



Monitoring Irigasi Otomatis Berbasis *IoT* dan *Android*

Untung Suwardoyo^{1*} , Raf Sanjani MK²

^{1*,2}Program Studi Teknik Informatika, Universitas Muhammadiyah Parepare, Indonesia

*Email : untung.suwardoyo@gmail.com

Abstract:

Irrigation management is still often done manually, especially in Pinrang Regency. Managing irrigation efficiently has challenges due to distance and unpredictable weather conditions. The purpose of this design is to make it easier to open and close irrigation doors through Android. The method used is an experimental method with a design using Internet of Things-based tools and Android applications. The test results show that the system prototype can operate the automatic irrigation gate according to the required water height and volume. The design proves that it can be an effective and efficient solution in irrigation management.

Keywords: *Irrigation; Monitoring; Internet of Things; Android.*

1. PENDAHULUAN

Irigasi merupakan upaya yang dilakukan untuk mengalirkan air dengan membangun bangunan atau saluran-saluran ke sawah-sawah dengan cara terstruktur dan membuang air yang tidak diperlukan lagi, setelah air di pergunakan dengan sebaik-baiknya (Samsugi et al., 2020). Dalam dunia modern, saat ini sudah banyak model irigasi yang dapat di lakukan manusia (Najimuddin, 2019).

Sistem irigasi saat ini sebagian besar masih menggunakan sistem manual (Izulyansah, 2020), salah satunya di Kabupaten Pinrang, yaitu di mana membuka dan menutup saluran irigasi masih secara tradisional. Ketika terjadi peluapan air di akibatkan hujan deras, petugas merasa kesulitan untuk mengatur irigasi dikarenakan jarak yang jauh. Oleh karena itu, 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 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* beroperasi dengan memanfaatkan argumen pemrograman, di mana setiap instruksi menghasilkan interaksi langsung antara mesin tanpa memerlukan campur tangan manusia dan tanpa batasan jarak (Sari et al., 2024). *IoT* menggunakan beberapa teknologi yang digabungkan menjadi satu kesatuan diantaranya sensor, koneksi internet, platform cloud dan perangkat keras sehingga menjadi sebuah rancangan sesuai dengan kebutuhan (Irmayani Pawelloi, 2023). *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.

Berdasarkan latar belakang, maka penelitian ini bertujuan untuk mengembangkan sistem monitoring irigasi otomatis berbasis *Android* yang memiliki potensi untuk memberikan manfaat signifikan, seperti kemampuan mengetahui ketinggian air dan debit air secara *real-time*, serta untuk memudahkan petugas dalam mengelola irigasi.

2. METODOLOGI PENELITIAN

Metode yang digunakan adalah metode eksperimental yang melibatkan perancangan sistem, pengumpulan data dari sensor, implementasi dan pengujian prototipe, serta analisis kerja sistem. Lokasi penelitian dilakukan di Dinas Pengairan Kabupaten Pinrang dan waktu pelaksanaan penelitian dilakukan selama tiga bulan, pada bulan Agustus-Oktober tahun 2023. Pada penelitian yang telah dilakukan, dibutuhkan beberapa alat dan bahan. Adapun alat dan bahan yang dibutuhkan dapat dilihat pada tabel 1.

Tabel 1. Alat dan Bahan

No.	Alat dan Bahan	Spesifikasi
1	Laptop	<i>Asus X550V Inter i7-7700HQ 2.8HZ (8 CPUs) up to 3,8Ghz</i>
2	Ponsel <i>Android</i>	<i>INFINIX HOT 40 PRO</i>
3	Kabel	<i>Type C , Kabel Jumper Male, Female & Micro USB</i>
4	Perangkat Tambahan	<i>Air, Pipa, Selang, Breadboard & WiFi Router, Servo SG90</i>
5	<i>Node MCU</i>	<i>ESP 32, ESP32-CAM</i>
6	Bahasa Program	<i>C++ & Kotlin</i>
7	Sensor	<i>Water Flow YF-S201, Water Level</i>

2.1 Metode Pengujian

a. *White box*

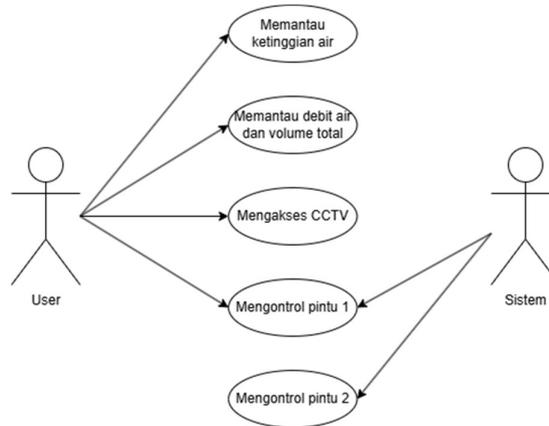
White Box adalah metode pengujian aplikasi dengan memeriksa modul untuk menilai dan menganalisis kode program guna mendeteksi apakah terdapat kesalahan atau tidak (Anggi Andriyadi et al., 2022). Pada pengujian ini melibatkan pemahaman mendalam tentang bagaimana aplikasi itu dibuat, termasuk kode, sumber, algoritma dan jalur logika.

b. *Black Box*

Black Box merupakan teknik pengujian yang bertujuan untuk mengidentifikasi kesalahan pada sistem aplikasi, seperti kesalahan dalam fungsi sistem atau hilangnya menu aplikasi. Oleh karena itu, *Black Box* Testing adalah metode yang digunakan untuk menguji fungsionalitas sistem aplikasi (Muhammad Arofiq et al., 2023).

3. HASIL DAN PEMBAHASAN

3.1 Rancangan Sistem

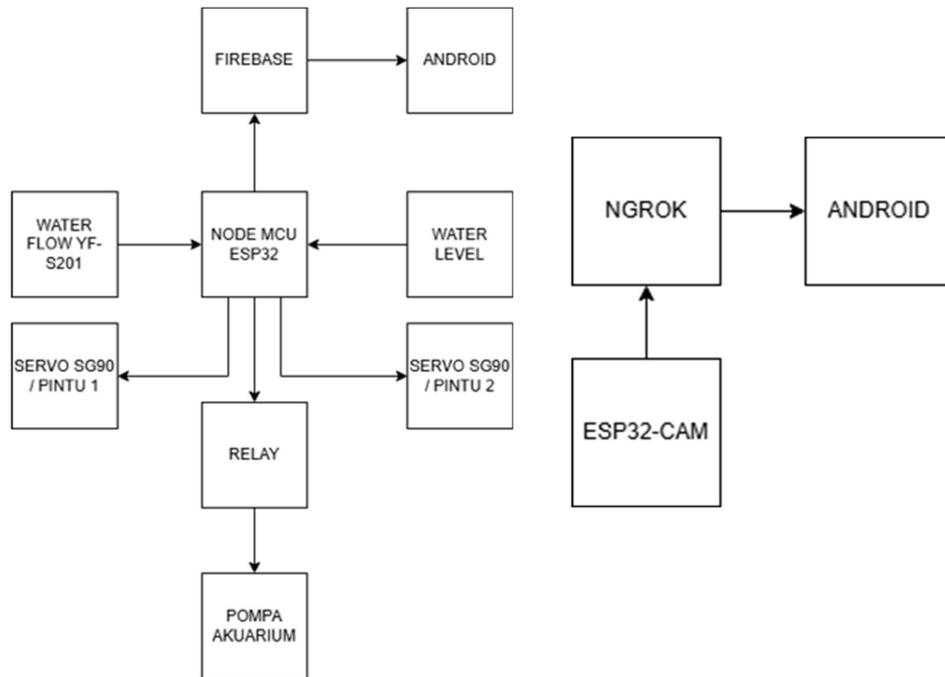


Gambar 1. *Use case Diagram*

Pada gambar 1, menjelaskan bagaimana *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 secara otomatis. *User* dapat mengontrol pintu 1 dan 2 serta dapat mengubah pintu 1 menjadi *mode manual* atau otomatis yang akan di kontrol oleh sistem berdasarkan *level* ketinggian air.

3.2 Implementasi Perangkat Keras (*Hardware*)

Komponen perangkat keras (*Hardware*) yang digunakan yaitu: *Node MCU ESP32*; *Sensor Water Level*; *Sensor Water Flow*; *Motor Servo*; *Relay*. Pengimplementasian komponen akan dilakukan sesuai dengan rancangan sistem, yang berupa alat jadi dan siap untuk digunakan sesuai kebutuhan, seperti pada gambar 2.



Gambar 2. Block Diagram Sistem ESP32 dan CAM

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

a) 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 2. Penggunaan *Pin* Sensor *Water Level* (Pintu 1 Irigasi)

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

Pada Tabel 2 menunjukkan sensor *water level* dihubungkan ke ESP32 dengan output sensor ke *Pin 33*, *VCC* sensor ke *VCC 5V* pada ESP32, dan *GND* sensor ke *GND* ESP32. Koneksi ini memungkinkan pembacaan ketinggian air untuk mengontrol Pintu 1 irigasi.

b) *Motor Servo*

Motor servo adalah perangkat yang dirancang dengan sistem kontrol umpan balik *loop* tertutup sehingga posisinya dapat diatur sesuai dengan poros derajatnya (Susanta, M. H. 2024). Adapun pin yang digunakan untuk Motor servo sebagai berikut :

Tabel 3. Penggunaan Pin Motor Servo (Pintu 1 Irigasi)

No.	ESP32	Motor Servo
1.	Output	Pin 27
2.	VCC	VCC 5V
3.	GND	GND

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

No.	ESP32	Motor Servo
1.	Output	Pin 14
2.	VCC	VCC 5V
3.	GND	GND

Pada tabel 3 dan 4 menunjukkan penggunaan *pin* untuk *motor servo* yang mengatur Pintu 1 dan Pintu 2 irigasi. Pada Tabel 3, motor servo Pintu 1 dihubungkan ke *ESP32* dengan *pin output* pada *Pin 27*, *VCC* pada *VCC 5V*, dan *GND* pada *GND*. Pada Tabel 4, *motor servo* Pintu 2 dihubungkan dengan *pin output* pada *Pin 14*, *VCC* pada *VCC 5V*, dan *GND* pada *GND*. Koneksi ini memungkinkan kontrol masing-masing pintu irigasi melalui *motor servo*.

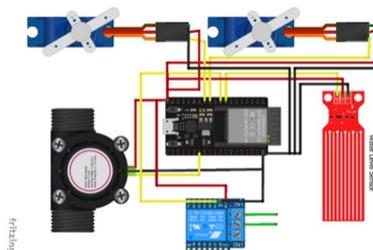
c) *Relay*

Relay merupakan sebuah komponen yang berfungsi sebagai saklar yang dioperasikan secara elektronik yang terdiri dari dua bagian utama, yaitu *electromagnet (koi)* dan mekanisme (kontak saklar) (Budiyanto *et al.*, 2020). *Relay* sering digunakan untuk mengontrol suatu perangkat yang membutuhkan arus atau tegangan yang lebih tinggi dari mikrokontroler.

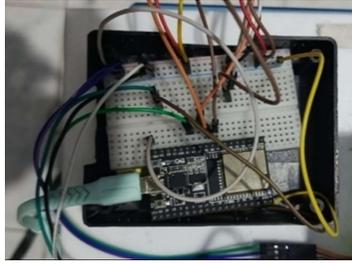
Tabel 5. Penggunaan Pin Relay

No.	ESP32	Relay
1.	Output	Pin 25
2.	VCC	VCC 5V
3.	GND	GND

Pada tabel 5 menunjukkan penggunaan *pin* untuk *relay* yang dikendalikan oleh *ESP32*. Relay dihubungkan ke *ESP32* dengan *pin output* pada Pin 25, *VCC* pada *VCC 5V*, dan *GND* pada *GND*. Koneksi ini memungkinkan *ESP32* untuk mengendalikan *relay* untuk mengaktifkan atau menonaktifkan perangkat eksternal. Adapun rangkaian perancangan sistem *monitoring* irigasi otomatis berbasis android pada gambar berikut :



Gambar 3. Rangkain sistem irigasi otomatis



Gambar 4. Rancangan Sistem

Berdasarkan pada gambar 3 dan 4, ada beberapa komponen yang digunakan dan saling terhubung dengan kabel *jumper*.

3.3 Rancangan Perangkat Lunak (*Software*)

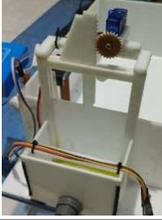
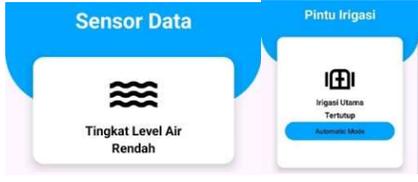
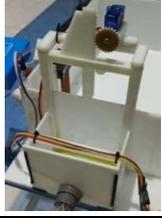
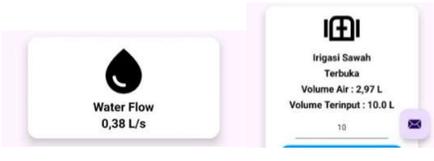
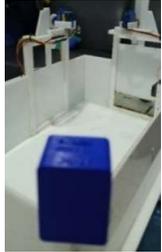
Pada perancangan perangkat lunak merupakan pembuatan program yang akan di upload pada mikrokontroler yang merupakan sebuah perintah untuk mengolah data pada aplikasi. Bahasa pemrograman yang digunakan berupa *C/C++* di *arduino IDE* dan untuk membuat aplikasi *android* menggunakan bahasa pemrograman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

3.4 Pengujian Aplikasi

a. Pengujian *Black-Box*

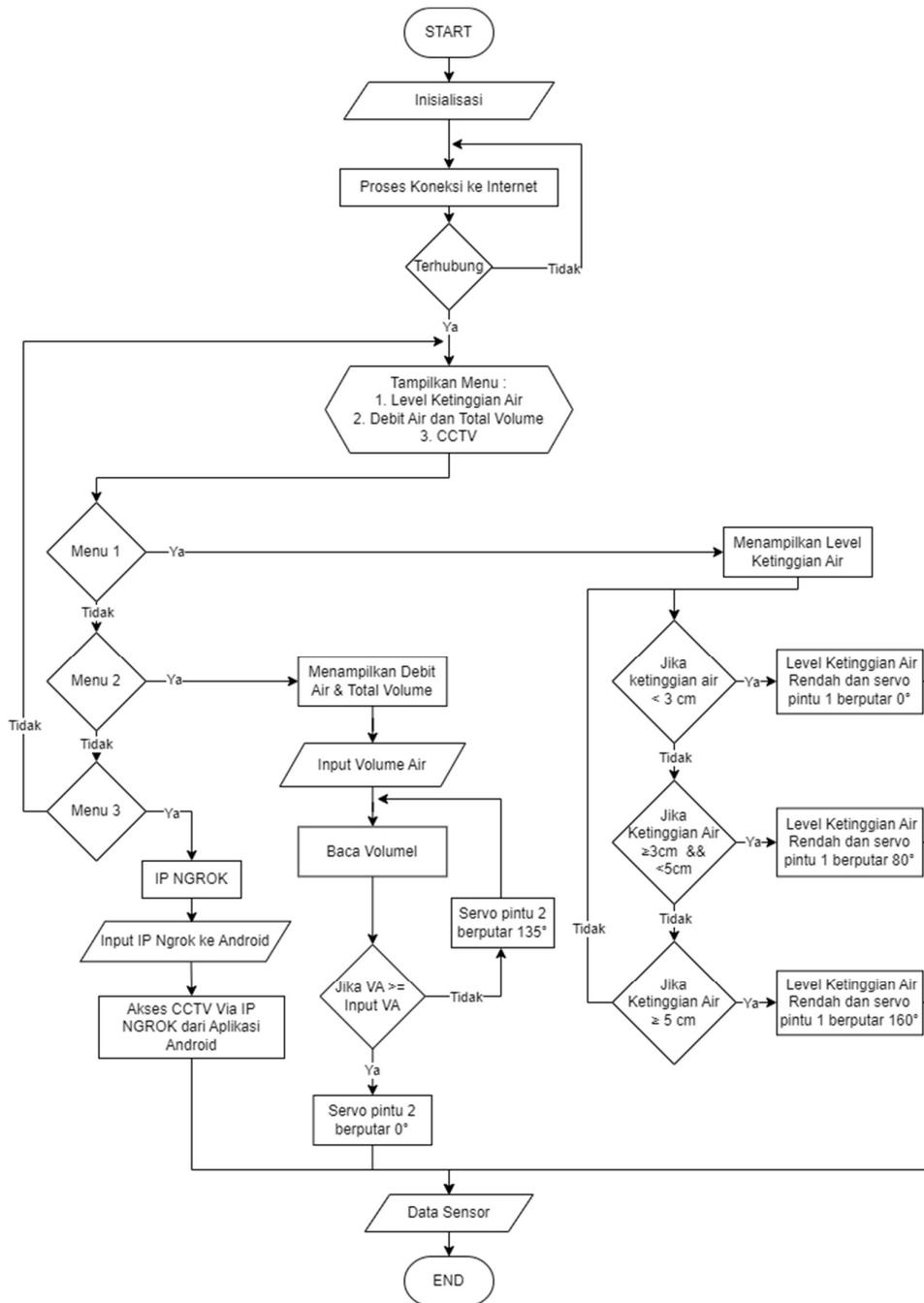
Tabel 6. Pengujian *black-box*

Uji coba tahap 1	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
Uji coba tahap 2	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

		
<p>Uji coba tahap 3</p> <p>Jika sensor mendeteksi ketinggian level air rendah maka motor servo akan berputar 0° secara otomatis</p>	<p>Hasil</p> <p>✓</p>	<p>Keterangan</p> <p>Berhasil, dikarenakan sensor mendeteksi ketinggian air dan servo berputar sesuai dengan yang diperintahkan</p>
		
<p>Uji coba tahap 4</p> <p>Jika sensor mendeteksi aliran air dan menghitung total volume maka pintu 2 akan terbuka</p>	<p>Hasil</p> <p>✓</p>	<p>Keterangan</p> <p>Berhasil, dikarenakan sensor mendeteksi aliran air dan servo berputar sesuai dengan yang diperintahkan</p>
		
<p>Uji coba tahap 5</p> <p>Menyalakan esp32 cam dan dapat memonitoring melalu aplikasi secara realtime dengan menggunakan <i>ngrok</i> sebagai ip address online</p>	<p>Hasil</p> <p>✓</p>	<p>Keterangan</p> <p>Berhasil, video dari esp32 cam melalui engrok ip address telah terhubung ke aplikasi sesuai dengan perintah</p>
		

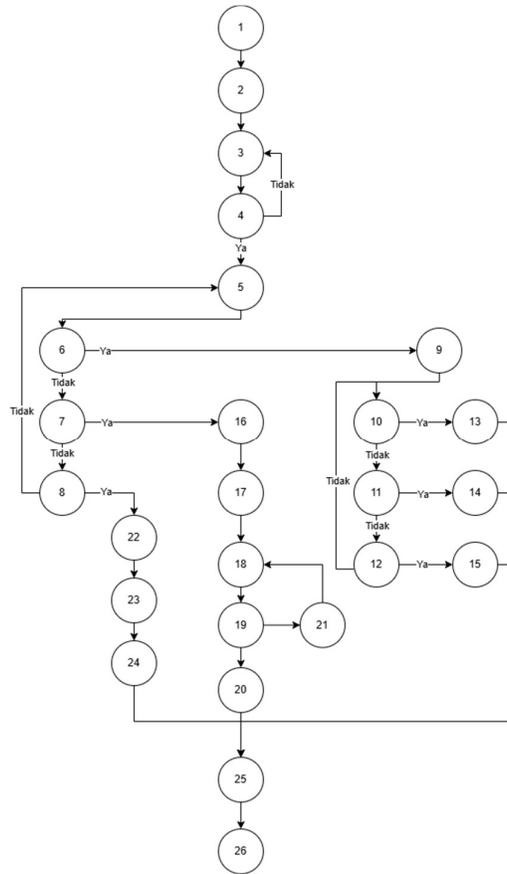
b. Pengujian White Box

a) *Flowchart dan Flowgraph*



Gambar 5. *Flowchart*

Berdasarkan pada gambar 5 menunjukkan alur proses kerja sistem irigasi otomatis. Proses dimulai dari koneksi ke *internet*, menampilkan menu, membuka tutup pintu irigasi hingga menampilkan data sensor dan selesai.



Gambar 6. *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 (Billiyon et al., 2024). Semakin tinggi nilai *cyclomatic complexity*, semakin kompleks dan sulit dipahami program tersebut. Berdasarkan pada gambar 5 dan 6 dapat dilakukan penyelesaian sebagai berikut:

$V(G)$ 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 7. *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	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-0=0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-0=0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-0=0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-0=0
5	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-3=2
6	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-2=1
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1-2=1
8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1-2=1
9	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-3=2
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-0=0
12	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-0=0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1-0=0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1-0=0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1-0=0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1-0=0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1-0=0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1-2=0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1-0=0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1-0=0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1-0=0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1-0=0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1-0=0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Pada tabel 7 menunjukkan matriks grafik dengan 26 kolom dan 26 baris yang berisi nilai 0 dan 1. Setiap baris mewakili status dari 26 elemen pada waktu tertentu, dengan

kolom-kolom yang menunjukkan status pada elemen-elemen tersebut. Di bagian akhir setiap baris, terdapat hasil perhitungan 1 - jumlah nilai 1 pada baris tersebut. Nilai-nilai ini menggambarkan seberapa banyak elemen yang tidak aktif atau tidak memenuhi kriteria tertentu pada setiap baris. Misalnya, pada baris ke-5, terdapat 3 elemen yang aktif yang bernilai 1 setiap elemennya, sehingga hasil perhitungannya adalah $1 - 3 = -2$, yang menunjukkan ketidaksesuaian dengan kriteria yang ditetapkan.

4. KESIMPULAN

Pada pengujian ini menunjukkan bahwa prototipe sistem ini dapat mengoperasikan pintu irigasi secara otomatis, monitoring, kontrol sesuai dengan ketinggian dan volume air yang dibutuhkan melalui aplikasi *android*. Rancangan ini membuktikan dapat menjadi solusi efektif dan efisien dalam pengelolaan, pengawasan irigasi oleh petugas.

REFERENSI

- Anggi Andriyadi, O., Rizal Nul Fikri, R., & Friska Saputri, E. (2022). *Informatika Darmajaya Dengan Whitebox Testing*. 8. <https://doi.org/10.53625/jirk.v1i8.1132>
- Billiyani, R., Ghifari, M., Fitri, S., Ardhyandoko, A., & Yaqin, M. A. (2024). Analisis Dan Perancangan Software Pengukuran Matriks Skala Dan Kompleksitas Kode Program Analysis And Design Of Scale Matrix Measurement Software And Program Code Complexity. In *JACIS: Journal Automation Computer Information System* (Vol. 4, Issue 1). <https://doi.org/10.53625/jirk.v1i8.1132>
- Budiyanto, A., Pramudita, G. B., Adinandra, S., Studi, P., & Elektro, T. (N.D.). *Kontrol Relay Dan Kecepatan Kipas Angin Direct Current (DC) Dengan Sensor Suhu LM35 Berbasis Internet Of Things (Iot)*. Retrieved August 31, 2024, From <https://doi.org/10.31358/techne.v19i01.224>
- Irmayani Pawelloi, A. (2023). *Rancang Bangun Sistem Monitoring Suhu Dan Salinitas Air Pada Lahan Rumput Berbasis Internet Of Things (Iot) Informasi Artikel* (Vol. 3, Issue 1). <http://jurnal.umpar.ac.id/index/jmosfet-5>
- Izulyansah, Y. (2020). *Monitoring Sistem Irigasi Sawah Menggunakan Android Tugas Akhir*. <https://repository.dinamika.ac.id/id/eprint/4216/1/15410200050-2020-UNIVERSITAS-DINAMIKA.Pdf>
- Muhammad Arofiq, N., Ferdo Erlangga, R., Irawan, A., & Saifudin, A. (2023). *Oktal: Jurnal Ilmu Komputer Dan Science Pengujian Fungsional Aplikasi Inventory Barang Kedatangan Dengan Metode Black Box Testing Bagi Pemula*. 2(5). <https://journal.mediapublikasi.id/index.php/oktal>

- Najimuddin, D. (2019). *Buku Ajar Irigasi Pedesaan*. Deepublish.
- Samsugi, S., Mardiyansyah, Z., & Nurkholis, A. (2020). Sistem Pengontrol Irigasi Otomatis Menggunakan Mikrokontroler Arduino Uno. *Jurnal Teknologi Dan Sistem Tertanam*, 1(1). <https://doi.org/10.33365/jtst.v1i1.719>
- Sari, I. P., Novita, A., Al-Khowarizmi, A.-K., Ramadhani, F., & Satria, A. (2024). Pemanfaatan Internet Of Things (Iot) Pada Bidang Pertanian Menggunakan Arduino Uno. *Blend Sains Jurnal Teknik*, 2(4), 337–343. <https://doi.org/10.56211/blendsains.v2i4.505>
- Susanta, M. H. (2024). *Prototype Penggunaan Empat Sensor Ultrasonik Pada Palang Parkir Otomatis Berbasis Arduino Uno*. <https://doi.org/10.572349/scientica.v2i10.2641>
- 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.