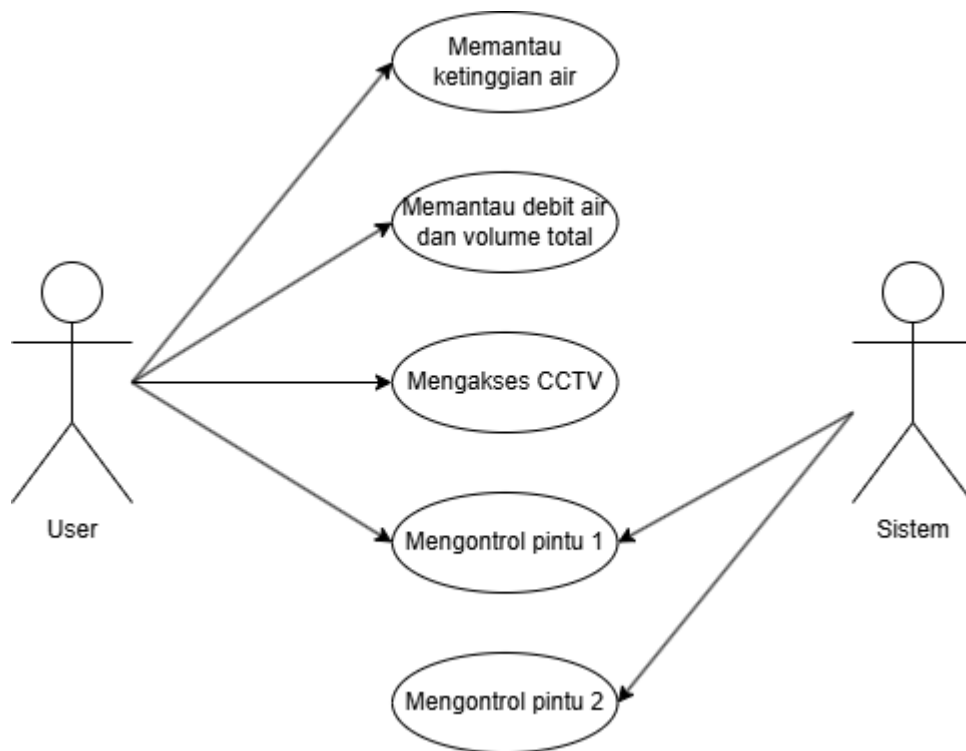
Selanjutnya jika memilih menu Level Ketinggian Air akan menampilkan data *Level* Ketinggian air dan melakukan perintah sesuai dengan level ketinggian. Jika level air rendah $< 3\text{cm}$, pintu 1 akan tertutup (0°), jika level air sedang $\geq 3\text{cm}$ & $< 5\text{cm}$, pintu 1 akan terbuka sebagian (80°) dan jika level air tinggi $\geq 5\text{ cm}$ maka pintu 1 akan terbuka penuh (160°).

Jika memilih menu Debit Air dan Total Volume, maka sistem akan menampilkan informasi debit air dan volume total. Kemudian menginput volume air yang diinginkan, dan sistem akan membandingkannya dengan volume air yang ada. Jika volume air yang lebih besar atau sama dengan volume yang diinginkan maka pintu 2 akan tertutup (0°) dan jika tidak, maka pintu 2 akan terbuka (135°).

Jika memilih menu CCTV , terlebih dahulu menginput IP Ngrok ke aplikasi android agar dapat mengakses CCTV melalui IP Ngrok. Setelah semua proses selesai, sistem akan menyimpan data dari sensor dan mengakhiri proses.

Pemodelan dalam pengembangan sistem informasi menggunakan UML yang merupakan sebuah metode pemodelan visual untuk merancang sistem berbasis objek. Salah satu jenis diagram UML yang digunakan dalam pengembangan sistem informasi adalah *Use Case*. Berikut adalah gambaran *use case* :

1. *Use Case*

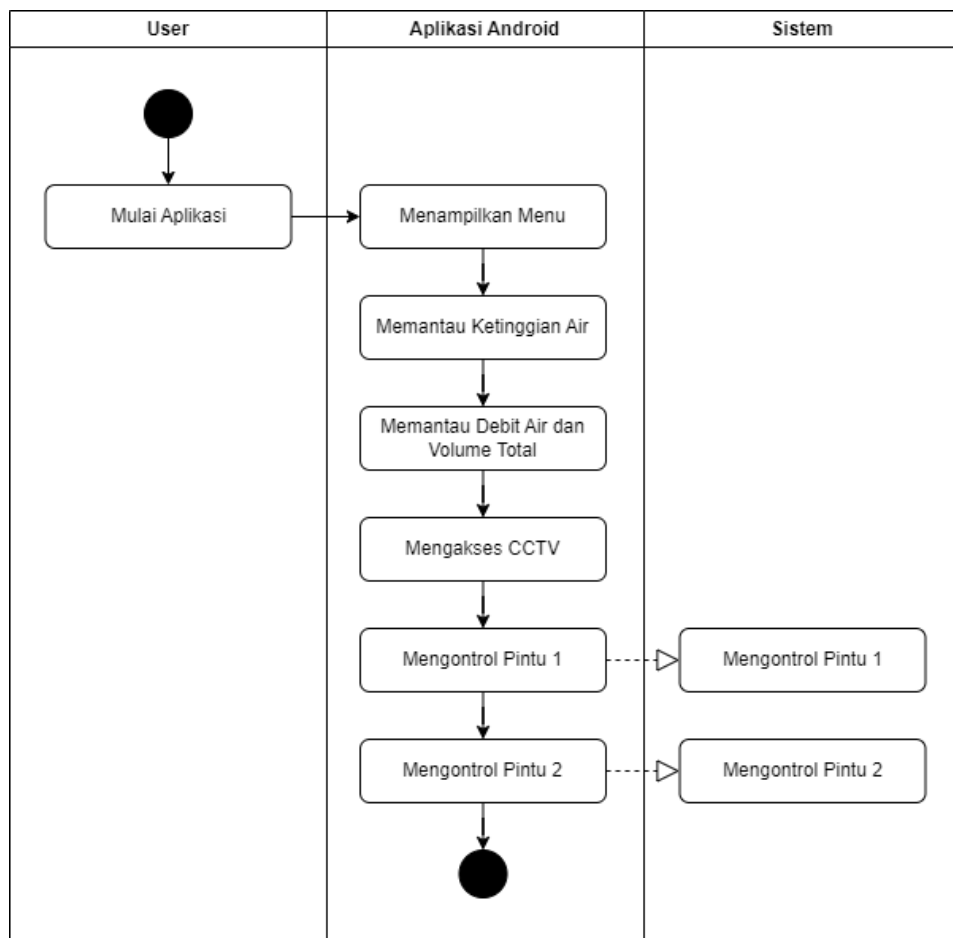


Gambar 4. 3 *Use Case Diagram*

Pada gambar 4.3 adalah sebuah perancangan sistem yang dibuat menggunakan diagram *use case*. *User* berinteraksi melalui aplikasi android yang memungkinkan untuk memantau ketinggian air, debit air dan volume total secara *real time* serta dapat mengakses cctv. Sistem bertanggung jawab untuk memproses data sensor dan mengontrol pintu air. *User* dapat mengontrol pintu 1 dan pintu 2 secara otomatis berdasarkan ketinggian air

dan volume air serta dapat mengontrol pintu 1 secara manual melalui aplikasi *android*.

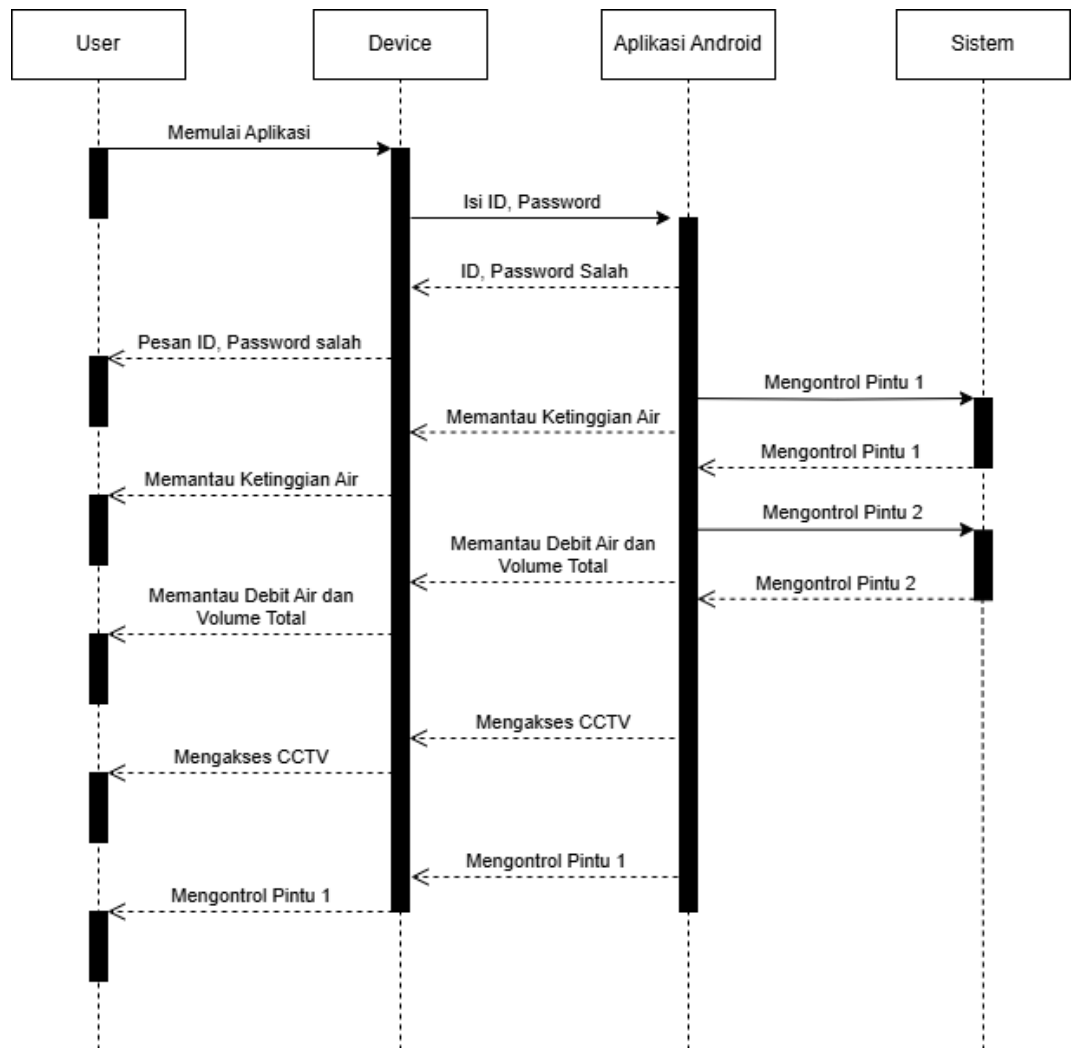
2. Activity Diagram



Gambar 4. 4 Activity Diagram

Pada gambar 4.4 *Activity Diagram* menjelaskan *User* berinteraksi melalui aplikasi *android* untuk memantau ketinggian air, debit air dan volume total air secara *real time* serta dapat mengakses cctv. Selain itu juga *user* dapat mengontrol pintu 1 dan 2 melalui aplikasi secara otomatis atau secara manual sesuai kebutuhan.

3. Sequence Diagram



Gambar 4. 5 *Sequence Diagram*

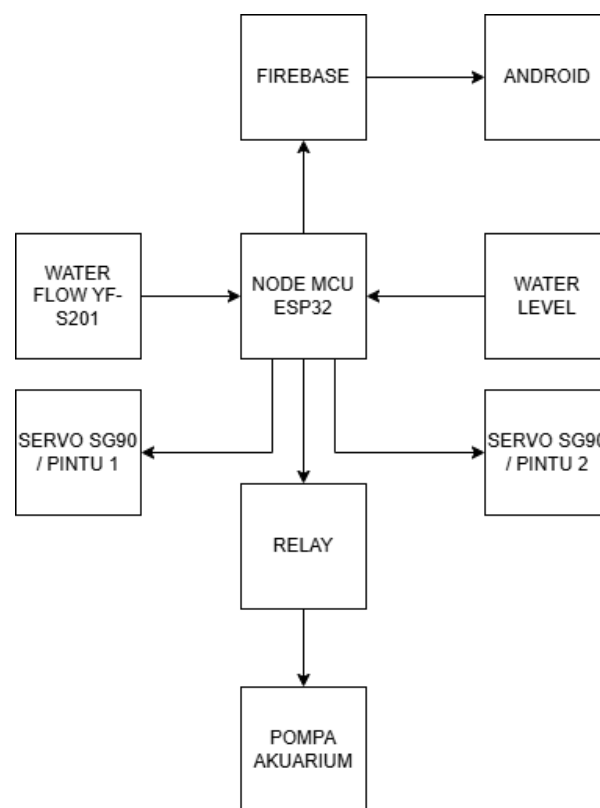
Pada gambar 4.5 *Sequence Diagram* berguna untuk memvisualisasikan dan menganalisis aliran kontrol dan komunikasi dalam sebuah sistem, serta untuk memastikan bahwa semua elemen berinteraksi dengan cara yang benar dan efisien.

C. Implementasi Perangkat Keras (*Hardware*)

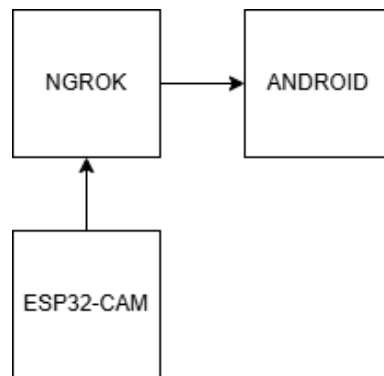
Berdasarkan pada penjelasan diatas terdapat beberapa komponen perangkat keras (*Hardware*) sebagai berikut :

1. Node MCU ESP32
2. Sensor Water Level
3. Sensor Water Flow
4. Motor Servo
5. Relay

Pengimpelemntasi komponen diatas akan dilakukan sesuai dengan rancangan sistem, yang berupa alat jadi dan siap untuk digunakan sesuai kebutuhan.



Gambar 4. 6 Block Diagram Sistem ESP32



Gambar 4. 7 Block Diagram Sistem ESP32-CAM

Adapun uraian beberapa komponen perangkat keras (*hardware*) :

1. Sensor Water Level

Sensor Water Level merupakan sensor yang digunakan untuk mendeteksi ketinggian air. Sensor ini memiliki jejak konduktif yang terbuka pada PCB yang menghasilkan resistansi yang berbeda-beda berdasarkan Tingkat air yang menutupi sensor. Adapun pin yang digunakan untuk sensor water level sebagai berikut :

Tabel 4. 1 Penggunaan Pin Sensor Water Level (Pintu 1 Irigasi)

No.	ESP32	Water Level Sensor
1.	<i>Output</i>	Pin 33
2.	<i>VCC</i>	<i>VCC 5V</i>
3.	<i>GND</i>	<i>GND</i>

2. Motor Servo

Motor servo adalah perangkat yang dirancang dengan sistem kontrol umpan balik *loop* tertutup sehingga posisinya dapat diatur sesuai dengan poros derajatnya. Adapun pin yang digunakan untuk Motor servo sebagai berikut :

Tabel 4. 2 Penggunaan Pin Motor Servo (Pintu 1 Irigasi)

No.	ESP32	Motor Servo
1.	<i>Output</i>	Pin 27
2.	<i>VCC</i>	<i>VCC 5V</i>
3.	<i>GND</i>	<i>GND</i>

Tabel 4. 3 Penggunaan Pin Motor Servo (Pintu 2 Irigasi)

No.	ESP32	Motor Servo
1.	<i>Output</i>	Pin 14
2.	<i>VCC</i>	<i>VCC 5V</i>
3.	<i>GND</i>	<i>GND</i>

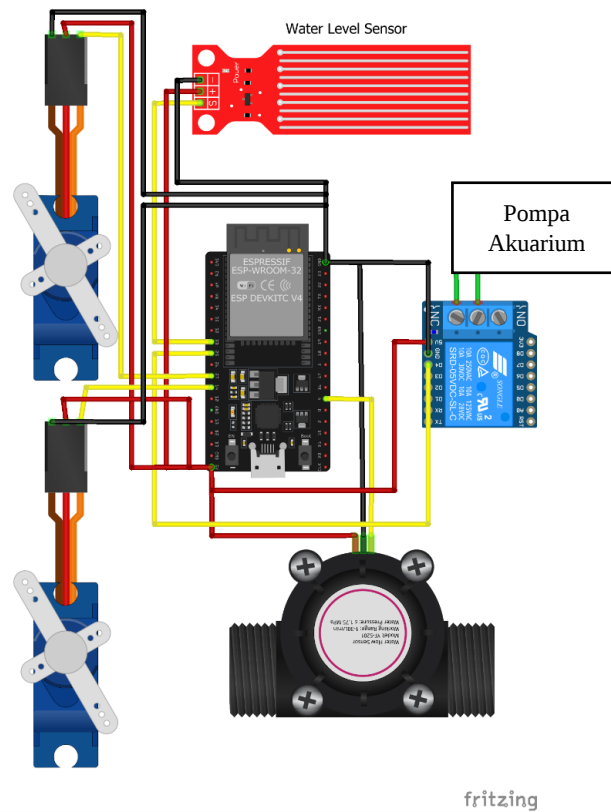
3. Relay

Relay adalah komponen yang berfungsi sebagai saklar yang di operasikan secara elektronik. Relay sering digunakan untuk mengontrol suatu perangkat yang membutuhkan arus atau tegangan yang lebih tinggi dari mikrokontroler.

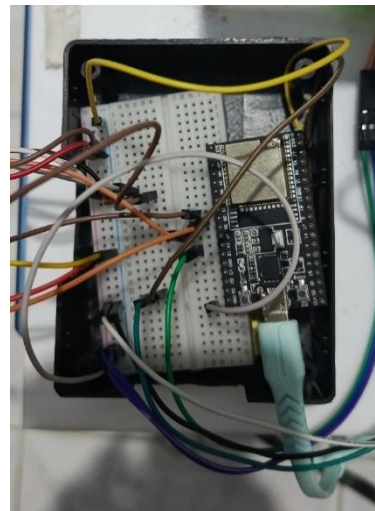
Tabel 4. 4 Penggunaan Pin Relay

No.	ESP32	Relay
4.	<i>Output</i>	Pin 25
5.	<i>VCC</i>	<i>VCC 5V</i>
6.	<i>GND</i>	<i>GND</i>

Adapun rangkaian perancangan sistem monitoring irigasi otomatis berbasis android pada gambar berikut :



Gambar 4. 8 Rangkain Sistem



Gambar 4. 9 Rancangan Sistem

Berdasarkan pada gambar diatas yaitu rancangan sistem, ada beberapa komponen yang digunakan dan saling terhubung dengan kabel jumper.

D. Rancangan Perangkat Lunak (*Software*)

Pada perancangan perangkat lunak merupakan pembuatan program yang akan di upload pada mikrokontroler yang merupakan sebuah perintah untuk mengolah data pada aplikasi. Bahasa pemrograman yang digunakan berupa C/C++ di arduino IDE dan untuk membuat aplikasi android menggunakan bahasa pemrograman kotlin di android studio.

```
#include <Arduino.h>
#include <WiFi.h>
#include <ESP32Servo.h>
#include <Firebase_ESP_Client.h>

// Insert your network credentials
#define WIFI_SSID "NierJur"
#define WIFI_PASSWORD "kucingberanak"

// Insert Firebase project API Key
#define API_KEY "AIzaSyB46dwY1BrR180WEVANvhIqiShNUVdNWmY"

// Insert RTDB URL
#define DATABASE_URL "https://kontrol-irigasi-v1-default-rtdb.asia-southeast1.firebaseio.com"

// PIN
#define PIN_PINTU1 27 // Pin servo pintu 1
#define PIN_PINTU2 14 // Pin servo pintu 2
#define relayPin 25 // Relay
const int sensorPin = 4; // Define pin untuk sensor YF-S201 atau sensor aliran air
const int pinSensor = 33; // Pin sensor water level

// Variabel global untuk sensor flow
volatile int pulseCount = 0;
float debit = 0.0; // Debit awal
const float calibrationFactor = 0.00316; // Faktor kalibrasi (L/pulse)
float totalVolume = 0.0; // Total volume air yang telah mengalir
float totalVolumeHarian = 0.0; // Total volume per hari
unsigned long lastPulseTime = 0; // Waktu dari detik terakhir terdeteksi pulsa
unsigned long resetTime = 0; // Waktu untuk reset totalVolume
const unsigned long resetInterval = 5 * 60 * 1000; // 5 menit dalam milidetik

// Variabel global untuk sensor water level
int tinggiAir = 0;

// Define Firebase Data object
FirebaseData fbdo, fbdo_s1, fbdo_s2, fbdo_s3;
FirebaseAuth auth;
```

```

FirebaseConfig config;

// Define servo
Servo pintu1;
Servo pintu2;

// Define manual override flag
bool manual = false;

int status1;

// Variables for volume edit
float volumeEdit = 0; // Total volume air perubahan data
float initialTotalVolume = 0;
float initialVolumeEdit = 0;

// Timer for sending data
unsigned long sendDataPrevMillis = 0;
bool signupOK = false;

void IRAM_ATTR pulseCounter() {
    pulseCount++;
}

void openGate() {
    Serial.println("Opening gate...");
    pintu1.write(20);
}

void closeGate() {
    Serial.println("Closing gate...");
    pintu1.write(180); // Rotate servo to 180 degrees
}

void setup() {
    Serial.begin(115200);

    // Initialize Wi-Fi
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Connecting to Wi-Fi");
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(300);
    }
    Serial.println();
    Serial.print("Connected with IP: ");
    Serial.println(WiFi.localIP());
    Serial.println();

    // Initialize Firebase
    config.api_key = API_KEY;
    config.database_url = DATABASE_URL;

    if (Firebase.signUp(&config, &auth, "", "")) {
        Serial.println("Firebase signup successful");
        signupOK = true;
    } else {
        Serial.printf("Firebase signup failed: %s\n", config.signer.signupError.message.c_str());
    }

    Firebase.begin(&config, &auth);
    Firebase.reconnectWiFi(true);

    pintu1.attach(PIN_PINTU1);
    pintu1.write(180); // Titik awal servo
    pintu2.attach(PIN_PINTU2);
    pintu2.write(0);

    // Set pin sensor sebagai input dan aktifkan internal pull-up resistor
    pinMode(sensorPin, INPUT_PULLUP);
    pinMode(pinSensor, INPUT);
    pinMode(relayPin, OUTPUT);

    // Reset variabel
    pulseCount = 0;
    lastPulseTime = millis();

```

```

lastPulseTime = millis();
resetTime = millis();

// Attach interrupt untuk sensor YF-S201
attachInterrupt(digitalPinToInterrupt(sensorPin), pulseCounter, FALLING);

// Stream volume air dan manualOverride
if (!Firebase.RTDB.beginStream(&fbdo_s1, "SensorWaterFlow/volumeEdit")) {
    Serial.printf("Stream 1 begin error, %s\n\n", fbdo_s1.errorReason().c_str());
}

if (!Firebase.RTDB.beginStream(&fbdo_s2, "Control/manualOverride")) {
    Serial.printf("Stream manualOverride begin error, %s\n\n", fbdo_s2.errorReason().c_str());
}
if (!Firebase.RTDB.beginStream(&fbdo_s3, "Pintu1/Status")) {
    Serial.printf("Stream Pintu 1 begin error, %s\n\n", fbdo_s3.errorReason().c_str());
}

d loop() {
    unsigned long currentTime = millis();

    // Stream manualOverride
    if (Firebase.RTDB.readStream(&fbdo_s2)) {
        if (fbdo_s2.streamAvailable()) {
            if (fbdo_s2.dataPath() == "/" ) {
                bool newmanual = fbdo_s2.boolData();
                Serial.print("Status Manual (boolean): ");
                Serial.println(newmanual);
                manual = newmanual;
            }
        }
    } else {
        Serial.printf("Stream read error, %s\n", fbdo_s2.errorReason().c_str());
    }

    // Stream manualOverride
    if (Firebase.RTDB.readStream(&fbdo_s3)) {
        if (fbdo_s3.streamAvailable()) {
            if (fbdo_s3.dataPath() == "/" ) {
                int newstatus = fbdo_s3.boolData();
                Serial.print("Status Pintu : ");
                Serial.println(newstatus);
                status1 = newstatus;
            }
        }
    } else {
        Serial.printf("Stream read error, %s\n", fbdo_s3.errorReason().c_str());
    }

    // Sensor Water Level
    int nilaiAir = analogRead(pinSensor);
    Serial.print("Sensor Air Value = ");
    Serial.println(nilaiAir);

    // Kontrol servo pintu berdasarkan status manual dan level air

    if (manual) {
        Serial.println("Manual Override is ON");
        if (status1 == 1) {
            openGate();
        } else if (status1 == 0) {
            closeGate();
        }
    } else {
        if (nilaiAir >= 1700) {
            pintu1.write(20); // Fully open position
            status1 = 1;
            Serial.println("Pintu Terbuka");
        } else if (nilaiAir > 1600 && nilaiAir < 1700) {
            pintu1.write(80); // Half open position
            status1 = 2;
            Serial.println("Pintu Terbuka Setengah");
        } else {
            pintu1.write(180); // Closed position
        }
    }
}

```

```

    } else {
        pintu1.write(180); // Closed position
        status1 = 0;
        Serial.println("Pintu Tertutup");
    }

    // Update status to Firebase
    if (Firebase.RTDB.setInt(&fbdo, "/Pintu1/Status", status1)) {
        Serial.println("Status Pintu1 berhasil diperbarui di Firebase");
    } else {
        Serial.printf("Gagal memperbarui status Pintu1: %s\n", fbdo.errorReason().c_str());
    }
}

if (nilaiAir >= 1700) {
    Serial.println("Relay OFF");
    digitalWrite(relayPin, LOW);
    tinggiAir = 1;
    Serial.println("Status Level Air Tinggi");
} else if (nilaiAir > 1600 && nilaiAir < 1700) {
    tinggiAir = 2;
    Serial.println("Status Level Air Sedang");
} else {
    Serial.println("Relay Nyala");
    digitalWrite(relayPin, HIGH);
    tinggiAir = 0;
    Serial.println("Status Level Air Rendah");
}

// Sensor Waterflow
if (millis() - lastPulseTime > 2000) {
    detachInterrupt(digitalPinToInterrupt(sensorPin));

    debit = calibrationFactor * pulseCount / (2000 / 1000.0); // Menggunakan calibrationFactor
    float volume = debit * (2000 / 1000.0); // Volume dalam interval waktu
    totalVolumeHarian += volume;
    totalVolume += volume;

    Serial.print("Debit: ");
    Serial.print(debit, 4);
    Serial.println(" L/s");

    Serial.print("Total Volume Harian: ");
    Serial.print(totalVolumeHarian, 4);
    Serial.println(" L");

    Serial.print("Total Volume: ");
    Serial.print(totalVolume, 4);
    Serial.println(" L");

    pulseCount = 0;
    lastPulseTime = millis();
    attachInterrupt(digitalPinToInterrupt(sensorPin), pulseCounter, FALLING);
}

// Reset totalVolumeHarian
if ((millis() - resetTime) > resetInterval) {
    totalVolumeHarian = initialTotalVolume;
    Serial.println("5 menit berlalu. Reset totalVolume ke nilai awal.");
    resetTime = millis(); // Set reset time baru
}

// Mengambil data volumeEdit dari Firebase
if (Firebase.RTDB.readStream(&fbdo_s1)) {
    if (fbdo_s1.streamAvailable()) {
        if (fbdo_s1.dataPath() == "/" ) {
            float newVolumeEdit = fbdo_s1.floatData();
            Serial.print("Volume Edit Value (float): ");
            Serial.println(newVolumeEdit);

            if (volumeEdit != newVolumeEdit) {
                volumeEdit = newVolumeEdit;
            }
        }
    }
}
}

```

```

}
Serial.print("Volume Edit: ");
Serial.println(volumeEdit);

// Kontrol servo pintu kedua
int status2;
if (totalVolume < volumeEdit) {
  pintu2.write(135); // Open position
  status2 = 1; // Pintu terbuka
  Serial.println("Pintu 2 terbuka.");
} else {
  pintu2.write(0); // Closed position
  status2 = 0; // Pintu tertutup
  Serial.println("Pintu 2 tertutup.");
  totalVolume = initialTotalVolume;
  volumeEdit = initialVolumeEdit;
  if (Firebase.RTDB.setFloat(&fbdo, "SensorWaterFlow/volumeEdit", volumeEdit)) {
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
  } else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
  }
  Serial.println("Total volume tercapai. Reset totalVolume ke nilai awal.");
}

// Mengirim data ke Firebase
if (Firebase.ready() && signupOK && (millis() - sendDataPrevMillis > 2000 || sendDataPrevMillis == 0)) {
  sendDataPrevMillis = millis();

  if (Firebase.RTDB.setInt(&fbdo, "Sensor/Value", nilaiAir)) {
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
  } else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
  }

  Serial.println("REASON: " + fbdo.errorReason());
}

if (Firebase.RTDB.setInt(&fbdo, "Sensor/TinggiAir", tinggiAir)) {
  Serial.println("PASSED");
  Serial.println("PATH: " + fbdo.dataPath());
  Serial.println("TYPE: " + fbdo.dataType());
} else {
  Serial.println("FAILED");
  Serial.println("REASON: " + fbdo.errorReason());
}

if (Firebase.RTDB.setInt(&fbdo, "Pintu1/Status", status1)) {
  Serial.println("PASSED");
  Serial.println("PATH: " + fbdo.dataPath());
  Serial.println("TYPE: " + fbdo.dataType());
} else {
  Serial.println("FAILED");
  Serial.println("REASON: " + fbdo.errorReason());
}

if (Firebase.RTDB.setInt(&fbdo, "Pintu2/Status", status2)) {
  Serial.println("PASSED");
  Serial.println("PATH: " + fbdo.dataPath());
  Serial.println("TYPE: " + fbdo.dataType());
} else {
  Serial.println("FAILED");
  Serial.println("REASON: " + fbdo.errorReason());
}

if (Firebase.RTDB.setFloat(&fbdo, "SensorWaterFlow/debit", debit)) {
  Serial.println("PASSED");
  Serial.println("PATH: " + fbdo.dataPath());
  Serial.println("TYPE: " + fbdo.dataType());
} else {
  Serial.println("FAILED");
  Serial.println("REASON: " + fbdo.errorReason());
}
}

```

```

        Serial.println("PASSED");
        Serial.println("PATH: " + fbdo.dataPath());
        Serial.println("TYPE: " + fbdo.dataType());
    } else {
        Serial.println("FAILED");
        Serial.println("REASON: " + fbdo.errorReason());
    }
}

if (Firebase.RTDB.setInt(&fbdo, "Pintu2/Status", status2)) {
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
} else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}

if (Firebase.RTDB.setFloat(&fbdo, "SensorWaterFlow/debit", debit)) {
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
} else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}

if (Firebase.RTDB.setFloat(&fbdo, "SensorWaterFlow/volume", totalVolumeHarian)) {
    Serial.println("PASSED");
    Serial.println("PATH: " + fbdo.dataPath());
    Serial.println("TYPE: " + fbdo.dataType());
} else {
    Serial.println("FAILED");
    Serial.println("REASON: " + fbdo.errorReason());
}
}

delay(100); // Mengurangi delay untuk loop yang lebih cepat
}

```

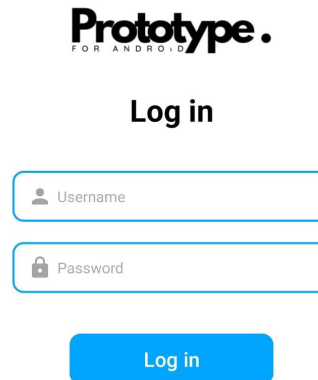
Gambar 4. 10 Rancangan Perangkat Lunak

Pada gambar di atas adalah sebuah sketch pada software arduino IDE yang berfungsi untuk memprogram komponen perangkat keras dengan menggunakan sintaks pemrograman bahasa C.

E. Rancangan Aplikasi Android

Pada rancangan sistem monitoring irigasi otomatis yang dituangkan ke dalam sebuah bentuk *prototype* dengan seluruh proses dapat di kontrol maupun dimonotoring melalui aplikasi *android*.

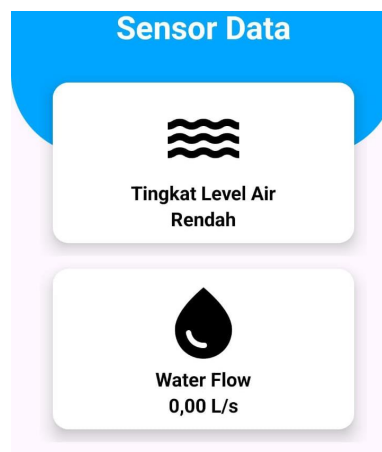
1. Tampilan Halaman Login



Gambar 4. 11 Tampilan Halaman Login

Pada halaman login ini terdiri dari logo, input username dan password yang memerlukan otentikasi pengguna sebelum mengakses aplikasi. Input username dan password menggunakan mode default yang berupa username admin dan password admin.

2. Tampilan Menu Sensor

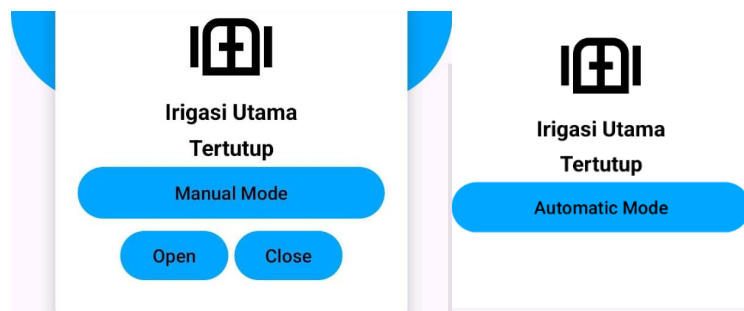


Gambar 4. 12 Tampilan Halaman Sensor Data

Pada tampilan menu ini menampilkan data *level* ketinggian air rendah, sedang, tinggi dan debit air secara realtime. Pada bagian

ketinggian air menampilkan status level air yang dikategorikan sebagai rendah, sedang, atau tinggi. Kategori ini didasarkan pada data yang diambil dari sensor dan dikirimkan ke Firebase, menunjukkan apakah ketinggian air berada dibawah batas minimal, dalam rentang normal atau melebihi batas maksimal. Selain itu, pada bagian Debit air menampilkan informasi mengenai aliran air saat ini berlangsung, yang juga diukur oleh sensor dan diupdate secara *real-time* ke *Firebase*. Nilai ini memberikan Gambaran mengenai seberapa cepat air mengalir melalui sistem irigasi, membantu dalam pengaturan dan pemantauan sistem secara efisien.

3. Tampilan menu Pintu 1 Irigasi



Gambar 4. 13 Tampilan Status Pintu 1 Irigasi

Pada menu Pintu 1 Irigasi ini memungkinkan untuk memantau dan mengendalikan status pintu irigasi dengan mudah. Status pintu akan ditampilkan Terbuka atau tertutup berdasarkan data yang diambil dari *firebase*. Pada menu ini memiliki mode otomatis dan manual. Dalam

mode otomatis, pintu 1 irigasi akan diatur secara otomatis berdasarkan data ketinggian air yang diterima dari sensor. Pintu akan terbuka sepenuhnya jika sensor mengirimkan data ketinggian air tinggi, terbuka sebagian jika ketinggian air sedang, dan tertutup jika ketinggian air rendah. Sementara itu, dalam mode manual, dua tombol akan muncul yang memungkinkan pengguna untuk mengatur posisi pintu secara langsung sesuai keinginan, terbuka atau tertutup dan memberikan kontrol penuh atas pengaturan pada pintu 1 irigasi.

4. Tampilan menu Pintu 2 Irigasi

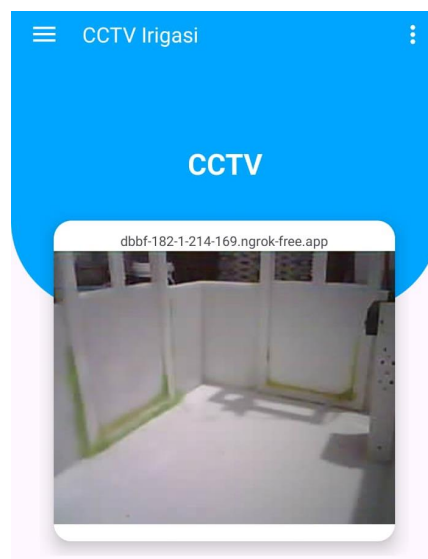


Gambar 4. 14 Tampilan Status Pintu 2 Irigasi

Pada menu Pintu 2 Irigasi ini menampilkan status pintu dan pengaturan volume air. Tampilan status pintu dua akan menampilkan status terbuka atau tertutup berdasarkan dari input volume air telah terpenuhi atau belum. Jika input volume air terpenuhi, pintu dua akan menampilkan status tertutup, sedangkan jika belum terpenuhi maka pintu akan menampilkan status terbuka. Selain itu, menu ini juga

menampilkan total volume air yang dihitung berdasarkan aliran air yang tercatat oleh sensor dan menampilkan volume air yang diinput dari aplikasi, yang menunjukkan jumlah volume air yang telah diatur. Pada menu ini juga menampilkan inputan volume yang diinginkan, yaitu volume air yang diinginkan untuk diatur dalam sistem irigasi.

5. Tampilan Menu CCTV



Gambar 4. 15 Tampilan Menu CCTV

Pada menu CCTV Irigasi dalam aplikasi ini dirancang untuk mempermudah pemantauan visual sistem irigasi dan menampilkan IP Address dari NGROK yang diperlukan untuk mengakses streaming video secara langsung. IP Address ini digunakan untuk menghubungkan dan menampilkan streaming video dari ESP32-CAM melalui Ngrok dengan mengakses dan memantau video live dari kamera ESP32-CAM

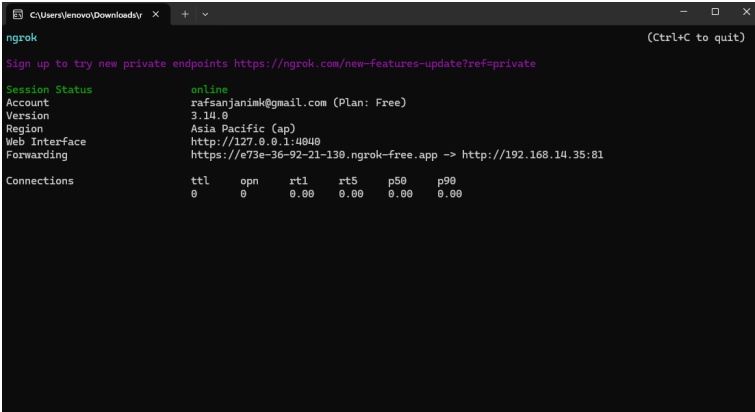
yang terpasang di *prototype* irigasi, memberikan visibilitas langsung terhadap kondisi dan aktivitas area irigasi.

6. Tampilan Menu Pengaturan



Gambar 4. 16 Tampilan Menu Pengaturan

Menu pengaturan dalam aplikasi ini memungkinkan untuk mengkonfigurasi dan mengelola ip address yang digunakan untuk mengakses streaming video dari kamera ESP32-CAM. Pada menu ini memungkinkan untuk menginput IP Address dari NGROK yang berfungsi untuk mengenkripsi dan mengonlinekan IP Address dari ESP32-CAM agar dapat dipantau secara online. Setelah menginput IP Address , data akan dikirim ke *Firebase* untuk disimpan dan diakses oleh aplikasi. Kemudian, IP Address yang baru akan dikembalikan ke menu CCTV Irigasi di aplikasi Android, agar memungkinkan melihat dan mengakses streaming video melalui Alamat yang sudah datur.



```

ngrok
Sign up to try new private endpoints https://ngrok.com/new-features-update?ref=private

Session Status      online
Account             rafsanjanmk@gmail.com (Plan: Free)
Version             3.14.0
Region              Asia Pacific (ap)
Web Interface        http://127.0.0.1:4040
Forwarding            https://e73e-36-92-21-130.ngrok-free.app -> http://192.168.14.35:81

Connections
  ttl   opn   rt1   rt5   p50   p99
   0     0    0.00  0.00  0.00  0.00

```

Gambar 4. 20 Tampilan Ngrok

F. Pembahasan

1. Metode Pengujian

Metode yang digunakan untuk menguji adalah metode eksekusi. Pada metode ini, pengujian dilakukan secara keseluruhan untuk menunjukkan fungsi perangkat keras dan perangkat lunak bekerja dengan baik dan sesuai spesifikasi persyaratan kinerja yang ditentukan.

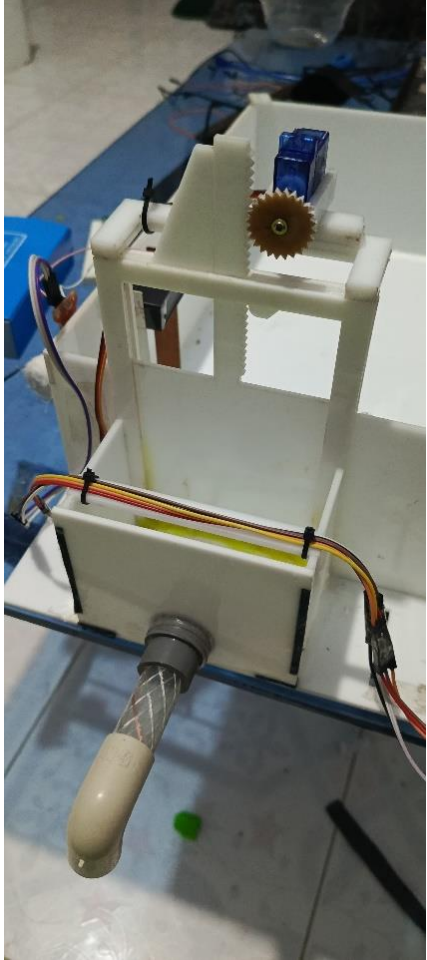
2. Pengujian *Black-Box*

Pengujian *black-box* adalah metode penujian perangkat lunak untuk mengevaluasi serangkaian kondisi input yang sepenuhnya menggunakan seluruh persyaratan fungsional suatu program maupun struktur control.

Pengujian *black-box* menguji beberapa aspek dasar suatu sistem dengan memperhatikan *logika internal* pada perangkat lunak.

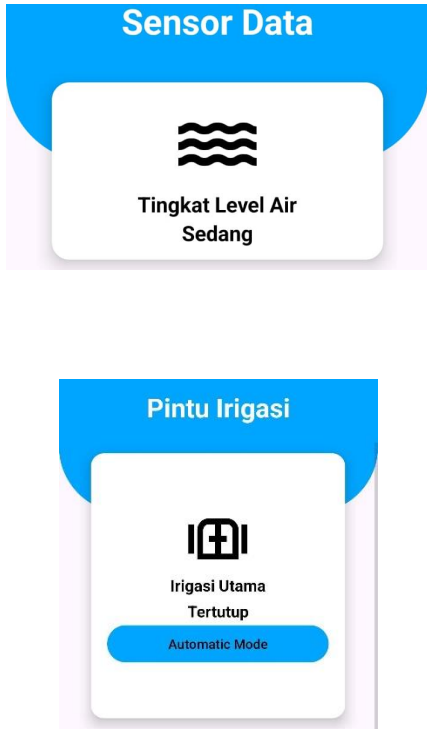
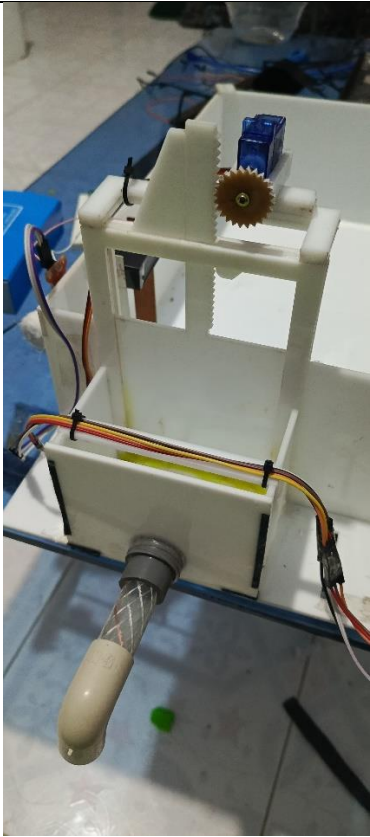
a. Pengujian tahap pertama

Tabel 4. 5 Pengujian tahap pertama

Uji coba	Hasil	Keterangan
Jika sensor mendeteksi ketinggian level air tinggi maka motor servo akan berputar 160° secara otomatis apabila dalam keadaan mode otomatis	✓	Berhasil, dikarenakan sensor mendeteksi ketinggian air dan servo berputar sesuai dengan yang diperintahkan
 The screenshot shows two panels. The top panel, titled 'Sensor Data', displays a water level icon and the text 'Tingkat Level Air Tinggi'. The bottom panel, titled 'Pintu Irigasi', displays a gate icon, the text 'Irigasi Utama Tertutup', and a blue button labeled 'Automatic Mode'.		 The photograph shows a white plastic frame housing a servo motor and a gear mechanism. A blue servo motor is mounted at the top, connected to a gear system. A yellow liquid is visible in a reservoir below the servo. A white tube with a grey cap is connected to the bottom of the frame. Wires are connected to the servo and other components.

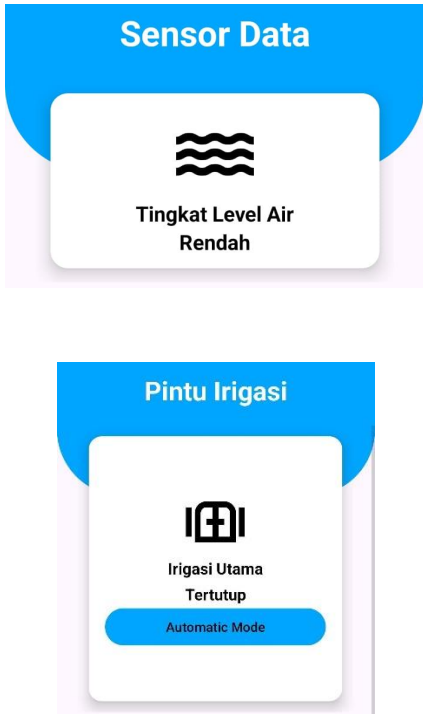
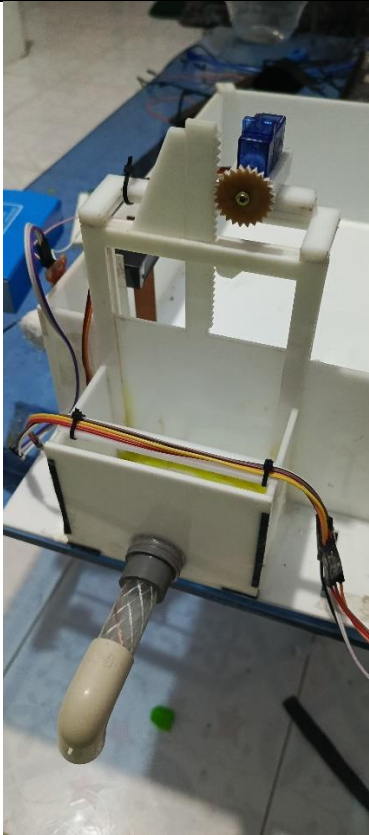
b. Pengujian tahap kedua

Tabel 4. 6 Pengujian tahap kedua

Uji coba	Hasil	Keterangan
Jika sensor mendeteksi ketinggian level air sedang maka motor servo akan berputar 80° secara otomatis apabila dalam keadaan mode otomatis	✓	Berhasil, dikarenakan sensor mendeteksi ketinggian air dan servo berputar sesuai dengan yang diperintahkan
 		

c. Pengujian tahap ketiga

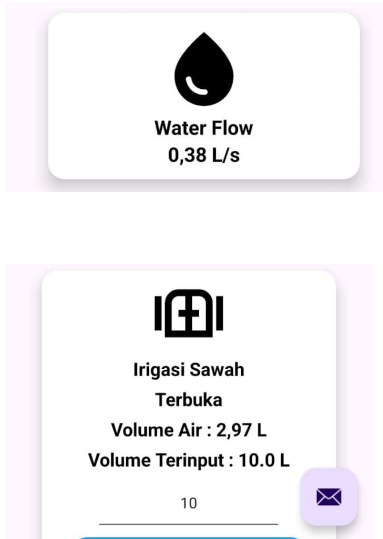
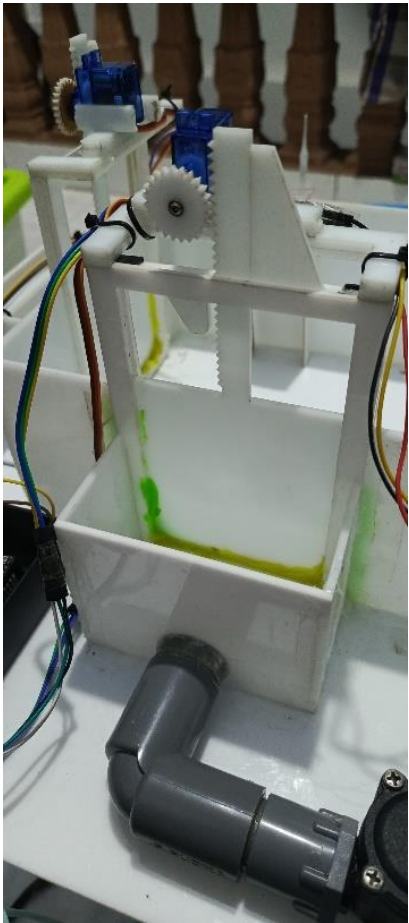
Tabel 4. 7 Pengujian Tahap Ketiga

Uji coba	Hasil	Keterangan
Jika sensor mendeteksi ketinggian level air rendah maka motor servo akan berputar 0° secara otomatis	✓	Berhasil, dikarenakan sensor mendeteksi ketinggian air dan servo berputar sesuai dengan yang diperintahkan
 		

d. Pengujian tahap keempat

Tabel 4. 8 Pengujian Tahap Keempat

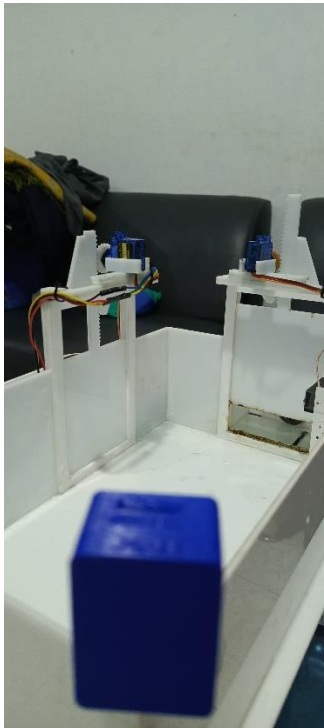
Uji coba	Hasil	Keterangan
Jika sensor mendeteksi aliran air dan menghitung total volume maka pintu 2 akan terbuka	✓	Berhasil, dikarenakan sensor mendeteksi aliran air dan servo berputar sesuai dengan yang diperintahkan

e. Pengujian tahap kelima

Tabel 4. 9 Pengujian Tahap Ketiga

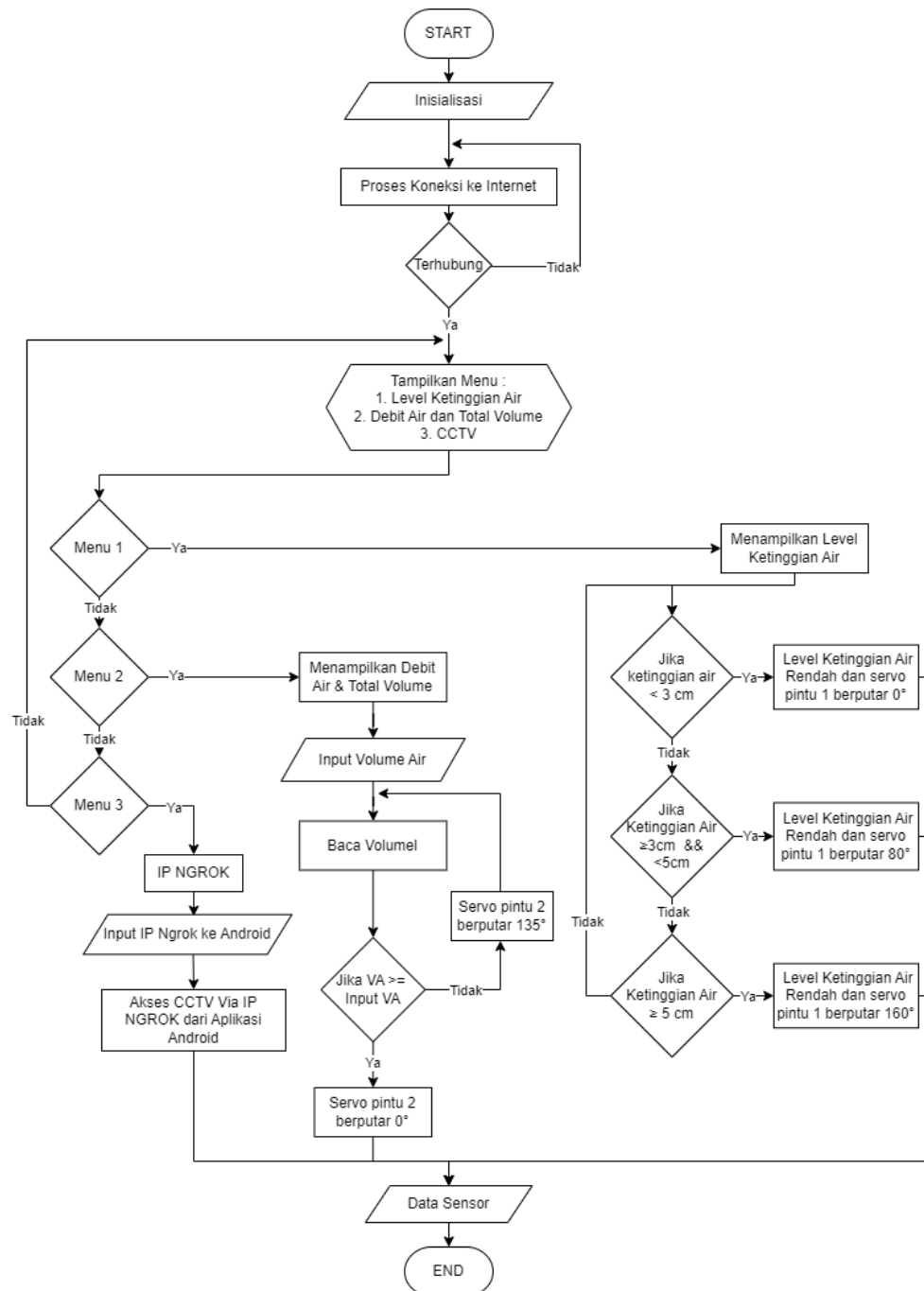
Uji coba	Hasil	Keterangan
Menyalakan esp32 cam dan dapat memonitoring melalui aplikasi secara realtime dengan menggunakan engrok sebagai ip address online	✓	Berhasil, video dari esp32 cam melalui engrok ip address telah terhubung ke aplikasi sesuai dengan perintah

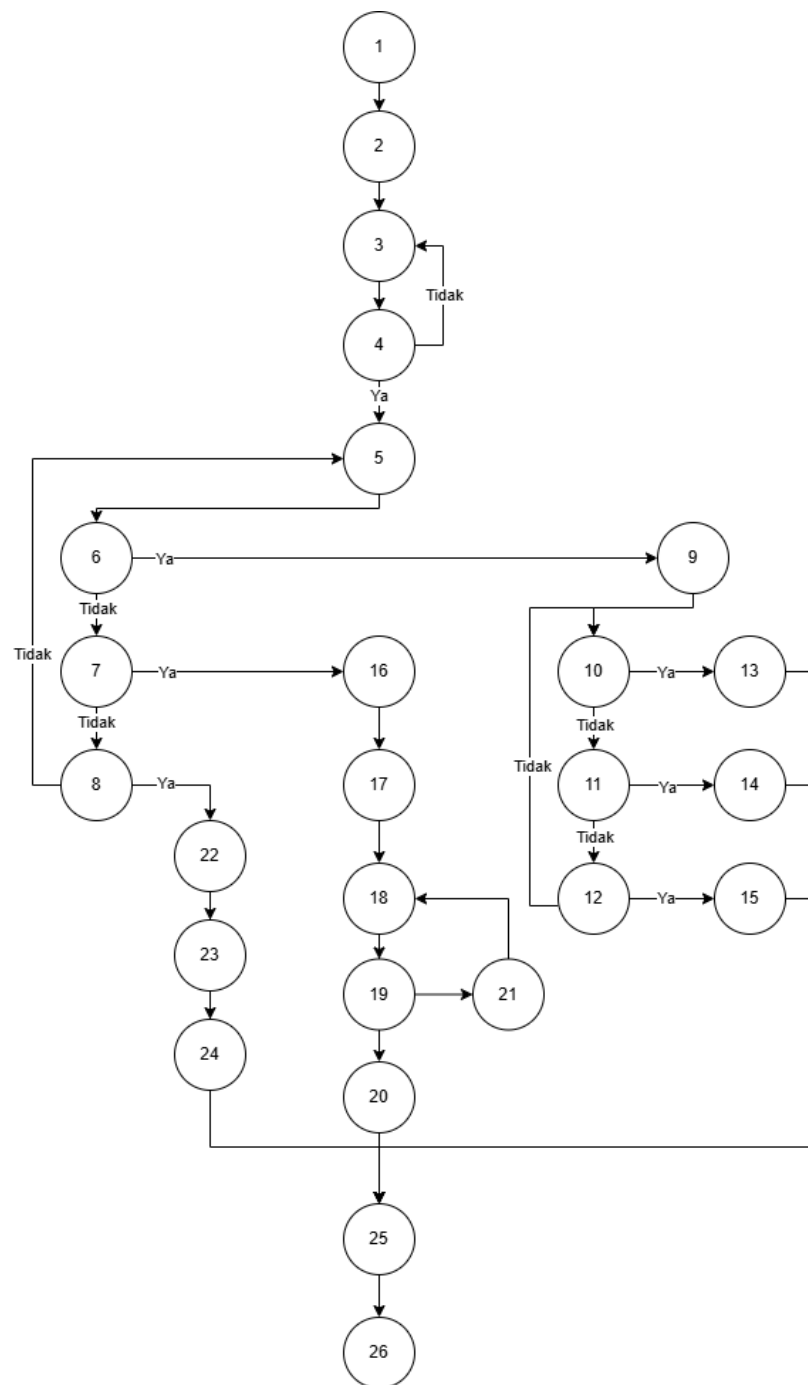
2. Pengujian White Box

Pengujian *white box* adalah Teknik pengujian yang berfokus pada struktur internal atau logika pada kode program yang diuji secara detail. Pada pengujian ini melibatkan pemahaman mendalam tentang bagaimana aplikasi itu dibuat, termasuk kode, sumber, algoritma dan jalur logika.

a. *Flowchart dan Flowgraph*



Gambar 4. 17 Flowchart



Gambar 4. 18 Flowgraph

b. Menghitung *Cyclomatic Complexity* $V(G)$ dari *Edge* dan *Node*

Cyclomatic complexity adalah metrik yang digunakan untuk mengukur kompleksitas suatu program atau sistem berdasarkan jumlah jalur

independen melalui kode. Semakin tinggi nilai cyclomatic complexity, semakin kompleks dan sulit dipahami program tersebut.

$V(G)$ Menu Sensor dan Pintu :

Rumus : $V(G) = E - N + 2$

Diketahui : $E (edge) = 33$

$N (Node) = 26$

Penyelesaian : $V(G) = E - N + 2$

$$= 33 - 26 + 2$$

$$= 9$$

c. Berdasarkan perhitungan *Cyclomatic Complexity* $V(G)$, *Flowchart* diatas memiliki Region = 9

d. *Indepent Path* yang terdapat pada *flowchart* adalah :

Path 1 : 1 – 2 – 3 – 4 – 3

Path 2 : 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 5

Path 3 : 1 – 2 – 3 – 4 – 5 – 6 – 9 – 10 – 11 – 12 – 9

Path 4 : 1 – 2 – 3 – 4 – 5 – 6 – 9 – 10 – 13 – 25 – 26

Path 5 : 1 – 2 – 3 – 4 – 5 – 6 – 9 – 11 – 14 – 25 – 26

Path 6 : 1 – 2 – 3 – 4 – 5 – 6 – 9 – 12 – 15 – 25 – 26

Path 7 : 1 – 2 – 3 – 4 – 5 – 7 – 16 – 17 – 18 – 19 – 21 – 18

Path 8 : 1 – 2 – 3 – 4 – 5 – 7 – 16 – 17 – 18 – 19 – 25 – 26

Path 9 : 1 – 2 – 3 – 4 – 5 – 8 – 22 – 23 – 24 – 25 – 26

e. *Grafik Matriks***Tabel 4. 8 Grafik Matriks**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	E-1
1		1																									1-0=0
2			1																								1-0=0
3				1																							1-0=0
4					1																						1-0=0
5						1	1	1																			1-3=2
6							1		1																		1-2=1
7								1								1											1-2=1
8					1																	1					1-2=1
9										1	1	1															1-3=2
10																											0
11												1															1-0=0
12									1																		1-0=0
13																									1		1-0=0
14																									1		1-0=0
15																									1		1-0=0
16																	1										1-0=0
17																											0
18																			1								1-0=0
19																											0
20																					1				1		1-2=0
21																			1								1-0=0
22																							1				1-0=0
23																											1-0=0
24																									1		1-0=0
25																										1	1-0=0
26																											0

3. Kebutuhan Air pada Petak Sawah

Dalam penelitian ini, dilakukan analisis kebutuhan air pada petak sawah seluas 48 m² yang ditanami padi, dengan memperhatikan suhu, kelembapan, kecepatan angin, durasi sinar matahari berdasarkan pada tabel 4.9 dan fase pertumbuhan vegetatif selama satu bulan pada tabel 4.10. Analisis ini penting untuk memastikan bahwa sistem irigasi otomatis yang dirancang mampu menyediakan jumlah air yang sesuai dengan kebutuhan tanaman, sehingga dapat mendukung pertumbuhan padi secara optimal.

a) Analisis Evapotranspirasi dan Koefisien Tanaman

Evapotranspirasi referensi (ET_0) digunakan sebagai dasar untuk menentukan kebutuhan air tanaman. Berdasarkan data pada tabel 4.9, Rata-rata ET_0 diperkirakan 3.18 mm/hari. Selanjutnya, koefisien tanaman (K_c) untuk padi pada fase vegetatif ditetapkan pada 1,1, sesuai dengan nilai yang umum digunakan dalam literatur agronomi.

Tabel 4. 9 Data Iklim Harian dan ET_0

Tanggal	Tn (°C)	Tx (°C)	Tavg (°C)	RH_avg (%)	ff_avg (m/s)	ss (jam)	ET_0 (mm/hari)
1 Juni 2024	24.1	33.7	29	77	1	5.1	2.85
2 Juni 2024	25.1	29.7	26.4	88	1	7.5	2.78
3 Juni 2024	25	30.7	26.8	87	1	0	2
4 Juni 2024	24.9	33.2	28.8	76	1	1	2.42
5 Juni 2024	25.5	32.2	28.4	74	1	8.2	3.21
6 Juni 2024	24.3	34	28.3	72	1	6.8	3.14
7 Juni 2024	23.7	33.4	27.7	77	2	6.7	3.34
8 Juni 2024	24.3	33.2	28.5	75	1	7.5	3.13
9 Juni 2024	24.6	32.7	27.9	78	1	7.7	3.09
10 Juni 2024	23.9	34.7	28	74	1	7.3	3.19
11 Juni 2024	23.5	33.7	28.5	74	1	5.5	2.93
12 Juni 2024	25.1	32.6	26.3	86	1	8.7	3.08
13 Juni 2024	23.8	32.8	28.2	75	1	3.9	2.72
14 Juni 2024	24.6	32.3	27.9	75	1	7.2	3.07
15 Juni 2024	23.7	33	27.1	74	2	9.1	3.68
16 Juni 2024	22.4	32.8	27.8	72	1	7.4	3.1
17 Juni 2024	23	33	27.3	72	1	6.7	3.03
18 Juni 2024	21.8	33.7	27.4	66	1	9.1	3.4
19 Juni 2024	22.3	32.9	26.6	67	2	9.3	3.9
20 Juni 2024	20.6	32.3	26.3	67	1	9.6	3.33
21 Juni 2024	21	33.5	27.7	69	2	8.6	3.75
22 Juni 2024	23.3	33	28.6	67	2	9.6	3.97
23 Juni 2024	24.4	31.8	27	80	1	7.5	2.99
24 Juni 2024	23.5	33.4	28.1	74	2	4.6	3.24
25 Juni 2024	24.3	32.4	27.9	75	2	8.7	3.58

Tanggal	Tn (°C)	Tx (°C)	Tavg (°C)	RH_avg (%)	ff_avg (m/s)	ss (jam)	ET _o (mm/hari)
26 Juni 2024	23.7	33	28	74	1	9.6	3.35
27 Juni 2024	23.3	33.6	28.4	76	1	9.5	3.32
28 Juni 2024	23.7	34.2	29.2	70	1	10.4	3.55
29 Juni 2024	23.4	32.7	27.8	80	1	10.4	3.31
30 Juni 2024	24.3	32.3	27.5	82	1	3.3	2.51
Rata-rata ET _o	-	-	-	-	-	-	3.17

(Sumber : BMKG Maros)

Berdasarkan pada table 4.9 mencakup beberapa informasi berupa tanggal menunjukkan tanggal pencatatan data. Tn (°C) dan Tx (°C) mencatat suhu minimum dan maksimum harian dalam derajat Celsius. Tavg (°C) adalah suhu rata-rata harian, dihitung sebagai rata-rata dari suhu maksimum dan minimum. RH_avg (%) mengindikasikan kelembapan relatif rata-rata harian, menunjukkan kadar uap air di udara. ff_avg (m/s) mengukur kecepatan angin rata-rata dalam meter per detik, sedangkan ss (jam) mencatat durasi sinar matahari harian dalam jam. Terakhir, ET_o (mm/hari) menggambarkan evapotranspirasi referensi harian dalam milimeter, mencerminkan laju penguapan air dari permukaan tanah dan tanaman dalam kondisi ideal.

Tabel 4. 10 Koefisien Tanaman (Kc) Padi

Fase Pertumbuhan	Koefisien Tanaman (Kc)
Fase Awal	0.4
Fase Vegetatif	1.1
Fase Pembungaan	1.2
Fase Pematangan	0.8

Dengan menggunakan ET_0 dan K_c yang telah ditentukan, kebutuhan air spesifik untuk tanaman padi (ET_c) dihitung sebagai berikut:

Rata-rata $ET_0 = 3.17 \text{ mm/hari}$

K_c Fase Vegetatif $K_c = 1.1$

Rumus :

$$ET_c = ET_0 \times K_c$$

Penyelesaian :

$$ET_c = ET_0 \times K_c$$

$$= 3.17 \times 1.1$$

$$= 3.487 \text{ mm/hari}$$

Nilai ET_c ini menunjukkan jumlah air yang dibutuhkan oleh tanaman padi per hari.

b) Perhitungan Kebutuhan Air per Petak Sawah

Perkiraan luas petak sawah yang dianalisis adalah 48 m^2 . Dengan nilai ET_c sebesar 3.487 mm/hari , volume air yang dibutuhkan per hari untuk satu petak sawah dihitung sebagai berikut:

$$\text{Luas Sawah} = 48 \text{ m}^2$$

$$ET_c = 3.487 \text{ mm/hari}$$

Rumus :

$$\text{Kebutuhan Air} = ET_c \times \text{Luas Sawah}$$

Penyelesaian :

$$\text{Konversi nilai } ET_c \text{ ke meter} = 3.487 \text{ mm} = \frac{3.487}{1000}$$

$$= 0.003487 \text{ m}$$

$$\begin{aligned}\text{Kebutuhan Air} &= 0.003487 \, m \times 48 \, m^2 \\ &= 0.1674 \, m^3\end{aligned}$$

Konversi ke Liter :

$$0.1674 \, m^3 \times 1000 = 167,4 \, L/hari$$

Selanjutnya, untuk menghitung total kebutuhan air selama satu bulan (30 hari) pada fase vegetatif, volume air harian tersebut dikalikan dengan jumlah hari:

$$167,4 \times 30 = 5022 \, L/bulan$$

Hasil perhitungan ini menunjukkan bahwa setiap petak sawah seluas 48m² membutuhkan sekitar 167.4 liter air dalam sehari dan 5022 liter air dalam sebulan selama fase vegetatif.

c) Efisiensi Irigasi dan Penyesuaian

Dalam praktik irigasi, tidak semua air yang disalurkan ke sawah akan langsung digunakan oleh tanaman. Sebagian air dapat hilang melalui perkolasi, penguapan, atau distribusi yang tidak merata. Oleh karena itu, efisiensi irigasi perlu diperhitungkan. Dalam penelitian ini, efisiensi irigasi diasumsikan sebesar 80%, yang merupakan nilai umum untuk sistem irigasi konvensional.

Dengan efisiensi irigasi 80%, volume air yang harus disalurkan oleh sistem irigasi otomatis dihitung sebagai berikut:

$$\begin{aligned}\text{Volume air yang disalurkan} &= \frac{5022 \, L/bulan}{0,8} \\ &= 6277,5 \, L/bulan\end{aligned}$$

Hasil perhitungan menunjukkan bahwa untuk mendukung pertumbuhan tanaman padi selama fase vegetatif, setiap petak sawah dengan luas 48 m² memerlukan sekitar 6277,5 liter air per bulan. Jumlah ini sudah mempertimbangkan efisiensi irigasi, sehingga memastikan bahwa volume air yang disalurkan mencukupi kebutuhan tanaman.

Data ini menjadi acuan penting dalam pengoperasian sistem irigasi otomatis yang dikembangkan, dimana penyesuaian volume air yang disalurkan dapat dilakukan secara presisi berdasarkan kebutuhan aktual tanaman. Selain itu, hasil ini juga memberikan panduan bagi petani dalam mengelola sumber daya air secara lebih efisien, terutama pada musim kemarau ketika ketersediaan air menjadi lebih terbatas.

Dengan demikian, sistem irigasi otomatis yang diimplementasikan dapat berperan secara efektif dalam mendukung pertumbuhan padi, sekaligus mengoptimalkan penggunaan air di lahan sawah, yang merupakan salah satu sumber daya yang sangat berharga, khususnya di wilayah dengan iklim kering.

4. Analisis Hasil Pengukuran Debit Air Menggunakan Sensor dan Manual

Sebelum membahas detail hasil pengukuran, perlu dilakukan perbandingan antara debit air yang diukur menggunakan sensor dengan debit yang diukur secara manual. Pengukuran dilakukan pada beberapa percobaan dengan durasi waktu yang sama, yaitu satu menit, untuk memastikan konsistensi data. Hasil

pengukuran ini kemudian dibandingkan untuk mengidentifikasi perbedaan atau kesamaan dalam akurasi dan kinerja kedua metode. Pada tabel 11 Perbandingan Sensor dan Manual akan menampilkan hasil pengukuran volume air serta perhitungan debit untuk masing-masing metode pada setiap percobaan.

Rumus yang digunakan :

- a. Debit (L/detik) :

$$Debit = \frac{Volume (L)}{Waktu (detik)}$$

- b. Selisih Volume (L)

$$Selisih Volume = Volume Manual - Volume Sensor$$

- c. Selisih Debit (L/detik)

$$Selisih Debit = Debit Manual - Debit Sensor$$

Tabel 4. 11 Perbandingan Hasil Pengukuran

No.	Volume Sensor (L)	Volume Manual (L)	Waktu (detik)	Debit Sensor (L/detik)	Debit Manual (L/detik)	Selisih Volume (L)	Selisih Debit (L/detik)
1	2.11	2.20	60	0.0325	0.0367	0.09	0.0015
2	2.14	2.28	60	0.0357	0.0380	0.14	0.0023
3	2.10	2.20	60	0.0350	0.0367	0.10	0.0017
4	2.10	2.25	60	0.0350	0.0375	0.15	0.0025
5	2.15	2.18	60	0.0358	0.0363	0.03	0.0005
6	2.03	2.12	60	0.0338	0.353	0.09	0.0015
Rata – rata				0.03508	0.03675	0.10	0.00167

Dari tabel di atas, terlihat bahwa terdapat selisih pada volume maupun debit antara pengukuran menggunakan sensor dan pengukuran manual. Rata-rata selisih volume adalah 0.10 L, sementara rata-rata selisih debit adalah 0.00167

L/detik. Perbedaan ini mungkin disebabkan oleh berbagai faktor seperti akurasi sensor, metode manual yang digunakan.

BAB V

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dijelaskan pada bab sebelumnya, maka dapat disimpulkan sebagai berikut :

1. Sistem monitoring irigasi otomatis berbasis *android* menghasilkan jika level ketinggian air Tinggi maka motor servo akan berputar 160°, dan jika level ketinggian air Sedang maka motor servo akan berputar 80°, kemudian jika level ketinggian air Rendah maka motor servo akan menutup pintu 1, dan jika menekan button mode manual pada aplikasi android maka pintu 1 dapat di kontrol secara manual.
2. Pada pembahasan diatas pengembangan dalam penelitian ini juga menghasilkan mekanisme pengaturan pintu ke 2 berdasarkan volume air yang diinginkan. Pintu 2 akan terbuka sehingga air akan mengalir dan memberikan data aliran air jika volume air belum terpenuhi. Setelah volume air tercapai maka pintu 2 akan menutup secara otomatis. Mekanisme ini memastikan bahwa irigasi yang dilakukan sesuai dengan kebutuhan yang spesifik, sehingga meningkatkan efesiensi penggunaan air dan memudahkan pengelola irigasi.
3. Proses pengtrolan, penerimaan dan pengiriman data memerlukan delay sebesar 0.1 detik setiap mengirim data dari mikrokontroler ke *firebase*, sehingga respon terhadap aplikasi dan mikrokontroler lebih cepat, dalam proses ini kecepatan konektivitas sangat berperan penting agar monitoring terasa lebih realtime.

4. Integrasi mikrokontroler cam dengan aplikasi android melalui NGROK berhasil meningkatkan kemampuan monitoring sistem irigasi otomatis. Dalam hal ini, memberikan akses secara real-time dan aman, serta meningkatkan kontrol dan pengawasan oleh pengguna.

B. Saran

Adapun saran yang kami berikan yaitu :

1. Diharapkan kedepannya untuk menggunakan sensor dengan Tingkat presisi yang tinggi atau kalibrasi berkala pada sensor yang digunakan.
2. Jika ingin diterapkan dilapangan terutama didaerah dengan koneksi internet yang tidak stabil disarankan menggunakan konektifitas yang lebih stabil.
3. Untuk pengembangan berikutnya menyediakan opsi penyimpanan data jangka Panjang akan memungkinkan analisis tren penggunaan air dan kinerja sistem waktu ke waktu.

DAFTAR PUSTAKA

- Aljundi, M. I., & Akbar, M. A. (2018). *Kotlin Zero to Hero: Membuat Aplikasi Android dengan Kotlin Cocok untuk Pemula-UDACODING*. Udacoding.
- Bariyah, S. H., & Imania, K. A. N. (2022). Pengembangan Virtual Assistant Chatbot Berbasis Whatsapp Pada Pusat Layanan Informasi Mahasiswa Institut Pendidikan Indonesia-Garut. *J. Petik*, 8(1), 66–79.
- Fauzan, M. N., & Adiputri, L. C. (2020). *Tutorial Membuat Prototipe Prediksi Ketinggian Air (PKA) Untuk Pendeteksi Banjir Peringatan Dini Berbasis IOT* (Vol. 1). Kreatif.
- Hakim, D. P. A. R., Budijanto, A., & Widjanarko, B. (2018). Sistem Monitoring Penggunaan Air PDAM pada Rumah Tangga Menggunakan Mikrokontroler NODEMCU Berbasis Smartphone ANDROID. *Jurnal Iptek*, 22(2), 9–18.
- Hasiholan, C., Primananda, R., & Amron, K. (2018). Implementasi Konsep Internet of Things pada Sistem Monitoring Banjir menggunakan Protokol MQTT. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2(12), 6128–6135.
- Herlinah, S., & Musliadi, K. H. (2019). *Pemrograman Aplikasi Android dengan Android Studio, Photoshop, dan Audition*. Elex Media Komputindo.
- Hutagaol, J. V., Setiawan, D., & Eteruddin, H. (2022). Perancangan Sistem Monitoring Kendaraan Listrik. *Jurnal Teknik*, 16(1), 96–102.
- Masruri, M. H., & Creativity, J. (2015). *Buku Pintar Android*. Elex Media Komputindo.
- Muslihudin, M. (2016). *Analisis dan perancangan Sistem Informasi menggunakan model Terstruktur dan UML*. Penerbit Andi.
- Najimuddin, D. (2019). *Buku Ajar Irigasi Pedesaan*. Deepublish.
- Parlika, R., Wijaya, D. C. M., Nisaa, T. A., & Rahmawati, S. (2021). Sistem Integrasi BOT Register Terhadap Website Pengolah Data Menggunakan Akses NGROK. *Jurnal Ilmiah SINUS*, 19(2), 1–16.
- Ray, P. P. (2018). A survey on Internet of Things architectures. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 30, Issue 3). <https://doi.org/10.1016/j.jksuci.2016.10.003>
- Sanad, E. A. W. (2019). Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire. *Jurnal Penelitian Enjiniring*, 22(1). <https://doi.org/10.25042/jpe.052018.04>
- Voutama, A. (2022). Sistem Antrian Cucian Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML. *Komputika : Jurnal Sistem Komputer*, 11(1). <https://doi.org/10.34010/komputika.v11i1.4677>
- Wasista, S., Saraswati, D. A., & Susanto, E. (2019). *Aplikasi Internet of Things (IOT) dengan Arduino dan Android “Membangun Smart Home Dan Smart Robot Berbasis Arduino Dan Android.”* Deepublish.

Wilnaldo, A., Satria Putra, Y., Adriat Program Studi Geofisika, R., Universitas Tanjungpura, F., & Hadari Nawawi, J. (2020). Perbandingan Metode Perhitungan Evapotranspirasi Potensial di Paloh Kabupaten Sambas Kalimantan Barat. *PRISMA FISIKA*, 8(3), 165–171.