

**PERANCANGAN SISTEM KONTROL OTOMATIS POMPA  
AIR BERBASIS LOGIKA FUZZY DENGAN INTEGRASI IOT  
STUDI KASUS: STASIUN POMPA CIDENG JAKARTA**

TESIS S-2

Untuk memenuhi persyaratan memperoleh gelar Magister Teknik  
Program Studi Magister Teknik Elektro



**Oleh:**

**IVAN ADE PRADIPTA**

**NIM: 20602300010**

**PROGRAM STUDI MAGISTER TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM SULTAN AGUNG  
SEMARANG  
2025**

## HALAMAN PERSETUJUAN

### PERANCANGAN SISTEM KONTROL OTOMATIS POMPA AIR BERBASIS LOGIKA FUZZY DENGAN INTEGRASI IOT STUDI KASUS: STASIUN POMPA CIDENG JAKARTA

Yang dipersiapkan dan disusun oleh:

**IVAN ADE PRADIPTA**

**NIM: 20602300010**

Tesis ini telah disetujui oleh pembimbing dan selanjutnya

Dapat diajukan kehadapan sidang panitia ujian tesis

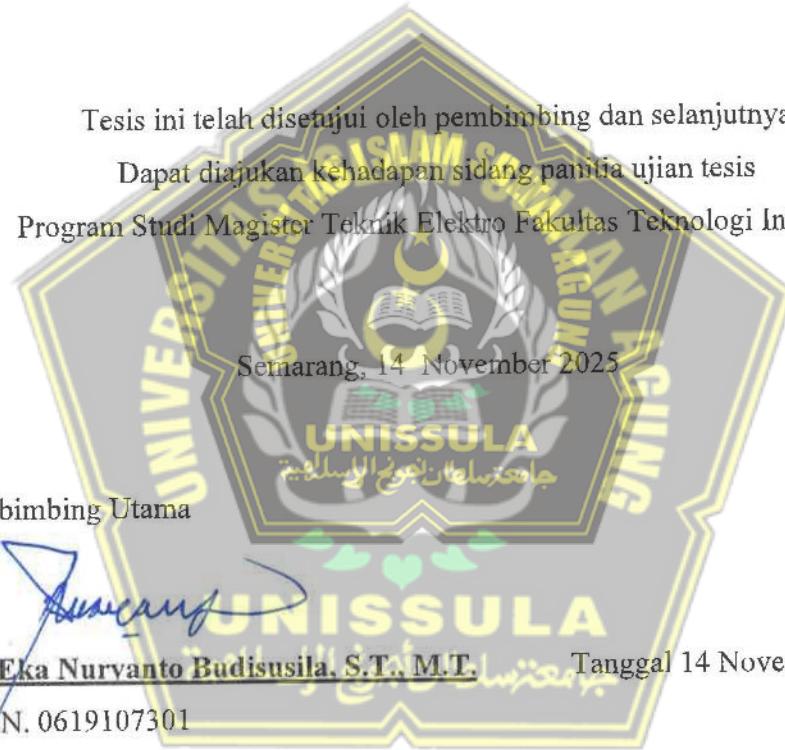
Program Studi Magister Teknik Elektro Fakultas Teknologi Industri

Semarang, 14 November 2025

Pembimbing Utama

Dr. Eka Nurvanto Budisusila, S.T., M.T. Tanggal 14 November 2025

NIDN. 0619107301



## LEMBAR PENGESAHAN

### PERANCANGAN SISTEM KONTROL OTOMATIS POMPA AIR BERBASIS LOGIKA FUZZY DENGAN INTEGRASI IOT STUDI KASUS: STASIUN POMPA CIDENG JAKARTA

Yang dipersiapkan dan disusun oleh:

**IVAN ADE PRADIPTA**

**NIM: 20602300010**

Telah dipertahankan di depan Dewan Penguji

Pada tanggal 21 November 2025

Susunan Dewan Penguji

Pembimbing Utama

Penguji 1

Dr. Eka Nurvanto Budisusila, S.T., M.T. Dr. Bustanul Arifin, S.T., M.T.

NIDN. 0619107301

NIDN. 0614117701

Ketua Penguji

Prof. Dr. Ir. Sri Artini Dwi Prasetyowati, M.Si.

NIDN. 0620026501

Tesis ini diterima sebagai salah satu pernyataan

Untuk memperoleh gelar Magister Teknik

Tanggal 21 November 2025

Ketua Program Studi Magister Teknik Elektro

Prof. Dr. Ir. Sri Artini Dwi Prasetyowati, M.Si.

NIDN. 0620026501

## PERNYATAAN KEASLIAN TESIS

Yang bertandatangan dibawah ini:

Nama : Ivan Ade Pradipta  
NIM : 20602300010  
Jurusan : Magister Teknik Elektro  
Fakultas : Fakultas Teknologi Industri

Dengan ini saya menyatakan bahwa Tesis yang diajukan kepada Program Studi Magister Teknik Elektro dengan judul:

### **“PERANCANGAN SISTEM KONTROL OTOMATIS POMPA AIR BERBASIS LOGIKA FUZZY DENGAN INTEGRASI IOT STUDI KASUS: STASIUN POMPA CIDENG JAKARTA”**

Adalah hasil karya sendiri, judul tersebut belum pernah diajukan untuk memperoleh gelar Magister Teknik pada Program Studi Magister Teknik Elektro Universitas Islam Sultan Agung ataupun pada universitas lain serta belum pernah ditulis maupun diterbitkan oleh orang lain kecuali secara tertulis diacu, disitasikan dan ditunjuk dalam daftar pustaka. Tesis ini adalah milik saya, segala bentuk kesalahan dan kekeliruan dalam Tesis ini adalah tanggung jawab saya.

Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggungjawab.

Surat ini dibuat di Semarang, 21 November 2025

Yang Menyatakan,

  
Ivan Ade Pradipta

NIM. 20602300010

## PERNYATAAN PERSETUJUAN UNGGAH KARYA ILMIAH

Saya yang bertandatangan dibawah ini:

Nama : Ivan Ade Pradipta  
NIM : 20602300010  
Jurusan : Magister Teknik Elektro  
Fakultas : Fakultas Teknologi Industri

Dengan ini menyerahkan karya ilmiah berupa Tesis dengan Judul:

### “PERANCANGAN SISTEM KONTROL OTOMATIS POMPA AIR BERBASIS LOGIKA FUZZY DENGAN INTEGRASI IOT STUDI KASUS: STASIUN POMPA CIDENG JAKARTA”

Dan menyetujuinya menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak Bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dalam pangkalan data, dan dipublikasikannya di internet atau media lain untuk kepentingan akademis selama tetap mencantumkan nama penulis sebagai pemilik Hak Cipta.

Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan pihak Universitas Islam Sultan Agung.

Semarang, 21 November 2025

Yang Menyatakan,



Ivan Ade Pradipta

NIM. 20602300010

## KATA PENGANTAR

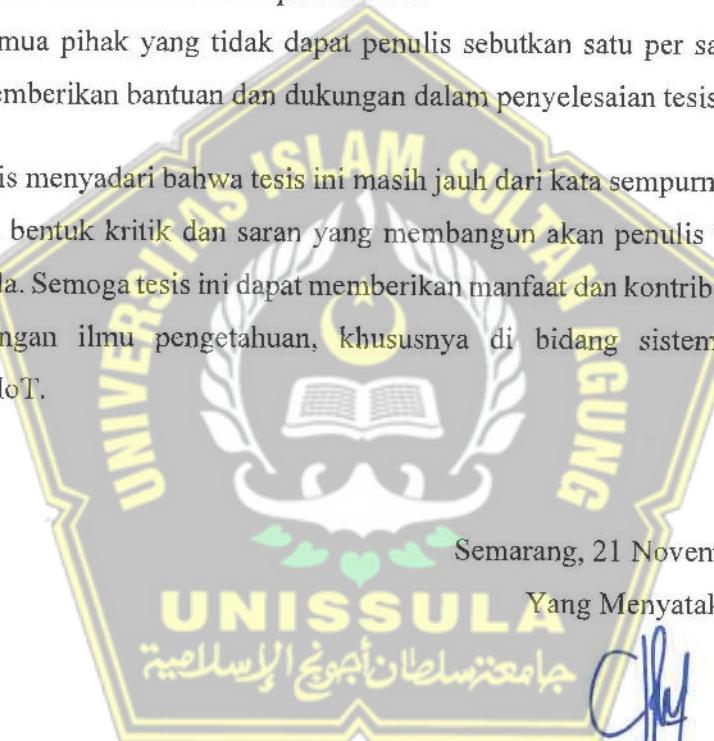
Alhamdulillahirabbil'alamin, segala puji dan syukur penulis panjatkan kehadirat Allah Subhanahu Wa Ta'ala atas segala rahmat, hidayah, dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan tesis yang berjudul “PERANCANGAN SISTEM KONTROL OTOMATIS POMPA AIR BERBASIS LOGIKA FUZZY DENGAN INTEGRASI IOT” dengan baik dan lancar. Tesis ini disusun sebagai salah satu syarat untuk memperoleh gelar Magister Teknik (M.T.) pada Program Studi Teknik Elektro, Universitas Islam Sultan Agung Kota Semarang.

Penulis menyadari bahwa dalam penyusunan tesis ini, terdapat banyak tantangan dan hambatan. Namun, berkat bimbingan, dukungan, serta doa dari berbagai pihak, semua kesulitan tersebut dapat teratasi. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang tulus kepada:

1. Bapak Dr. Eka Nuryanto Budisusila, S.T., M.T., selaku Dosen Pembimbing Utama, yang telah meluangkan waktu, tenaga, dan pikiran untuk memberikan bimbingan, arahan, serta masukan yang sangat berharga sejak awal hingga akhir penyusunan tesis ini.
2. Bapak Dr. Bustanul Arifin, S.T., M.T., selaku Anggota Pengaji 1, yang telah memberikan kritik dan saran yang membangun demi kesempurnaan tesis ini.
3. Ibu Prof. Dr. Ir. Sri Arttini Dwi Prasetyowati, M.Si., selaku Ketua Program Studi Magister Teknik Elektro Universitas Islam Sultan Agung dan Ketua Pengaji, atas segala kemudahan dan dukungan dalam proses administrasi akademik serta penyempurnaan tesis ini.
4. Ibu Dr. Novi Marlyana S.T., M.T. IPU, ASEAN Eng., selaku Dekan Fakultas Teknologi Industri Universitas Islam Sultan Agung yang telah mendukung segala kegiatan di lingkungan Fakultas Teknologi Industri.
5. Istri dan Anak tercinta, Rizka Zulfiyani dan Ibrahim Khayr Pradipta yang telah membantu banyak dengan menerahkan tenaga dan pikirannya selama ini.

6. Kedua orang tua tercinta, Ayahanda Ansory (Rahimahullah) dan Ibunda Siti Solechah, serta seluruh keluarga besar yang senantiasa memberikan doa dan dukungan.
7. Pimpinan dan seluruh staf Dinas Sumber Daya Air DKI Jakarta khususnya Unit Pusat Data dan Informasi Sumber Daya Air serta operator Stasiun Pompa Cideng yang telah memberikan izin dan bantuan selama proses pengambilan data dan observasi lapangan.
8. Rekan-rekan seperjuangan angkatan 2023 Program Magister Teknik Elektro, terima kasih atas persahabatan, semangat, serta diskusi yang bermanfaat selama masa perkuliahan.
9. Semua pihak yang tidak dapat penulis sebutkan satu per satu, yang telah memberikan bantuan dan dukungan dalam penyelesaian tesis ini.

Penulis menyadari bahwa tesis ini masih jauh dari kata sempurna. Oleh karena itu, segala bentuk kritik dan saran yang membangun akan penulis terima dengan lapang dada. Semoga tesis ini dapat memberikan manfaat dan kontribusi positif bagi pengembangan ilmu pengetahuan, khususnya di bidang sistem kontrol dan teknologi IoT.



Semarang, 21 November 2025

Yang Menyatakan,

Ivan Ade Pradipta

NIM. 20602300010

## DAFTAR ISI

|   |             |
|---|-------------|
| <b>COVER .....</b>                                      | <b>i</b>    |
| <b>HALAMAN PERSETUJUAN .....</b>                        | <b>ii</b>   |
| <b>LEMBAR PENGESAHAN .....</b>                          | <b>iii</b>  |
| <b>PERNYATAAN KEASLIAN TESIS .....</b>                  | <b>iv</b>   |
| <b>PERNYATAAN PERSETUJUAN UNGGAH KARYA ILMIAH .....</b> | <b>v</b>    |
| <b>KATA PENGANTAR.....</b>                              | <b>vi</b>   |
| <b>DAFTAR ISI.....</b>                                  | <b>viii</b> |
| <b>DAFTAR GAMBAR.....</b>                               | <b>xi</b>   |
| <b>DAFTAR TABEL .....</b>                               | <b>xiv</b>  |
| <b>ABSTRACT .....</b>                                   | <b>xv</b>   |
| <b>ABSTRAK .....</b>                                    | <b>xvi</b>  |
| <b>BAB I PENDAHULUAN.....</b>                           | <b>1</b>    |
| 1.1.    Latar Belakang .....                            | 1           |
| 1.2.    Perumusan Masalah .....                         | 3           |
| 1.3.    Tujuan .....                                    | 3           |
| 1.4.    Batasan Masalah.....                            | 4           |
| 1.5.    Manfaat .....                                   | 4           |
| 1.6.    Keaslian Penelitian.....                        | 5           |
| 1.7.    Sistematika Penulisan .....                     | 7           |
| <b>BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....</b>  | <b>9</b>    |
| 2.1.    Tinjauan Pustaka .....                          | 9           |
| 2.2.    Landasan Teori.....                             | 11          |
| 2.2.1.    Pompa Air .....                               | 11          |
| 2.2.2.    Banjir di Jakarta .....                       | 13          |
| 2.2.3.    Stasiun Pompa.....                            | 14          |
| 2.2.3.1.    Stasiun Pompa Cideng .....                  | 16          |
| 2.2.4.    Logika Fuzzy.....                             | 17          |

|                     |   |           |
|---------------------|---|-----------|
| 2.2.4.1             | Logika Fuzzy Metode Sugeno .....                      | 20        |
| 2.2.5.              | <i>Mean Squared Error</i> .....                       | 23        |
| 2.2.6.              | Arduino Mega 2560 .....                               | 24        |
| 2.2.7.              | Sensor Ultrasonik (HC-SR04) .....                     | 25        |
| 2.2.8.              | Sensor Curah Hujan ( <i>Tipping Bucket</i> ) .....    | 27        |
| 2.2.9.              | Sensor <i>Water Flow</i> YF-S201 .....                | 29        |
| 2.2.10.             | Pompa Air DC <i>Submersible</i> .....                 | 30        |
| 2.2.11.             | Modul <i>Relay-2 Channel</i> .....                    | 31        |
| 2.2.12.             | <i>Real Time Clock</i> (RTC) DS3231 .....             | 32        |
| 2.2.13.             | NodeMCU ESP32 .....                                   | 33        |
| 2.2.14.             | Matlab .....  | 35        |
| 2.2.15.             | <i>Internet of Things</i> (IoT).....                  | 36        |
| 2.2.16.             | <i>Plafform</i> Blynk.....                            | 37        |
| <b>BAB III</b>      | <b>METODE PENELITIAN</b> .....                        | <b>39</b> |
| 3.1.                | Desain Penelitian.....                                | 39        |
| 3.2.                | Model Penelitian .....                                | 41        |
| 3.3.                | Objek Penelitian .....                                | 43        |
| 3.4.                | Pemodelan Kendali Pompa Air dengan Logika Fuzzy ..... | 44        |
| 3.5.1.              | Fuzzyifikasi .....                                    | 44        |
| 3.5.2.              | <i>Fuzzy Rule Base</i> .....                          | 50        |
| 3.5.3.              | Defuzzifikasi .....                                   | 52        |
| 3.5.                | Rencana Operasi dan Rotasi Pompa .....                | 54        |
| 3.5.1.              | Mekanisme Rotasi Harian .....                         | 54        |
| 3.5.2.              | Implementasi Logika Rotasi Pada Mikrokontroler .....  | 57        |
| 3.6.                | Pemodelan dengan Integrasi IoT .....                  | 58        |
| <b>BAB IV</b> ..... | <b>65</b>   |           |

|                       |   |            |
|-----------------------|---|------------|
| 4.1.                  | Pengujian Fungsionalitas Komponen.....              | 65         |
| 4.1.1.                | Pengujian Sensor.....                               | 65         |
| 4.1.2.                | Pengujian Mikrokontroler .....                      | 71         |
| 4.1.3.                | Pengujian Aktuator.....                             | 72         |
| 4.2.                  | Analisis Sistem Kontrol Logika Fuzzy .....          | 73         |
| 4.2.1.                | Fuzzifikasi .....                                   | 73         |
| 4.2.2.                | <i>Fuzzy Rule Base</i> .....                        | 75         |
| 4.2.3.                | Defuzzifikasi .....                                 | 82         |
| 4.3.                  | Pengujian Sistem Terintegrasi.....                  | 83         |
| 4.3.1.                | Studi Kasus .....                                   | 84         |
| 4.3.2.                | Pengujian Fungsionalitas Rotasi Pompa .....         | 87         |
| 4.4.                  | Analisis Integrasi dan Monitoring IoT (Blynk) ..... | 89         |
| 4.5.                  | Analisis Perbandingan Kinerja.....                  | 91         |
| 4.6.                  | Pembahasan Hasil .....                              | 92         |
| <b>BAB V</b>          | .....   | <b>94</b>  |
| 5.1.                  | Kesimpulan .....                                    | 94         |
| 5.2.                  | Saran.....  | 94         |
| <b>DAFTAR PUSTAKA</b> | .....   | <b>96</b>  |
| <b>LAMPIRAN</b>       | .....   | <b>101</b> |

## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 2.1 Pompa Air Stasioner .....                                       | 12 |
| Gambar 2.2 Skema Pompa Aliran Aksial .....                                 | 13 |
| Gambar 2.3 Kondisi Banjir di Bundaran Hotel Indonesia, Jakarta.....        | 14 |
| Gambar 2.4 Rumah Pompa Polder Kamal .....                                  | 15 |
| Gambar 2.5 Stasiun Pompa Cideng .....                                      | 17 |
| Gambar 2.6 Diagram Blok Sistem Kontrol Logika Fuzzy.....                   | 18 |
| Gambar 2.7 Bentuk Fungsi Keanggotaan Segitiga .....                        | 19 |
| Gambar 2.8 Bentuk Fungsi Keanggotaan Trapesium .....                       | 20 |
| Gambar 2.9 Proses Inferensi Logika Fuzzy .....                             | 22 |
| Gambar 2.10 Arduino Mega 2560 .....  | 24 |
| Gambar 2.11 Sensor Ultrasonik HC-SR04 .....                                | 26 |
| Gambar 2.12 Sensor Tipping Bucket.....                                     | 28 |
| Gambar 2.13 Sensor Water Flow YF-S201 .....                                | 29 |
| Gambar 2.14 Pompa Air DC Mini Tipe Submersible.....                        | 30 |
| Gambar 2.15 Modul Relay 2-Channel .....                                    | 32 |
| Gambar 2.16 Modul Real-Time Clock RTC DS3231.....                          | 33 |
| Gambar 2.17 NodeMCU ESP32 .....  | 34 |
| Gambar 2.18 Antarmuka Fuzzy Logic Designer pada Matlab .....               | 36 |
| Gambar 2.19 Arsitektur Umum Internet of Things.....                        | 37 |
| Gambar 2.20 Contoh Dasbor Aplikasi pada Platform Blynk.....                | 38 |
| <br>   |    |
| Gambar 3.1 Flowchart Penelitian.....                                       | 40 |
| Gambar 3.2 Diagram Blok Model Penelitian.....                              | 41 |
| Gambar 3.3 Diagram Rangkaian Keseluruhan Alat.....                         | 42 |
| Gambar 3.4 Lokasi Stasiun Pompa Cideng Jakarta .....                       | 43 |
| Gambar 3.5 Rumah Pompa Cideng Jakarta .....                                | 43 |
| Gambar 3.6 Tampak Dalam Rumah Pompa Cideng Jakarta.....                    | 44 |
| Gambar 3.7 Parameter Input-Output Logika Fuzzy/Fuzzy Inference System..... | 44 |
| Gambar 3.8 Kurva Input Curah Hujan .....                                   | 45 |
| Gambar 3.9 Variabel Fungsi Keanggotaan Curah Hujan dengan Irisan.....      | 45 |

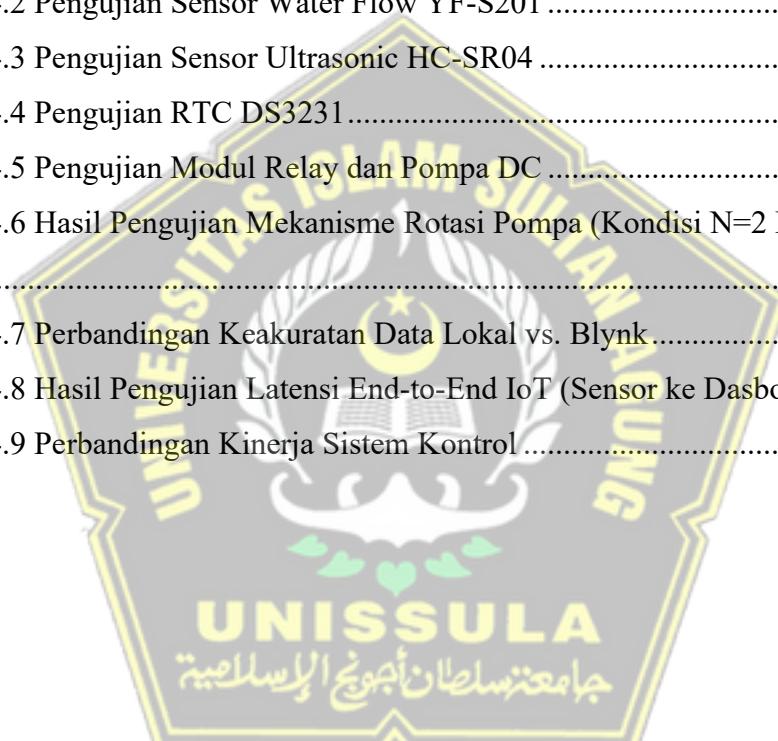
|  |    |
|--|----|
| Gambar 3.10 Kurva Input Debit Sungai .....   | 46 |
| Gambar 3.11 Variabel Fungsi Keanggotaan Debit Sungai dengan Irisan .....           | 47 |
| Gambar 3.12 Kurva Input Tinggi Muka Air .....                                      | 48 |
| Gambar 3.13 Variabel Fungsi Keanggotaan Tinggi Muka Air dengan Irisan .....        | 48 |
| Gambar 3.14 Variabel Output Pompa 1 sampai dengan Pompa 6 .....                    | 50 |
| Gambar 3.15 Fuzzy Rule Base pada Mode Rule Viewer di Matlab .....                  | 52 |
| Gambar 3.16 Diagram Alir Implementasi Sistem Operasi Pompa .....                   | 57 |
| Gambar 3.17 Penambahan Widget pada Dasbor Proyek Blynk di Website .....            | 60 |
| Gambar 3.18 Penambahan Widget pada Dasbor Proyek Blynk di Website .....            | 60 |
| Gambar 3.19 Konfigurasi Pin Virtual untuk Widget pada Website.....                 | 61 |
| Gambar 3.20 Konfigurasi Pin Virtual untuk Widget pada Aplikasi .....               | 61 |
| <br>   |    |
| Gambar 4.1 Pengujian Sensor Tipping Bucket.....                                    | 65 |
| Gambar 4.2 Pengujian Sensor Water Flow YF-S201 .....                               | 67 |
| Gambar 4.3 Pengujian Sensor <i>Ultrasonic</i> HC-SR04 .....                        | 68 |
| Gambar 4.4 Pengujian Timer Waktu Sensor RTC DS3231.....                            | 70 |
| Gambar 4.5 Done-Compiling Program pada Aplikasi Arduino IDE.....                   | 71 |
| Gambar 4.6 Serial Monitor Pengujian Arduino Mega 2560 .....                        | 71 |
| Gambar 4.7 Pengujian NodeMCU ESP32 .....   | 72 |
| Gambar 4.8 Pengujian Modul Relay dan Pompa DC .....                                | 72 |
| Gambar 4.9 Fuzzifikasi Curah Hujan Rendah .....                                    | 74 |
| Gambar 4.10 Fuzzifikasi Debit Sungai Sedang .....                                  | 74 |
| Gambar 4.11 Fuzzifikasi Tinggi Muka Air Tinggi .....                               | 75 |
| Gambar 4.12 Fuzzifikasi Output Simulasi Perhitungan.....                           | 75 |
| Gambar 4.13 Wujud Prototipe Sistem Kontrol Pompa Air Terintegrasi .....            | 83 |
| Gambar 4.14 Tampilan LCD pada Kondisi Normal (Tidak Ada Pompa Aktif)....           | 84 |
| Gambar 4.15 Tampilan Dasbor Blynk pada Kondisi Normal (Tidak Ada Pompa Aktif)..... | 85 |
| Gambar 4.16 Tampilan LCD pada Kondisi Waspada (2 Pompa Aktif) .....                | 85 |
| Gambar 4.17 Tampilan Dasbor Blynk pada Kondisi Waspada (2 Pompa Aktif) ..          | 85 |
| Gambar 4.18 Tampilan LCD pada Kondisi Siaga (4 Pompa Aktif) .....                  | 86 |
| Gambar 4.19 Tampilan Dasbor Blynk pada Kondisi Siaga (4 Pompa Aktif).....          | 86 |

|  |    |
|--|----|
| Gambar 4.20 Tampilan LCD pada Kondisi Bahaya (6 Pompa Aktif) .....                   | 87 |
| Gambar 4.21 Tampilan Dasbor Blynk pada Kondisi Bahaya (6 Pompa Aktif).....           | 87 |
| Gambar 4.22 Tampilan Aplikasi Blynk pada Hari Kamis (Pompa 2 dan Pompa 4<br>ON)..... | 88 |
| Gambar 4.23 Tampilan Aplikasi Blynk pada Hari Jumat (Pompa 3 dan Pompa 5<br>ON)..... | 88 |
| Gambar 4.24 Tampilan Aplikasi Blynk pada Hari Sabtu (Pompa 1 dan Pompa 6<br>ON)..... | 89 |



## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 3.1 Fuzzy Rule Base.....  | 50 |
| Tabel 3.2 Kategori Status Siaga Banjir .....  | 51 |
| Tabel 3.3 Jadwal Rotasi Pompa Harian .....  | 54 |
| Tabel 3.4 Frekuensi Akumulatif Pompa .....  | 56 |
| Tabel 3.12 Konfigurasi Mapping Pin Virtual Blynk.....                               | 61 |
| <br>  |    |
| Tabel 4.1 Pengujian Sensor Tipping Bucket.....                                      | 66 |
| Tabel 4.2 Pengujian Sensor Water Flow YF-S201 .....                                 | 68 |
| Tabel 4.3 Pengujian Sensor Ultrasonic HC-SR04 .....                                 | 69 |
| Tabel 4.4 Pengujian RTC DS3231.....   | 70 |
| Tabel 4.5 Pengujian Modul Relay dan Pompa DC .....                                  | 73 |
| Tabel 4.6 Hasil Pengujian Mekanisme Rotasi Pompa (Kondisi N=2 Pompa Aktif)<br>..... | 88 |
| Tabel 4.7 Perbandingan Keakuratan Data Lokal vs. Blynk .....                        | 90 |
| Tabel 4.8 Hasil Pengujian Latensi End-to-End IoT (Sensor ke Dasbor Blynk) ....      | 90 |
| Tabel 4.9 Perbandingan Kinerja Sistem Kontrol .....                                 | 92 |



## ***ABSTRACT***

*Conventional water pump stations in DKI Jakarta, such as the Cideng Pump Station, are often inefficient as they rely on reactive ON/OFF control systems based on a single sensor and lack pump load management. This research aims to design and implement an adaptive, anticipatory, and reliable automatic control system prototype for six water pump units. The method used is System Design integrating three main technologies: Fuzzy Logic, Internet of Things (IoT), and pump rotation management. This system's architecture utilizes a dual-microcontroller strategy, where an Arduino Mega 2560 serves as the main control unit (running Fuzzy Logic and rotation) and a NodeMCU ESP32 serves as the IoT gateway. Sugeno Fuzzy Logic with 27 rules is used to map three input variables (Rainfall, River Flow, Water Level) into a distinct singleton output level representing the number of active pumps (from 0 to 6). For long-term reliability, a 7-day pump rotation mechanism based on an RTC DS3231 is implemented. Laboratory-scale prototype testing showed the system successfully responded proportionally to four test scenarios (Normal, Waspada, Siaga, Bahaya), activating the corresponding number of pumps (0, 2, 4, and 6 pumps) for each tested scenario. The 7-day rotation analysis demonstrated balanced workload distribution with a maximum difference of only 1 ON cycle per week. IoT integration via the Blynk platform successfully displayed real-time sensor data and pump status with an average latency of 1.8 seconds. It is concluded that the proposed system is significantly superior to conventional methods as it is adaptive, anticipatory, proportional, and more reliable.*

**Keywords:** *Fuzzy Logic, Flood Mitigation, Internet of Things, Pump Rotation, Blynk.*

## ABSTRAK

Stasiun pompa air konvensional di DKI Jakarta, seperti Stasiun Pompa Cideng, seringkali tidak efisien karena mengandalkan sistem kontrol *ON/OFF* yang reaktif berbasis satu sensor dan tidak memiliki manajemen beban pompa. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sebuah prototipe sistem kontrol otomatis yang adaptif, antisipatif, dan andal untuk enam unit pompa air. Metode yang digunakan adalah perancangan sistem yang mengintegrasikan tiga teknologi utama: Logika Fuzzy, *Internet of Things* (IoT), dan manajemen rotasi pompa. Arsitektur sistem ini menggunakan strategi *dual-microcontroller*, di mana Arduino Mega 2560 berfungsi sebagai unit kontrol utama (menjalankan logika fuzzy dan rotasi) dan NodeMCU ESP32 berfungsi sebagai *gateway* IoT. Logika Fuzzy Sugeno dengan 27 aturan digunakan untuk memetakan tiga variabel masukan (curah hujan, debit sungai, tinggi muka air) menjadi sebuah nilai output tegas (*singleton*) yang merepresentasikan jumlah pompa aktif (dari 0 hingga 6). Untuk keandalan jangka panjang, diimplementasikan mekanisme rotasi pompa 7 hari berbasis RTC DS3231. Hasil pengujian prototipe skala laboratorium menunjukkan sistem berhasil merespons empat skenario pengujian (Normal, Waspada, Siaga, Bahaya) secara proporsional, dengan mengaktifkan jumlah pompa (0, 2, 4, dan 6 pompa) sesuai skenario yang diuji. Analisis rotasi 7 hari membuktikan pemerataan beban kerja pompa dengan selisih maksimum hanya 1 siklus pompa aktif per minggu. Integrasi IoT melalui *platform* Blynk berhasil menampilkan data sensor dan status pompa secara *real-time* dengan latensi rata-rata 1,8 detik. Disimpulkan bahwa sistem yang diusulkan terbukti unggul signifikan dibandingkan metode konvensional karena bersifat adaptif, antisipatif, proporsional, dan lebih andal.

**Kata Kunci:** Logika Fuzzy, Mitigasi Banjir, *Internet of Things*, Rotasi Pompa, Blynk.

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Pengendalian banjir merupakan salah satu tantangan krusial yang dihadapi oleh kota-kota besar di seluruh dunia, tidak terkecuali DKI Jakarta. Sebagai ibu kota negara dengan kepadatan penduduk yang tinggi dan topografi yang relatif datar, Jakarta sangat rentan terhadap genangan dan banjir, terutama saat musim hujan dengan intensitas curah hujan yang tinggi [1]. Untuk mengatasi masalah ini, pemerintah telah membangun berbagai infrastruktur pengendali banjir, di antaranya adalah stasiun pompa air yang berfungsi untuk membuang kelebihan air dari daerah yang lebih rendah ke saluran pembuangan utama atau laut, sebagai bagian dari strategi mitigasi struktural [2].

Stasiun Pompa Cideng adalah salah satu komponen vital dalam jaringan sistem drainase makro Jakarta yang perannya sangat penting untuk melindungi area pusat kota dari ancaman banjir [3]. Secara topografis, area layanan pompa ini merupakan kawasan sistem polder dengan elevasi muka tanah yang berada di bawah muka air laut pasang atau saluran pembuangan utama. Kondisi ini menyebabkan air tidak dapat mengalir secara gravitasi, sehingga operasional pompa menjadi kebutuhan absolut dan mendesak; kegagalan fungsi pompa di titik ini dapat berakibat fatal berupa genangan langsung di area vital pemerintahan dan ekonomi [4].

Namun, permasalahan utama yang sering dijumpai pada sistem kontrol pompa air konvensional di banyak stasiun pompa, termasuk di Stasiun Pompa Cideng, adalah kurangnya efisiensi dan kecepatan respons terhadap perubahan kondisi hidrologis yang dinamis. Sistem yang ada saat ini seringkali masih mengandalkan operasi manual atau semi-otomatis berbasis saklar *ON/OFF* sederhana yang bergantung pada satu variabel, yaitu ketinggian muka air [5]. Selain itu, mekanisme penyalaan pompa yang ada saat ini cenderung statis dan tidak memiliki sistem manajemen pergantian (rotasi) prioritas pompa. Hal ini menyebabkan beban kerja tidak terdistribusi merata, di mana satu unit pompa tertentu bekerja terus-menerus sementara unit lain jarang digunakan, yang berpotensi mempercepat keausan komponen dan memperpendek umur teknis peralatan [6].

Ketergantungan pada operator manusia dalam sistem yang kompleks membuka peluang terjadinya kesalahan akibat faktor manusia (*human error*), yang dapat mempengaruhi keandalan dan waktu respons sistem [7]. Selain itu, sistem kontrol konvensional tidak mampu beradaptasi dengan variabel-variabel noneksak seperti laju kenaikan air atau prediksi intensitas hujan, sehingga operasi pompa menjadi kurang optimal [8].

Akibat dari sistem kontrol yang kurang adaptif ini adalah timbulnya berbagai inefisiensi. Sebagai contoh, ketika curah hujan sangat deras dan debit air meningkat secara drastis, keterlambatan dalam mengaktifkan pompa dapat menyebabkan air meluap dan menimbulkan genangan di area sekitar. Sebaliknya, pengoperasian pompa pada kapasitas penuh secara terus-menerus saat debit air masuk tidak terlalu tinggi akan menyebabkan pemborosan energi listrik yang signifikan [9]. Sebab akibat ini menunjukkan bahwa sistem kontrol yang kaku dan tidak cerdas menjadi akar masalah dari pemborosan sumber daya dan peningkatan risiko kegagalan dalam manajemen banjir.

Untuk mengatasi kelemahan tersebut, diperlukan sebuah sistem kontrol yang cerdas, adaptif, dan otomatis. Logika fuzzy (*fuzzy logic*) menawarkan solusi yang sangat relevan untuk masalah ini, karena kemampuannya dalam menangani ketidakpastian dan non-linearitas yang melekat pada sistem dunia nyata, menjadikannya sangat cocok untuk aplikasi kontrol industri [10]. Logika fuzzy mampu mengolah data masukan yang bersifat tidak pasti atau ambigu (misalnya, level air “agak tinggi” atau curah hujan “cukup deras”) dan menerjemahkannya ke dalam keputusan kontrol yang presisi. Dengan menggunakan beberapa variabel masukan seperti ketinggian muka air dan laju perubahan ketinggian air, sistem kontrol berbasis logika fuzzy dapat menentukan kecepatan putaran pompa atau jumlah pompa yang harus aktif secara lebih optimal dan proporsional [11].

Selanjutnya, untuk meningkatkan keandalan dan kapabilitas *monitoring*, integrasi sistem dengan teknologi *Internet of Things* (IoT) menjadi sebuah keniscayaan [12]. Pemanfaatan sensor-sensor yang terhubung melalui jaringan IoT memungkinkan pengumpulan data secara real-time dan akurat dari lapangan. Data ini kemudian dapat dikirim ke ruang kontrol pusat atau bahkan dapat diakses dari jarak jauh melalui dasbor berbasis web atau aplikasi seluler [13]. Integrasi IoT tidak

hanya memfasilitasi automasi penuh, tetapi juga menyediakan data historis yang berharga untuk analisis dan pemeliharaan prediktif, serta memungkinkan operator untuk melakukan intervensi jarak jauh jika diperlukan dalam kondisi darurat [14].

Oleh karena itu, penelitian ini bertujuan untuk merancang sebuah sistem kontrol pompa air otomatis yang mengintegrasikan keunggulan logika fuzzy sebagai otak pengambilan keputusan dengan teknologi IoT sebagai sistem saraf sensorik dan komunikasinya. Penerapan sistem ini di Stasiun Pompa Cideng diharapkan dapat menjadi model percontohan yang mampu meningkatkan efektivitas dan efisiensi pengendalian banjir secara signifikan, mengurangi ketergantungan pada intervensi manual, menghemat konsumsi energi, dan pada akhirnya memberikan kontribusi nyata bagi sistem mitigasi banjir di DKI Jakarta.

### **1.2. Perumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, maka dapat diambil rumusan masalah yaitu:

1. Bagaimana merancang arsitektur sistem kontrol otomatis untuk mengelola enam pompa air secara efisien menggunakan tiga variabel masukan dinamis: curah hujan, debit sungai, dan ketinggian muka air?
2. Bagaimana mengembangkan aturan logika fuzzy yang mampu menerjemahkan kombinasi ketiga variabel tersebut menjadi keputusan kontrol *ON/OFF* yang optimal dan adaptif?
3. Bagaimana mengintegrasikan mikrokontroler Arduino dengan *platform* IoT (Blynk) agar status pompa dan data sensor dapat dimonitor secara *real-time*?
4. Bagaimana merancang mekanisme penjadwalan menggunakan RTC untuk mengatur rotasi kerja keenam pompa agar dapat beroperasi secara bergantian?

### **1.3. Tujuan**

Tujuan dari penelitian ini adalah:

1. Merancang dan membangun prototipe sistem kontrol pompa otomatis berbasis logika fuzzy yang terintegrasi IoT pada Stasiun Pompa Cideng, Jakarta.
2. Menyusun arsitektur perangkat keras dan lunak yang mencakup sensor curah hujan, debit sungai, ketinggian muka air, mikrokontroler, dan pompa.

3. Mengimplementasikan algoritma logika fuzzy untuk menghasilkan sinyal kontrol *ON/OFF* yang logis bagi enam pompa.
4. Mengembangkan sistem pemantauan jarak jauh melalui *platform* IoT Blynk untuk menampilkan data sensor dan status operasional secara *real-time*.
5. Menerapkan algoritma penjadwalan berbasis waktu (RTC) untuk mengelola distribusi pengaktifan pompa secara otomatis sesuai siklus hari.

#### **1.4. Batasan Masalah**

Agar penelitian ini lebih terarah dan fokus pada tujuan yang ingin dicapai, maka ruang lingkup permasalahan dibatasi sebagai berikut:

1. Penelitian ini berfokus pada perancangan dan pengujian dalam skala laboratorium (prototipe). Karakteristik dan parameter sistem dirancang berdasarkan studi kasus Stasiun Pompa Cideng, namun tidak diimplementasikan secara langsung di lokasi.
2. Variabel masukan untuk kontroler logika fuzzy terbatas pada tiga jenis, yaitu curah hujan, debit sungai, dan ketinggian muka air.
3. Keluaran sistem kontrol terbatas pada sinyal *ON/OFF* untuk enam buah pompa. Penelitian ini tidak membahas kontrol kecepatan variabel (*Variable Speed Drive*).
4. Platform IoT yang digunakan dikhkususkan pada Blynk. Penelitian tidak melakukan perbandingan atau implementasi pada platform IoT lainnya.
5. Perangkat keras utama yang digunakan adalah mikrokontroler Arduino, ESP32 dan sensor-sensor yang telah disebutkan.
6. Penelitian tidak membahas secara mendalam aspek keamanan siber dari transmisi data pada sistem IoT yang dibangun.

#### **1.5. Manfaat**

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Memberikan kontribusi pada pengembangan ilmu di bidang sistem kontrol cerdas, khususnya penerapan logika fuzzy pada sistem *multi-input* dan *multi-output*.

2. Menjadi studi kasus dan referensi ilmiah mengenai implementasi teknologi IoT, khususnya *platform* Blynk, untuk pemantauan dan manajemen infrastruktur pengendali banjir.
3. Menambah khazanah literatur penelitian terapan di bidang rekayasa sumber daya air dan mitigasi bencana perkotaan.
4. Menghasilkan sebuah model prototipe yang dapat menjadi acuan bagi Dinas Sumber Daya Air atau pengelola Stasiun Pompa Cideng untuk modernisasi sistem kontrol.
5. Meningkatkan efisiensi operasional dengan mengoptimalkan jumlah pompa yang aktif sesuai dengan beban air aktual, yang berpotensi menghemat konsumsi energi listrik.
6. Menyediakan sistem pemantauan terpusat yang mudah diakses, memungkinkan operator untuk mengambil keputusan yang lebih cepat dan tepat berdasarkan data yang akurat dan real-time.

### **1.6. Keaslian Penelitian**

Untuk menunjukkan orisinalitas dan posisi penelitian ini di antara penelitian-penelitian relevan yang telah ada sebelumnya, berikut adalah perbandingan dengan beberapa penelitian terkini yang memiliki topik serupa yaitu:

- A. Penelitian yang dilakukan oleh Riza Moh Nujumul Huda (2024) berjudul “Kajian Sistem Kendali Pompa Air Dengan Metode Logika Fuzzy Sebagai Upaya Memaksimalkan Pengendalian Banjir Kawasan Pelindo III Kota Semarang”. Penelitian ini berfokus pada perancangan sistem kontrol pompa air untuk mitigasi banjir yang spesifik terjadi di area Pelabuhan Pelindo III, Semarang. Sistem yang dirancang menggunakan tiga variabel masukan, yaitu curah hujan, debit air, dan ketinggian rob, yang merupakan faktor krusial di wilayah pesisir. Logika Fuzzy diimplementasikan untuk mengolah ketiga masukan tersebut dan menghasilkan keputusan kontrol untuk dua buah pompa air. Perangkat keras utama yang digunakan meliputi mikrokontroler Arduino Uno, sensor ultrasonik untuk mengukur ketinggian air, dan sensor water level sebagai pendukung. Proses perancangan dan simulasi aturan fuzzy dibantu oleh perangkat lunak Matlab. Output dari sistem ini adalah keputusan biner, yaitu mengaktifkan atau menonaktifkan (*ON/OFF*) Pompa 1 dan Pompa 2 secara

- terpisah atau bersamaan. Parameter Logika Fuzzy yang digunakan membagi setiap variabel masukan (tinggi air, curah hujan, dan rob) ke dalam tiga himpunan keanggotaan: rendah, sedang, dan tinggi. Penelitian ini memberikan kontribusi penting dalam penerapan Logika Fuzzy untuk penanganan banjir di lingkungan pelabuhan yang dipengaruhi oleh pasang surut air laut (rob) [15].
- B. Penelitian oleh Achmad Syamsuddin (2024) dengan judul “Kajian Sistem Kendali Pintu Air Berbasis Logika Fuzzy Pada Bendung Gerak Kanal Banjir Barat Kota Semarang”. Berbeda dengan penelitian sebelumnya yang berfokus pada pompa, objek kontrol pada penelitian ini adalah pintu air. Tujuan utamanya adalah mengatur bukaan pintu air secara otomatis untuk mengendalikan aliran di Kanal Banjir Barat, Semarang. Sistem ini menggunakan tiga variabel masukan: tinggi muka air di sisi hulu, tinggi muka air laut (rob), dan intensitas curah hujan. Sensor yang digunakan adalah dua sensor ultrasonik dan satu sensor curah hujan. Pengambilan keputusan dilakukan oleh *Fuzzy Logic Controller* yang ditanamkan pada mikrokontroler Arduino Uno. *Output* dari sistem ini bukanlah kontrol *ON/OFF*, melainkan kontrol proporsional yang mengatur sudut bukaan motor servo sebagai penggerak pintu air. Parameter Logika Fuzzy yang ditetapkan juga membagi variabel masukan (tinggi air, curah hujan, rob) menjadi tiga kategori: rendah, sedang, dan tinggi. Sementara itu, variabel *output* (bukaan pintu air) dibagi menjadi tiga level: tutup, sedang, dan buka lebar. Penelitian ini menunjukkan keunggulan logika fuzzy untuk sistem kontrol kontinu pada infrastruktur pengendali banjir [16].

Penelitian yang akan dilakukan ini berjudul “Perancangan Sistem Kontrol Otomatis Pompa Air Berbasis Logika Fuzzy Dengan Integrasi IoT” dengan studi kasus pada Stasiun Pompa Cideng Jakarta. Perbedaan mendasar dan kebaruan (*novelty*) penelitian ini dibandingkan kedua penelitian sebelumnya terletak pada beberapa aspek kunci. Pertama, lokasi dan skala objek penelitian yang diambil adalah Stasiun Pompa Cideng di Jakarta, sebuah infrastruktur vital berskala besar dengan karakteristik hidrologi yang berbeda dari area pelabuhan atau kanal banjir di Semarang. Kedua, kombinasi variabel *input* yang digunakan lebih spesifik untuk operasi pompa di daratan, yaitu tinggi muka air, debit air yang masuk (diukur

dengan *water flow sensor*), dan curah hujan. Penggunaan sensor debit air secara langsung memberikan data yang lebih akurat mengenai beban aktual yang harus ditangani pompa dibandingkan hanya mengandalkan ketinggian air.

Ketiga, dan yang menjadi kebaruan utama, adalah adanya integrasi sistem kontrol dengan platform *Internet of Things* (IoT) menggunakan Blynk. Hal ini memungkinkan seluruh data operasional, seperti status keenam pompa, level air, dan debit, dapat dipantau secara *real-time* dari jarak jauh melalui dasbor pada website atau aplikasi smartphone. Fitur ini tidak ada pada kedua penelitian sebelumnya dan merupakan sebuah lompatan menuju sistem manajemen infrastruktur cerdas (*smart infrastructure*). Integrasi IoT tidak hanya berfungsi untuk *monitoring*, tetapi juga untuk pengumpulan data historis yang dapat digunakan untuk analisis kinerja dan pemeliharaan prediktif. Terakhir, sistem ini dirancang untuk mengendalikan enam buah pompa air, yang menghadirkan tantangan perancangan aturan fuzzy yang lebih kompleks untuk menentukan prioritas dan kombinasi pompa yang harus aktif demi mencapai efisiensi energi yang maksimal.

### **1.7. Sistematika Penulisan**

Laporan tesis ini terdiri dari lima bab, dengan sistematika penulisan sebagai berikut:

## **BAB I PENDAHULUAN**

Bab ini berisi latar belakang permasalahan, perumusan masalah, tujuan penelitian, manfaat penelitian, keaslian penelitian, serta sistematika penulisan tesis secara keseluruhan.

## **BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI**

Bab ini membahas literatur dan penelitian-penelitian terdahulu yang relevan, serta teori-teori dasar yang mendukung perancangan sistem kontrol otomatis pompa air. Pembahasan mencakup teori sistem kontrol, dasar-dasar logika fuzzy, prinsip kerja pompa air, serta karakteristik dan kondisi Stasiun Pompa Cideng Jakarta.

## **BAB III METODOLOGI PENELITIAN**

Bab ini menjelaskan metode penelitian yang digunakan, tahapan perancangan sistem kontrol logika fuzzy, pemilihan variabel *input* dan *output*, perancangan *rule base*, perancangan sistem IoT, serta prosedur pengujian sistem. Selain itu, dijelaskan pula lokasi penelitian, alat dan bahan yang digunakan, serta diagram alir penelitian secara rinci.

#### **BAB IV HASIL PENELITIAN DAN PEMBAHASAN**

Bab ini berisi hasil perancangan sistem kontrol otomatis pompa air berbasis logika fuzzy serta analisis kinerjanya. Dibahas pula perbandingan performa antara sistem kontrol fuzzy dengan sistem kontrol konvensional, analisis respon sistem, serta dampaknya terhadap efektivitas pengendalian banjir di wilayah sekitar Stasiun Pompa Cideng.

#### **BAB V KESIMPULAN DAN SARAN**

Bab ini bersikan tentang kesimpulan dan saran dari penelitian yang telah dilakukan.



## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1. Tinjauan Pustaka

Penelitian terdahulu merupakan referensi untuk mengembangkan penelitian ini. Salah satu literatur yang mudah untuk ditemui mengenai topik yang diangkat pada penelitian ini berupa artikel dari jurnal penelitian antara lain:

Jurnal pertama ditulis oleh Baginda Olon Siregar (2024) yang meneliti tentang perancangan sistem pengendalian pompa air untuk mengatur ketinggian level air. Metode yang digunakan dalam penelitian ini adalah logika fuzzy sebagai pengganti saklar ON/OFF konvensional untuk mengatasi ketidaklinieran perubahan level air. Penulis menggunakan sensor ultrasonik sebagai masukan utama, di mana hasil penelitian menunjukkan bahwa penggunaan logika fuzzy mampu menciptakan kontrol level air yang lebih stabil dan mengurangi overshoot dibandingkan metode konvensional [17].

Jurnal kedua ditulis oleh Amaldi Baiki (2025) dengan topik sistem kendali pompa otomatis untuk penyiraman tanaman bawang merah. Penelitian ini menerapkan metode logika fuzzy Sugeno untuk menentukan durasi penyiraman berdasarkan variabel input kelembaban tanah dan suhu udara. Hasil akhir memperlihatkan bahwa sistem mampu mengelola sumber daya air secara efisien sesuai dengan kebutuhan spesifik tanaman bawang merah, sehingga mencegah terjadinya pembusukan akar akibat penyiraman berlebih [18].

Jurnal ketiga ditulis oleh Yoga Alif (2021) yang berfokus pada sistem monitoring kondisi perangkat keras pompa melalui judul penelitian terkait rancang bangun alat *monitoring* suhu. Metode yang digunakan adalah logika fuzzy untuk mengolah data suhu pada lilitan motor pompa guna mendeteksi gejala *overheating*. Hasil penelitian ini membuktikan bahwa logika fuzzy tidak hanya efektif untuk kontrol proses air, tetapi juga dapat digunakan untuk memutus arus secara otomatis saat suhu kritis tercapai demi keperluan pemeliharaan (*maintenance*) [19].

Jurnal keempat ditulis oleh Fahmi Akbar Rafsanjani (2025) yang meneliti tentang rancang bangun pintu air otomatis. Penelitian ini menggunakan kontroler logika fuzzy metode Mamdani untuk memetakan kondisi ketinggian air menjadi

derajat bukaan servo yang presisi. Hasil penelitian berupa prototipe pintu air yang dapat bekerja secara otomatis merespons kondisi lingkungan, di mana pergerakan pintu terbukti lebih halus (*smooth*) dibandingkan kontrol logika tegas (crisp) [20].

Jurnal kelima ditulis oleh Ach Habib Arrahman (2019) yang membahas mengenai perancangan pengontrol pintu air berdasarkan perubahan ketinggian air. Metode yang diterapkan berfokus pada aturan fuzzy untuk menyeimbangkan debit air yang masuk dengan sudut bukaan pintu keluar. Hasil penelitian menunjukkan efektivitas sistem dalam mengatur bukaan pintu air secara proporsional, sehingga mampu meminimalisir risiko luapan air yang mendadak akibat keterlambatan respon mekanik [21].

Jurnal keenam ditulis oleh Indra Hermawan (2022) dengan topik sistem irigasi cerdas. Penelitian ini merancang sistem menggunakan *fuzzy rule-based* yang mengolah input dari sensor kekeruhan (*turbidity*) untuk otomatisasi pintu air sekaligus mendeteksi adanya endapan lumpur. Hasil yang didapat adalah sistem mampu mengontrol aliran air sekaligus memberikan keputusan apakah pintu harus dibuka penuh untuk pembilasan sedimen atau ditutup untuk irigasi normal [22].

Jurnal ketujuh ditulis oleh Arunesh Kumar Singh (2022) membahas tentang simulasi metode *Fuzzy Logic Controller* (FLC) untuk mengoptimalkan manajemen irigasi menghadapi gangguan eksternal. Hasil penelitian memperkuat temuan bahwa logika fuzzy memiliki respon waktu (*response time*) yang lebih baik dibandingkan kontroler PID dan menjadi standar referensi internasional untuk sistem irigasi cerdas [23].

Jurnal kedelapan ditulis oleh Sabaruddin Nduru (2022) yang meneliti tentang sistem peringatan dini banjir. Penelitian ini mengintegrasikan metode fuzzy untuk mengklasifikasikan level bahaya (Aman, Siaga, Bahaya) dengan teknologi *Internet of Things* (IoT) sebagai media transmisi. Hasil penelitian menunjukkan bahwa fuzzy dapat menentukan status siaga secara akurat, sementara IoT berfungsi mengirimkan notifikasi peringatan tersebut kepada aplikasi pesan instan warga secara nirkabel [24].

Jurnal kesembilan ditulis oleh Ahmad Kurnia (2022) yang membahas rancang bangun sistem peringatan dini pencegahan banjir akibat luapan sungai. Metode yang digunakan menggabungkan pembacaan sensor level air secara kontinu dengan

logika fuzzy untuk mengurangi kesalahan pembacaan (false alarm). Hasilnya adalah sebuah sistem yang mampu memberikan peringatan dini yang valid sebelum banjir terjadi berdasarkan analisis laju kenaikan debit sungai [25].

Jurnal kesepuluh ditulis oleh Rinaldi (2022) dengan topik otomatisasi pertanian presisi. Penelitian ini mengintegrasikan logika fuzzy dan IoT untuk mengatur takaran nutrisi pada sistem penyiraman dan pemupukan tanaman secara otomatis. Hasil penelitian menunjukkan adanya sinergi yang baik di mana fuzzy bertindak sebagai otak pengambil keputusan dosis, sedangkan IoT memungkinkan petani memantau riwayat pemupukan dan kondisi tanaman dari jarak jauh [26].

Jurnal kesebelas ditulis oleh Siswanto Nurhadiyono (2024) yang berfokus pada pengembangan sistem peringatan dini banjir berbasis mikrokontroler ESP8266 dan Firebase. Metode yang digunakan menekankan pada kecepatan transmisi data sensor ke database awan (*cloud*) dengan latensi rendah. Hasil penelitian menunjukkan bahwa sistem mampu mengirimkan notifikasi secara *real-time* dan andal kepada pengguna, memastikan informasi potensi banjir diterima tanpa penundaan yang berarti [27].

Berdasarkan tinjauan terhadap penelitian-penelitian di atas, terdapat perbedaan mendasar dengan penelitian yang akan dilakukan. Mayoritas penelitian sebelumnya cenderung terpisah, yaitu berfokus pada kontrol aktuator tunggal (hanya pompa atau pintu air) atau sebatas sistem peringatan dini (*early warning system*) yang bersifat pasif dan informatif. Penelitian ini hadir untuk melengkapi kekurangan tersebut dengan merancang sistem kontrol aktif untuk enam pompa sekaligus menggunakan tiga variabel masukan (curah hujan, debit, tinggi air) serta mengintegrasikannya secara penuh dengan platform IoT Blynk. Pendekatan ini menghasilkan sistem manajemen banjir yang lebih komprehensif karena mampu melakukan tindakan eksekusi (pompa) sekaligus *monitoring real-time* dalam satu arsitektur terpadu.

## 2.2. Landasan Teori

### 2.2.1. Pompa Air

Pompa adalah suatu mesin atau alat yang digunakan untuk memindahkan fluida (cairan atau gas) dari suatu tempat ke tempat lain melalui pipa dengan cara menambahkan energi pada fluida tersebut. Energi ini digunakan untuk mengatasi

hambatan seperti perbedaan tekanan, perbedaan ketinggian (*head statis*), atau gesekan dalam pipa (*head dinamis*). Secara fundamental, kerja pompa mengubah energi mekanis dari putaran poros menjadi energi kinetik dan energi tekanan pada fluida [28].

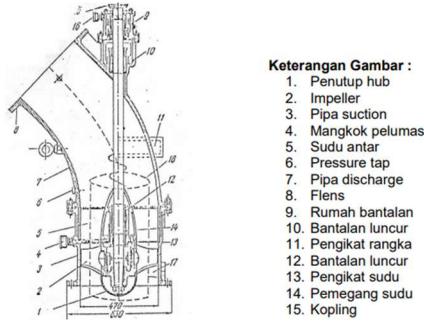
Pompa stasioner atau pompa tetap adalah jenis pompa yang dipasang secara permanen pada suatu lokasi atau fondasi untuk melayani kebutuhan spesifik, seperti pada sistem irigasi, drainase perkotaan, atau instalasi industri. Dalam konteks pengendalian banjir skala besar, jenis pompa stasioner yang paling efektif adalah pompa aliran aksial (*axial flow pump*). Prinsip kerja pompa aksial adalah mendorong fluida secara sejajar (aksial) dengan sumbu putaran impeler. Impeler pada pompa ini berbentuk seperti baling-baling (*propeler*) yang saat berputar akan memberikan gaya aksial pada fluida, mendorongnya ke arah *outlet*. Bentuk pompa air stasioner dapat dilihat pada Gambar 2.1.



Gambar 2.1 Pompa Air Stasioner

Sumber: Dokumentasi Dinas Sumber Daya Air DKI Jakarta (2025)

Karakteristik utama pompa aksial adalah kemampuannya menghasilkan debit aliran yang sangat tinggi (*high flow rate*) dengan ketinggian angkat atau *head* yang relatif rendah (*low head*). Karakteristik ini sangat sesuai untuk aplikasi penanganan banjir, di mana tujuannya adalah memindahkan volume air yang sangat besar dari sebuah polder atau kolam retensi ke sungai atau kanal yang elevasinya tidak jauh berbeda. Efisiensi pompa aksial akan maksimal pada kondisi *low head* dan akan menurun drastis jika dipaksa bekerja pada *head* yang tinggi [29]. Skema pompa air aliran aksial dapat dilihat pada Gambar 2.2.



Gambar 2.2 Skema Pompa Aliran Aksial

Sumber: Diadaptasi dari Rendi, dkk. (2023)

### 2.2.2. Banjir di Jakarta

Banjir di DKI Jakarta merupakan fenomena hidrometeorologi yang bersifat multidimensional, disebabkan oleh interaksi kompleks antara faktor alam dan aktivitas manusia (antropogenik). Pemahaman terhadap faktor-faktor penyebab banjir sangat penting dalam upaya mitigasi dan perencanaan tata kelola sumber daya air yang efektif. Secara geografis, sekitar 40% wilayah Jakarta merupakan dataran rendah dengan elevasi berada di bawah permukaan air laut saat pasang. Kondisi ini menjadikan Jakarta sangat rentan terhadap genangan, terutama ketika terjadi kombinasi antara curah hujan tinggi dan pasang air laut. Selain itu, wilayah ini dilalui oleh 13 sungai utama yang sebagian besar berhulu di daerah Bogor dan Puncak, kawasan dengan intensitas curah hujan yang tinggi. Ketika hujan deras terjadi di wilayah hulu, debit air sungai meningkat secara signifikan dan mengalir menuju hilir (Jakarta) [30].

Curah hujan sendiri merupakan salah satu faktor hidrometeorologis utama yang memicu terjadinya banjir. Secara definisi, curah hujan (H) adalah tinggi atau ketebalan lapisan air hujan yang terkumpul pada suatu bidang datar dalam periode waktu tertentu, apabila air tersebut tidak mengalami infiltrasi, penguapan, atau limpasan. Secara umum, curah hujan dapat dihitung menggunakan hubungan antara volume air hujan yang tertampung (V) dan luas bidang tangkapan (A) dengan rumus:

$$H = 10 \times \frac{V}{A} \quad (2.1)$$

Dimana:

$H$  : Curah hujan (mm)

V : Volume air (mL atau cm<sup>3</sup>)

A : Luas permukaan penampung (cm<sup>2</sup>)

10 : Faktor pengali

Faktor alam lainnya yang memperparah banjir di Jakarta meliputi pasang air laut (rob) yang dapat menghambat aliran sungai menuju Teluk Jakarta. Saat debit air sungai meningkat akibat curah hujan tinggi dan pada saat bersamaan terjadi pasang laut, air sungai tidak dapat mengalir ke laut dan menimbulkan efek backwater, yaitu air kembali meluap ke daratan. Kondisi ini sering menyebabkan genangan luas, terutama di wilayah Jakarta Utara dan Barat.

Sementara dari sisi antropogenik, penurunan muka tanah (*land subsidence*) menjadi faktor penting yang memperburuk dampak banjir. Penurunan ini terjadi akibat eksplorasi air tanah berlebihan dan proses pemasaran tanah, sehingga elevasi permukaan daratan menjadi semakin rendah dibandingkan muka air laut. Selain itu, alih fungsi lahan di daerah hulu dan perkotaan telah mengurangi area resapan air, meningkatkan limpasan permukaan, serta memperpendek waktu konsentrasi aliran hujan ke sungai. Sedimentasi dan penyempitan sungai akibat pembuangan sampah dan endapan tanah juga menurunkan kapasitas tumpang sungai, sehingga air lebih cepat meluap [30].

Salah satu contoh kejadian banjir besar di Jakarta pernah terjadi di kawasan Bundaran Hotel Indonesia, di mana kombinasi curah hujan tinggi, limpasan air dari hulu, dan sistem drainase yang tidak optimal menyebabkan genangan air yang meluas, dapat dilihat pada Gambar 2.3.



Gambar 2.3 Kondisi Banjir di Bundaran Hotel Indonesia, Jakarta

Sumber: Antara News (2020)

### 2.2.3. Stasiun Pompa

Stasiun pompa atau rumah pompa adalah sebuah bangunan infrastruktur pengendali banjir yang berfungsi sebagai benteng pertahanan terakhir untuk

melindungi sebuah kawasan (polder) dari genangan. Fungsi utamanya adalah memompa air dari area yang lebih rendah ke saluran drainase utama, sungai, atau laut yang memiliki elevasi muka air lebih tinggi. Rumah pompa menjadi komponen vital dalam sistem drainase perkotaan, terutama di wilayah dengan topografi datar atau berbentuk cekungan seperti Jakarta, di mana drainase gravitasi saja tidak cukup [31].

Sebuah stasiun pompa modern terdiri dari beberapa komponen utama yang bekerja secara terintegrasi. Di bagian depan, terdapat saringan sampah (*trash rack*), yaitu jeruji besi yang berfungsi menyaring sampah padat agar tidak masuk dan merusak impeler pompa. Air yang telah tersaring kemudian masuk ke ruang basah (*wet well*), sebuah bak penampung besar yang berfungsi untuk meredam fluktuasi aliran dan memastikan pompa selalu terendam air saat beroperasi untuk mencegah kavitas. Di dalam stasiun, terpasang unit-unit pompa stasioner (umumnya tipe aksial atau *mixed-flow*) yang menjadi jantung dari sistem.

Selain itu, terdapat panel kontrol listrik yang berisi sirkuit untuk mengoperasikan motor pompa, serta sistem kontrol (baik manual, semi-otomatis, maupun otomatis). Mengingat krusialnya fungsi stasiun pompa, ketersediaan daya listrik harus terjamin. Oleh karena itu, setiap stasiun pompa wajib dilengkapi dengan genset (*generator set*) berkapasitas besar sebagai sumber daya cadangan jika terjadi pemadaman listrik dari PLN. Pengoperasian rumah pompa yang efektif dan efisien sangat bergantung pada kecepatan dan ketepatan sistem kontrolnya dalam merespons perubahan level air di *wet well*. Bentuk rumah pompa dapat dilihat pada Gambar 2.4.



Gambar 2.4 Rumah Pompa Polder Kamal

Sumber: Dokumentasi Dinas Sumber Daya Air DKI Jakarta (2024)

### 2.2.3.1. Stasiun Pompa Cideng

Stasiun Pompa Cideng merupakan salah satu infrastruktur pengendali banjir vital yang berlokasi di wilayah Jakarta Pusat, yang dirancang sebagai bagian dari sistem polder untuk mengelola tata air di kawasan padat penduduk dan pusat bisnis. Secara geografis, stasiun ini berfungsi untuk mengendalikan debit air dari Kali Cideng sebelum dialirkan menuju saluran pembuangan utama atau kanal makro, seperti Kanal Banjir Barat (KBB). Stasiun ini menerapkan sistem drainase aktif, di mana air yang terkumpul di kolam penampungan (*retention basin*) atau *long storage* di pompa secara mekanis untuk melawan gravitasi (*backwater effect*) ketika tinggi muka air di saluran pembuangan utama lebih tinggi daripada di saluran drainase kota [32].

Secara teknis, Stasiun Pompa Cideng dilengkapi dengan sejumlah unit pompa berkapasitas besar yang dirancang untuk menangani debit banjir ekstrem. Pada umumnya, stasiun pompa skala ini memiliki kapasitas agregat mencapai puluhan meter kubik per detik ( $m^3/s$ ) yang terbagi ke dalam beberapa unit pompa (4 unit pompa *submersible* dan 6 unit pompa aksial) untuk memungkinkan manajemen operasi yang fleksibel sesuai beban air. Mekanisme operasional stasiun ini sangat bergantung pada ketersediaan energi listrik dan dukungan generator set (*genset*) cadangan untuk memastikan pompa tetap beroperasi saat terjadi pemadaman listrik akibat cuaca buruk. Keandalan komponen elektromekanis di stasiun ini sangat krusial karena kegagalan satu unit pompa saja dapat menyebabkan kenaikan muka air yang signifikan dalam waktu singkat [33].

Urgensi Stasiun Pompa Cideng tidak hanya terletak pada kapasitasnya, melainkan pada kawasan strategis yang dilindunginya, yang sering disebut sebagai area “Ring 1” Ibukota, termasuk kawasan Istana Negara dan sentra ekonomi Thamrin-Sudirman. Kondisi hidrologis daerah tangkapan air (*catchment area*) Cideng didominasi oleh permukaan kedap air (beton dan aspal) dengan koefisien limpasan (*runoff coefficient*) yang mendekati angka 1,0. Hal ini menyebabkan air hujan sangat cepat berubah menjadi aliran permukaan (*surface runoff*) yang membebani saluran drainase. Tanpa adanya intervensi pompa yang bekerja secara real-time dan responsif, kawasan ini memiliki risiko tinggi mengalami genangan

atau banjir, terutama saat terjadi fenomena hujan lokal intensitas tinggi yang disertai dengan pasang air laut atau kiriman air dari hulu [34].

Oleh karena itu, efektivitas operasional Stasiun Pompa Cideng sangat bergantung pada strategi manajemen pola operasi pompa yang presisi. Keterlambatan dalam pengambilan keputusan pengaktifan pompa sering kali menjadi faktor penyebab terjadinya genangan. Kondisi inilah yang mendasari perlunya modernisasi sistem kontrol dari manual menjadi otomatisasi berbasis kecerdasan buatan atau logika fuzzy, guna meminimalisir *human error* dan meningkatkan kecepatan respons terhadap perubahan elevasi muka air yang dinamis [35]. Bentuk Stasiun Pompa Cideng dapat dilihat pada Gambar 2.5



Gambar 2.5 Stasiun Pompa Cideng

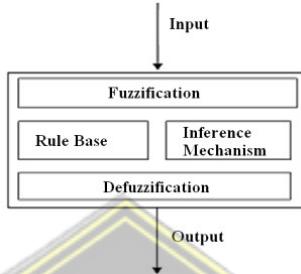
Sumber: Dokumentasi Dinas Sumber Daya Air DKI Jakarta (2023)

#### 2.2.4. Logika Fuzzy

Logika fuzzy adalah suatu cabang dari kecerdasan buatan (*Artificial Intelligence*) yang merupakan perluasan dari logika Boolean klasik. Berbeda dengan logika Boolean yang hanya mengenal dua nilai kebenaran mutlak, yaitu benar (1) atau salah (0), logika fuzzy memungkinkan adanya nilai kebenaran di antara keduanya. Konsep ini disebut derajat keanggotaan (*degree of membership*), yang dapat bernilai real antara 0 dan 1. Logika fuzzy dirancang untuk dapat memproses dan memodelkan penalaran manusia yang seringkali bersifat ambigu, tidak pasti, dan kualitatif, seperti “agak tinggi”, “cukup panas”, atau “sangat cepat” [36].

Metodologi ini sangat berguna untuk menangani sistem-sistem yang kompleks, non-linear, atau sulit untuk dimodelkan secara matematis dengan persamaan diferensial konvensional. Dengan menggunakan variabel linguistik (kata-kata)

untuk merepresentasikan data numerik, logika fuzzy memungkinkan perancangan sistem kontrol yang lebih intuitif karena didasarkan pada pengetahuan dan pengalaman seorang pakar. Sebuah Sistem Kontrol Logika Fuzzy (*Fuzzy Logic Controller/FLC*) secara umum terdiri dari empat komponen utama: fuzzifikasi, basis aturan, mesin inferensi, dan defuzzifikasi [37]. Diagram blok sistem kontrol logika fuzzy dapat dilihat pada Gambar 2.6.



Gambar 2.6 Diagram Blok Sistem Kontrol Logika Fuzzy

Sumber: Diadaptasi dari Murtadlo, dkk (2020)

### 1) Himpunan Fuzzy

Dasar dari logika fuzzy adalah teori himpunan fuzzy (*fuzzy set*). Dalam teori himpunan klasik atau *crisp set*, keanggotaan sebuah elemen dalam suatu himpunan bersifat tegas: sebuah elemen bisa menjadi anggota atau bukan anggota, tidak ada nilai di antaranya. Sebagai contoh, jika kita mendefinisikan himpunan “Tinggi Badan di Atas 170 cm”, maka seseorang dengan tinggi 170,1 cm adalah anggota, sementara seseorang dengan tinggi 169,9 cm bukan anggota. Transisi dari “bukan anggota” menjadi “anggota” terjadi secara tiba-tiba dan kaku.

Himpunan fuzzy mengatasi kekakuan ini dengan memperkenalkan konsep derajat keanggotaan ( $\mu$ ). Dalam himpunan fuzzy, setiap elemen memiliki derajat keanggotaan yang menunjukkan seberapa besar elemen tersebut menjadi bagian dari himpunan. Nilai  $\mu$  berkisar antara 0 (bukan anggota sama sekali) hingga 1 (anggota penuh). Menggunakan contoh yang sama, dalam himpunan fuzzy “Orang Tinggi”, seseorang dengan tinggi 170,1 cm mungkin memiliki derajat keanggotaan  $\mu=0,51$ , sementara yang tingginya 169,9 cm memiliki  $\mu=0,49$ . Transisi keanggotaan terjadi secara gradual, yang lebih mencerminkan cara manusia berpikir [36].

Sebuah himpunan fuzzy A dalam semesta pembicaraan U secara matematis didefinisikan sebagai himpunan pasangan berurutan seperti pada Persamaan 2.2.

$$A = \{ (x, \mu_a(x)) \mid x \in U \} \quad (2.2)$$

Dimana:

$x$  : Elemen dari semesta pembicaraan  $U$

$\mu_a(x)$  : Fungsi keanggotaan dari  $x$  dalam himpunan fuzzy  $A$ , dengan nilai antara  $[0, 1]$

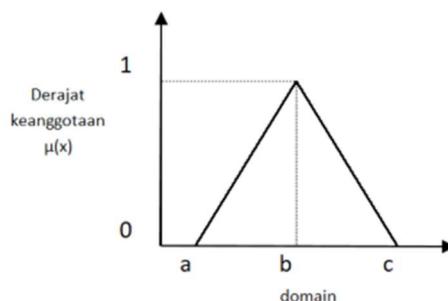
## 2) Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah sebuah kurva atau grafik yang mendefinisikan bagaimana setiap titik dalam semesta pembicaraan (input) dipetakan ke sebuah derajat keanggotaan antara 0 dan 1. Bentuk dari fungsi keanggotaan ini sangat penting karena ia merepresentasikan secara matematis konsep linguistik yang digunakan. Pemilihan bentuk fungsi keanggotaan bersifat subjektif dan bergantung pada pengalaman atau data yang tersedia, namun ada beberapa bentuk standar yang umum digunakan [38]. Beberapa bentuk fungsi keanggotaan yang populer antara lain:

a. Fungsi Keanggotaan Segitiga (*Triangular*) : Bentuk ini adalah yang paling sederhana dan umum digunakan. Didefinisikan oleh tiga parameter  $(a, b, c)$ , di mana  $a$  dan  $c$  adalah kaki-kaki segitiga (nilai keanggotaan = 0) dan  $b$  adalah puncak segitiga (nilai keanggotaan = 1). Rumus matematisnya dapat dilihat pada Persamaan 2.3.

$$\mu(x) = \begin{cases} 0 & \text{jika } x \leq a \\ \frac{x-a}{b-a} & \text{jika } a < x \leq b \\ \frac{c-x}{c-b} & \text{jika } b < x < c \\ 0 & \text{jika } x \geq c \end{cases} \quad (2.3)$$

Kurva fungsi keanggotaan segitiga dapat dilihat pada Gambar 2.7.



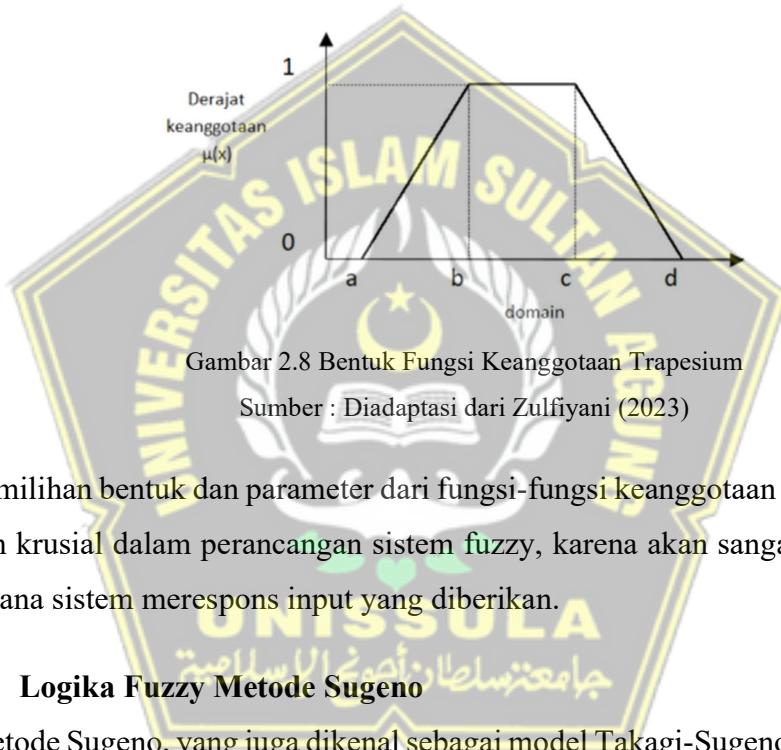
Gambar 2.7 Bentuk Fungsi Keanggotaan Segitiga

Sumber : Diadaptasi dari Zulfiyani (2023)

- b. Fungsi Keanggotaan Trapesium (*Trapezoidal*) : Fungsi ini mirip dengan segitiga tetapi memiliki puncak yang datar, yang berarti ada rentang nilai di mana derajat keanggotaan bernilai 1 penuh. Didefinisikan oleh empat parameter ( $a, b, c, d$ ). Rumus matematisnya dapat dilihat pada Persamaan 2.4.

$$x \begin{cases} 0 & \text{jika } x \leq a \\ \frac{x-a}{b-a} & \text{jika } a < x \leq b \\ 1 & \text{jika } b < x \leq c \\ \frac{d-x}{d-c} & \text{jika } c < x < d \\ 0 & \text{jika } x \geq d \end{cases} \quad (2.4)$$

Kurva fungsi keanggotaan trapesium dapat dilihat pada Gambar 2.8.



Pemilihan bentuk dan parameter dari fungsi-fungsi keanggotaan ini merupakan langkah krusial dalam perancangan sistem fuzzy, karena akan sangat menentukan bagaimana sistem merespons input yang diberikan.

#### 2.2.4.1 Logika Fuzzy Metode Sugeno

Metode Sugeno, yang juga dikenal sebagai model Takagi-Sugeno-Kang (TSK), adalah salah satu dari dua model inferensi fuzzy yang paling umum digunakan, selain metode Mamdani. Diperkenalkan oleh Takagi, Sugeno, dan Kang, model ini memiliki perbedaan fundamental pada bagian konsekuensi (bagian *THEN*) dari aturan fuzzy-nya. Jika pada metode Mamdani output dari setiap aturan adalah sebuah himpunan fuzzy (misalnya, Kecepatan Pompa adalah Cepat), maka pada metode Sugeno, output dari setiap aturan adalah sebuah fungsi matematika dari variabel *input*, bukan himpunan fuzzy [36].

Perbedaan ini membuat model Sugeno sangat efisien secara komputasi dan cocok untuk aplikasi sistem kontrol yang memerlukan respons cepat dan presisi

matematis. Karena outputnya sudah berupa fungsi, proses defuzzifikasi menjadi jauh lebih sederhana dibandingkan dengan metode Mamdani yang memerlukan perhitungan centroid yang kompleks. Model Sugeno sangat ideal untuk dimodelkan secara matematis dan bekerja dengan baik bersama teknik optimisasi dan adaptif, seperti pada sistem *Adaptive Neuro-Fuzzy Inference System (ANFIS)* [5].

Sistem inferensi Sugeno memiliki tiga komponen utama: fuzzifikasi input, aplikasi aturan inferensi, dan penentuan output sebagai berikut:

### 1. Fuzzifikasi *Input*

Tahap ini identik dengan metode Mamdani. Setiap variabel *input* numerik (*crisp*) yang diterima dari sensor dipetakan ke dalam himpunan-himpunan fuzzy menggunakan fungsi keanggotaan (misalnya, segitiga, trapesium, atau *gaussian*). Hasil dari tahap ini adalah derajat keanggotaan (nilai antara 0 dan 1) untuk setiap himpunan fuzzy dari setiap variabel *input*.

### 2. Basis Aturan dan Mesin Inferensi

Ini adalah bagian yang paling membedakan model Sugeno. Struktur aturan IF-THEN pada model Sugeno memiliki antecedent (bagian IF) yang bersifat fuzzy, namun konsekuensi (bagian THEN) yang bersifat *crisp* berupa fungsi matematika. Terdapat dua model Sugeno yang umum digunakan, yaitu:

- Model Sugeno Orde-Nol (*Zero-Order*) Dalam model ini, output dari setiap aturan adalah sebuah nilai konstan (polinomial orde-nol). Model ini adalah bentuk paling sederhana dari Sugeno. Bentuk aturannya dapat dilihat pada Persamaan 2.5.

$$\text{IF } x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z = k \quad (2.5)$$

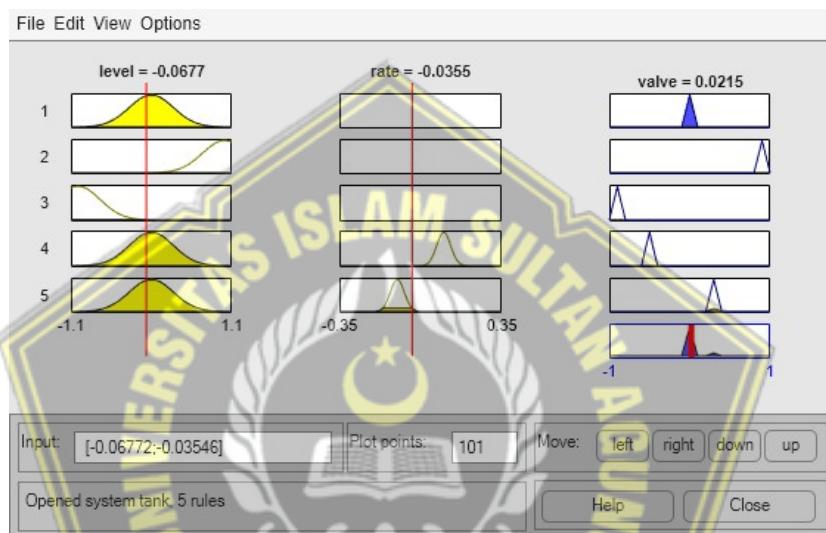
Dimana  $k$  adalah sebuah konstanta. Setiap aturan akan menghasilkan *output* berupa nilai numerik tunggal.

- Model Sugeno Orde-Satu (*First-Order*) Dalam model ini, output dari setiap aturan adalah sebuah fungsi linear dari variabel-variabel input (polinomial orde-satu). Model ini lebih canggih dan mampu merepresentasikan sistem yang lebih kompleks. Bentuk aturannya dapat dilihat pada Persamaan 2.6.

$$\text{IF } x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z = px + qy + r \quad (2.6)$$

Dimana  $p$ ,  $q$ , dan  $r$  adalah koefisien konstanta yang ditentukan selama proses perancangan. Nilai *output*  $z$  dari aturan ini tidak hanya konstan, tetapi bergantung pada nilai *input*  $x$  dan  $y$ .

Mesin inferensi bekerja dengan menghitung kekuatan pemicu (*firing strength*) atau  $\alpha$ -predikat dari setiap aturan. Nilai ini, dilambangkan dengan  $w$ , dihitung dengan menerapkan operator fuzzy (biasanya MIN untuk AND) pada derajat keanggotaan dari bagian anteseden. Untuk aturan ke- $i$ ,  $w_i = \min(\mu_a(x), \mu_e(y))$ . Ilustrasi proses inferensi logika fuzzy pada Matlab dapat dilihat pada Gambar 2.9.



Gambar 2.9 Proses Inferensi Logika Fuzzy

Sumber: Dokumentasi Matlab

### 3. Penentuan *Output* (Defuzzifikasi)

Proses penentuan *output* akhir pada metode Sugeno secara inheren merupakan proses defuzzifikasi yang sangat efisien. Berbeda dengan fuzzy Mamdani yang menggabungkan himpunan fuzzy output dan mencari centroidnya,, metode Sugeno menggunakan rata-rata terbobot (*weighted average*) dari output setiap aturan.

Langkah-langkahnya adalah sebagai berikut:

- Untuk setiap aturan ke- $i$ , hitung kekuatan pemicunya,  $w_i$ .
- Untuk setiap aturan ke- $i$ , hitung level outputnya,  $z_i$ .
  - a. Jika menggunakan model orde-nol,  $z_i = k_i$ .
  - b. Jika menggunakan model orde-satu,  $z_i = p_i x + q_i y + r_i$ , di mana  $x$  dan  $y$  adalah nilai *crisp* input awal.

- Output akhir sistem (*crisp output*),  $z$ , dihitung dengan rumus rata-rata terbobot seperti pada Persamaan 2.7.

$$z = \frac{\sum \alpha_i z_i}{\sum \alpha_i} \quad (2.7)$$

Dimana:

$Z$  : Nilai crisp

$\alpha_i$  : Kekuatan pemicu dari aturan ke- $i$

$z_i$  : Level output dari aturan ke- $i$

Proses ini secara signifikan mengurangi beban komputasi karena tidak melibatkan operasi integral yang kompleks. Hasilnya adalah nilai numerik tunggal yang dapat langsung digunakan untuk mengendalikan aktuator [37].

### 2.2.5. Mean Squared Error

*Mean Squared Error* (MSE) adalah metrik statistik fundamental yang digunakan untuk mengukur “kualitas” sebuah model atau estimator. Secara spesifik, MSE mengukur rata-rata dari kuadrat selisih (*error*) antara nilai yang diestimasi atau diprediksi oleh sebuah model dengan nilai aktual atau observasi. Dalam bidang rekayasa dan ilmu data, MSE menjadi salah satu metode evaluasi kinerja yang paling populer karena sifat matematisnya yang baik dan interpretasinya yang jelas [39]. Rumus untuk menghitung MSE dapat dilihat pada Persamaan 2.8.

$$MSE = (1/n) * \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (2.8)$$

Dimana:

$n$  : Jumlah total titik data atau observasi

$Y_i$  : Nilai observasi atau nilai aktual ke- $i$

$\hat{Y}_i$  : Nilai yang diprediksi atau dihasilkan oleh model untuk observasi ke- $i$

$(Y_i - \hat{Y}_i)$  : Selisih atau error untuk satu observasi

$\Sigma$  : Notasi sigma yang berarti menjumlahkan semua kuadrat error dari  $i=1$  hingga  $n$

Alasan utama mengapa selisihnya dikuadratkan adalah dua hal. Pertama, pengkuadratan memastikan bahwa semua nilai error (baik positif maupun negatif) menjadi positif, sehingga tidak terjadi saling meniadakan saat dijumlahkan. Kedua,

pengkuadratan memberikan “bobot” yang lebih besar pada error yang lebih besar. Artinya, selisih sebesar 4 akan memberikan kontribusi 16 terhadap total MSE, sementara selisih sebesar 2 hanya berkontribusi 4. Hal ini membuat MSE sangat sensitif terhadap *outlier* atau prediksi yang sangat jauh dari nilai sebenarnya. Dalam konteks sistem kontrol, MSE dapat digunakan pada tahap analisis pasca-simulasi untuk mengevaluasi seberapa baik kinerja kontroler dalam menjaga variabel proses (misalnya, ketinggian air) agar tetap dekat dengan setpoint yang diinginkan. Nilai MSE yang mendekati nol mengindikasikan kinerja kontrol yang sangat baik.

### 2.2.6. Arduino Mega 2560

Arduino Mega 2560 adalah papan sirkuit mikrokontroler yang didasarkan pada ATmega2560. Papan ini merupakan versi yang lebih canggih dari Arduino Uno, dirancang khusus untuk proyek-proyek yang membutuhkan jumlah pin *Input/Output* (I/O) yang jauh lebih banyak, memori yang lebih besar, dan lebih banyak antarmuka komunikasi. Karena kapasitasnya yang besar, Arduino Mega sangat ideal untuk aplikasi kompleks seperti robotika, sistem otomasi, dan proyek yang melibatkan banyak sensor dan aktuator secara bersamaan [40].

Fungsinya sebagai unit pemroses sentral sama seperti Arduino Uno, namun dengan kapabilitas yang ditingkatkan. Arduino Mega mampu mengelola lebih banyak tugas secara simultan, menyimpan program yang lebih besar, dan berinteraksi dengan lebih banyak perangkat keras, menjadikannya pilihan yang tepat untuk sistem kontrol pompa berskala besar dalam penelitian ini. Bentuk fisik Arduino Mega 2560 dapat dilihat pada Gambar 2.10 dan konfigurasi pinnya dapat dilihat pada Tabel 2.1.



Gambar 2.10 Arduino Mega 2560

Sumber: Arduino Indonesia

Spesifikasi teknis Arduino Mega:

- Mikrokontroler: ATmega2560
- Tegangan Operasi: 5V
- Tegangan *Input* (direkomendasikan): 7-12V
- Pin I/O Digital: 54 (15 di antaranya menyediakan output PWM)
- Pin Input Analog: 16 (A0-A15)
- Arus DC per Pin I/O: 20 mA
- Memori Flash: 256 KB (8 KB digunakan oleh bootloader)
- SRAM: 8 KB
- EEPROM: 4 KB
- Kecepatan Clock: 16 MHz
- Komunikasi: 4x UART (Serial Hardware), I2C, SPI

Tabel 2.1 Konfigurasi Pin Arduino Mega 2560

| Pin                                    | Fungsi   |
|--|--|
| <i>Power</i> (Daya)                    | Menyediakan pin 5V, 3.3V, GND, dan Vin   |
| GND                                    | Pin <i>Ground</i> atau kutub negatif   |
| <i>Input</i> Analog (A0-A15)           | Menyediakan 16 pin untuk membaca sinyal analog, jauh lebih banyak dari Uno   |
| <i>Input/Output</i> Digital (Pin 0-53) | Menyediakan 54 pin digital yang dapat berfungsi sebagai input atau output  |
| PWM (~)                                | Pin 2 hingga 13 dan 44 hingga 46 menyediakan sinyal PWM  |
| Serial ( <i>Hardware</i> )             | Memiliki 4 port serial terpisah (Serial0 pada pin 0,1; Serial1 pada pin 19,18; Serial2 pada pin 17,16; Serial3 pada pin 15,14) |

### 2.2.7. Sensor Ultrasonik (HC-SR04)

Sensor ultrasonik adalah sebuah perangkat yang digunakan untuk mengukur jarak ke sebuah objek tanpa kontak fisik. Sensor ini bekerja berdasarkan prinsip pantulan gelombang suara ultrasonik (gelombang suara dengan frekuensi di atas jangkauan pendengaran manusia). Model HC-SR04 adalah salah satu sensor

ultrasonik yang paling umum digunakan dalam proyek-proyek mikrokontroler karena harganya yang terjangkau dan kemudahan penggunaannya [41].

Cara kerja sensor ini adalah dengan memancarkan semburan pendek gelombang suara ultrasonik dari *transmitter*. Gelombang ini kemudian akan merambat di udara, dan jika mengenai sebuah objek, gelombang tersebut akan dipantulkan kembali dan ditangkap oleh *receiver*. Mikrokontroler kemudian mengukur selang waktu antara saat gelombang dipancarkan hingga saat gelombang pantul diterima kembali. Jarak ke objek dapat dihitung menggunakan rumus: Jarak = (Waktu Tempuh × Kecepatan Suara) / 2. Bentuk fisik sensor ultrasonik dapat dilihat pada Gambar 2.11 dan konfigurasi pinnya dapat dilihat pada Tabel 2.2.

Gambar 2.11 Sensor Ultrasonik HC-SR04

Sumber: Arduino Indonesia

Spesifikasi teknis sensor Ultrasonic HC-SR04:

- Tegangan Operasi: 5V DC
- Arus Kerja: < 15 mA
- Frekuensi Kerja: 40 kHz
- Jarak Deteksi: 2 cm – 450 cm
- Sudut Deteksi:  $\pm 15$  derajat
- Akurasi:  $\pm 0.3$  cm

Tabel 2.2 Konfigurasi Pin HC-SR04

| Pin         | Fungsi   |
|-------------|--|
| VCC         | Pin catu daya, dihubungkan ke 5V pada Arduino  |
| <i>Echo</i> | Pin <i>output</i> . Setelah sensor memancarkan gelombang, pin ini akan menghasilkan sinyal <i>HIGH</i> . Durasi sinyal <i>HIGH</i> ini sebanding dengan waktu tempuh gelombang suara bolak-balik |

| Pin            | Fungsi  |
|----------------|---|
| <i>Trigger</i> | Pin <i>input</i> pemicu. Untuk memulai pengukuran, pin ini harus diberi sinyal <i>HIGH</i> selama minimal 10 mikrodetik oleh mikrokontroler |
| GND            | Pin <i>Ground</i> , dihubungkan ke GND pada Arduino   |

### 2.2.8. Sensor Curah Hujan (*Tipping Bucket*)

Sensor curah hujan tipe *tipping bucket* (ember jungkit) adalah instrumen meteorologi standar yang digunakan untuk mengukur curah hujan secara kuantitatif dan otomatis. Berbeda dengan sensor *raindrop* yang hanya mendeteksi ada atau tidaknya hujan secara kualitatif, *tipping bucket* mampu memberikan data presisi mengenai jumlah curah hujan dalam satuan milimeter (mm) [42]. Instrumen ini bekerja berdasarkan prinsip mekanis sederhana yang mengubah volume air hujan yang terkumpul menjadi sinyal pulsa digital yang dapat dihitung.

Prinsip kerja alat ini dimulai ketika air hujan masuk melalui sebuah corong penampung (*funnel*) dengan luas area yang terstandarisasi (misalnya,  $200 \text{ cm}^2$ ). Air tersebut kemudian diarahkan untuk jatuh ke salah satu dari dua wadah kecil (ember) yang terpasang pada sebuah mekanisme jungkit. Setiap ember dirancang untuk menampung volume air yang setara dengan curah hujan tertentu, misalnya 0.2 mm. Ketika satu ember telah terisi penuh, berat air akan menyebabkan mekanisme tersebut berjungkit (terbalik), menumpahkan air yang terkumpul, dan secara bersamaan memposisikan ember kosong lainnya di bawah corong untuk mulai menampung air hujan berikutnya [43].

Setiap kali mekanisme ini berjungkit, sebuah magnet kecil yang terpasang pada porosnya akan melewati sebuah saklar buluh (*reed switch*). *Reed switch* adalah saklar yang akan menutup (terhubung) sesaat ketika dilewati medan magnet. Penutupan saklar sesaat ini menghasilkan satu pulsa sinyal listrik. Dengan menghitung jumlah pulsa yang terjadi dalam periode waktu tertentu, sebuah mikrokontroler atau data *logger* dapat menentukan total curah hujan. Sebagai contoh, jika sensor memiliki resolusi 0,2 mm per pulsa dan tercatat 10 pulsa, maka total curah hujan yang terukur adalah 2,0 mm [44]. Bentuk fisik *tipping bucket* dapat dilihat pada Gambar 2.12.



Gambar 2.12 Sensor *Tipping Bucket*

Sumber: Dokumentasi Pribadi

Spesifikasi teknis tipping bucket:

- Tegangan *Input*: 3,3V - 5V DC
- Material: Plastik PLA/ASA
- Menggunakan pin *interrupt*.
- Luas kolektor  $5,4\text{cm} \times 3,6\text{cm} = 19,44 \text{ cm}^2$
- Resolusi: 1,42mL per tip. (1 tip bernilai 0,7 mm curah hujan)
- Akurasi:  $\pm 2\%$  hingga  $\pm 4\%$  tergantung pada intensitas hujan.
- Tipe Output: Kontak saklar buluh (*reed switch*) sesaat (tipe *Normally Open*).
- Suhu Operasional:  $0^\circ\text{C}$  hingga  $70^\circ\text{C}$  (model tanpa pemanas).

Sensor *tipping bucket* profesional biasanya tidak memiliki “pin” seperti komponen elektronik kecil, melainkan dilengkapi dengan kabel dua inti (*two-core cable*) untuk koneksi. Karena *output*-nya hanyalah sebuah saklar mekanis, sensor ini tidak memerlukan catu daya eksternal untuk beroperasi (kecuali untuk model yang dilengkapi pemanas). Kabel 1 dan kabel 2 adalah terminal dari *reed switch* sehingga tidak ada polaritas (tidak masalah mana yang positif atau negatif).

Untuk menghubungkannya ke mikrokontroler seperti Arduino atau ESP32, salah satu kabel dihubungkan ke GND, dan kabel lainnya dihubungkan ke salah satu pin digital *input*. Pin *input* pada mikrokontroler harus dikonfigurasi dengan *pull-up* resistor internal (`pinMode(pin, INPUT_PULLUP);`). Dengan konfigurasi ini, pin akan membaca *HIGH* saat saklar terbuka (normal) dan akan terbaca *LOW* sesaat ketika *tipping bucket* berjungkit dan saklar menutup. Perubahan dari *HIGH* ke *LOW* inilah yang dideteksi sebagai satu pulsa atau satu “tip”.

### 2.2.9. Sensor Water Flow YF-S201

Sensor aliran air atau *water flow sensor* YF-S201 adalah sebuah perangkat yang digunakan untuk mengukur laju aliran (debit) cairan. Sensor ini bekerja berdasarkan prinsip “*Hall Effect*”. Di dalam badan sensor, terdapat sebuah rotor (kincir) kecil yang dilengkapi dengan magnet. Ketika air mengalir melalui sensor, air akan memutar rotor tersebut. Sebuah sensor *Hall effect* yang ditempatkan di dekat rotor akan mendeteksi putaran medan magnet ini dan menghasilkan pulsa sinyal digital [45].

Frekuensi dari pulsa sinyal digital yang dihasilkan berbanding lurus dengan kecepatan putaran rotor, yang juga sebanding dengan laju aliran air. Mikrokontroler dapat menghitung jumlah pulsa dalam satu satuan waktu untuk menentukan frekuensi, dan kemudian mengonversinya menjadi laju aliran dalam satuan liter per menit atau liter per jam. Sensor YF-S201 umumnya digunakan untuk pipa berukuran  $\frac{1}{2}$  inci. Bentuk fisik sensor *water flow* dapat dilihat pada Gambar 2.13 dan konfigurasi pinnya dapat dilihat pada Tabel 2.3.

Gambar 2.13 Sensor Water Flow YF-S201

Sumber: Eco Robotics

Spesifikasi Teknis Sensor Water Flow YF-S201:

- Tegangan Operasi: 5V – 18V DC
- Rentang Aliran: 1 – 30 Liter/Menit
- Tekanan Air Maksimal:  $< 1.75$  MPa
- Karakteristik Pulsa: Frekuensi (Hz)  $\approx 7.5 \times$  Laju Aliran (Q dalam L/min).  
(Catatan: Nilai konstanta ini dapat bervariasi antar produsen dan harus dikalibrasi).
- Ukuran Pipa: G1/2" (Standar  $\frac{1}{2}$  inci)

Tabel 2.3 Konfigurasi Pin *Water Flow YF-S401*

| Pin          | Fungsi  |
|--------------|---|
| Kabel Merah  | Pin catu daya (VCC), dihubungkan ke 5V  |
| Kabel Hitam  | Pin <i>Ground</i> (GND)   |
| Kabel Kuning | Pin output sinyal (pulsa), dihubungkan ke salah satu pin digital Arduino (sebaiknya yang mendukung <i>interrupt</i> ) |

### 2.2.10. Pompa Air DC *Submersible*

Pompa air DC mini tipe submersible adalah sebuah perangkat elektromekanis yang berfungsi untuk memindahkan air menggunakan motor listrik yang ditenagai oleh arus searah (DC). Pompa jenis ini sangat umum digunakan dalam aplikasi skala kecil dan prototipe karena ukurannya yang ringkas, kebutuhan tegangan yang rendah (biasanya 3-12V), dan kemudahan dalam pengendalian. Tipe yang sering digunakan adalah *submersible*, yang berarti seluruh badan pompa dapat dicelupkan ke dalam air [46].

Prinsip kerjanya sederhana: motor DC memutar sebuah impeler (baling-baling kecil) di dalam rumah pompa. Putaran impeler ini menciptakan area bertekanan rendah di bagian *inlet* (masukan), sehingga air dari luar terisap masuk. Selanjutnya, impeler mendorong air tersebut dengan gaya sentrifugal menuju outlet (keluaran) dengan tekanan yang lebih tinggi. Bentuk fisik pompa air DC dapat dilihat pada Gambar 2.14 dan konfigurasi pinnya dapat dilihat pada Tabel 2.4.



Gambar 2.14 Pompa Air DC Mini Tipe *Submersible*

Sumber: Digiware

Spesifikasi Teknis Pompa Air DC mini *submersible*:

- Tegangan: 2,5-6 Volt
- Daya angkat maksimum: 40-110 cm / 15,75"-43,4"

- Laju aliran: 80-120 L/H
- Diameter luar: 7,5 mm / 0,3"
- Diameter dalam: 5 mm / 0,2"
- Diameter: Sekitar 24 mm / 0,95"
- Panjang: Sekitar 45 mm / 1,8"
- Tinggi: Sekitar 30 mm / 1,2"
- Material: Plastik rekayasa
- Mode penggerak: Desain DC, penggerak magnetik

Tabel 2.4 Konfigurasi Pin Pompa DC *Submersible*

| Pin             | Fungsi  |
|-----------------|---|
| Kabel Merah (+) | Terminal positif, dihubungkan ke kutub positif dari sumber daya DC                |
| Kabel Hitam (-) | Terminal negatif, dihubungkan ke kutub negatif ( <i>Ground</i> ) dari sumber daya |

### 2.2.11. Modul *Relay-2 Channel*

Modul *relay 2-channel* adalah papan sirkuit yang mengintegrasikan dua buah relay independen ke dalam satu modul. Modul ini memungkinkan mikrokontroler untuk mengontrol dua perangkat atau sirkuit bertegangan tinggi secara terpisah hanya dengan menggunakan dua pin output digital. Modul ini sangat efisien dari segi ruang dan perkabelan dibandingkan menggunakan dua modul *relay 1-channel* terpisah [47].

Prinsip kerjanya sama persis dengan *relay 1-channel*, di mana setiap *relay* memiliki kumparan elektromagnetik yang dapat diaktifkan oleh sinyal 5V dari mikrokontroler. Modul ini biasanya dilengkapi dengan LED indikator untuk setiap *relay*, yang akan menyala ketika *relay* dalam kondisi aktif, sehingga memudahkan proses debugging. Bentuk fisik modul *relay 2-channel* dapat dilihat pada Gambar 2.15 dan konfigurasi pinnya dapat dilihat pada Tabel 2.5.



Gambar 2.15 Modul *Relay 2-Channel*

Sumber: Digiware

Spesifikasi teknis Modul *Relay 2-Channel*:

- Tegangan Pemicu (*Coil*): 5V DC
- Jumlah Channel: 2
- Kapasitas Kontak Saklar: Umumnya 10A 250V AC per *channel*
- Arus Pemicu:  $\pm 15-20$  mA per *channel*

Tabel 2.5 Konfigurasi Pin Modul *Relay 2-Channel*

| Pin                     | Fungsi                                       |
|-------------------------|--|
| VCC                     | Pin catu daya untuk modul, dihubungkan ke 5V |
| GND                     | Pin <i>Ground</i>                            |
| IN1                     | Pin sinyal pemicu untuk <i>Relay 1</i>       |
| IN2                     | Pin sinyal pemicu untuk <i>Relay 2</i>       |
| Terminal <i>Relay 1</i> | Terdiri dari COM1, NO1, dan NC1              |
| Terminal <i>Relay 2</i> | Terdiri dari COM2, NO2, dan NC2              |

### 2.2.12. *Real Time Clock* (RTC) DS3231

Modul *Real-Time Clock* (RTC) DS3231 adalah sebuah sirkuit terintegrasi (IC) yang berfungsi untuk menyediakan informasi waktu dan tanggal yang sangat akurat, mulai dari detik, menit, jam, hari, tanggal, bulan, hingga tahun. Keunggulan utama dari modul DS3231 adalah akurasinya yang tinggi berkat adanya *Temperature Compensated Crystal Oscillator* (TCXO) internal, yang menjaga osilator tetap stabil meskipun terjadi perubahan suhu lingkungan [48].

Modul ini dilengkapi dengan baterai cadangan (biasanya tipe CR2032) yang memastikan waktu akan terus berjalan meskipun catu daya utama ke mikrokontroler terputus. Hal ini membuatnya sangat vital untuk aplikasi yang memerlukan pencatatan waktu (*time stamping*) atau penjadwalan, seperti pada sistem rotasi

pompa harian dalam penelitian ini. Komunikasi dengan mikrokontroler dilakukan melalui antarmuka I2C (*Inter-Integrated Circuit*). Bentuk fisik RTC DS3231 dapat dilihat pada Gambar 2.16 dan konfigurasi pinnya dapat dilihat pada Tabel 2.6.



Gambar 2.16 Modul *Real-Time Clock* RTC DS3231

Sumber: Digiware

Spesifikasi Teknis RTC DS3231:

- Tegangan Operasi: 3.3V – 5.5V
- Antarmuka Komunikasi: I2C (SCL, SDA)
- Akurasi:  $\pm 2\text{ppm}$  (parts per million) pada rentang suhu  $0^\circ\text{C}$  hingga  $+40^\circ\text{C}$
- Fitur: Kalender lengkap hingga tahun 2100 dengan kompensasi tahun kabisat, dua alarm yang dapat diprogram, sensor suhu digital internal
- Baterai Cadangan: *Input* untuk baterai 3V (misal: CR2032)

Tabel 2.6 Konfigurasi Pin RTC DS3231

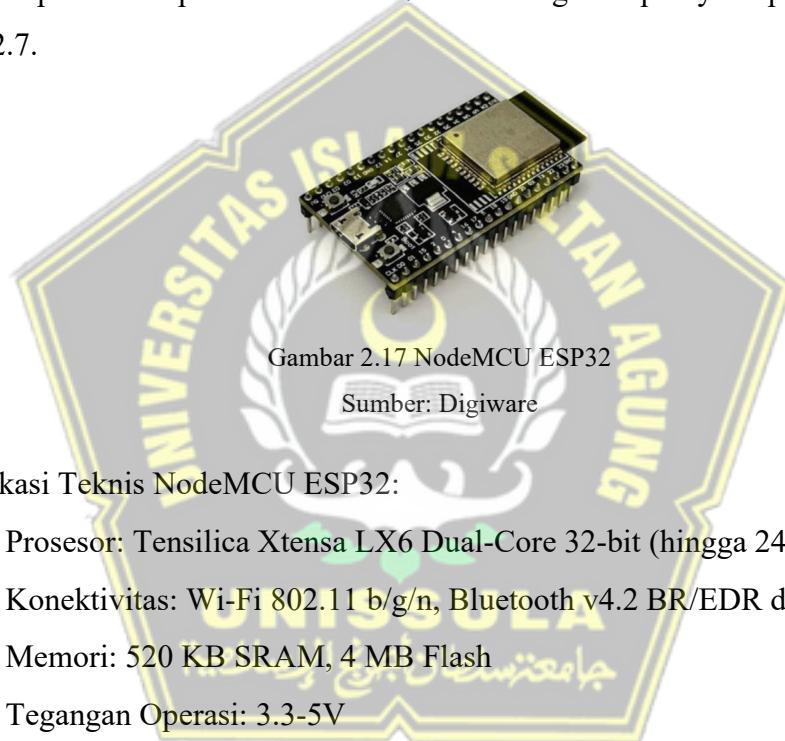
| Pin                         | Fungsi   |
|-----------------------------|--|
| VCC                         | Pin catu daya untuk modul, dihubungkan ke 5V atau 3,3V                       |
| GND                         | Pin <i>Ground</i>  |
| SCL ( <i>Serial Clock</i> ) | Jalur clock untuk komunikasi I2C, dihubungkan ke pin SCL pada mikrokontroler |
| SDA ( <i>Serial Data</i> )  | Jalur data untuk komunikasi I2C, dihubungkan ke pin SDA pada mikrokontroler  |
| SQW ( <i>Square Wave</i> )  | Pin <i>output</i> gelombang persegi yang dapat diprogram (opsional)          |
| 32K                         | <i>Output</i> frekuensi 32kHz (opsional)                                     |

### 2.2.13. NodeMCU ESP32

NodeMCU ESP32 adalah sebuah papan pengembangan *open-source* berbiaya rendah yang didasarkan pada modul ESP32 dari Espressif Systems. ESP32 adalah

sebuah *System on a Chip* (SoC) yang sangat powerful, karena mengintegrasikan mikrokontroler *dual-core* dengan konektivitas Wi-Fi dan Bluetooth dalam satu chip. Hal ini menjadikannya pilihan ideal untuk proyek-proyek *Internet of Things* (IoT) [49].

Mirip dengan Arduino, NodeMCU ESP32 dapat diprogram menggunakan Arduino IDE (setelah menginstal paket *board* ESP32). Keunggulan utamanya dibandingkan Arduino adalah kecepatan prosesor yang jauh lebih tinggi, memori yang lebih besar, serta fitur nirkabel yang sudah terintegrasi, sehingga tidak memerlukan modul tambahan untuk terhubung ke internet. Bentuk fisik NodeMCU ESP32 dapat dilihat pada Gambar 2.17 dan konfigurasi pinnya dapat dilihat pada Tabel 2.7.



Gambar 2.17 NodeMCU ESP32

Sumber: Digiware

#### Spesifikasi Teknis NodeMCU ESP32:

- Prosesor: Tensilica Xtensa LX6 Dual-Core 32-bit (hingga 240 MHz)
- Konektivitas: Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR dan BLE
- Memori: 520 KB SRAM, 4 MB Flash
- Tegangan Operasi: 3.3-5V
- Pin I/O Digital: Sebagian besar pin dapat berfungsi sebagai I/O digital
- Pin Input Analog: 18 channel ADC 12-bit
- Fitur Lain: Sensor sentuh kapasitif, sensor efek Hall, antarmuka SD card, Ethernet, SPI, I2C, UART

Tabel 2.7 Konfigurasi Pin NodeMCU ESP32

| Pin | Fungsi  |
|-----|---|
| VIN | Pin untuk catu daya eksternal 5V (akan diregulasi menjadi 3.3V) |

| Pin                  | Fungsi                              |
|----------------------|-------------------------------------|
| OUT                  | <i>Output</i> tegangan 3.3V         |
| GND                  | <i>Ground</i>                       |
| Digital I/O          | Membaca atau menulis sinyal digital |
| Analog Input (ADC)   | Membaca sinyal analog               |
| PWM                  | Menghasilkan sinyal PWM             |
| EN ( <i>Enable</i> ) | Pin untuk mereset mikrokontroler    |

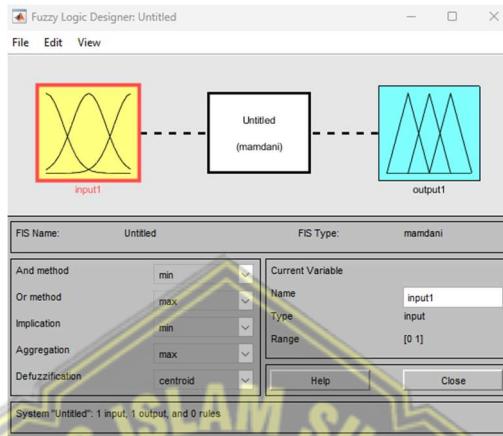
#### 2.2.14. Matlab

Matlab (*Matrix Laboratory*) adalah sebuah platform komputasi numerik dan bahasa pemrograman tingkat tinggi yang dikembangkan oleh MathWorks. *Platform* ini sangat populer di kalangan insinyur, ilmuwan, dan peneliti karena kemampuannya yang luar biasa dalam menangani operasi matriks, analisis data, visualisasi, dan pengembangan algoritma. Arsitektur Matlab yang berpusat pada matriks membuatnya sangat efisien untuk menyelesaikan masalah-masalah matematis yang kompleks [38].

Dalam konteks penelitian ini, komponen Matlab yang paling relevan adalah *Fuzzy Logic Toolbox*. *Toolbox* ini menyediakan lingkungan pengembangan interaktif yang komprehensif untuk merancang, menyimulasikan, dan mengimplementasikan sistem berbasis logika fuzzy. Pengguna dapat memanfaatkan antarmuka grafis (GUI) seperti *Fuzzy Inference System* (FIS) Editor untuk mendefinisikan variabel *input* dan *output*, metode inferensi (Mamdani atau Sugeno), dan metode defuzzifikasi. Selanjutnya, *Membership Function Editor* memungkinkan pengguna untuk merancang bentuk fungsi keanggotaan secara visual, sementara *Rule Editor* menyediakan antarmuka yang intuitif untuk menulis dan mengelola basis aturan IF-THEN.

Keunggulan utama penggunaan Matlab adalah kemampuannya untuk melakukan simulasi yang mendalam sebelum sistem diimplementasikan pada perangkat keras. Dengan menggunakan Matlab atau lingkungannya yang berbasis blok, Simulink, perancang dapat membuat model matematis dari proses fisik (misalnya, dinamika level air dalam tangki) dan menguji respons kontroler fuzzy terhadap berbagai skenario *input*. Hasil simulasi ini, yang dapat divisualisasikan

dalam bentuk grafik, memungkinkan perancang untuk menyempurnakan (*tuning*) parameter fuzzy (seperti bentuk fungsi keanggotaan dan aturan) untuk mencapai kinerja optimal, sehingga secara signifikan mengurangi waktu dan risiko pada tahap implementasi perangkat keras. Tampilan antarmuka (*interface*) *Fuzzy Logic Designer* dapat dilihat pada Gambar 2.18.



Gambar 2.18 Antarmuka *Fuzzy Logic Designer* pada Matlab

Sumber: Dokumentasi Matlab

### 2.2.15. *Internet of Things* (IoT)

*Internet of Things* (IoT) merepresentasikan sebuah evolusi dari internet, di mana koneksi tidak lagi terbatas pada perangkat komputasi konvensional (seperti komputer dan *smartphone*), melainkan meluas ke miliaran objek fisik di sekitar kita. Objek-objek ini—mulai dari sensor lingkungan, peralatan rumah tangga, hingga mesin industri—dilengkapi dengan sirkuit elektronik, perangkat lunak, dan kemampuan komunikasi jaringan yang memungkinkan mereka untuk mengumpulkan data, bertukar informasi, dan bahkan bertindak berdasarkan data tersebut tanpa memerlukan intervensi manusia [12].

Arsitektur sistem IoT secara konseptual dapat dibagi menjadi beberapa lapisan fungsional. Lapisan paling bawah adalah Lapisan Persepsi atau Perangkat Fisik, yang terdiri dari sensor untuk mengumpulkan data dari lingkungan (misalnya, suhu, kelembaban, ataupun ketinggian air) dan aktuator untuk melakukan tindakan fisik (misalnya, menyalakan relay dan menggerakkan motor). Di atasnya adalah Lapisan Jaringan atau Koneksi, yang bertanggung jawab untuk mentransmisikan data dari perangkat ke cloud. Lapisan ini mencakup teknologi koneksi nirkabel (seperti Wi-Fi, Bluetooth, LoRaWAN) dan protokol komunikasi data yang efisien

seperti MQTT (*Message Queuing Telemetry Transport*), yang dirancang khusus untuk IoT karena ringan dan andal.

Lapisan berikutnya adalah Lapisan Platform atau Pengolahan Data, yang seringkali berada di cloud. Di sinilah data mentah dari sensor diterima, diproses, disimpan, dan dianalisis. Lapisan terakhir adalah Lapisan Aplikasi, yang merupakan antarmuka bagi pengguna akhir. Ini bisa berupa dasbor web, aplikasi seluler, atau sistem notifikasi yang menyajikan informasi yang telah diolah dalam format yang mudah dipahami dan memungkinkan pengguna untuk berinteraksi dengan sistem. Dalam penelitian ini, IoT berperan sebagai tulang punggung komunikasi yang memungkinkan prototipe di laboratorium dapat dipantau dan divisualisasikan dari mana saja, mengubahnya dari sistem otomatis lokal menjadi sistem cerdas yang terhubung. Arsitektur umum IoT dapat dilihat pada Gambar 2.19.



Gambar 2.19 Arsitektur Umum *Internet of Things*

Sumber: Dokumentasi Komara (2023)

### 2.2.16. Platform Blynk

Blynk adalah sebuah *platform Internet of Things* (IoT) yang dirancang untuk menyederhanakan proses pembuatan aplikasi seluler atau web untuk mengontrol dan memonitor perangkat keras dari jarak jauh. *Platform* Blynk terdiri dari tiga komponen utama yang bekerja secara sinergis untuk menyediakan solusi IoT yang lengkap dan mudah digunakan, bahkan bagi pemula [50]. Tiga komponen utama tersebut adalah:

- Blynk *App* yaitu aplikasi pada smartphone (iOS atau Android) yang memungkinkan pengguna membuat antarmuka grafis (dasbor) dengan cara *drag-and-drop widget*. Pengguna dapat memilih berbagai *widget* seperti

tombol, *slider*, grafik, *gauge*, dan terminal untuk berinteraksi dengan perangkat kerasnya.

- Blynk *Server* bertanggung jawab untuk menangani semua komunikasi antara perangkat keras dan aplikasi smartphone. *Server* ini bertindak sebagai perantara yang memastikan data dapat dikirim dan diterima dengan andal. Blynk menyediakan server publik yang dapat digunakan secara gratis (Blynk Cloud), namun juga memberikan opsi bagi pengguna untuk menginstal server lokal sendiri untuk privasi dan kontrol penuh.
- Blynk *Libraries* berfungsi kumpulan pustaka perangkat lunak untuk berbagai platform perangkat keras (seperti Arduino, ESP32, Raspberry Pi). Pustaka ini sangat menyederhanakan proses koneksi perangkat ke *server* Blynk dan proses pengiriman/penerimaan data, sehingga pengembang hanya perlu fokus pada logika utama proyeknya.

Contoh dasbor aplikasi pada *platform* Blynk dapat dilihat pada Gambar 2.20.



Gambar 2.20 Contoh Dasbor Aplikasi pada *Platform* Blynk

Sumber: Dokumentasi Pribadi

## **BAB III**

### **METODE PENELITIAN**

Bab ini menguraikan langkah-langkah sistematis yang dilakukan dalam penelitian, mulai dari desain penelitian, objek yang diteliti, alat dan bahan yang digunakan, hingga perancangan model kontrol logika fuzzy dan integrasi sistem *Internet of Things* (IoT).

#### **3.1. Desain Penelitian**

Penelitian ini menggunakan metode penelitian dan pengembangan (*Research and Development* - R&D) dengan pendekatan eksperimental di laboratorium. Tujuannya adalah untuk menghasilkan sebuah produk berupa prototipe sistem kontrol pompa air otomatis dan menguji kinerjanya. Tahapan penelitian dirancang secara sistematis untuk memastikan tujuan dapat tercapai dengan baik, yang meliputi beberapa langkah utama sebagai berikut:

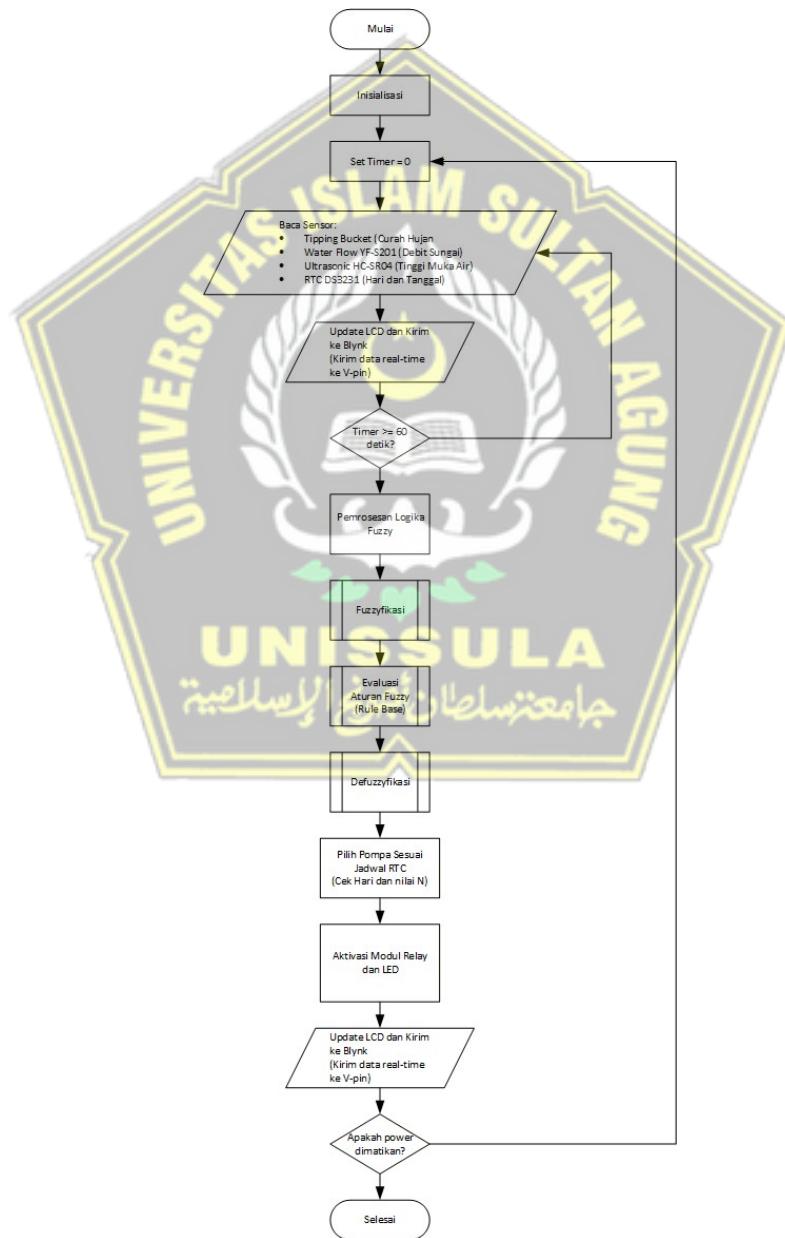
1. Studi Literatur  
Mengumpulkan dan menganalisis teori, konsep, dan penelitian terdahulu yang relevan dengan sistem kontrol, logika fuzzy, sensor, mikrokontroler, dan platform IoT. Tahap ini menjadi dasar untuk perancangan sistem dan penentuan kebaruan penelitian.
2. Perancangan Sistem  
Merancang arsitektur perangkat keras (*hardware*) dan perangkat lunak (*software*). Perancangan perangkat keras mencakup pemilihan komponen dan skema rangkaian elektronik. Perancangan perangkat lunak meliputi pembuatan diagram alir (*flowchart*), perancangan basis aturan logika fuzzy, dan desain antarmuka pada *platform* IoT Blynk.
3. Implementasi dan Perakitan  
Merakit seluruh komponen perangkat keras sesuai dengan skema rangkaian yang telah dirancang. Selanjutnya, melakukan implementasi kode program yang telah dikembangkan ke dalam mikrokontroler Arduino Mega 2560.
4. Pengujian dan Analisis

Melakukan serangkaian pengujian terhadap prototipe yang telah dibuat. Pengujian mencakup fungsionalitas setiap sensor, aktuator (pompa), koneksi IoT, dan yang terpenting, pengujian logika kontrol fuzzy dalam merespons berbagai skenario *input*. Data hasil pengujian kemudian dianalisis untuk mengevaluasi kinerja dan efektivitas sistem.

## 5. Penarikan Kesimpulan

Merumuskan kesimpulan berdasarkan hasil analisis data dan pengujian, serta memberikan saran untuk pengembangan sistem di masa mendatang.

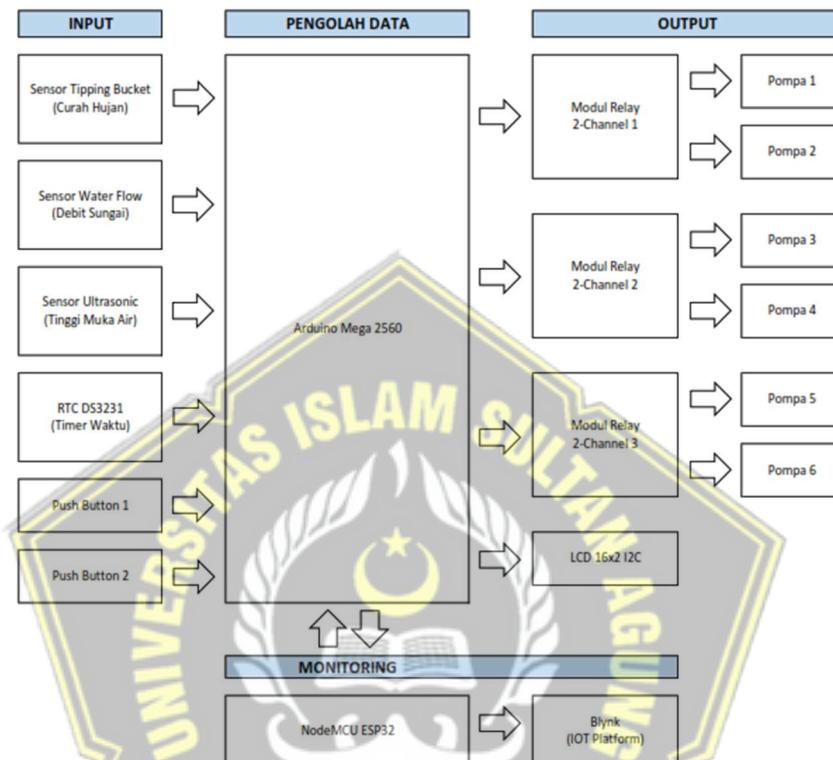
Diagram alir (*flowchart*) penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 *Flowchart* Penelitian

### 3.2. Model Penelitian

Model penelitian ini menggambarkan arsitektur sistem secara keseluruhan, mulai dari *input*, proses, hingga *output*. Sistem ini dirancang sebagai sistem kontrol loop tertutup (*closed-loop*) yang terintegrasi dengan *platform monitoring* jarak jauh. Diagram blok sistem keseluruhan penelitian ini dapat dilihat pada Gambar 3.2.



Gambar 3.2 Diagram Blok Model Penelitian

Secara garis besar, alur kerja sistem adalah sebagai berikut:

#### 1. Blok Input

Terdiri dari tiga jenis sensor, yaitu sensor ultrasonik (mengukur ketinggian muka air), sensor *water flow* (mengukur debit sungai), dan sensor *tipping bucket* (mendeteksi intensitas hujan). Data dari ketiga sensor ini menjadi masukan utama bagi sistem.

#### 2. Blok Proses

Jantung dari sistem ini adalah mikrokontroler **Arduino Mega 2560**. Arduino bertugas untuk membaca data dari semua sensor, kemudian memproses data tersebut menggunakan algoritma logika fuzzy yang telah ditanamkan untuk mengambil keputusan. Selain itu, Arduino juga terhubung ke internet melalui modul Wi-Fi untuk berkomunikasi dengan *server Blynk*.

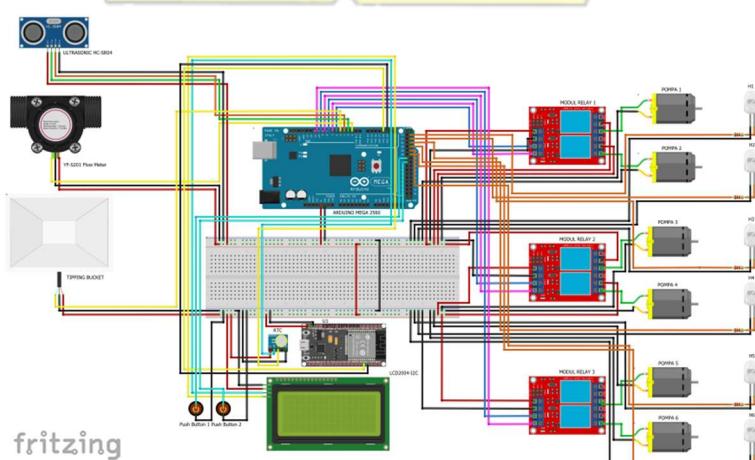
### 3. Blok *Output*

Terdiri dari enam buah pompa air DC yang dikendalikan oleh sebuah modul relay. Keputusan yang dihasilkan oleh logika fuzzy akan dieksekusi oleh Arduino dengan mengirimkan sinyal untuk mengaktifkan atau menonaktifkan *relay*, yang selanjutnya akan menyalakan atau mematikan pompa.

### 4. Blok *Monitoring (IoT)*

Arduino secara periodik mengirimkan data sensor dan status setiap pompa ke *server* Blynk. Data ini kemudian dapat diakses dan divisualisasikan oleh pengguna dari mana saja melalui Aplikasi Blynk di *smartphone* atau dasbor web.

Untuk merealisasikan prototipe sistem pengendalian pompa berbasis logika fuzzy dan *Internet of Things* (IoT), diperlukan rancangan rangkaian elektronik yang mengintegrasikan seluruh sensor, modul kendali, serta aktuator pompa. Gambar berikut menampilkan diagram rangkaian keseluruhan yang dibuat menggunakan perangkat lunak Fritzing sebagai representasi model penelitian. Rangkaian ini mencakup sensor curah hujan, sensor debit sungai, sensor tinggi muka air, mikrokontroler Arduino Mega 2560, ESP32, modul RTC DS3231, *push button*, LCD 20x4, serta enam pompa DC yang dikendalikan melalui modul *relay*. Koneksi catu daya, jalur komunikasi data, dan hubungan antarmodul disusun sedemikian rupa agar sesuai dengan kebutuhan proses akuisisi data, pengolahan logika fuzzy, dan pengiriman informasi ke aplikasi Blynk IoT. Diagram rangkaian keseluruhan alat dapat dilihat pada Gambar 3.3.



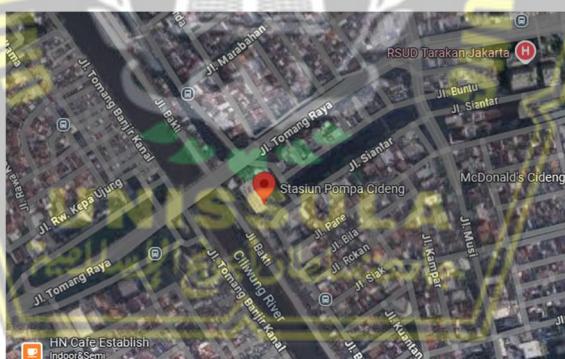
Gambar 3.3 Diagram Rangkaian Keseluruhan Alat

### 3.3. Objek Penelitian

Objek dari penelitian ini adalah sebuah prototipe sistem kontrol otomatis untuk stasiun pompa pengendali banjir. Prototipe ini dirancang dalam skala laboratorium untuk mensimulasikan fungsi dan logika operasional dari stasiun pompa di dunia nyata, dengan mengambil studi kasus dan parameter dari Stasiun Pompa Cideng, Jakarta. Objek ini mencakup integrasi perangkat keras (sensor, mikrokontroler, aktuator) dan perangkat lunak (algoritma Logika Fuzzy dan *platform IoT*) menjadi satu kesatuan sistem yang fungsional.

Pengambilan data dilakukan secara berulang-ulang dengan berbagai kondisi. Data yang diterima kemudian dibuat dalam perbandingan dengan kondisi sebenarnya. Dalam penelitian ini digunakan perbandingan 1:100 untuk data pada rancangan bangun dengan data asli.

Stasiun Pompa Cideng Jakarta sendiri memiliki 10 pompa stasioner yang terdiri dari 6 pompa aksial dengan kapasitas  $6,7 \text{ m}^3/\text{s}$  dan 4 pompa *submersible* dengan kapasitas  $2 \text{ m}^3/\text{s}$ . Lokasi penelitian berada pada Stasiun Pompa Cideng di Jakarta Pusat, DKI Jakarta seperti terlihat pada Google Maps pada Gambar 3.4 serta bangunan stasiun pompa tersebut dapat dilihat pada Gambar 3.5 dan Gambar 3.6.



Gambar 3.4 Lokasi Stasiun Pompa Cideng Jakarta

Sumber : Google Maps



Gambar 3.5 Rumah Pompa Cideng Jakarta



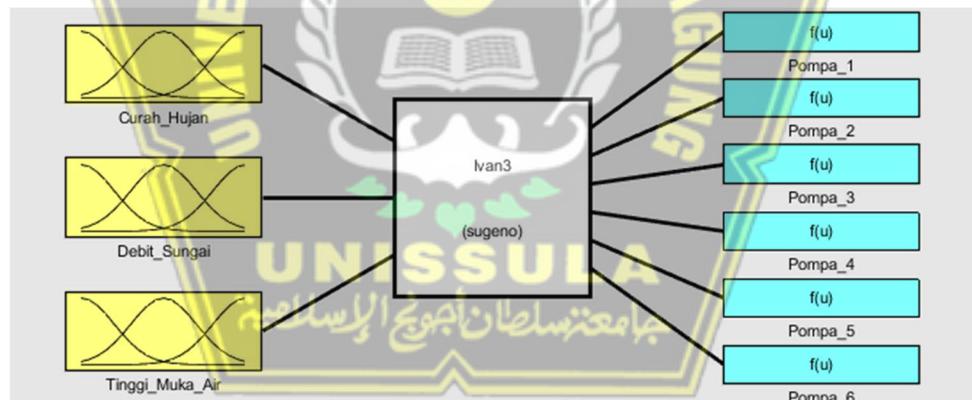
Gambar 3.6 Tampak Dalam Rumah Pompa Cideng Jakarta

Sumber : Dokumentasi Dinas Sumber Daya Air DKI Jakarta

### 3.4. Pemodelan Kendali Pompa Air dengan Logika Fuzzy

#### 3.5.1. Fuzzifikasi

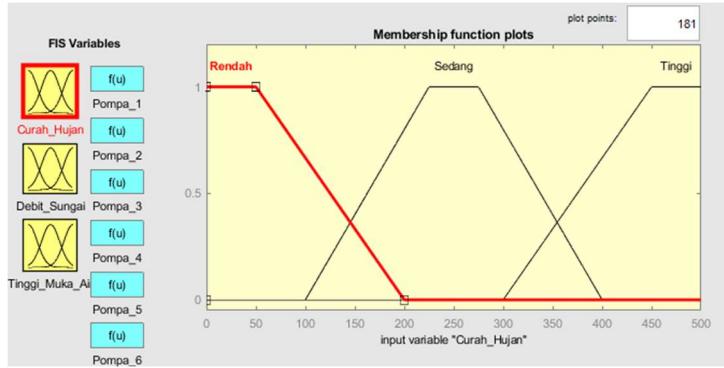
Dalam model penelitian ini, variabel masukan yang digunakan dalam metode fuzzy adalah: Curah Hujan (*input 1*), Debit Sungai (*input 2*), dan Tinggi Muka Air (*input 3*). Sedangkan *output* terdiri dari enam sinyal kontrol pompa: Pompa 1 sampai dengan Pompa 6. Skema fuzzifikasi untuk *input* dan *output* ditunjukkan pada Gambar 3.7.



Gambar 3.7 Parameter *Input-Output* Logika Fuzzy/Fuzzy Inference System

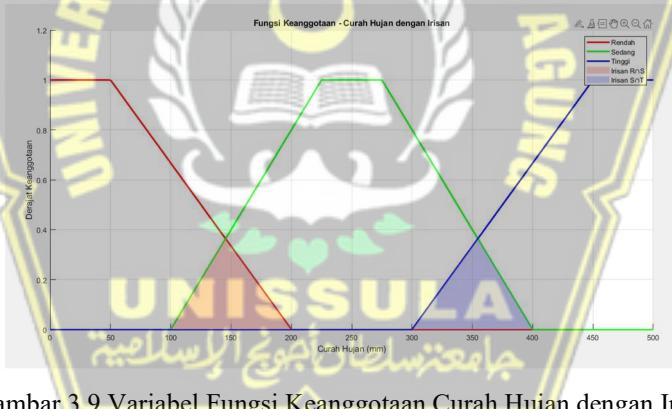
#### 1. Variabel Curah Hujan

Variabel Curah Hujan difuzzykan menjadi tiga nilai linguistik, yaitu: Rendah, Sedang, dan Tinggi. Fungsi keanggotaannya digambarkan sebagai kurva yang memetakan tiap nilai input ke derajat keanggotaan pada rentang 0 sampai 500. Agar nilai yang awalnya kabur (fuzzy) dapat diolah secara komputasional, kurva tersebut dirumuskan dalam bentuk fungsi matematis. Kurva *input* Curah Hujan dapat dilihat pada Gambar 3.8.



Gambar 3.8 Kurva Input Curah Hujan

Pada fungsi keanggotaan Curah Hujan terdapat area tumpang tindih antar himpunan linguistik: sebagian nilai yang termasuk kategori Rendah juga masuk kategori Sedang, dan sebaliknya — tumpang tindih ini berada pada rentang 100 – 200. Pola serupa terjadi antara Sedang dan Tinggi, dengan tumpang tindih pada rentang 300–400. Kondisi tumpang tindih ini diartikan sebagai irisan, yaitu kasus di mana X “dekat dengan A” dan sekaligus “dekat dengan B”. Irisan pada variabel *input* Curah Hujan dapat dilihat pada Gambar 3.9.



Gambar 3.9 Variabel Fungsi Keanggotaan Curah Hujan dengan Irisan

Fungsi matematis dari variabel Curah Hujan [x] yaitu:

- Curah Hujan [Rendah]

Parameter: trapmf[0 0 50 200]

Fungsi matematis:

$$\mu_{\text{Rendah}}(x) \begin{cases} \frac{1}{200-x} & \text{jika } 0 \leq x \leq 50 \\ \frac{150}{150} & \text{jika } 50 < x < 200 \\ 0 & \text{jika } x \geq 200 \end{cases}$$

- Curah Hujan [Sedang]

Parameter: trapmf[100 225 275 400]

Fungsi matematis:

$$\mu_{\text{Sedang}}(x) = \begin{cases} 0 & \text{jika } x \leq 100 \text{ atau } x \geq 400 \\ \frac{x - 100}{125} & \text{jika } 100 < x \leq 225 \\ \frac{1}{400 - x} & \text{jika } 225 < x \leq 275 \\ \frac{125}{125} & \text{jika } 275 < x < 400 \end{cases}$$

c. Curah Hujan [Tinggi]

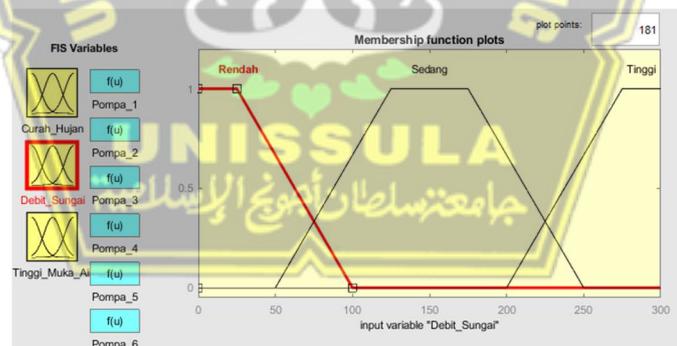
Parameter: trapmf[300 450 500 500]

Fungsi matematis:

$$\mu_{\text{Tinggi}}(x) = \begin{cases} 0 & \text{jika } x \leq 300 \\ \frac{x - 300}{150} & \text{jika } 300 < x < 450 \\ 1 & \text{jika } x \geq 450 \end{cases}$$

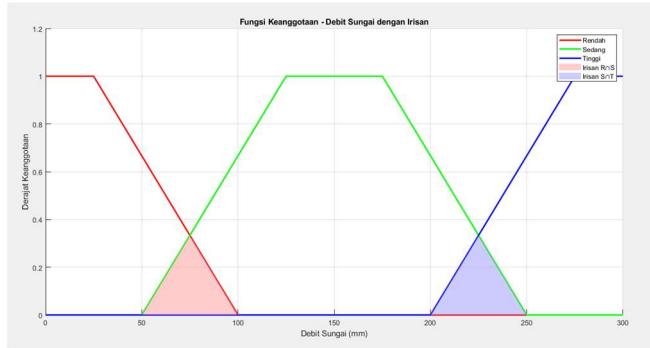
## 2. Variabel Debit Sungai

Variabel Debit Sungai juga memiliki tiga nilai linguistik, yaitu: Rendah, Sedang, dan Tinggi. Fungsi keanggotaannya berupa kurva yang memetakan nilai *input* ke derajat keanggotaan pada rentang 0 sampai 300. Untuk mengubah *output* menjadi nilai pasti yang dapat diproses komputer, kurva tersebut dinyatakan dalam bentuk fungsi matematis. Kurva *input* Debit Sungai dapat dilihat pada Gambar 3.10.



Gambar 3.10 Kurva *Input* Debit Sungai

Pada fungsi keanggotaan Debit Sungai terdapat tumpang tindih antara kategori Rendah dan Sedang pada rentang 50–100, dan tumpang tindih antara Sedang dan Tinggi pada rentang 200–250. Tumpang tindih ini dianggap sebagai irisan, artinya X bisa “dekat dengan A” dan “dekat dengan B” secara bersamaan. Irisan pada variabel input Debit Sungai dapat dilihat pada Gambar 3.11.



Gambar 3.11 Variabel Fungsi Keanggotaan Debit Sungai dengan Irisan

Fungsi matematis dari variabel Debit Sungai [y] yaitu:

- a. Debit Sungai [Rendah]

Parameter: trapmf[ 0 0 25 100]

Fungsi matematis:

$$\mu_{\text{Rendah}}(y) \begin{cases} 1 & \text{jika } 0 \leq y \leq 25 \\ \frac{100-y}{75} & \text{jika } 25 < y < 100 \\ 0 & \text{jika } y \geq 100 \end{cases}$$

- b. Debit Sungai [Sedang]

Parameter: trapmf[50 125 175 250]

Fungsi matematis:

$$\mu_{\text{Sedang}}(y) \begin{cases} 0 & \text{jika } y \leq 50 \text{ atau } y \geq 250 \\ \frac{y-50}{75} & \text{jika } 50 < y \leq 125 \\ \frac{1}{250-y} & \text{jika } 125 < y \leq 175 \\ \frac{75}{250-y} & \text{jika } 175 < y < 250 \end{cases}$$

- c. Debit Sungai [Tinggi]

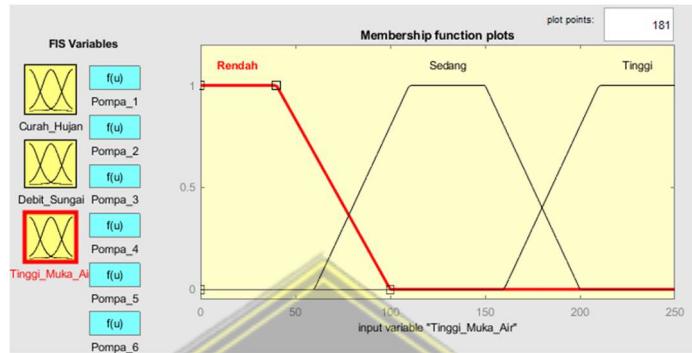
Parameter: trapmf[200 275 300 300]

Fungsi matematis:

$$\mu_{\text{Tinggi}}(y) \begin{cases} 0 & \text{jika } y \leq 200 \\ \frac{y-200}{75} & \text{jika } 200 < y < 275 \\ 1 & \text{jika } y \geq 275 \end{cases}$$

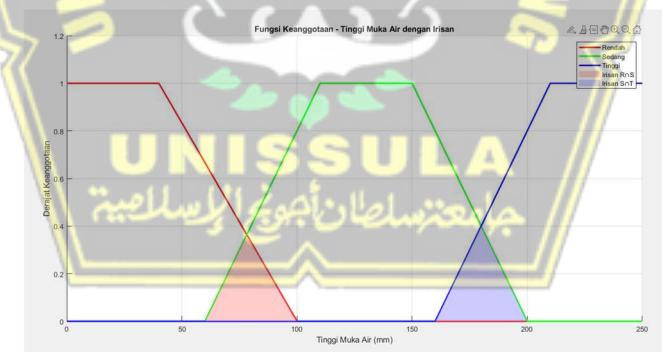
### 3. Variabel Tinggi Muka Air

Variabel Tinggi Muka Air juga difuzzikan menjadi tiga nilai linguistik, yaitu: Rendah, Sedang, dan Tinggi. Fungsi keanggotaannya menggambarkan derajat keanggotaan pada rentang 0 sampai 250, dan untuk memproses data yang bersifat kabur diperlukan representasi fungsi matematis. Kurva *input* Tinggi Muka Air dapat dilihat pada Gambar 3.12.



Gambar 3.12 Kurva *Input* Tinggi Muka Air

Pada fungsi keanggotaan Tinggi Muka Air terdapat tumpang tindih antara Rendah dan Sedang pada rentang 60–100, serta antara Sedang dan Tinggi pada rentang 160–200. Seperti sebelumnya, kondisi tumpang tindih ini dipahami sebagai irisan: X bisa “dekat dengan A” sekaligus “dekat dengan B”. Irisan pada variabel input Tinggi Muka Air dapat dilihat pada Gambar 3.13.



Gambar 3.13 Variabel Fungsi Keanggotaan Tinggi Muka Air dengan Irisan

Fungsi matematis dari variabel Tinggi Muka Air [z] yaitu:

- Tinggi Muka Air [Rendah]

Parameter: `trapmf[ 0 0 40 100 ]`

Fungsi matematis:

$$\mu_{Rendah}(z) \begin{cases} \frac{1}{100-z} & \text{jika } 0 \leq z \leq 40 \\ \frac{60}{60} & \text{jika } 40 < z < 100 \\ 0 & \text{jika } z \geq 100 \end{cases}$$

b. Tinggi Muka Air [Sedang]

Parameter: trapmf[60 110 150 200]

Fungsi matematis:

$$\mu_{Sedang}(z) \begin{cases} 0 & \text{jika } z \leq 60 \text{ atau } z \geq 200 \\ \frac{z-60}{50} & \text{jika } 60 < z \leq 110 \\ 1 & \text{jika } 110 < z \leq 150 \\ \frac{200-z}{50} & \text{jika } 150 < z < 200 \end{cases}$$

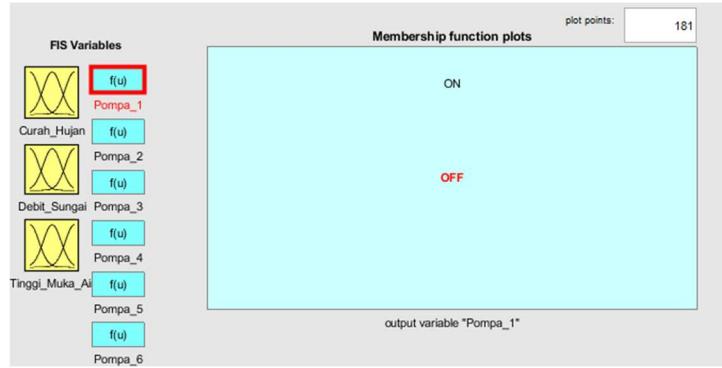
c. Tinggi Muka Air [Tinggi]

Parameter: trapmf[160 210 250 250]

Fungsi matematis:

$$\mu_{Tinggi}(z) \begin{cases} 0 & \text{jika } z \leq 160 \\ \frac{z-160}{50} & \text{jika } 160 < z < 210 \\ 1 & \text{jika } z \geq 210 \end{cases}$$

*Output* sistem adalah enam variabel pompa (Pompa 1 sampai dengan Pompa 6). Masing-masing variabel *output* pompa memiliki dua nilai linguistik yaitu *ON* dan *OFF*. Enam variabel keluaran pompa merepresentasikan unit pompa yang dikendalikan secara individual sesuai keputusan logika fuzzy. Nilai linguistik *ON* dan *OFF* menandakan kondisi kerja setiap pompa dalam merespons tingkat curah hujan, debit sungai, dan tinggi muka air. Fungsi keanggotaan berbentuk dua tingkat (biner) memastikan bahwa keputusan pengaktifan pompa bersifat tegas, sehingga memudahkan integrasi dengan modul *relay*. Melalui pendekatan ini, sistem mampu menyalakan kombinasi pompa tertentu sesuai kebutuhan kapasitas pembuangan, sekaligus dapat dipantau melalui antarmuka Blynk IoT. Bentuk fungsi keanggotaannya untuk tiap pompa ditampilkan pada Gambar 3.14.



Gambar 3.14 Variabel *Output* Pompa 1 sampai dengan Pompa 6

### 3.5.2. *Fuzzy Rule Base*

*Fuzzy rule base* adalah kumpulan aturan fuzzy yang menghubungkan kondisi *input (antecedent)* dengan aksi/keputusan *output (consequent)*. Pada penelitian ini menggunakan aturan berupa pernyataan *If-Then*. *Fuzzy rule base* direpresentasikan dalam bentuk yang lebih sederhana dapat dilihat pada Tabel 3.1.

Tabel 3.1 *Fuzzy Rule Base*

| NO | CURAH HUJAN | DEBIT SUNGAI | TINGGI MUKA AIR | P1  | P2  | P3  | P4  | P5  | P6  |
|----|-------------|--------------|-----------------|-----|-----|-----|-----|-----|-----|
| 1  | RENDAH      | RENDAH       | RENDAH          | OFF | OFF | OFF | OFF | OFF | OFF |
| 2  | RENDAH      | RENDAH       | SEDANG          | ON  | ON  | OFF | OFF | OFF | OFF |
| 3  | RENDAH      | RENDAH       | TINGGI          | ON  | ON  | ON  | ON  | OFF | OFF |
| 4  | RENDAH      | SEDANG       | RENDAH          | ON  | OFF | OFF | OFF | OFF | OFF |
| 5  | RENDAH      | SEDANG       | SEDANG          | ON  | ON  | ON  | OFF | OFF | OFF |
| 6  | RENDAH      | SEDANG       | TINGGI          | ON  | ON  | ON  | ON  | ON  | OFF |
| 7  | RENDAH      | TINGGI       | RENDAH          | ON  | ON  | OFF | OFF | OFF | OFF |
| 8  | RENDAH      | TINGGI       | SEDANG          | ON  | ON  | ON  | ON  | OFF | OFF |
| 9  | RENDAH      | TINGGI       | TINGGI          | ON  | ON  | ON  | ON  | ON  | ON  |
| 10 | SEDANG      | RENDAH       | RENDAH          | ON  | ON  | OFF | OFF | OFF | OFF |
| 11 | SEDANG      | RENDAH       | SEDANG          | ON  | ON  | ON  | ON  | OFF | OFF |
| 12 | SEDANG      | RENDAH       | TINGGI          | ON  | ON  | ON  | ON  | ON  | ON  |
| 13 | SEDANG      | SEDANG       | RENDAH          | ON  | ON  | ON  | OFF | OFF | OFF |
| 14 | SEDANG      | SEDANG       | SEDANG          | ON  | ON  | ON  | ON  | ON  | OFF |

| NO | CURAH HUJAN | DEBIT SUNGAI | TINGGI MUKA AIR | P1 | P2 | P3 | P4 | P5  | P6  |
|----|-------------|--------------|-----------------|----|----|----|----|-----|-----|
| 15 | SEDANG      | SEDANG       | TINGGI          | ON | ON | ON | ON | ON  | ON  |
| 16 | SEDANG      | TINGGI       | RENDAH          | ON | ON | ON | ON | OFF | OFF |
| 17 | SEDANG      | TINGGI       | SEDANG          | ON | ON | ON | ON | ON  | ON  |
| 18 | SEDANG      | TINGGI       | TINGGI          | ON | ON | ON | ON | ON  | ON  |
| 19 | TINGGI      | RENDAH       | RENDAH          | ON | ON | ON | ON | ON  | OFF |
| 20 | TINGGI      | RENDAH       | SEDANG          | ON | ON | ON | ON | ON  | ON  |
| 21 | TINGGI      | RENDAH       | TINGGI          | ON | ON | ON | ON | ON  | ON  |
| 22 | TINGGI      | SEDANG       | RENDAH          | ON | ON | ON | ON | ON  | ON  |
| 23 | TINGGI      | SEDANG       | SEDANG          | ON | ON | ON | ON | ON  | ON  |
| 24 | TINGGI      | SEDANG       | TINGGI          | ON | ON | ON | ON | ON  | ON  |
| 25 | TINGGI      | TINGGI       | RENDAH          | ON | ON | ON | ON | ON  | ON  |
| 26 | TINGGI      | TINGGI       | SEDANG          | ON | ON | ON | ON | ON  | ON  |
| 27 | TINGGI      | TINGGI       | TINGGI          | ON | ON | ON | ON | ON  | ON  |

Kemudian dari fuzzy rule base tersebut akan ditampilkan status siaga banjir berdasarkan jumlah pompa yang *ON*. Status tersebut berupa: Normal, Waspada, Siaga, atau Bahaya yang akan ditampilkan melalui platform IoT Blynk. Daftar kategori status siaga banjir dapat dilihat pada Tabel 3.2.

Tabel 3.2 Kategori Status Siaga Banjir

| No | Jumlah Pompa ON | Status  |
|----|-----------------|---------|
| 1  | Tidak Ada       | Normal  |
| 2  | Satu            | Waspada |
| 3  | Dua             | Waspada |
| 4  | Tiga            | Siaga   |
| 5  | Empat           | Siaga   |
| 6  | Lima            | Bahaya  |
| 7  | Enam            | Bahaya  |

### 3.5.3. Defuzzifikasi

Tahap akhir setelah pembentukan aturan pada logika fuzzy Sugeno adalah defuzzifikasi, yaitu proses menghasilkan keluaran tegas (*crisp output*) dari hasil inferensi fuzzy. Berbeda dengan metode Mamdani yang menggunakan fungsi keanggotaan pada himpunan fuzzy keluaran, pada metode Sugeno bagian *consequent* aturan direpresentasikan dalam bentuk fungsi linear atau konstanta. Dengan demikian, hasil akhir diperoleh melalui perhitungan rata-rata berbobot (*weighted average*), di mana bobot ditentukan oleh derajat kebenaran (*firing strength*) dari setiap aturan. Secara matematis, nilai crisp  $z^*$  ditentukan dengan persamaan 2.6.

Dengan  $w_i$  menyatakan nilai *firing strength* aturan ke- $i$  dan  $z_i$  adalah nilai konsekuensi (konstanta atau fungsi linear) dari aturan tersebut. Pendekatan ini relatif lebih efisien secara komputasional dan sangat sesuai untuk sistem kontrol *real-time* karena menghasilkan keluaran numerik secara langsung tanpa perlu membentuk himpunan fuzzy pada tahap akhir. Contoh hasil defuzzifikasi menggunakan *Fuzzy Logic Toolbox* ditampilkan pada Gambar 3.15.



Gambar 3.15 *Fuzzy Rule Base* pada Mode *Rule Viewer* di Matlab

Dalam penelitian kali ini, defuzzifikasi didefinisikan dalam beberapa langkah, yaitu:

1. Mendefinisikan Himpunan *Output (Singleton)*

Berbeda dengan input yang memiliki rentang nilai, *output* dari sistem ini adalah keputusan diskrit yang merepresentasikan jumlah pompa yang aktif. Oleh karena itu, didefinisikan tujuh level *output (singleton)* sebagai berikut:

- $z_0 = 0$  (Tidak ada pompa yang aktif)
- $z_1 = 1$  (Satu pompa aktif)
- $z_2 = 2$  (Dua pompa aktif)
- $z_3 = 3$  (Tiga pompa aktif)
- $z_4 = 4$  (Empat pompa aktif)
- $z_5 = 5$  (Lima pompa aktif)
- $z_6 = 6$  (Enam pompa aktif)

Setiap aturan dalam basis aturan (*rule base*) yang telah dibuat akan menghasilkan salah satu dari nilai  $z$  ini. Sebagai contoh, Aturan #2 (IF Curah Hujan is Rendah AND Debit Sungai is Rendah AND Tinggi Muka Air is Sedang) yang sebelumnya menghasilkan keputusan Pompa 1 *ON*, kini diubah konsekuensinya menjadi *Output* = 1 ( $z_1$ ).

## 2. Metode Center Average Defuzzifier

Untuk mengakomodasi kemungkinan adanya beberapa aturan yang aktif secara bersamaan, metode Center Average Defuzzifier (sering juga disebut Center of Gravity untuk singleton) digunakan. Metode ini menghitung rata-rata tertimbang dari semua kontribusi output aturan, sehingga menghasilkan keputusan yang lebih halus dan representatif. Rumus yang digunakan adalah seperti pada Persamaan 2.6.

## 3. Aturan Keputusan Akhir dan Integrasi Rotasi

Hasil perhitungan  $Z$  akan menghasilkan sebuah nilai numerik yang bisa jadi bukan bilangan bulat (contoh: 3,75). Karena jumlah pompa harus berupa bilangan bulat, maka dilakukan pembulatan ke bilangan bulat terdekat untuk menentukan jumlah pompa akhir ( $N$ ) yang akan diaktifkan, dengan persamaan:

$$N = \text{round}(Z)$$

Nilai  $N$  inilah yang kemudian menjadi input untuk sistem rotasi pompa. Contoh implementasi yaitu misalkan setelah melalui proses fuzzy, didapatkan hasil  $Z = 3,75$ . Maka nilai  $Z$  dibulatkan menjadi  $N = \text{round}(3,75) = 4$ . Ini berarti sistem memutuskan untuk mengaktifkan 4 pompa. Sistem kemudian memeriksa counter hari rotasi (misal hari ke-2).

Berdasarkan Tabel Rencana Operasi (lihat Tabel 3.9), untuk  $N = 4$  pada hari ke-2, pompa yang akan diaktifkan adalah {Pompa 3, Pompa 4, Pompa 5, Pompa 6}.

Mikrokontroler kemudian mengirim sinyal untuk mengaktifkan keempat pompa tersebut.

Dengan metode ini, sistem tidak hanya cerdas dalam menentukan “berapa banyak” pompa yang dibutuhkan, tetapi juga andal dalam menentukan “pompa yang mana” yang harus beroperasi, sesuai dengan prinsip pemerataan beban kerja.

### 3.5. Rencana Operasi dan Rotasi Pompa

Salah satu tantangan dalam sistem multi-pompa yang beroperasi secara kontinu adalah pemerataan beban kerja untuk menghindari keausan dini pada unit pompa tertentu. Jika sistem hanya mengacu pada urutan (misalnya, kondisi 2 pompa selalu menyalakan Pompa 1 dan Pompa 2), maka pompa-pompa tersebut akan memiliki jam operasional yang jauh lebih tinggi dibandingkan pompa lainnya, sehingga meningkatkan risiko kerusakan dan mengurangi keandalan sistem secara keseluruhan.

Untuk mengatasi masalah ini, dirancang sebuah mekanisme rotasi pompa yang bertujuan untuk mendistribusikan beban kerja secara merata. Logika Fuzzy tetap bertugas menentukan jumlah pompa yang harus aktif berdasarkan kondisi hidrologis. Namun, keputusan mengenai pompa spesifik mana yang akan diaktifkan akan diatur oleh skema rotasi berbasis waktu (harian).

#### 3.5.1. Mekanisme Rotasi Harian

Sistem akan mengimplementasikan siklus rotasi selama 6 hari, di mana setiap hari kombinasi pompa yang aktif akan diubah. Siklus ini akan dikelola oleh sebuah counter internal pada mikrokontroler yang diinisialisasi berdasarkan waktu operasional (misalnya, diatur ulang setiap 24 jam dengan bantuan modul Real-Time Clock (RTC) atau disimulasikan). Tabel berikut merinci kombinasi pompa yang akan diaktifkan untuk setiap kemungkinan output dari sistem Logika Fuzzy selama siklus 7 hari. Tabel rencana operasi pompa dapat dilihat pada Tabel 3.3.

Tabel 3.3 Jadwal Rotasi Pompa Harian

| Hari  | Output Fuzzy<br>(N = X) | Kombinasi Pompa Aktif |
|-------|-------------------------|-----------------------|
| Senin | 1                       | Pompa 1               |

| <b>Hari</b> | <b>Output Fuzzy<br/>(N = X)</b> | <b>Kombinasi Pompa Aktif</b>                |
|-------------|---------------------------------|---|
| Senin       | 2                               | Pompa 2, Pompa 3                            |
|             | 3                               | Pompa 4, Pompa 5, Pompa 6                   |
|             | 4                               | Pompa 1, Pompa 2, Pompa 3, Pompa 6          |
|             | 5                               | Pompa 2, Pompa 3, Pompa 4, Pompa 5, Pompa 6 |
|             | 6                               | (Semua Pompa)                               |
| Selasa      | 1                               | Pompa 2                                     |
|             | 2                               | Pompa 4, Pompa 5                            |
|             | 3                               | Pompa 1, Pompa 2, Pompa 3                   |
|             | 4                               | Pompa 1, Pompa 2, Pompa 3, Pompa 5          |
|             | 5                               | Pompa 1, Pompa 3, Pompa 4, Pompa 5, Pompa 6 |
|             | 6                               | (Semua Pompa)                               |
| Rabu        | 1                               | Pompa 3                                     |
|             | 2                               | Pompa 1, Pompa 6                            |
|             | 3                               | Pompa 4, Pompa 5, Pompa 6                   |
|             | 4                               | Pompa 1, Pompa 2, Pompa 3, Pompa 4          |
|             | 5                               | Pompa 1, Pompa 2, Pompa 4, Pompa 5, Pompa 6 |
|             | 6                               | (Semua Pompa)                               |
| Kamis       | 1                               | Pompa 4                                     |
|             | 2                               | Pompa 2, Pompa 4                            |
|             | 3                               | Pompa 1, Pompa 2, Pompa 3                   |
|             | 4                               | Pompa 1, Pompa 2, Pompa 3, Pompa 6          |
|             | 5                               | Pompa 2, Pompa 3, Pompa 4, Pompa 5, Pompa 6 |
|             | 6                               | (Semua Pompa)                               |
| Jumat       | 1                               | Pompa 5                                     |
|             | 2                               | Pompa 3, Pompa 5                            |
|             | 3                               | Pompa 4, Pompa 5, Pompa 6                   |
|             | 4                               | Pompa 2, Pompa 3, Pompa 4, Pompa 5          |
|             | 5                               | Pompa 1, Pompa 3, Pompa 4, Pompa 5, Pompa 6 |
|             | 6                               | (Semua Pompa)                               |

| Hari   | Output Fuzzy<br>(N = X) | Kombinasi Pompa Aktif                       |
|--------|-------------------------|---|
| Sabtu  | 1                       | Pompa 6                                     |
|        | 2                       | Pompa 1, Pompa 6                            |
|        | 3                       | Pompa 1, Pompa 2, Pompa 3                   |
|        | 4                       | Pompa 1, Pompa 3, Pompa 5, Pompa 6          |
|        | 5                       | Pompa 1, Pompa 2, Pompa 4, Pompa 5, Pompa 6 |
|        | 6                       | (Semua Pompa)                               |
| Minggu | 1                       | Pompa 1                                     |
|        | 2                       | Pompa 2, Pompa 3                            |
|        | 3                       | Pompa 4, Pompa 5, Pompa 6                   |
|        | 4                       | Pompa 1, Pompa 2, Pompa 4, Pompa 6          |
|        | 5                       | Pompa 1, Pompa 2, Pompa 3, Pompa 4, Pompa 5 |
|        | 6                       | (Semua Pompa)                               |

Tabel 3.4 dibawah menyajikan hasil perhitungan frekuensi aktif (jumlah  $ON$ ) untuk setiap pompa dalam satu siklus 7 hari, dibagi per skenario kebutuhan. Kolom “Total” menunjukkan akumulasi beban kerja hipotetis jika setiap skenario ( $N=1$  s/d  $N=5$ ) terjadi masing-masing satu kali dalam seminggu.

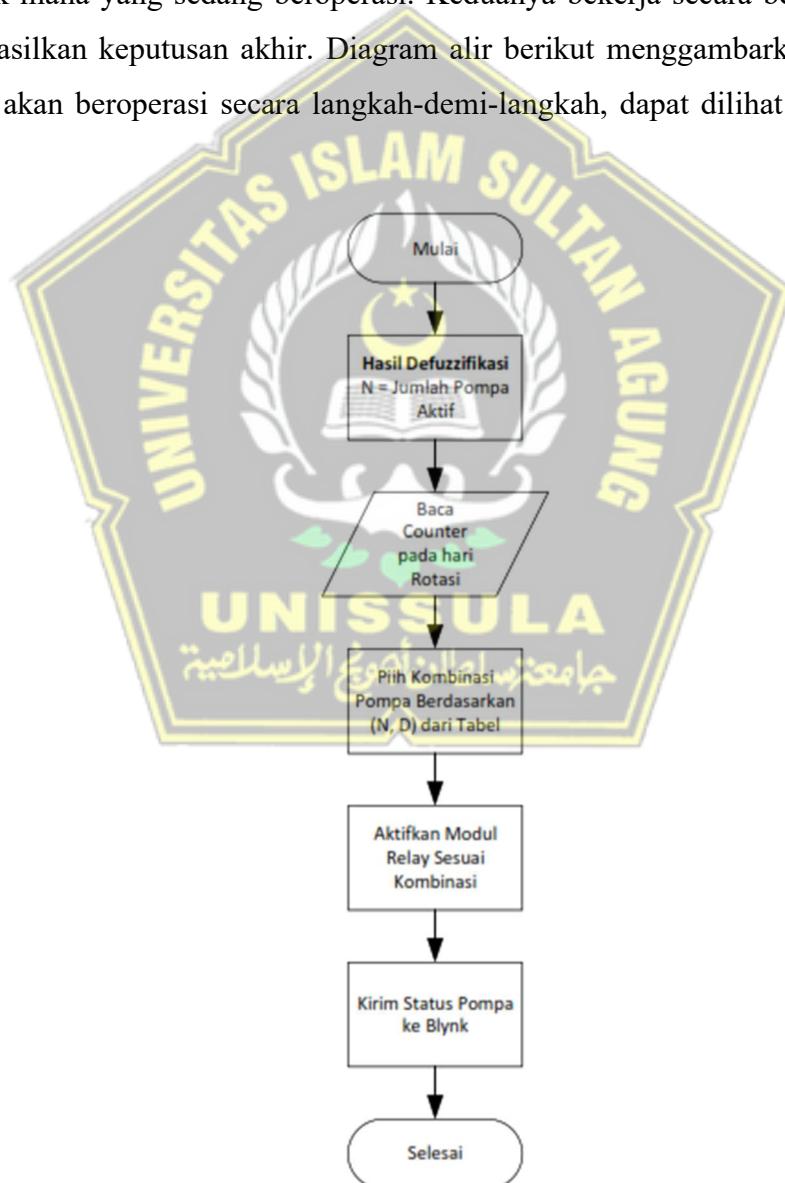
Tabel 3.4 Frekuensi Akumulatif Pompa

| Pompa        | Frekuensi Aktif |           |           |           |           | Total<br>Akumulatif |
|--------------|-----------------|-----------|-----------|-----------|-----------|---------------------|
|              | N=1             | N=2       | N=3       | N=4       | N=5       |                     |
| Pompa 1      | 2               | 2         | 3         | 6         | 5         | <b>18</b>           |
| Pompa 2      | 1               | 3         | 3         | 6         | 5         | <b>18</b>           |
| Pompa 3      | 1               | 3         | 3         | 6         | 5         | <b>18</b>           |
| Pompa 4      | 1               | 2         | 4         | 3         | 7         | <b>17</b>           |
| Pompa 5      | 1               | 2         | 4         | 3         | 7         | <b>17</b>           |
| Pompa 6      | 1               | 2         | 4         | 4         | 6         | <b>17</b>           |
| <b>Total</b> | <b>7</b>        | <b>14</b> | <b>21</b> | <b>28</b> | <b>35</b> | <b>105</b>          |

### 3.5.2. Implementasi Logika Rotasi Pada Mikrokontroler

Logika rotasi ini diimplementasikan sebagai lapisan tambahan setelah proses defuzzifikasi. Arsitektur ini memisahkan dua otak dalam pengambilan Keputusan, yaitu logika fuzzy bertugas menentukan berapa banyak pompa yang harus aktif (misalnya, 4 unit), berdasarkan kondisi hidrologis. Setelah nilai kuantitas ( $N=4$ ) ini diperoleh, logika rotasi bertindak sebagai manajer operasional yang bertugas menentukan pompa mana saja yang akan dieksekusi.

Pemisahan logika ini sangat penting untuk modularitas sistem. Logika rotasi tidak mengintervensi perhitungan fuzzy, dan logika fuzzy tidak perlu tahu pompa spesifik mana yang sedang beroperasi. Keduanya bekerja secara berurutan untuk menghasilkan keputusan akhir. Diagram alir berikut menggambarkan bagaimana sistem akan beroperasi secara langkah-demi-langkah, dapat dilihat pada Gambar 3.16.



Gambar 3.16 Diagram Alir Implementasi Sistem Operasi Pompa

Penjelasan Diagram Alir Implementasi Sistem Operasi Pompa:

1. Hasil defuzzifikasi, yaitu sistem logika fuzzy menghasilkan nilai *crisp* (N) yang merepresentasikan jumlah pompa yang harus aktif.
2. Baca *counter* hari rotasi, yaitu mikrokontroler memeriksa nilai *counter* hari saat ini (D), yang bernilai dari 1 hingga 6.
3. Pilih kombinasi, dimana berdasarkan nilai N dan D, sebuah fungsi atau struktur kontrol (*switch-case atau if-else if*) akan memilih himpunan pompa spesifik yang harus diaktifkan sesuai dengan yang tertera pada Tabel 3.3.
4. Aktifkan *relay*, dimana mikrokontroler mengirimkan sinyal *HIGH* hanya ke pin-pin *relay* yang terhubung dengan pompa yang terpilih dalam kombinasi tersebut.
5. Kirim status ke Blynk, dimana status *ON/OFF* dari keenam pompa kemudian dikirim ke *platform* Blynk untuk memastikan data monitoring sesuai dengan kondisi aktuasi di lapangan.

Dengan mengimplementasikan rencana operasi ini, penelitian tidak hanya berfokus pada kecerdasan sistem dalam merespons kondisi lingkungan, tetapi juga mempertimbangkan aspek keandalan dan keberlanjutan operasional perangkat dalam jangka panjang.

### 3.6. Pemodelan dengan Integrasi IoT

Tahap selanjutnya setelah perancangan sistem kontrol adalah mengintegrasikan prototipe dengan platform *Internet of Things* (IoT). Tujuan dari integrasi ini adalah untuk memungkinkan pemantauan (*monitoring*) dan visualisasi data secara *real-time* dari jarak jauh. Platform yang dipilih untuk penelitian ini adalah Blynk, karena kemudahan dalam penyiapan, antarmuka yang ramah pengguna, dan dukungan pustaka yang lengkap untuk berbagai jenis mikrokontroler, termasuk Arduino dan ESP32.

Proses integrasi ini mengubah sistem kontrol yang bersifat lokal menjadi sistem cerdas yang terhubung ke internet. Data dari sensor ketinggian air, debit sungai, dan curah hujan, serta status operasional dari keenam pompa dan status siaga banjir, akan dikirim secara periodik ke server cloud Blynk. Data ini kemudian dapat diakses melalui sebuah dasbor kustom yang dirancang pada aplikasi *smartphone*, memberikan gambaran lengkap mengenai kondisi stasiun pompa kepada operator

di mana saja dan kapan saja. Langkah-langkah pengintegrasian sistem dengan platform Blynk diuraikan sebagai berikut:

### **Langkah 1: Penyiapan Proyek pada Platform Blynk.Cloud**

Langkah pertama adalah penyiapan ruang kerja digital pada platform Blynk.Cloud. Proses ini berpusat pada *web dashboard* (Blynk.Cloud) untuk konfigurasi templat dan perangkat, serta pada aplikasi smartphone untuk perancangan antarmuka *mobile*, dengan urutan sebagai berikut:

1. Pembuatan *template* dan *datastream*, dimana proses diawali pada website *dashboard* Blynk.Cloud dengan membuat sebuah “*Template*” baru, yang diberi nama “Kontrol Pompa Cideng”. Template ini berfungsi sebagai cetak biru untuk semua perangkat keras. Di dalam *template* ini, “*Datastream*” (saluran data) didefinisikan untuk setiap variabel yang akan di-*monitor*. *Datastream* ini memetakan pin virtual (Vpin) yang akan menghubungkan data dari *hardware* ke *widget* antarmuka.
2. Pembuatan perangkat (*device*) dan kredensial otentikasi, yaitu setelah *template* selesai, sebuah “*Device*” baru (misalnya “ESP32 Cideng”) dibuat berdasarkan *template* tersebut. Pada tahap ini, Blynk.Cloud menghasilkan tiga kredensial otentikasi unik yang dapat diakses melalui tab “*Device Info*” pada dasbor *device*. Kredensial dibawah ini:
  - BLYNK\_TEMPLATE\_ID,
  - BLYNK\_TEMPLATE\_NAME, dan
  - BLYNK\_AUTH\_TOKENmerupakan kunci penting yang harus dimasukkan ke dalam kode program mikrokontroler (ESP32) untuk proses autentikasi koneksi.
3. Perancangan dasbor antarmuka (*web* dan *mobile*), dimana Blynk 2.0 memiliki kanvas desain yang terpisah untuk antarmuka *web* dan *mobile*. Keduanya perlu dirancang secara terpisah, meskipun terhubung ke *datastream* (V0-V9) yang sama. Karena sistem ini bersifat *monitoring-only*, *widget* yang digunakan difokuskan untuk visualisasi data, bukan control, yaitu:
  - *Gauge* (3 buah): Dihubungkan ke V0, V1, dan V2 untuk menampilkan data numerik curah hujan, debit sungai, dan tinggi muka air.

- LED (6 buah): Dihubungkan ke V3 hingga V8. *Widget* ini berfungsi sebagai lampu indikator yang merepresentasikan status *ON/OFF* aktual dari setiap pompa yang dilaporkan oleh Arduino Mega.
- Tampilan Label (*Label*) (1 buah): Dihubungkan ke V9 untuk menampilkan teks status siaga secara penuh (misal: “WASPADA”).

Tampilan dasbor Blynk pada *website* dapat dilihat pada Gambar 3.17 dan tampilan pada aplikasi smartphone dapat dilihat pada Gambar 3.18.



Gambar 3.17 Penambahan *Widget* pada Dasbor Proyek Blynk di *Website*



Gambar 3.18 Penambahan *Widget* pada Dasbor Proyek Blynk di *Website*

## Langkah 2: Konfigurasi Pin Virtual

Setiap *widget* yang ditambahkan ke dasbor harus dihubungkan ke sebuah pin virtual. Pin virtual adalah saluran data antara perangkat keras dan aplikasi. Konfigurasi dilakukan dengan mengetuk setiap *widget* dan mengatur pin virtual yang akan digunakan. Konfigurasi *mapping* pin virtual pada Blynk dapat dilihat pada Tabel 3.5.

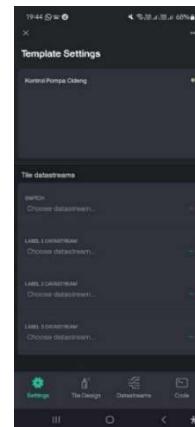
Tabel 3.5 Konfigurasi *Mapping* Pin Virtual Blynk

| Pin | Nama            | Tipe           | Keterangan                             |
|-----|-----------------|----------------|--|
| V0  | Curah Hujan     | <i>Double</i>  | Units: mm, Min: 0                      |
| V1  | Debit Sungai    | <i>Double</i>  | Units: m <sup>3</sup> /s, Min: 0       |
| V2  | Tinggi Muka Air | <i>Double</i>  | Units: m, Min: 0                       |
| V3  | P1              | <i>Integer</i> | Min:0, Maks:1 (Untuk ON/OFF)           |
| V4  | P2              | <i>Integer</i> | Min:0, Maks:1                          |
| V5  | P3              | <i>Integer</i> | Min:0, Maks:1                          |
| V6  | P4              | <i>Integer</i> | Min:0, Maks:1                          |
| V7  | P5              | <i>Integer</i> | Min:0, Maks:1                          |
| V8  | P6              | <i>Integer</i> | Min:0, Maks:1                          |
| V9  | Status Siaga    | <i>String</i>  | “NORMAL”, “WASPADA”, “SIAGA”, “BAHAYA” |

Tampilan konfigurasi pin virtual Blynk pada *website* dapat dilihat pada Gambar 3.19 dan tampilan pada aplikasi smartphone dapat dilihat pada Gambar 3.20.



Gambar 3.19 Konfigurasi Pin Virtual untuk *Widget* pada *Website*



Gambar 3.20 Konfigurasi Pin Virtual untuk *Widget* pada Aplikasi

### Langkah 3: Implementasi Kode Program untuk Konektivitas IoT

Fungsionalitas IoT pada sistem ini diimplementasikan menggunakan arsitektur gateway, yang memisahkan unit pemrosesan utama dengan unit konektivitas.

1. Unit Pemrosesan Utama (Arduino Mega 2560): Tetap menjalankan logika inti, membaca sensor, dan menentukan status pompa. Modifikasi kode difokuskan untuk mengirimkan seluruh data sistem secara periodik melalui Serial1.
2. Unit Konektivitas (ESP32 *Gateway*): Sebuah ESP32 didedikasikan sebagai jembatan (*gateway*) antara Arduino Mega dan *cloud*. ESP32 bertugas menerima data serial dari Mega dan meneruskannya ke *server* Blynk.

Implementasi kode dibagi menjadi dua bagian:

#### A. Kode Program ESP32 (Gateway)

Program pada ESP32 bertujuan untuk terhubung ke WiFi dan Blynk, serta mendengarkan data dari Mega.

- 1) Inisialisasi Pustaka dan Kredensial: Pustaka WiFi.h dan BlynkSimpleEsp32.h diikutsertakan. Kredensial Blynk (diperoleh dari dasbor web Blynk) dan kredensial WiFi (SSID & *password*) didefinisikan di awal program.

C++

```
#define BLYNK_TEMPLATE_ID "..."
#define BLYNK_TEMPLATE_NAME "..."
#define BLYNK_AUTH_TOKEN "..."
char ssid[] = "Nama_WiFi";
char pass[] = "Password_WiFi";
```

- 2) Koneksi Server dan Serial: Di dalam `setup()`, koneksi ke Blynk diinisialisasi menggunakan `Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass)`. Selain itu, `HardwareSerial SerialMega(2)` diinisialisasi untuk memulai komunikasi hardware serial (pada Pin 16, 17) dengan Arduino Mega pada baud rate 9600.
- 3) Menjaga Koneksi: Di dalam `loop()`, fungsi `Blynk.run()` dipanggil secara kontinu untuk menjaga koneksi dan sinkronisasi dengan server Blynk.

## B. Modifikasi Kode Arduino Mega (Logika Utama)

Program pada Arduino Mega dimodifikasi untuk secara aktif melaporkan seluruh status sistem ke ESP32.

- 1) Inisialisasi Serial: Komunikasi `Serial1.begin(9600)` diinisialisasi di dalam `setup()` untuk mengirim data ke ESP32.
- 2) Pengiriman Data Periodik: Sebuah fungsi pengiriman data (`kirimDataKeESP32()`) dibuat, yang memanfaatkan timer non-blocking (`millis()`). Interval pengiriman diatur pada 1000 ms (1 detik) untuk menyediakan pemantauan near real-time tanpa membanjiri (*flooding server*).
- 3) Protokol Pengiriman Data: Fungsi tersebut mengirimkan data sensor (Curah Hujan V0, Debit Sungai V1, Tinggi Air V2), status siaga (V9), serta status aktual setiap pompa (P1-P6 pada V3-V8) yang dibaca melalui `digitalRead()`. Data diformat menggunakan protokol string sederhana (contoh: “V0,10.5”), di mana setiap data dikirim pada baris baru (`println`).
- 4) Fungsi Penerimaan Data Dihilangkan: Karena sistem ini dirancang sebagai monitoring-only, fungsi untuk membaca data dari ESP32 (seperti `bacaPerintahDariESP32()`) dihilangkan untuk mencegah kontrol eksternal.

## Langkah 4: Pengujian dan Verifikasi Sistem

Tahap pengujian dilakukan untuk memvalidasi fungsionalitas sistem secara menyeluruh, dari pembacaan sensor hingga tampilan data di platform IoT.

1. Verifikasi Koneksi: Pengujian koneksi awal dilakukan dengan mengisolasi ESP32 dan menghubungkannya ke PC. Serial Monitor (*baud rate* 115200) digunakan untuk mengamati proses booting. Keberhasilan koneksi ditandai dengan munculnya pesan “*Blynk Connected!*”, yang mengonfirmasi bahwa kredensial WiFi dan Blynk valid.
2. Verifikasi Status *Online*: Setelah kedua mikrokontroler dinyalakan menggunakan suplai daya final (ESP32 dan Mega ditenagai secara terpisah dengan *common ground*), dasbor web Blynk pada *device* “ESP32 Cideng” diamati. Status perangkat diverifikasi telah berubah dari “*Offline*” menjadi “*Online*”.

3. Uji Fungsionalitas *Monitoring* (Sensor): Pengujian fungsionalitas dilakukan dengan memberikan *input* simulasi pada sensor (misalnya, mengubah jarak pada sensor ultrasonik). Perubahan nilai yang ditampilkan real-time pada LCD alat dibandingkan dengan data yang muncul pada widget (Label/Gauge) di dasbor web dan mobile Blynk. Hasil pengujian menunjukkan data berhasil diperbarui pada kedua platform dengan latensi sekitar 1 detik, sesuai dengan interval pengiriman yang ditetapkan.
4. Verifikasi Status Pompa (*Monitoring*): Skenario spesifik dijalankan pada Arduino Mega untuk memicu aktivasi pompa (misalnya, logika internal menyalakan Pompa 1). Status pin pompa (*HIGH/LOW*) yang terbaca oleh Mega diverifikasi telah berhasil dikirim dan ditampilkan dengan benar pada *widget* LED (V3-V8) di dasbor Blynk (LED menyala saat pompa aktif).
5. Verifikasi Status Siaga: Dilakukan simulasi kondisi bahaya (misalnya, input data ketinggian air melebihi ambang batas “BAHAYA”). Teks status yang dihasilkan oleh logika Arduino Mega (misalnya “BAHAYA” pada V9) diverifikasi telah tampil dengan benar dan lengkap pada *widget* “Label” di dasbor *web* dan *mobile*.
6. Pengujian Latensi: Pengujian dilakukan dengan cara memberikan perubahan *input* yang drastis pada satu sensor (misalnya debit sungai) dan secara bersamaan memulai *stopwatch*. *Stopwatch* dihentikan tepat pada saat *widget* (*Gauge/Label*) yang sesuai pada aplikasi Blynk di *smartphone* berubah nilainya. Pengujian ini dilakukan sebanyak 10 kali dalam kondisi koneksi WiFi yang stabil.

Dengan menyelesaikan keempat langkah ini, sistem kontrol pompa air telah berhasil diintegrasikan dengan platform IoT, sehingga siap untuk pemantauan dan analisis kinerja secara *real-time* dan jarak jauh.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil pengujian, analisis data, dan pembahasan dari prototipe sistem kontrol pompa air otomatis yang telah dirancang dan dibangun. Pembahasan akan mencakup pengujian fungsionalitas setiap komponen perangkat keras, analisis kinerja algoritma logika fuzzy, pengujian sistem secara terintegrasi dalam berbagai skenario simulasi, serta analisis sistem monitoring IoT.

#### **4.1. Pengujian Fungsionalitas Komponen**

##### **4.1.1. Pengujian Sensor**

###### **1. *Tipping Bucket***

Pengujian sensor *tipping bucket* dilakukan menggunakan air dengan volume terukur (misalnya 10 mL) secara perlahan diteteskan ke dalam corong *tipping bucket*. Jumlah jungkitan (*tip*) yang terjadi dihitung oleh mikrokontroler. Proses ini diulang beberapa kali untuk memastikan konsistensi. Hasil pengukuran sensor tersebut kemudian dibandingkan dengan perhitungan curah hujan manual pada gelas ukur. Pengujian sensor *tipping bucket* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Pengujian Sensor *Tipping Bucket*

Dalam proses kalibrasi sensor *tipping bucket*, penentuan luas area corong tangkapan menjadi faktor penting dalam perhitungan curah hujan (mm). Secara fisik, alat *tipping bucket* yang digunakan memiliki luas area corong sebesar 19,44 cm<sup>2</sup> (dengan dimensi 5,4 cm × 3,6 cm). Namun, pada tahap kalibrasi ini, digunakan luas area sebesar 78,5 cm<sup>2</sup>, yaitu luas setara dengan alat pembanding manual (gelas ukur berbentuk corong dengan diameter 10 cm).

Penyesuaian luas area ini dilakukan agar hasil perhitungan curah hujan antara sensor tipping bucket dan alat pembanding dapat dibandingkan secara langsung dalam satuan yang sepadan. Dengan demikian, setiap volume air (V) yang digunakan pada pengujian dikonversi menjadi nilai curah hujan (H) menggunakan persamaan 2.1.

Sensor *Tipping Bucket* secara konsisten menghasilkan 1 *tip* untuk setiap 0,172 mm air yang masuk, sesuai dengan luas area corong yang digunakan sebagai pembanding sebesar  $78,5 \text{ cm}^2$ . Sensor ini berfungsi dengan baik untuk memberikan data *input* curah hujan secara kuantitatif. Hasil pengujian sensor *tipping bucket* dapat dilihat pada Tabel 4.1.

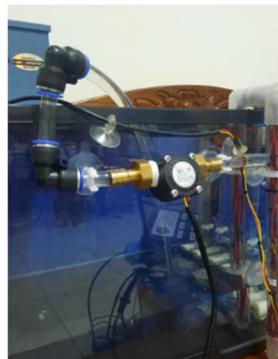
Tabel 4.1 Pengujian Sensor *Tipping Bucket*

| Pengujian        | Volume Air (mL) | Jumlah Tip | Curah Hujan (mm) | Tipping Bucket (mm) | Selisih (mm) | Error (%)   |
|------------------|-----------------|------------|------------------|---------------------|--------------|-------------|
| 1                | 5               | 4          | 63,69            | 69,00               | 0,05         | 8,33        |
| 2                | 10              | 8          | 127,39           | 120,00              | -0,07        | -5,80       |
| 3                | 15              | 12         | 191,08           | 200,00              | 0,09         | 4,67        |
| 4                | 20              | 15         | 254,78           | 275,00              | 0,20         | 7,94        |
| 5                | 25              | 18         | 318,47           | 344,00              | 0,26         | 8,02        |
| 6                | 30              | 22         | 382,17           | 413,00              | 0,31         | 8,07        |
| 7                | 35              | 25         | 445,86           | 464,00              | 0,18         | 4,07        |
| 8                | 40              | 28         | 509,55           | 516,00              | 0,06         | 1,27        |
| 9                | 45              | 32         | 573,25           | 585,00              | 0,12         | 2,05        |
| 10               | 50              | 35         | 636,94           | 636,00              | -0,01        | -0,15       |
| <b>Rata-rata</b> |                 |            |                  |                     | <b>0,12</b>  | <b>3,85</b> |

## 2. Sensor Water Flow YF-S201

Pengujian sensor *Water Flow* YF-S201 dilakukan dengan cara menghubungkannya ke sebuah pompa kecil dan selang. Air dengan volume yang diketahui (misalnya 1 liter) dialirkan melalui sensor. Jumlah pulsa yang dihasilkan oleh sensor dihitung oleh mikrokontroler dan dikonversi menjadi volume. Hasil pengukuran sensor tersebut kemudian dibandingkan dengan volume manual

menggunakan timer dan gelas ukur. Pengujian sensor *Water Flow* YF-S201 dapat dilihat pada Gambar 4.2.



Gambar 4.2 Pengujian Sensor *Water Flow* YF-S201

Kalibrasi dilakukan untuk mengetahui hubungan antara jumlah pulsa yang dihasilkan sensor *Water Flow* YF-S201 dengan volume air yang sebenarnya mengalir, sehingga diperoleh faktor konversi yang akurat. Proses kalibrasi dilakukan dengan cara mengalirkan air melalui sensor dan menampung keluaran air hingga mencapai volume tertentu (1 liter) yang diukur menggunakan gelas ukur. Selama proses tersebut, jumlah pulsa yang dihasilkan oleh sensor dihitung menggunakan mikrokontroler.

Pengujian dilakukan sebanyak lima kali pengulangan dengan hasil jumlah pulsa yang terukur untuk setiap satu liter air adalah 461, 463, 460, 459, dan 463. Dari hasil pengujian tersebut diperoleh nilai rata-rata jumlah pulsa untuk setiap liter air sebesar 461,2. Dengan demikian, dapat disimpulkan bahwa sensor *Water Flow* YF-S201 menghasilkan rata-rata 461,2 pulsa untuk setiap liter air yang mengalir. Nilai ini selanjutnya digunakan sebagai faktor kalibrasi sensor untuk konversi jumlah pulsa menjadi satuan debit (liter per menit) dalam sistem.

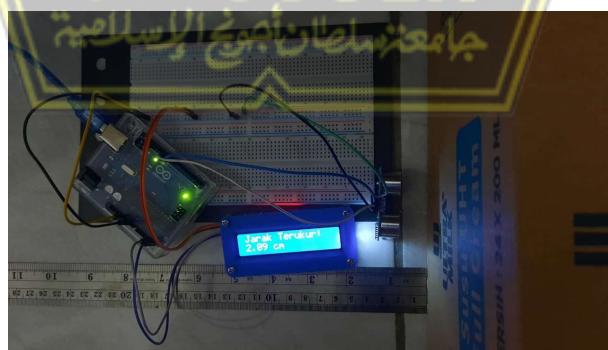
Setelah proses kalibrasi, pengujian menunjukkan rata-rata *error* pengukuran volume sebesar 6,41%. Sensor ini terbukti andal untuk mengukur laju aliran air sebagai input bagi sistem fuzzy. Hasil pengujian sensor *Water Flow* YF-S201 dapat dilihat pada Tabel 4.2.

Tabel 4.2 Pengujian Sensor *Water Flow* YF-S201

| Pengujian        | Gelas Ukur<br>(mL) | Water Flow<br>(mL) | Selisih<br>(mL) | Error<br>(%) |
|------------------|--------------------|--------------------|-----------------|--------------|
| 1                | 100                | 109                | 9               | 9,00         |
| 2                | 200                | 211                | 11              | 5,50         |
| 3                | 300                | 316                | 16              | 5,33         |
| 4                | 400                | 425                | 25              | 6,25         |
| 5                | 500                | 530                | 30              | 6,00         |
| 6                | 600                | 636                | 36              | 6,00         |
| 7                | 700                | 748                | 48              | 6,86         |
| 8                | 800                | 852                | 52              | 6,50         |
| 9                | 900                | 959                | 59              | 6,56         |
| 10               | 1000               | 1061               | 61              | 6,10         |
| <b>Rata-rata</b> |                    |                    | <b>34,7</b>     | <b>6,41</b>  |

### 3. Sensor Ultrasonik HC-SR04

Pengujian sensor Ultrasonik HC-SR04 dilakukan dengan membandingkannya dengan penggaris yang ditempatkan di depan kardus (sebagai acuan jarak). Pengujian ini mengukur jarak aktual dari sensor ke sisi kardus menggunakan alat ukur manual (penggaris), kemudian membandingkannya dengan data yang dibaca oleh sensor dan ditampilkan pada LCD. Gambar pengujian sensor Ultrasonik HC-SR04 dapat dilihat pada Gambar 4.3.



Gambar 4.3 Pengujian Sensor *Ultrasonic* HC-SR04

Hasilnya, dari sepuluh kali percobaan dengan jarak yang berbeda diperoleh rata-rata error pembacaan sebesar 1,58%. Nilai ini menunjukkan bahwa sensor

*ultrasonic* memiliki tingkat akurasi yang sangat baik untuk digunakan dalam sistem ini. Hasil pengujian sensor *Ultrasonic* HC-SR04 dapat dilihat pada Tabel 4.3.

Tabel 4.3 Pengujian Sensor *Ultrasonic* HC-SR04

| Pengujian        | Penggaris<br>(cm) | HC-SR04<br>(cm) | Selisih<br>(cm) | Error<br>(%) |
|------------------|-------------------|-----------------|-----------------|--------------|
| 1                | 2                 | 2,09            | 0,09            | 4,50         |
| 2                | 4                 | 4,56            | 0,56            | 14,00        |
| 3                | 6                 | 6,28            | 0,28            | 4,67         |
| 4                | 8                 | 8,21            | 0,21            | 2,63         |
| 5                | 10                | 9,57            | -0,43           | -4,30        |
| 6                | 12                | 11,3            | -0,7            | -5,83        |
| 7                | 14                | 13,98           | -0,02           | -0,14        |
| 8                | 16                | 16,49           | 0,49            | 3,06         |
| 9                | 18                | 17,63           | -0,37           | -2,06        |
| 10               | 20                | 19,86           | -0,14           | -0,70        |
| <b>Rata-rata</b> |                   |                 | <b>-0,003</b>   | <b>1,58</b>  |

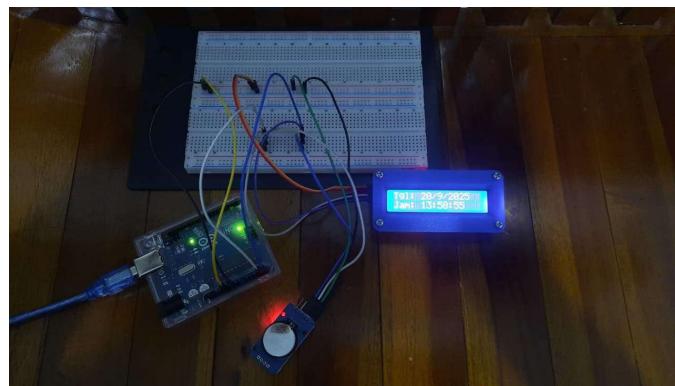
Setelah melakukan pengujian sensor-sensor input yang digunakan dalam penelitian ini, didapatkan rata-rata selisih masing-masing sensor yaitu sensor Tippung Bucket sebesar 3,85%, sensor Water Flow YF-S401 sebesar 8,82%, dan sensor Ultrasonik HC-SR04 sebesar 1,58%.

#### 4. Sensor RTC DS3231

Pengujian sensor RTC DS3231, yang berfungsi sebagai penunjuk waktu real-time, dilakukan dengan cara menghubungkannya ke mikrokontroler Arduino melalui jalur komunikasi I<sup>2</sup>C (pin SDA dan SCL). Tujuan pengujian adalah memastikan bahwa pencatatan jam, menit, dan detik yang ditampilkan pada Serial Monitor Arduino IDE sesuai dengan waktu aktual.

Prosedur pengujian dimulai dengan sinkronisasi waktu pada modul RTC menggunakan sketch Arduino, kemudian dilakukan pemantauan selama beberapa jam. Waktu aktual dibandingkan dengan jam referensi (misalnya jam digital atau jam tangan yang sudah diverifikasi). Pencatatan dilakukan setiap interval 10 menit

untuk melihat kemungkinan deviasi atau drift. Skema rangkaian pengujian RTC DS3231 dapat dilihat pada Gambar 4.4.



Gambar 4.4 Pengujian *Timer* Waktu Sensor RTC DS3231

Hasilnya, setelah pemantauan selama kurang lebih dua jam, perbedaan pembacaan antara RTC DS3231 dan jam referensi hanya sekitar  $\pm 1$  detik, yang masih berada dalam batas toleransi dan sesuai dengan spesifikasi. Nilai ini menunjukkan bahwa modul RTC DS3231 memiliki akurasi tinggi dan dapat diandalkan sebagai penentu waktu pada sistem ini. Hasil pengujian sensor RTC DS3231 dapat dilihat pada Tabel 4.4

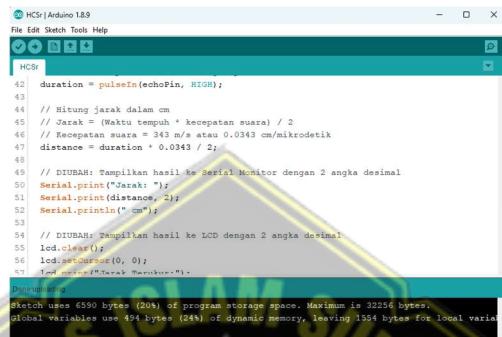
Tabel 4.4 Pengujian RTC DS3231

| No               | Waktu Aktual<br>(Jam:Menit:Detik) | RTC DS3231<br>(Jam:Menit:Detik) | Selisih<br>(Detik) |
|------------------|-----------------------------------|---------------------------------|--------------------|
| 1                | 16:14:10                          | 16:14:09                        | 1                  |
| 2                | 16:25:04                          | 16:25:03                        | 1                  |
| 3                | 16:35:58                          | 16:35:57                        | 1                  |
| 4                | 16:46:52                          | 16:46:51                        | 1                  |
| 5                | 16:57:46                          | 16:57:45                        | 1                  |
| 6                | 17:08:40                          | 17:08:39                        | 1                  |
| 7                | 17:19:34                          | 17:19:33                        | 1                  |
| 8                | 17:30:28                          | 17:30:27                        | 1                  |
| 9                | 17:41:22                          | 17:41:21                        | 1                  |
| 10               | 17:52:16                          | 17:52:15                        | 1                  |
| <b>Rata-rata</b> |                                   |                                 | <b>1,00</b>        |

#### 4.1.2. Pengujian Mikrokontroler

##### 1. Pengujian Arduino Mega 2560

Pengujian Arduino Mega 2560 dilakukan dengan cara menghubungkan dengan suatu sensor (misalkan sensor Ultrasonik). Sebuah program uji diunggah untuk membaca data dari setiap sensor secara kontinu dan menampilkannya pada *Serial Monitor*. Hasil pengujian Arduino Mega 2560 dapat dilihat pada Gambar 4.5 dan Gambar 4.6.



```
HCsr | Arduino 1.8.9
File Edit Sketch Tools Help
HCsr
42 duration = pulseIn(echoPin, HIGH);
43
44 // Hitung jarak dalam cm
45 // Jarak = (Waktu tenggu * kecakapan suara) / 2
46 // Kecakapan suara = 343 m/s atau 0.0343 cm/mikrodetik
47 distance = duration * 0.0343 / 2;
48
49 // DIUBAH: Tampilan hasil ke Serial Monitor dengan 2 angka desimal
50 Serial.print("Jarak: ");
51 Serial.print(distance, 2);
52 Serial.println(" cm");
53
54 // DIUBAH: Tampilan hasil ke LCD dengan 2 angka decimal
55 lcd.clear();
56 lcd.setCursor(0, 0);
57 lcd.print("Jarak: " + String(distance));
D:\sketches\HCsr\HCsr.ino
Sketch uses 6590 bytes (20%) of program storage space. Maximum is 32256 bytes.
Global variables use 494 bytes (24%) of dynamic memory, leaving 1554 bytes for local variables.
```

Gambar 4.5 Done-Compiling Program pada Aplikasi Arduino IDE



```
16:00:24.305 -> Jarak: 6.83 cm
16:00:24.866 -> Jarak: 7.24 cm
16:00:25.384 -> Jarak: 8.75 cm
16:00:25.552 -> Jarak: 10.60 cm
16:00:26.513 -> Jarak: 9.55 cm
16:00:27.029 -> Jarak: 10.15 cm
16:00:27.593 -> Jarak: 10.62 cm
16:00:28.058 -> Jarak: 11.20 cm
16:00:28.671 -> Jarak: 10.51 cm
16:00:29.239 -> Jarak: 10.50 cm
16:00:30.745 -> Jarak: 10.60 cm
16:00:30.310 -> Jarak: 10.96 cm
16:00:30.870 -> Jarak: 9.01 cm
16:00:31.386 -> Jarak: 10.26 cm
16:00:31.951 -> Jarak: 9.91 cm
```

Gambar 4.6 Serial Monitor Pengujian Arduino Mega 2560

Hasilnya, Arduino Mega 2560 berhasil membaca dan mengolah data mentah dari sensor tanpa kendala. Program berhasil di upload dan data yang ditampilkan pada Serial Monitor konsisten dengan input fisik yang diberikan pada sensor, membuktikan bahwa Arduino Mega 2560 dapat berfungsi sebagai unit akuisisi data yang andal.

##### 2. Pengujian NodeMCU ESP32

Pengujian ini bertujuan untuk memvalidasi kelayakan penggunaan modul NodeMCU ESP32 sebagai *co-processor* atau modem eksternal guna menyediakan fungsionalitas Wi-Fi untuk Arduino. Mengingat Arduino tidak memiliki kemampuan koneksi nirkabel secara internal, diperlukan sebuah jembatan data

yang andal antara kedua perangkat. Pengujian ini berfokus pada verifikasi lapisan fisik dan data link melalui komunikasi serial (UART) sebelum implementasi protokol yang lebih kompleks.

Pengujian dilakukan dengan mengimplementasikan arsitektur komunikasi *master-slave*. Arduino bertindak sebagai *master* yang menginisiasi komunikasi dengan mengirimkan perintah, sementara ESP32 bertindak sebagai *slave* yang merespons perintah tersebut.

Hasilnya pada *serial monitor*, tercatat bahwa Arduino Uno secara konsisten mengirimkan perintah dan menerima balasan yang valid dari ESP32. Maka dapat disimpulkan bahwa komunikasi serial dua arah antara Arduino Uno dan modul ESP32 telah berhasil diimplementasikan. Dengan terbentuknya jembatan data ini, ESP32 terbukti dapat difungsikan sebagai modem Wi-Fi eksternal yang andal untuk proyek berbasis Arduino. Hasil pengujian NodeMCU ESP32 dapat dilihat pada Gambar 4.7.



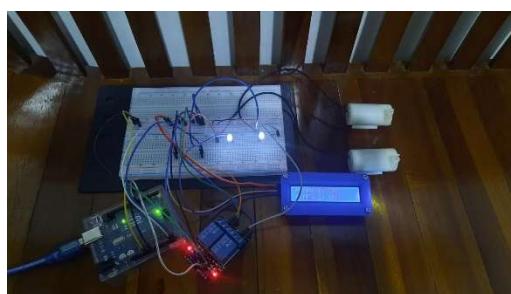
```
15:48:16.489 -> UNO <- Menerima balasan: ESP32: SIAP
15:48:21.170 -> UNO <- Mengirim perintah: 'CER_STATUS'
15:48:21.272 -> UNO <- Menerima balasan: ESP32: SIAP
15:48:26.304 -> UNO <- Mengirim perintah: 'CER_STATUS'
15:48:26.397 -> UNO <- Menerima balasan: ESP32: SIAP
15:48:31.119 -> UNO <- Mengirim perintah: 'CER_STATUS'
15:48:31.111 -> UNO <- Menerima balasan: ESP32: SIAP
15:48:36.123 -> UNO <- Mengirim perintah: 'CER_STATUS'
15:48:36.423 -> UNO <- Menerima balasan: ESP32: SIAP
15:48:41.673 -> UNO <- Mengirim perintah: 'CER_STATUS'
15:48:41.768 -> UNO <- Menerima balasan: ESP32: SIAP
15:48:46.787 -> UNO <- Mengirim perintah: 'CER_STATUS'
15:48:46.881 -> UNO <- Menerima balasan: ESP32: SIAP
15:48:51.850 -> UNO <- Mengirim perintah: 'CER_STATUS'
15:48:51.962 -> UNO <- Menerima balasan: ESP32: SIAP
```

Gambar 4.7 Pengujian NodeMCU ESP32

#### 4.1.3. Pengujian Aktuator

##### 1. Pengujian Modul Relay dan Pompa DC

Pengujian ini dilakukan dengan cara menghubungkan *modul relay* dengan Pompa DC, yang kemudian dikendalikan oleh mikrokontroler. Sinyal digital sederhana (*HIGH/LOW*) dikirimkan ke pin *input relay* untuk menguji fungsi *ON/OFF* pada pompa. Pengujian *modul relay* dapat dilihat pada Gambar 4.8.



Gambar 4.8 Pengujian Modul Relay dan Pompa DC

Hasilnya, *modul relay* berhasil menerjemahkan sinyal dari mikrokontroler untuk mengaktifkan dan menonaktifkan Pompa DC. Pompa DC berhasil menyala saat diberi sinyal *HIGH* dan mati saat diberi sinyal *LOW*. Sistem aktuator ini berfungsi sesuai harapan dan siap untuk dikendalikan oleh logika fuzzy. Hasil pengujian *modul relay* dan pompa DC dapat dilihat pada Tabel 4.5.

Tabel 4.5 Pengujian Modul Relay dan Pompa DC

| No | Modul Relay |           | Pompa 1 | Pompa 2 | Keterangan |
|----|-------------|-----------|---------|---------|------------|
|    | Channel 1   | Channel 2 |         |         |            |
| 1  | LOW         | LOW       | OFF     | OFF     | Berhasil   |
| 2  | LOW         | HIGH      | OFF     | ON      | Berhasil   |
| 3  | HIGH        | LOW       | ON      | OFF     | Berhasil   |
| 4  | HIGH        | HIGH      | ON      | ON      | Berhasil   |

## 4.2. Analisis Sistem Kontrol Logika Fuzzy

### 4.2.1. Fuzzifikasi

Fuzzifikasi merupakan pemetaan nilai *crisp*/nilai tegas kedalam variabel fuzzy set serta menentukan derajat keanggotaan didalam *fuzzy set*. Dalam analisis fuzzifikasi ada variabel *input* yaitu: Curah Hujan, Debit Sungai, dan Tinggi Muka Air serta *output* variabel yaitu Pompa 1 sampai Pompa 6.

#### 1. Curah Hujan

Penghitungan fuzzifikasi Curah Hujan dengan nilai 100 mm:

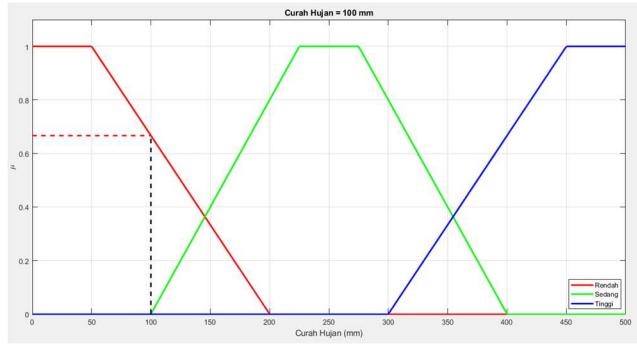
$$\mu_{\text{Rendah}}[100] = \frac{200 - 100}{150} ; 50 < 100 < 200$$

$$= 0,67$$

$$\mu_{\text{Sedang}}[100] = 0 ; 100 \leq 100$$

$$\mu_{\text{Tinggi}}[100] = 0 ; 100 \leq 300$$

Perolehan perhitungan diatas dapat disimpulkan masing-masing derajat keanggotaan tiap nilai linguistik yaitu  $\mu_{\text{Rendah}}[100]=0,5$ ;  $\mu_{\text{Sedang}}[100]=0$ ; dan  $\mu_{\text{Tinggi}}[100]=0$ . Maka derajat keanggotaan fuzzy pada nilai 100 adalah Rendah. Gambar fungsi keanggotaan fuzzy variabel Curah Hujan dengan nilai 100 dapat dilihat pada Gambar 4.9.



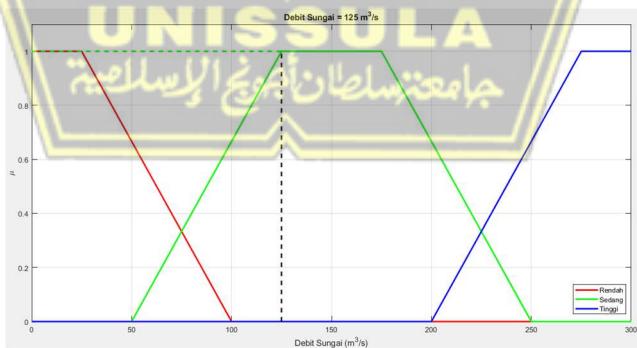
Gambar 4.9 Fuzzifikasi Curah Hujan Rendah

## 2. Debit Sungai

Penghitungan fuzzifikasi Debit Sungai dengan nilai 125 mm:

$$\begin{aligned}
 \mu_{\text{Rendah}}[125] &= 0 & ; 125 \geq 100 \\
 \mu_{\text{Sedang}}[125] &= \frac{125 - 50}{75} & ; 50 < 125 \leq 125 \\
 &= 1 \\
 \mu_{\text{Tinggi}}[125] &= 0 & ; 125 \leq 200
 \end{aligned}$$

Perolehan perhitungan diatas dapat disimpulkan masing-masing derajat keanggotaan tiap nilai linguistik yaitu  $\mu_{\text{Rendah}}[125]=0$ ;  $\mu_{\text{Sedang}}[125]=1$ ; dan  $\mu_{\text{Tinggi}}[125]=0$ . Maka derajat keanggotaan fuzzy pada nilai 125 adalah Sedang. Gambar fungsi keanggotaan fuzzy variabel Debit Sungai dengan nilai 125 dapat dilihat pada Gambar 4.10.



Gambar 4.10 Fuzzifikasi Debit Sungai Sedang

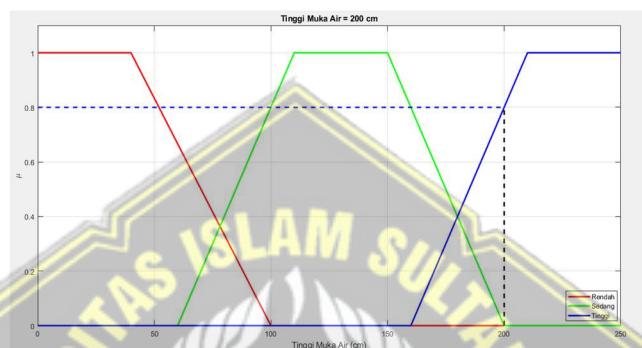
## 3. Tinggi Muka Air

Penghitungan fuzzifikasi Tinggi Muka Air dengan nilai 200 mm:

$$\begin{aligned}
 \mu_{\text{Rendah}}[200] &= 0 & ; 200 \geq 100 \\
 \mu_{\text{Sedang}}[200] &= 0 & ; 200 \geq 200
 \end{aligned}$$

$$\begin{aligned}\mu_{\text{Tinggi}}[200] &= \frac{200 - 160}{50} & ; 160 < 200 < 210 \\ &= 0,8\end{aligned}$$

Perolehan perhitungan diatas dapat disimpulkan masing-masing derajat keanggotaan tiap nilai linguistik yaitu  $\mu_{\text{Rendah}}[200]=0$ ;  $\mu_{\text{Sedang}}[200]=0$ ; dan  $\mu_{\text{Tinggi}}[200]=0,8$ . Maka derajat keanggotaan fuzzy pada nilai 200 adalah Tinggi. Gambar fungsi keanggotaan fuzzy variabel Tinggi Muka Air dengan nilai 200 dapat dilihat pada Gambar 4.11.



Gambar 4.11 Fuzzifikasi Tinggi Muka Air Tinggi

### Fuzzifikasi Output

Hasil perhitungan yang diperoleh dari input Curah Hujan 100 mm, Debit Sungai 125 mm, dan Tinggi Muka Air 200 mm dapat dilihat pada Gambar 4.12.



Gambar 4.12 Fuzzifikasi Output Simulasi Perhitungan

#### 4.2.2. Fuzzy Rule Base

Terdapat 27 aturan fuzzy yang disusun sebagai basis pengendalian untuk menentukan perilaku Pompa 1 hingga Pompa 6. Mesin inferensi ini menggunakan fungsi minimum (min) sebagai metode implikasi pada setiap aturan berbentuk IF–

THEN. Oleh karena itu, ketika perhitungan dilakukan secara manual, bentuk fungsi fuzzy rule base dapat dinyatakan sebagai berikut:

1. IF (Curah Hujan is Rendah) AND (Debit Sungai is Rendah) AND (Tinggi Muka Air is Rendah) THEN (Pompa 1 is OFF)(Pompa 2 is OFF)(Pompa 3 is OFF)(Pompa 4 is OFF)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_1 &= \mu_{\text{Curah Hujan is Rendah}}[100] \wedge \mu_{\text{Debit Sungai is Rendah}}[0] \\
 &\quad \mu_{\text{Tinggi Muka Air is Rendah}}[0] \\
 &= \min(0,67; 0; 0) \\
 &= 0
 \end{aligned}$$

2. IF (Curah Hujan is Rendah) AND (Debit Sungai is Rendah) AND (Tinggi Muka Air is Sedang) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is OFF)(Pompa 4 is OFF)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_2 &= \mu_{\text{Curah Hujan is Rendah}}[100] \wedge \mu_{\text{Debit Sungai is Rendah}}[0] \\
 &\quad \mu_{\text{Tinggi Muka Air is Sedang}}[0] \\
 &= \min(0,67; 0; 0) \\
 &= 0
 \end{aligned}$$

3. IF (Curah Hujan is Rendah) AND (Debit Sungai is Rendah) AND (Tinggi Muka Air is Tinggi) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_3 &= \mu_{\text{Curah Hujan is Rendah}}[100] \wedge \mu_{\text{Debit Sungai is Rendah}}[0] \\
 &\quad \mu_{\text{Tinggi Muka Air is Tinggi}}[200] \\
 &= \min(0,67; 0; 0,8) \\
 &= 0
 \end{aligned}$$

4. IF (Curah Hujan is Rendah) AND (Debit Sungai is Sedang) AND (Tinggi Muka Air is Rendah) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is OFF)(Pompa 4 is OFF)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_4 &= \mu_{\text{Curah Hujan is Rendah}}[100] \wedge \mu_{\text{Debit Sungai is Sedang}}[125] \wedge \mu_{\text{Tinggi Muka Air is Rendah}}[0] \\
 &= \min(0,67; 1; 0) \\
 &= 0
 \end{aligned}$$

5. IF (Curah Hujan is Rendah) AND (Debit Sungai is Sedang) AND (Tinggi Muka Air is Sedang) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_5 &= \mu \text{Curah Hujan is Rendah} [100] \wedge \mu \text{Debit Sungai is Sedang} \\
 &\quad [125] \wedge \mu \text{Tinggi Muka Air is Sedang} [0] \\
 &= \min (0,67 ; 1 ; 0) \\
 &= 0
 \end{aligned}$$

6. IF (Curah Hujan is Rendah) AND (Debit Sungai is Sedang) AND (Tinggi Muka Air is Tinggi) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}
 \alpha_6 &= \mu \text{Curah Hujan is Rendah} [100] \wedge \mu \text{Debit Sungai is Sedang} \\
 &\quad [125] \wedge \mu \text{Tinggi Muka Air is Tinggi} [200] \\
 &= \min (0,67 ; 1 ; 0,8) \\
 &= 0,67
 \end{aligned}$$

7. IF (Curah Hujan is Rendah) AND (Debit Sungai is Tinggi) AND (Tinggi Muka Air is Rendah) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_7 &= \mu \text{Curah Hujan is Rendah} [100] \wedge \mu \text{Debit Sungai is Tinggi} [0] \wedge \\
 &\quad \mu \text{Tinggi Muka Air is Rendah} [0] \\
 &= \min (0,67 ; 0 ; 0) \\
 &= 0
 \end{aligned}$$

8. IF (Curah Hujan is Rendah) AND (Debit Sungai is Tinggi) AND (Tinggi Muka Air is Sedang) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}
 \alpha_8 &= \mu \text{Curah Hujan is Rendah} [100] \wedge \mu \text{Debit Sungai is Tinggi} [0] \wedge \\
 &\quad \mu \text{Tinggi Muka Air is Sedang} [0] \\
 &= \min (0,67 ; 0 ; 0) \\
 &= 0
 \end{aligned}$$

9. IF (Curah Hujan is Rendah) AND (Debit Sungai is Tinggi) AND (Tinggi Muka Air is Tinggi) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\alpha_9 = \mu \text{Curah Hujan is Rendah} [100] \wedge \mu \text{Debit Sungai is Tinggi} [0]$$

$$\begin{aligned}
 & \mu \text{ Tinggi Muka Air is Tinggi [200]} \\
 &= \min (0,67 ; 0 ; 0,8) \\
 &= 0
 \end{aligned}$$

10. IF (Curah Hujan is Sedang) AND (Debit Sungai is Rendah) AND (Tinggi Muka Air is Rendah) THEN (Pompa 1 is ON)(Pompa 2 is OFF)(Pompa 3 is OFF)(Pompa 4 is OFF)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_{10} &= \mu \text{ Curah Hujan is Sedang [0]} \wedge \mu \text{ Debit Sungai is Rendah [0]} \wedge \\
 &\quad \mu \text{ Tinggi Muka Air is Rendah [0]} \\
 &= \min (0 ; 0 ; 0) \\
 &= 0
 \end{aligned}$$

11. IF (Curah Hujan is Sedang) AND (Debit Sungai is Rendah) AND (Tinggi Muka Air is Sedang) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is OFF)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_{11} &= \mu \text{ Curah Hujan is Sedang [0]} \wedge \mu \text{ Debit Sungai is Rendah [0]} \wedge \\
 &\quad \mu \text{ Tinggi Muka Air is Sedang [0]} \\
 &= \min (0 ; 0 ; 0) \\
 &= 0
 \end{aligned}$$

12. IF (Curah Hujan is Sedang) AND (Debit Sungai is Rendah) AND (Tinggi Muka Air is Tinggi) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}
 \alpha_{12} &= \mu \text{ Curah Hujan is Sedang [0]} \wedge \mu \text{ Debit Sungai is Rendah [0]} \wedge \\
 &\quad \mu \text{ Tinggi Muka Air is Tinggi [200]} \\
 &= \min (0 ; 0 ; 0,8) \\
 &= 0
 \end{aligned}$$

13. IF (Curah Hujan is Sedang) AND (Debit Sungai is Sedang) AND (Tinggi Muka Air is Rendah) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is OFF)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_{13} &= \mu \text{ Curah Hujan is Sedang [0]} \wedge \mu \text{ Debit Sungai is Sedang [125]} \wedge \mu \text{ Tinggi Muka Air is Rendah [0]} \\
 &= \min (0 ; 1 ; 0) \\
 &= 0
 \end{aligned}$$

14. IF (Curah Hujan is Sedang) AND (Debit Sungai is Sedang) AND (Tinggi Muka Air is Sedang) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_{14} &= \mu \text{Curah Hujan is Sedang [0]} \wedge \mu \text{Debit Sungai is Sedang [125]} \wedge \mu \text{Tinggi Muka Air is Sedang [0]} \\
 &= \min(0; 1; 0) \\
 &= 0
 \end{aligned}$$

15. IF (Curah Hujan is Sedang) AND (Debit Sungai is Sedang) AND (Tinggi Muka Air is Tinggi) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}
 \alpha_{15} &= \mu \text{Curah Hujan is Sedang [0]} \wedge \mu \text{Debit Sungai is Sedang [125]} \wedge \mu \text{Tinggi Muka Air is Tinggi [200]} \\
 &= \min(0; 1; 0,8) \\
 &= 0
 \end{aligned}$$

16. IF (Curah Hujan is Sedang) AND (Debit Sungai is Tinggi) AND (Tinggi Muka Air is Rendah) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is OFF).

$$\begin{aligned}
 \alpha_{16} &= \mu \text{Curah Hujan is Sedang [0]} \wedge \mu \text{Debit Sungai is Tinggi [0]} \wedge \mu \text{Tinggi Muka Air is Rendah [0]} \\
 &= \min(0; 0; 0) \\
 &= 0
 \end{aligned}$$

17. IF (Curah Hujan is Sedang) AND (Debit Sungai is Tinggi) AND (Tinggi Muka Air is Sedang) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}
 \alpha_{17} &= \mu \text{Curah Hujan is Sedang [0]} \wedge \mu \text{Debit Sungai is Tinggi [0]} \wedge \mu \text{Tinggi Muka Air is Sedang [0]} \\
 &= \min(0; 0; 0) \\
 &= 0
 \end{aligned}$$

18. IF (Curah Hujan is Sedang) AND (Debit Sungai is Tinggi) AND (Tinggi Muka Air is Tinggi) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\alpha_{18} = \mu \text{Curah Hujan is Sedang [0]} \wedge \mu \text{Debit Sungai is Tinggi [0]} \wedge \mu \text{Tinggi Muka Air is Tinggi [0]}$$

$$\mu \text{ Tinggi Muka Air is Tinggi} [200]$$

$$= \min (0 ; 0 ; 0,8)$$

$$= 0$$

19. IF (Curah Hujan is Tinggi) AND (Debit Sungai is Rendah) AND (Tinggi Muka Air is Rendah) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is OFF)(Pompa 4 is OFF)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}\alpha_{19} &= \mu \text{ Curah Hujan is Tinggi} [0] \wedge \mu \text{ Debit Sungai is Rendah} [0] \wedge \\ &\quad \mu \text{ Tinggi Muka Air is Rendah} [0] \\ &= \min (0 ; 0 ; 0) \\ &= 0\end{aligned}$$

20. IF (Curah Hujan is Tinggi) AND (Debit Sungai is Rendah) AND (Tinggi Muka Air is Sedang) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}\alpha_{20} &= \mu \text{ Curah Hujan is Tinggi} [0] \wedge \mu \text{ Debit Sungai is Rendah} [0] \wedge \\ &\quad \mu \text{ Tinggi Muka Air is Sedang} [0] \\ &= \min (0 ; 0 ; 0) \\ &= 0\end{aligned}$$

21. IF (Curah Hujan is Tinggi) AND (Debit Sungai is Rendah) AND (Tinggi Muka Air is Tinggi) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}\alpha_{21} &= \mu \text{ Curah Hujan is Tinggi} [0] \wedge \mu \text{ Debit Sungai is Rendah} [0] \wedge \\ &\quad \mu \text{ Tinggi Muka Air is Tinggi} [200] \\ &= \min (0 ; 0 ; 0,8) \\ &= 0\end{aligned}$$

22. IF (Curah Hujan is Tinggi) AND (Debit Sungai is Sedang) AND (Tinggi Muka Air is Rendah) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is OFF)(Pompa 6 is OFF).

$$\begin{aligned}\alpha_{22} &= \mu \text{ Curah Hujan is Tinggi} [0] \wedge \mu \text{ Debit Sungai is Sedang} [125] \\ &\quad \wedge \mu \text{ Tinggi Muka Air is Rendah} [0] \\ &= \min (0 ; 1 ; 0) \\ &= 0\end{aligned}$$

23. IF (Curah Hujan is Tinggi) AND (Debit Sungai is Sedang) AND (Tinggi Muka Air is Sedang) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}
 \alpha_{23} &= \mu \text{Curah Hujan is Tinggi [0]} \wedge \mu \text{Debit Sungai is Sedang [125]} \\
 &\quad \wedge \mu \text{Tinggi Muka Air is Sedang [0]} \\
 &= \min (0 ; 1 ; 0) \\
 &= 0
 \end{aligned}$$

24. IF (Curah Hujan is Tinggi) AND (Debit Sungai is Sedang) AND (Tinggi Muka Air is Tinggi) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}
 \alpha_{24} &= \mu \text{Curah Hujan is Tinggi [0]} \wedge \mu \text{Debit Sungai is Sedang [125]} \\
 &\quad \wedge \mu \text{Tinggi Muka Air is Tinggi [200]} \\
 &= \min (0 ; 1 ; 0,8) \\
 &= 0
 \end{aligned}$$

25. IF (Curah Hujan is Tinggi) AND (Debit Sungai is Tinggi) AND (Tinggi Muka Air is Rendah) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}
 \alpha_{25} &= \mu \text{Curah Hujan is Tinggi [0]} \wedge \mu \text{Debit Sungai is Tinggi [0]} \wedge \\
 &\quad \mu \text{Tinggi Muka Air is Rendah [0]} \\
 &= \min (0 ; 0 ; 0) \\
 &= 0
 \end{aligned}$$

26. IF (Curah Hujan is Tinggi) AND (Debit Sungai is Tinggi) AND (Tinggi Muka Air is Sedang) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\begin{aligned}
 \alpha_{26} &= \mu \text{Curah Hujan is Tinggi [0]} \wedge \mu \text{Debit Sungai is Tinggi [0]} \wedge \\
 &\quad \mu \text{Tinggi Muka Air is Sedang [0]} \\
 &= \min (0 ; 0 ; 0) \\
 &= 0
 \end{aligned}$$

27. IF (Curah Hujan is Tinggi) AND (Debit Sungai is Tinggi) AND (Tinggi Muka Air is Tinggi) THEN (Pompa 1 is ON)(Pompa 2 is ON)(Pompa 3 is ON)(Pompa 4 is ON)(Pompa 5 is ON)(Pompa 6 is ON).

$$\alpha_{27} = \mu \text{Curah Hujan is Tinggi [0]} \wedge \mu \text{Debit Sungai is Tinggi [0]} \wedge$$

$$\begin{aligned}
 \mu \text{ Tinggi Muka Air is Tinggi [200]} \\
 = \min (0 ; 0 ; 0,8) \\
 = 0
 \end{aligned}$$

#### 4.2.3. Defuzzifikasi

Proses defuzzifikasi merupakan tahap krusial yang menerjemahkan hasil inferensi fuzzy menjadi keputusan kontrol yang tegas. Penelitian ini menggunakan metode *Center Average Defuzzifier* untuk menghitung nilai output akhir. Analisis pada sub-bab ini akan menjabarkan penerapan metode tersebut pada salah satu studi kasus pengujian untuk memvalidasi proses pengambilan keputusan sistem. Proses perhitungan defuzzifikasi dilakukan dengan menggunakan Persaman 2.7 sebagai berikut:

$$\begin{aligned}
 Z &= \frac{\sum \alpha_i z_i}{\sum \alpha_i} \\
 &= \frac{(0 \times 0) + (0 \times 2) + (0 \times 5) + (0 \times 2) + (0 \times 4) + (0,5 \times 6) + (0 \times 4)}{0 + 0 + 0 + 0 + 0 + 0,67 + 0} \\
 &\quad \frac{(0 \times 6) + (0 \times 6) + (0 \times 1) + (0 \times 3) + (0 \times 6) + (0 \times 3) + (0 \times 5)}{0 + 0 + 0 + 0 + 0 + 0 + 0} \\
 &\quad \frac{(0 \times 6) + (0 \times 5) + (0 \times 6) + (0 \times 6) + (0 \times 2) + (0 \times 4) + (0 \times 6)}{0 + 0 + 0 + 0 + 0 + 0 + 0} \\
 &\quad \frac{(0 \times 4) + (0 \times 6) + (0 \times 6) + (0 \times 6) + (0 \times 6) + (0 \times 6)}{0 + 0 + 0 + 0 + 0 + 0} \\
 &= \frac{3}{0,67} \\
 &= 4,47
 \end{aligned}$$

Hasil defuzzifikasi (Z) adalah 4,47. Karena jumlah pompa harus berupa bilangan bulat, nilai ini dibulatkan ke bilangan bulat terdekat:

$$N = \text{round}(4,47) = 4$$

Hasil akhir yang diperoleh adalah 4, sehingga aksi yang dilakukan adalah 4 unit pompa *ON*. Hasil perhitungan menunjukkan bahwa untuk kondisi siaga, misalnya saat Tinggi Muka Air telah memasuki level “Tinggi” namun Curah Hujan dan Debit Sungai masih berada pada level “Sedang”, sistem secara cerdas memutuskan untuk mengaktifkan 4 dari 6 pompa. Keputusan ini sangat logis dan menunjukkan kemampuan sistem untuk merespons secara proporsional tanpa reaksi yang

berlebihan. Ini adalah cerminan dari kemampuan interpolasi logika fuzzy, di mana sistem tidak langsung melompat ke respons maksimal (6 pompa) hanya karena satu variabel input mencapai level tinggi. Sistem tidak hanya memilih antara respons normal (2 pompa) atau respons darurat (6 pompa), tetapi secara cerdas menentukan level penanganan “Siaga” yang tepat. Nilai  $N = 4$  ini kemudian diteruskan ke modul rotasi untuk menentukan 4 pompa spesifik mana yang akan dinyalakan sesuai jadwal harianya. Analisis ini membuktikan bahwa metode defuzzifikasi yang dipilih mampu menghasilkan keputusan kontrol yang halus, proporsional, dan sesuai dengan kondisi lapangan yang dinamis.

Dengan menggunakan metode *Mean Squared Error* (MSE) dapat diketahui selisih rata-rata perhitungan yang dihasilkan. Perhitungan yang dihasilkan pada manual adalah 4, sedangkan hasil perhitungan yang dilakukan pada analisis Matlab adalah 4. Sehingga diperoleh hasil dari model kendali pompa air dengan kendali logika fuzzy Sugeno dengan menggunakan *Mean Squared Error* (MSE) seperti pada Persamaan 2.8 adalah sebagai berikut:

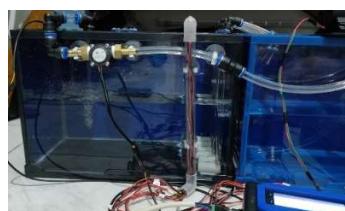
$$MSE = \frac{(4 - 4)^2}{1} = 0$$

Sehingga nilai rata-rata MSE adalah 0.

#### 4.3. Pengujian Sistem Terintegrasi

Pengujian ini dilakukan dengan menggabungkan seluruh komponen menjadi satu sistem utuh. Beberapa skenario kondisi hidrologis disimulasikan untuk mengamati respons sistem secara keseluruhan.

Pengujian ini dilakukan dengan menggabungkan seluruh komponen menjadi satu sistem utuh. Untuk keperluan validasi, sebuah prototipe fisik skala laboratorium telah dibangun, yang mengintegrasikan semua sensor, mikrokontroler (Arduino Mega 2560 dan NodeMCU ESP32), modul *relay*, dan pompa. Wujud fisik dari prototipe sistem yang telah terintegrasi dapat dilihat pada Gambar 4.13.



Gambar 4.13 Wujud Prototipe Sistem Kontrol Pompa Air Terintegrasi

Prototipe ini dirancang untuk mensimulasikan stasiun pompa air secara fungsional, di mana data dari sensor (curah hujan, debit sungai dan tinggi muka air) diproses oleh logika fuzzy untuk mengendalikan enam pompa DC secara otomatis, sementara data operasional dikirim ke dasbor Blynk. Pada prototipe inilah beberapa skenario kondisi hidrologis disimulasikan untuk mengamati respons sistem secara keseluruhan. Rentang input untuk setiap skenario dipilih berdasarkan area inti (nilai keanggotaan = 1.0) dari fungsi keanggotaan fuzzy yang telah didefinisikan. Hal ini dilakukan untuk memastikan bahwa pengujian memberikan hasil yang pasti, tidak ambigu, dan mewakili setiap aturan utama dari sistem.

#### 4.3.1. Studi Kasus

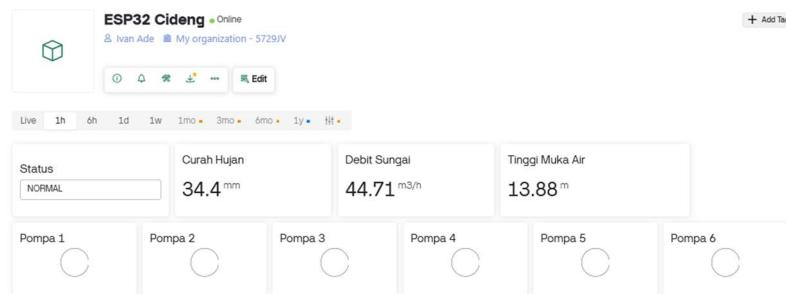
Untuk memvalidasi kinerja sistem secara menyeluruh, dilakukan pengujian berdasarkan empat skenario yang mewakili kondisi hidrologis dari normal hingga darurat. Pengujian ini bertujuan untuk menganalisis respons logika fuzzy dan kesesuaian output aktuator (pompa) terhadap kondisi input yang disimulasikan.

##### 1. Skenario 1: Kondisi Normal (Semua Pompa OFF)

Skenario ini merepresentasikan kondisi harian di mana tidak ada ancaman banjir, dengan ketinggian air rendah, hujan intensitas rendah, dan debit sungai rendah. Untuk pengujian, digunakan *input* spesifik Curah Hujan = 34, Debit Sungai = 48, dan Tinggi Muka Air = 14. Berdasarkan input tersebut, hasil defuzzifikasi (Z) secara konsisten menghasilkan nilai yang mendekati 0,0. Setelah dibulatkan, nilai N = 0, yang diterjemahkan menjadi 0 unit pompa aktif. Sistem merespons dengan benar sesuai kondisi “Normal”, dan tidak ada pompa aktif. Hasil percobaan Skenario 1 dapat dilihat pada Gambar 4.14 dan Gambar 4.15.



Gambar 4.14 Tampilan LCD pada Kondisi Normal (Tidak Ada Pompa Aktif)



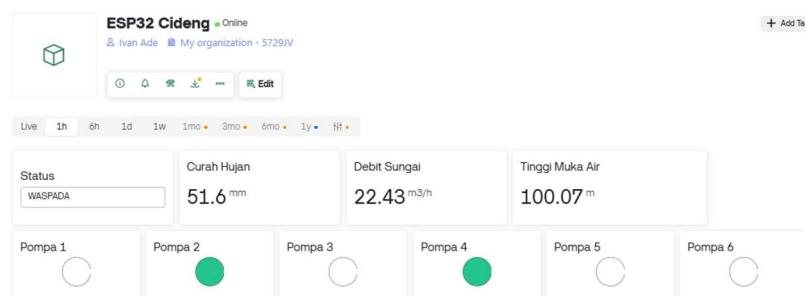
Gambar 4.15 Tampilan Dasbor Blynk pada Kondisi Normal (Tidak Ada Pompa Aktif)

## 2. Skenario 2: Kondisi Waspada (1-2 Pompa Aktif)

Skenario ini mensimulasikan kondisi di mana hujan mulai turun dengan intensitas sedang, atau ada sedikit peningkatan pada debit dan ketinggian air. Input spesifik yang digunakan adalah Curah Hujan = 52, Debit Sungai = 22, dan Tinggi Muka Air = 100. Dengan ketiga input tersebut dan berdasarkan perhitungan logika fuzzy, lalu diinterpretasikan sistem sebagai kondisi “Waspada” di mana jumlah pompa harus ditambah. Hasil defuzzifikasi (Z) secara konsisten menghasilkan nilai yang mendekati 2,0. Setelah dibulatkan nilai N = 2, yang mengaktifkan 2 unit pompa yang kemudian ditentukan oleh modul rotasi harian (misalnya Pompa 2 & 4 di Hari Kamis). Hasil percobaan Skenario 2 dapat dilihat pada Gambar 4.16 dan Gambar 4.17.



Gambar 4.16 Tampilan LCD pada Kondisi Waspada (2 Pompa Aktif)



Gambar 4.17 Tampilan Dasbor Blynk pada Kondisi Waspada (2 Pompa Aktif)

### 3. Skenario 3: Kondisi Siaga (3-4 Pompa Aktif)

Skenario ini merepresentasikan kondisi siaga banjir, seperti akibat hujan deras atau adanya banjir kiriman. Input spesifik yang digunakan adalah Curah Hujan = 103, Debit Sungai = 246, dan Tinggi Muka Air = 130. Kombinasi input ini memicu aturan-aturan pada level siaga. Hasil defuzzifikasi (Z) secara konsisten menghasilkan nilai yang mendekati 4,0. Setelah dibulatkan nilai  $N = 4$ , yang diterjemahkan menjadi 4 unit pompa aktif. Ini adalah keputusan cerdas yang menunjukkan kemampuan interpolasi Fuzzy, memberikan respons yang kuat untuk segera menurunkan level air sebelum mencapai kondisi bahaya. Percobaan Skenario 3 dapat dilihat pada Gambar 4.18 dan Gambar 4.19.



Gambar 4.18 Tampilan LCD pada Kondisi Siaga (4 Pompa Aktif)



Gambar 4.19 Tampilan Dasbor Blynk pada Kondisi Siaga (4 Pompa Aktif)

### 4. Skenario 4: Kondisi Bahaya (5-6 Pompa Aktif)

Skenario ini adalah kondisi darurat banjir di mana semua parameter input berada pada level maksimal. Input spesifik yang diuji adalah Curah Hujan = 413, Debit Sungai = 300, dan Tinggi Muka Air = 197. Dengan ketiga input berada di kategori "Tinggi", aturan-aturan dengan konsekuensi tertinggi (output 6 pompa) aktif dengan kekuatan penuh. Hasil defuzzifikasi (Z) secara konsisten menghasilkan nilai yang mendekati 6,0. Setelah dibulatkan nilai  $N = 6$ , yang mengaktifkan seluruh 6 unit pompa yang tersedia. Ini membuktikan bahwa sistem dapat

memberikan respons maksimal dan bertindak secara tepat dalam menangani kondisi paling kritis sekalipun untuk menguras air secepat mungkin. Percobaan Skenario 4 dapat dilihat pada Gambar 4.20 dan Gambar 4.21.



Gambar 4.20 Tampilan LCD pada Kondisi Bahaya (6 Pompa Aktif)



Gambar 4.21 Tampilan Dasbor Blynk pada Kondisi Bahaya (6 Pompa Aktif)

#### 4.3.2. Pengujian Fungsionalitas Rotasi Pompa

Setelah memverifikasi fungsionalitas pompa setiap kondisi, pengujian dilanjutkan pada salah satu fitur utama sistem, yaitu mekanisme rotasi pompa. Pengujian ini bertujuan untuk memastikan bahwa sistem mampu menjalankan jadwal rotasi harian secara akurat sesuai dengan logika yang telah dirancang guna mencapai pemerataan beban kerja antar pompa.

#### Hasil dan Analisis Pengujian Rotasi

Hasil pengujian selama satu siklus rotasi penuh (6 hari) dicatat dan disajikan pada Tabel 4.6. Hasil observasi dibandingkan dengan kombinasi pompa yang diharapkan sesuai dengan rencana operasi yang telah ditetapkan.

Tabel 4.6 Hasil Pengujian Mekanisme Rotasi Pompa (Kondisi N=2 Pompa Aktif)

| No | Hari   | Kombinasi Pompa yang Diharapkan | Kombinasi Pompa Aktif | Status |
|----|--------|---------------------------------|-----------------------|--------|
| 1  | Senin  | Pompa 2, Pompa 3                | Pompa 2, Pompa 3      | Sesuai |
| 2  | Selasa | Pompa 4, Pompa 5                | Pompa 4, Pompa 5      | Sesuai |
| 3  | Rabu   | Pompa 1, Pompa 6                | Pompa 1, Pompa 6      | Sesuai |
| 4  | Kamis  | Pompa 2, Pompa 4                | Pompa 2, Pompa 4      | Sesuai |
| 5  | Jumat  | Pompa 3, Pompa 5                | Pompa 3, Pompa 5      | Sesuai |
| 6  | Sabtu  | Pompa 1, Pompa 6                | Pompa 1, Pompa 6      | Sesuai |
| 7  | Minggu | Pompa 2, Pompa 3                | Pompa 2, Pompa 3      | Sesuai |

Hasil percobaan perubahan skema rotasi pompa dapat dilihat pada Gambar 4.22, Gambar 4.23 dan Gambar 4.24.



Gambar 4.22 Tampilan Aplikasi Blynk pada Hari Kamis (Pompa 2 dan Pompa 4 ON)



Gambar 4.23 Tampilan Aplikasi Blynk pada Hari Jumat (Pompa 3 dan Pompa 5 ON)



Gambar 4.24 Tampilan Aplikasi Blynk pada Hari Sabtu (Pompa 1 dan Pompa 6 ON)

Berdasarkan data pada Tabel 4.6, dapat disimpulkan bahwa mekanisme rotasi pompa berfungsi dengan sempurna. Sistem berhasil membaca informasi waktu dari modul RTC DS3231 dan menggunakan untuk memilih kombinasi pompa yang tepat sesuai dengan jadwal yang telah ditentukan untuk setiap harinya. Pada hari ketujuh, sistem juga terbukti berhasil kembali ke jadwal hari pertama, yang menandakan bahwa siklus rotasi dapat berjalan secara kontinu dan berkelanjutan.

Keberhasilan pengujian ini membuktikan bahwa sistem tidak hanya cerdas dalam pengambilan keputusan sesaat (melalui fuzzy), tetapi juga andal dalam manajemen operasional jangka panjang. Implementasi fitur ini secara signifikan meningkatkan nilai kebaruan dan potensi penerapan sistem di lapangan, karena dapat memperpanjang umur pakai aktuator dan mengurangi frekuensi pemeliharaan.

#### 4.4. Analisis Integrasi dan Monitoring IoT (Blynk)

Setelah memverifikasi fungsionalitas sistem kontrol secara internal, pengujian dilanjutkan pada aspek eksternal, yaitu kemampuan sistem untuk berkomunikasi dengan *platform Internet of Things* (IoT). Pengujian ini bertujuan untuk menganalisis kinerja integrasi antara mikrokontroler NodeMCU ESP32 dengan server dan aplikasi Blynk, yang berfungsi sebagai antarmuka monitoring jarak jauh. Dua parameter utama yang dievaluasi adalah keakuratan data yang ditampilkan dan kecepatan transmisi data (latensi). Hasil pengujian integrasi dengan *platform* Blynk menunjukkan kinerja yang sangat baik dan sesuai dengan yang diharapkan, yaitu berupa:

## 1. Keakuratan Data

Perbandingan data antara pembacaan lokal (LCD) dan tampilan jarak jauh (Blynk) menunjukkan tidak ada perbedaan nilai (*error* 0%). Data numerik dari sensor dan status logika (*ON/OFF*) dari setiap pompa berhasil ditransmisikan dan ditampilkan dengan akurat, seperti yang dirangkum pada Tabel 4.7.

Tabel 4.7 Perbandingan Keakuratan Data Lokal vs. Blynk

| No | Parameter       | Nilai Aktual<br>(LCD) | Nilai Terbaca (Blynk) |          | Status |
|----|-----------------|-----------------------|-----------------------|----------|--------|
|    |                 |                       | Website               | Aplikasi |        |
| 1  | Curah Hujan     | 103                   | 103,20                | 103,20   | Sesuai |
| 2  | Debit Sungai    | 246                   | 246,31                | 246,31   | Sesuai |
| 3  | Tinggi Muka Air | 130                   | 129,82                | 129,82   | Sesuai |
| 4  | Pompa 1         | ON                    | ON                    | ON       | Sesuai |
| 5  | Pompa 2         | ON                    | ON                    | ON       | Sesuai |
| 6  | Pompa 3         | ON                    | ON                    | ON       | Sesuai |
| 7  | Pompa 4         | OFF                   | OFF                   | OFF      | Sesuai |
| 8  | Pompa 5         | OFF                   | OFF                   | OFF      | Sesuai |
| 9  | Pompa 6         | ON                    | ON                    | ON       | Sesuai |
| 10 | Status Siaga    | Siaga                 | Siaga                 | Siaga    | Sesuai |

## 2. Latensi Transmisi Data

Dari 10 kali percobaan pengukuran waktu tunda, diperoleh latensi transmisi data dengan rentang antara 1,6 detik hingga 2,2 detik. Waktu tunda rata-rata yang tercatat adalah 1,8 detik seperti dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Pengujian Latensi *End-to-End* IoT (Sensor ke Dasbor Blynk)

| Percobaan | Waktu (detik) |
|-----------|---------------|
| 1         | 1,7           |
| 2         | 1,9           |
| 3         | 1,8           |
| 4         | 2,0           |
| 5         | 1,6           |
| 6         | 1,8           |

| Percobaan        | Waktu (detik) |
|------------------|---------------|
| 7                | 1,9           |
| 8                | 1,7           |
| 9                | 1,8           |
| 10               | 1,8           |
| <b>Rata-rata</b> | <b>1,8</b>    |

### 3. Stabilitas Koneksi

Selama pengujian stabilitas yang berlangsung selama 30 menit, koneksi antara NodeMCU ESP32 dan *server* Blynk tetap terhubung secara stabil tanpa adanya insiden terputus.

#### 4.4.1. Analisis Sistem IoT

Hasil pengujian membuktikan bahwa integrasi sistem dengan platform IoT Blynk berhasil dan berjalan dengan sangat baik. Keakuratan data 100% memastikan bahwa operator atau pemangku kepentingan dapat memercayai informasi yang ditampilkan pada dasbor untuk pengambilan keputusan.

Latensi rata-rata sebesar 1,8 detik dianggap sangat cepat dan dapat diterima untuk aplikasi monitoring banjir. Waktu tunda di bawah 2 detik memastikan bahwa informasi yang diterima oleh pengguna bersifat *near real-time* dan relevan dengan kondisi aktual di lapangan. Stabilitas koneksi yang terjaga juga menunjukkan keandalan sistem untuk operasional pemantauan jangka panjang.

Secara keseluruhan, implementasi IoT ini berhasil memenuhi tujuannya, yaitu menyediakan sebuah antarmuka monitoring jarak jauh yang akurat, cepat, dan andal, yang merupakan peningkatan signifikan dibandingkan pemantauan manual konvensional.

### 4.5. Analisis Perbandingan Kinerja

Salah satu tolok ukur utama keberhasilan penelitian ini adalah untuk membuktikan bahwa sistem yang diusulkan menawarkan peningkatan kinerja yang signifikan dibandingkan dengan sistem kontrol konvensional. Untuk analisis ini, sistem konvensional didefinisikan sebagai sistem *ON/OFF* sederhana yang umum digunakan, yang bersifat reaktif dan hanya bergantung pada satu ambang batas

ketinggian air (TMA) tanpa mempertimbangkan variabel antisipatif lainnya. Perbandingan kinerja tersebut dapat dilihat pada Tabel 4.9.

Tabel 4.9 Perbandingan Kinerja Sistem Kontrol

| No | Aspek Kinerja      | Sistem Kontrol Konvensional (ON/OFF)   | Sistem Kontrol Logika Fuzzy (Penelitian Ini)   |
|----|--------------------|--|--|
| 1  | Dasar Keputusan    | Satu variabel (tinggi muka air) dengan ambang batas kaku.  | Tiga variabel (curah hujan, debit sungai, dan tinggi muka air) dengan transisi halus.  |
| 2  | Waktu Respons      | Cenderung lambat karena menunggu air mencapai ambang batas tertentu.   | Lebih cepat dan antisipatif karena mempertimbangkan laju kenaikan (debit) dan curah hujan.   |
| 3  | Stabilitas Operasi | Rentan terhadap fluktuasi (efek hunting), di mana pompa sering menyala-mati saat level air berada di sekitar ambang batas. | Operasi lebih stabil dan halus. Jumlah pompa disesuaikan secara proporsional, sehingga dapat menghindari siklus ON-OFF yang tidak perlu. |
| 4  | Adaptabilitas      | Rendah karena tidak dapat beradaptasi dengan skenario berbeda (misalnya: banjir kiriman tanpa hujan).                      | Sangat adaptif, yaitu mampu memberikan respon yang berbeda untuk setiap kombinasi kondisi input.   |

#### 4.6. Pembahasan Hasil

Berdasarkan Tabel 4.9, terlihat jelas bahwa sistem yang diusulkan unggul secara komprehensif di setiap parameter evaluasi. Keunggulan paling fundamental terletak pada dasar kontrol dan sifat respons. Sistem konvensional bersifat reaktif; ia tidak akan melakukan tindakan apa pun sampai tinggi muka air secara fisik

menyentuh ambang batas kritis. Hal ini sangat berisiko di kondisi nyata, karena pada saat tinggi muka air sudah tinggi, mungkin sudah terlambat untuk mencegah genangan. Sebaliknya, sistem Fuzzy-IoT yang diusulkan bersifat antisipatif dan adaptif. Seperti dibuktikan pada pengujian skenario 2 (kondisi waspada), sistem fuzzy sudah mengaktifkan 2 unit pompa hanya karena melihat adanya peningkatan pada curah hujan dan debit sungai, bahkan sebelum tinggi muka air mencapai level mengkhawatirkan. Ini membuktikan sistem usulan memiliki responsivitas yang jauh lebih cepat dan mampu melakukan mitigasi lebih dini.

Perbedaan signifikan berikutnya adalah pada strategi aktivasi. Sistem konvensional yang sederhana hanya memiliki kondisi pompa *OFF* atau *ON* berdasarkan perkiraan saja dimana ini sangat tidak efisien. Sebaliknya dengan sistem fuzzy, seperti divalidasi pada Skenario 3 (kondisi siaga), mampu memberikan respons proporsional dengan mengaktifkan 4 dari 6 pompa. Kemampuan ini membuka potensi efisiensi energi yang besar, karena sistem tidak membuang daya untuk mengaktifkan seluruh pompa jika kondisi lapangannya tidak memerlukan respons maksimal.

Terakhir, dari sisi operasional dan manajemen aset, sistem usulan menawarkan dua keunggulan modern. Pertama, ketersediaan informasi yang dapat diakses melalui *platform* Blynk dimana operator dapat memantau kondisi stasiun pompa secara *real-time* dari jarak jauh. Ini menggantikan kebutuhan pemantauan lokal yang tidak efisien. Kedua, keandalan aktuator (pompa) jauh lebih terjamin. Seperti dibuktikan pada analisis distribusi beban kerja pada Tabel 3.4, implementasi rotasi 7 hari memastikan setiap pompa mendapatkan beban kerja yang hampir seimbang. Hal ini secara langsung akan memperpanjang umur pakai pompa dan mengurangi biaya pemeliharaan jangka panjang, sebuah aspek yang sering diabaikan oleh sistem konvensional.

Secara keseluruhan, analisis perbandingan ini mengonfirmasi bahwa sistem kontrol Fuzzy-IoT tidak hanya fungsional, tetapi juga menawarkan peningkatan kinerja yang holistik, mencakup aspek kecerdasan (adaptif), kecepatan (antisipatif), efisiensi (proporsional), dan keandalan (rotasi).

## **BAB V**

### **KESIMPULAN DAN SARAN**

Bab ini menyajikan kesimpulan dari keseluruhan hasil penelitian dan analisis yang telah dilakukan, serta merumuskan saran untuk pengembangan penelitian di masa mendatang.

#### **5.1. Kesimpulan**

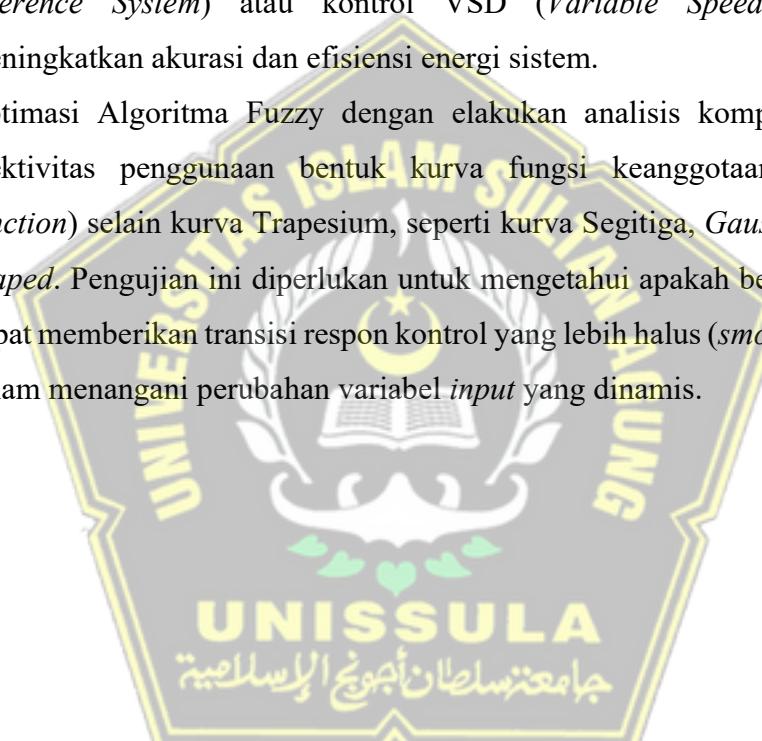
Berdasarkan hasil perancangan, implementasi, dan pengujian sistem kontrol otomatis pompa air berbasis logika fuzzy dan IoT, dapat disimpulkan bahwa:

1. Arsitektur sistem kontrol otomatis telah berhasil dirancang dan diimplementasikan dengan enam pompa air yang dikendalikan oleh tiga variabel masukan dinamis. Sistem menggunakan *dual-microcontroller* (Arduino Mega 2560 dan NodeMCU ESP32) yang terbukti stabil dan efektif dalam memisahkan tugas pemrosesan serta komunikasi IoT.
2. Penerapan 27 aturan logika fuzzy berhasil mengendalikan enam pompa secara proporsional dan adaptif, menghasilkan kombinasi *ON/OFF* yang sesuai dengan kondisi *input* (curah hujan, debit sungai, dan tinggi muka air). Validasi melalui empat skenario menunjukkan sistem bekerja sesuai logika yang dirancang.
3. Integrasi sistem dengan platform IoT Blynk memungkinkan pemantauan real-time seluruh sensor dan status pompa dengan latensi rata-rata 1,8 detik dan akurasi 100%.
4. Algoritma penjadwalan menggunakan modul RTC DS3231 berhasil diterapkan untuk manajemen rotasi pompa. Sistem terbukti dapat mengaktifkan kombinasi *relay* pompa yang berbeda bergantung pada pembacaan hari (Senin-Minggu), sehingga mencegah penggunaan pompa tertentu secara berlebihan dan memastikan keenam pompa beroperasi secara bergantian.

#### **5.2. Saran**

Berdasarkan hasil implementasi prototipe dan potensi pengembangan lebih lanjut, maka disarankan:

1. Mengembangkan rancangan PCB kustom untuk menggantikan rangkaian kabel jumper agar sistem lebih andal, rapi, dan tahan gangguan.
2. Menggunakan power supply terintegrasi yang mampu menyediakan semua kebutuhan tegangan (5V, 6V, dan 12V) dalam satu unit aman, serta menambahkan proteksi dan monitoring daya seperti sekring dan sensor INA219 agar lebih aman dan mudah dipantau melalui Blynk.
3. Penelitian lanjutan disarankan untuk implementasi skala penuh di stasiun pompa sebenarnya, melakukan validasi dengan data hidrologis riil, serta mengembangkan model prediktif berbasis ANFIS (*Adaptive Neuro Fuzzy Inference System*) atau kontrol VSD (*Variable Speed Drive*) guna meningkatkan akurasi dan efisiensi energi sistem.
4. Optimasi Algoritma Fuzzy dengan elakukan analisis komparatif terhadap efektivitas penggunaan bentuk kurva fungsi keanggotaan (*membership function*) selain kurva Trapesium, seperti kurva Segitiga, *Gaussian*, atau *Bell-shaped*. Pengujian ini diperlukan untuk mengetahui apakah bentuk kurva lain dapat memberikan transisi respon kontrol yang lebih halus (*smooth*) dan efisien dalam menangani perubahan variabel *input* yang dinamis.



## DAFTAR PUSTAKA

- [1] D. Takagi, E. M. F. Estupinan, and M. S. Babel, “A Review of Flood Management in Jakarta, Indonesia: The Impact of Structural and Non-Structural Measures,” *J. Southeast Asian Earth Sci.*, vol. 197, 2020.
- [2] M. Chaerul, Y. Putra, and H. Wigono, “Flood Risk Management in Jakarta: The Role of Structural and Non-structural Measures,” *J. Nat. Resour. Environ. Manag.*, vol. 11, pp. 245–256, 2021.
- [3] S. A. Sagala, F. Fathoni, and D. A. Puspasari, “A System Dynamics Approach to Assess Flood Vulnerability in Jakarta,” *IOP Conf. Ser. Earth Enviromental Sci.*, vol. 737, 2021.
- [4] H. A. Simanjuntak and P. A. Wibowo, “Evaluasi Kinerja Sistem Polder dalam Pengendalian Banjir di DKI Jakarta,” *J. Tek. Sipil dan Lingkungan*, vol. 5, pp. 89–98, 2021.
- [5] S. N. Sivanandam and S. N. Deepa, *Introduction to Fuzzy Logic using MATLAB*. Springer, 2007.
- [6] R. S. Khurmi and J. K. Gupta, *A Textbook of Machine Design*. New Delhi: Eurasia Publishing House, 2005.
- [7] F. Ghasemi, M. Kalatpour, and M. Moghiman, “A Review on Human Reliability Analysis Methods in Process Industries,” *Int. J. Qual. Reabil. Manag.*, vol. 30, pp. 1009–2034, 2013.
- [8] A. A. El-Fergany, “Adaptive Fuzzy Logic Control of Submersible Pumping System Fed by Photovoltaic Power,” *IET Renew. Power Gener.*, vol. 12, pp. 111–119, 2018.
- [9] B. R. P. Ferreira, D. A. De Souza, and R. A. S. De Souza, “Energy Efficiency in Pumping Systems: A Case Study in a Water Supply Company,” *IEEE Lat. Am. Trans.*, vol. 18, pp. 1069–1076.
- [10] R. Precup and H. Hellendoorn, “A Survey on Industrial Applications of Fuzzy Control,” *Comput. Ind.*, vol. 62, pp. 213–226, 2011.
- [11] I. Al-Mosawi, “Design and Implementation of a Fuzzy Logic Controller for a Water Pumping System,” *J. Eng. Sustain. Dev.*, vol. 22, no. 1–12, 2018.
- [12] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A Survey,”

- Comput. Networks*, vol. 54, pp. 2787–2805, 2010.
- [13] M. A. Al-Qaness, A. A. Ewees, and H. Fan, “IoT-Based Smart Water Management System,” *Comput. Electr. Eng.*, vol. 81, 2020.
  - [14] S. Rathore, P. K. Singh, and J. H. Park, “IoT-Based Infrastructure for Predictive Maintenance in Smart Pumping Systems,” *IEEE Trans. Ind. Informatics*, vol. 15, pp. 4047–4055, 2019.
  - [15] R. M. N. Huda, “Kajian Sistem Kendali Pompa Air Dengan Metode Logika Fuzzy Sebagai Upaya Memaksimalkan Pengendalian Banjir Kawasan Pelindo III,” Universitas Islam Sultan Agung Semarang, 2024.
  - [16] A. Syamsuddin, “Kajian Sistem Kendali Pintu Air Berbasis Logika Fuzzy Pada Bendung Gerak Kanal Banjir Barat Kota Semarang,” Universitas Islam Sultan Agung, 2024.
  - [17] B. Siregar and Muhammad Andre Apriansyah, “Sistem Pengendalian Water Pump Untuk Mengatur Tinggi Level Air Dengan Fuzzy,” *J. Softw. Eng. Comput. Intell.*, vol. 2, no. 01, pp. 1–8, 2024, doi: 10.36982/jseci.v2i01.3514.
  - [18] A. Baiki, I. Lutfi, and P. Risma, “Implementasi Sistem Kendali Pompa Otomatis Menggunakan Fuzzy Logic Sugeno Pada,” *J. Teliska*, vol. 18, no. I, pp. 39–47, 2025, doi: 10.5281/zenodo.15052688.
  - [19] Y. A. K. Utama, D. T. Hidayat, and N. Juniarto, “Implementasi Fuzzy Logic Pada Sistem Monitoring Suhu Pompa Air,” *J. Inform. Kaputama*, vol. 5, no. 1, pp. 154–163, 2021, doi: 10.59697/jik.v5i1.315.
  - [20] F. A. Rafsanjani, A. E. Wardoyo, H. Azizah, and A. Faruq, “Rancang Bangun Pintu Air Otomatis Berbasis Kontroler Logika Fuzzy Mamdani,” *J. Tek. Elektro dan Komputasi*, vol. 7, pp. 43–53, 2025.
  - [21] A. H. Arrahman, “Rancang Bangun Pengontrol Pintu Air Menggunakan Metode Fuzzy Berdasarkan Perubahan Ketinggian Air,” vol. 5, no. 1, pp. 1–11, 2019.
  - [22] I. Hermawan and D. Aulia Fachrudin, “Rancang Bangun Sistem Irigasi Cerdas Menggunakan Metode Fuzzy Rule-Based untuk Otomatisasi Pintu Air dan Pendekripsi Endapan,” *J. Komput. Terap.*, vol. 8, no. 1, pp. 1–11, 2022, doi: 10.35143/jkt.v8i1.5253.
  - [23] A. K. Singh and T. Tariq, “Intelligent Control of Irrigation Systems Using

- Fuzzy Logic Controller,” *Energies*, pp. 1642–1647, 2022.
- [24] S. Nduru, A. Al Hafiz, and D. H. Pane, “Implementasi Metode Fuzzy Berbasis Internet Of Things (IoT) Untuk Peringatan Dini Banjir,” *J. Sist. Komput. Triguna Dharma (JURSIK TGD)*, vol. 1, no. 1, pp. 26–33, 2022, doi: 10.53513/jursik.v1i1.4805.
- [25] A. Kurnia, R. Mustafa, and R. Listiana, “Rancang Bangun Sistem Peringatan Dini Dalam Pencegahan Banjir Akibat Luapan Sungai Berbasis Arduino Menggunakan Metode Fuzzy Logic,” *J. Informatics Electron. Eng.*, vol. 2, no. 2, pp. 60–65, 2022.
- [26] R. Rinaldi, T. Dewi, and Y. Oktarina, “Implementasi Fuzzy Logic dalam Mengendalikan Input dan Output pada Penyiraman dan Pemupukan Tanaman Otomatis Berbasis IoT,” *J. Appl. Smart Electr. Netw. Syst.*, vol. 3, no. 02, pp. 65–73, 2022, doi: 10.52158/jasens.v3i02.520.
- [27] S. Nurhadiyono, “Aplikasi Internet of Thing Berbasis Platform Firebase Dengan Menggunakan Mikrokontroler Esp 8266 Pada Sistem Peringatan Dini Bencana Banjir,” 2024, [Online]. Available: <http://repository.unissula.ac.id/38234/>
- [28] Sularso and H. Tahara, *Pompa dan Kompresor: Pemilihan, Pemakaian, dan Pemeliharaan*. Jakarta: Pradnya Paramita, 2013.
- [29] J. Karassik, J. P. Messina, P. Cooper, and C. C. Helad, *Pump Handbook*, 4th ed. New York: McGraw-Hill, 2008.
- [30] M. Chaerul, Y. T. W. Putra, and A. H. Wigono, “Flood Risk Management in Jakarta: The Role of Structural and Non-structural Measures,” *J. Nat. Resour. Environ. Manag.*, vol. 11, pp. 245–256, 2021.
- [31] L. G. Suwignyo, *Drainase Perkotaan*. Malang: UB Press, 2020.
- [32] “Profil Sistem Pengendalian Banjir dan Drainase Provinsi DKI Jakarta,” Jakarta, 2023.
- [33] R. J. Kodoatie and R. Sjarief, *Tata Ruang Air*. Yogyakarta: Andi Offset, 2010.
- [34] C. Asdak, *Hidrologi dan Pengelolaan Daerah Aliran Sungai*. Yogyakarta: Gadjah Mada University Press, 2010.
- [35] “The Project for Capacity Development of Jakarta Comprehensive Flood

- Management System in Indonesia,” Jakarta, 2018.
- [36] S. Kusumadewi and H. Purnomo, *Applikasi Logika Fuzzy untuk Mendukung Keputusan*. Yogyakarta: Graha Ilmu, 2010.
- [37] T. J. Ross, *Fuzzy Logic with Engineering Applications*, 4th ed. Chichester: John Wiley & Sons, 2016.
- [38] Mathworks, “Fuzzy Logic Toolbox - User’s Guide,” Natick, 2024. <https://www.mathworks.com/help/fuzzy/> (accessed Aug. 12, 2025).
- [39] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [40] M. Banzi and M. Shiloh, *Make: Getting Started with Arduino: The Open Source Electronics Prototyping Platform*, 3rd ed. Maker Media Inc., 2014.
- [41] S. Monk, *30 Arduino Projects for the Evil Genius*. New York: McGraw-Hill, 2010.
- [42] A. P. Sujalu, A. Y. P. Asih, and M. P. Biantary, *Instrumentasi Klimatologi dan Meteorologi*. Sleman: Zahir Publishing, 2022.
- [43] V. P. Bangun and K. L. Toruan, “Rancang Bangun Penakar Hujan Dengan Peringatan Dini Hujan Lebat Menggunakan Tipping Bucket dan Mikrokontroler ESP32,” *Emit. J. Tek. Elektro*, vol. 24, pp. 1–10, 2024.
- [44] A. F. Adzin, “Rancang Bangun Kalibrator Portable Penakar Hujan Tipping Bucket,” *Method. J. Tek. Inform. dan Sist. Inf.*, vol. 9, pp. 101–110, 2023.
- [45] D. H. S. S. Kalista, B. I. P. Siregar, and A. B. Pulungan, “Rancang Bangun Sistem Monitoring Pemakaian Air dan Kontrol Keran Air Otomatis Menggunakan Sensor YF-S201,” *J. Tek. Elektro dan Komput.*, vol. 7, pp. 23–30, 2020.
- [46] A. Hughes and B. Drury, *Electric Motors and Drivers: Fundamentals, Types and Applications*, 4th ed. Oxford: Newnes, 2013.
- [47] C. Platt, *Encyclopedia of Electronic Components Volume 1: Resistors, Capacitors, Inductors, Switches, Encoders, Relays, Transistors*. Maker Media Inc., 2012.
- [48] A. Prayitno and H. Santoso, “Rancang Bangun Sistem Penjadwalan Lampu Otomatis Berbasis Arduino dengan Modul RTC DS3231,” *J. Tek. Elektro dan Komput.*, vol. 8, pp. 71–78, 2019.

- [49] N. Kolban, *Kolban's Book on ESP32 & ESP8266*. Leanpub, 2018.
- [50] P. Putra and I. Setiawan, "Prototipe Sistem Monitoring dan Kontrol Jarak Jauh Berbasis IoT Menggunakan Blynk," *J. SPEKTRUM*, vol. 5, pp. 78–83, 2018.

