



Sistem Penyiraman Pintar dan Pemantauan Menggunakan Sistem IoT Berbasis *Smartphone* Pada Tanaman Cabai

Tugas Akhir

**Oleh:
Josua Cristian Tambunan (4212011019)**

**Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2024**

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan di bawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : "Sistem Penyiraman Pintar dan Pemantauan Menggunakan Sistem IoT Berbasis *Smartphone* Pada Tanaman Cabai" adalah hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 22 Juli 2024



Josua Cristian Tambunan
NIM: 4212011019

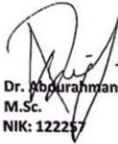
Lembar Pengesahan


Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Terapan Teknik (S.Tr.T)
di
Politeknik Negeri Batam


Oleh:
Josua Cristian Tambunan (4212011019)

Tanggal Sidang: 17 - 07 - 2024

Disetujui oleh :


1. Dr. Abdurrahman Dwijotomo, S.ST.,
M.Sc.
NIK: 122257


1. Adlian Jefiza, S.Pd., M.T.
NIK: 119220


2. Fadli Firdaus, M.Pd
NIK: 122271

Sistem Penyiraman Pintar dan Pemantauan Menggunakan Sistem IoT berbasis *Smartphone* pada Tanaman Cabai

Abstrak

Salah satu hal yang mempengaruhi kualitas dan tumbuh kembang tanaman cabai adalah penyiraman. Saat ini proses penyiraman pada tanaman cabai masih dilakukan dengan cara konvensional, yaitu dengan bantuan tangan manusia. Metode penyiraman ini kurang efektif dalam menjaga kelembaban tanah pada tanaman cabai, dan juga sulitnya untuk melakukan *monitoring* kondisi lingkungan pada tanaman cabai tersebut. Pada penelitian ini diusulkan sebuah sistem untuk mengatasi permasalahan tersebut, yaitu dengan membuat sebuah sistem penyiraman tanaman cabai otomatis menggunakan metode *fuzzy* dan juga terjadwal. Pada sistem penyiraman yang dibuat menggunakan dua parameter yaitu sensor DHT22 untuk mendeteksi suhu udara dan sensor YL-69 untuk mendeteksi kelembaban tanah, sedangkan penyiraman terjadwal akan diaplikasikan menggunakan modul RTC. Metode *fuzzy* digunakan untuk menghasilkan tiga durasi penyiraman yang berbeda berdasarkan kondisi yang terjadi. Sebuah sistem pemantauan juga diusulkan dengan memanfaatkan sistem IoT untuk melakukan *monitoring* kondisi lingkungan di sekitar tanaman cabai melalui aplikasi *Blynk* pada perangkat *smartphone*. Sistem penyiraman otomatis dan terjadwal yang dihasilkan dari penelitian ini dapat bekerja dengan baik dalam mengendalikan durasi penyiraman dengan rata-rata durasi yang dihasilkan 4.52 detik dan juga dalam mengendalikan kelembaban tanah, dimana sistem dapat menjaga kelembaban tanah pada tanaman cabai dengan nilai rata-rata pembacaan sensor kelembaban sebesar 73.2%, sistem penyiraman otomatis juga menunjukkan hasil yang lebih efektif dalam menjaga kelembaban tanah jika dibandingkan dengan sistem penyiraman manual. Sedangkan sistem IoT pemantauan atau *monitoring* yang menggunakan jaringan internet *hotspot* pada lingkungan tanaman cabai dapat berfungsi dengan baik dengan jarak akses sekitar 20m.

Kata kunci: *fuzzy*, kelembaban tanah, suhu udara, sistem IoT, tanaman cabai.

Smart Watering and Monitoring System Using a Smartphone-based IoT System for Chili Plants

Abstract

One of the things that affects the quality and growth and development of chili plants is watering. Currently, the process of watering chili plants is still carried out conventionally, namely with the help of human hands. This watering method is less effective in maintaining soil moisture in chili plants, and it is also difficult to monitor environmental conditions in chili plants. In this research, a system is proposed to overcome this problem, namely by creating an automatic chili plant watering system using a fuzzy and scheduled method. The watering system is created using two parameters, namely the DHT22 sensor to detect air temperature and the YL-69 sensor to detect soil moisture, while scheduled watering will be applied using the RTC module. The fuzzy method is used to produce three different watering durations based on the conditions that occur. A monitoring system is also proposed by utilizing an IoT system to monitor environmental conditions around chili plants via the Blynk application on smartphone devices. The automatic and scheduled watering system resulting from this research can work well in controlling watering duration with an average duration of 4.52 seconds and also in controlling soil moisture, where the system can maintain soil moisture in chili plants with an average value of sensor readings. humidity was 73.2%, the automatic watering system also showed more effective results in maintaining soil moisture when compared to the manual watering system. Meanwhile, the IoT monitoring system that uses an internet hotspot network in the chili plant environment can function well with an access distance of around 20m.

Keywords: fuzzy, soil moisture, air temperature, IoT system, chilli plant.

Kata Pengantar

Puji syukur kepada Tuhan Yang Maha Esa, atas segala rahmat dan kuasa-Nya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir dengan judul “Sistem Penyiraman Pintar dan Pemantauan Menggunakan Sistem IoT Berbasis *Smartphone* pada Tanaman Cabai”. Penulisan Laporan Tugas Akhir ini memiliki tujuan untuk melengkapi persyaratan kelulusan tingkat Diploma IV Program Studi Mekatronika Politeknik Negeri Batam.

Pada kesempatan ini penulis juga ingin menyampaikan ucapan terima kasih yang sebesar-besarnya atas segala dukungan dan juga bantuan yang telah diberikan selama proses penyusunan Tugas Akhir ini hingga selesai kepada:

1. Kedua orang tua dan juga keluarga besar yang telah memberikan semangat, doa dan juga dukungan
2. Bapak Uuf Brajawidagda, S.T., M.T., Ph.D. selaku Direktur Politeknik Negeri Batam.
3. Bapak Dr. Budi Sugandi, S.T., M.Eng. selaku ketua Jurusan Teknik Elektro Politeknik Negeri Batam.
4. Bapak Indra Hardian Mulyadi, S.T., M.Eng. selaku ketua Program Studi Teknik Mekatronika Politeknik Negeri Batam
5. Bapak Adlian Jefiza, S.Pd., M.T. selaku dosen pembimbing Tugas Akhir yang telah menyediakan waktu dan pikiran untuk mengarahkan dan membimbing penulis dalam menyelesaikan Tugas Akhir ini.
6. Bapak Muhammad Naufal Airlangga Diputra, S.Pd., M.P.H. selaku dosen pengampu mata kuliah Tugas Akhir.
7. Bapak Dr. Budi Sugandi, S.T., M.Eng. selaku wali dosen kelas mekatronika A malam.
8. Bapak Dr. Abdurahman Dwijotomo, S.ST., M.Sc. selaku dosen penguji 1 dan Bapak Fadli Firdaus, M.Pd selaku dosen penguji 2.
9. Seluruh dosen yang ada di Jurusan Teknik Elektro Politeknik Negeri Batam.
10. Seluruh rekan-rekan yang telah berpartisipasi dalam membantu penyelesaian Tugas Akhir ini.

Dalam penyusunan Laporan Tugas Akhir ini, penulis menyadari masih banyak hal yang perlu dikembangkan, oleh karena itu penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan Tugas Akhir ini. Penulis juga berharap Laporan Tugas Akhir ini dapat bermanfaat dan berguna bagi kita semua. Terima kasih atas segala perhatiannya.

Batam, 22 Juli 2024

Josua Cristian Tambunan

Daftar Isi

Pernyataan Keaslian Tugas Akhir.....	i
Lembar Pengesahan	ii
Abstrak	iii
<i>Abstract</i>	iv
Kata Pengantar	v
Daftar Isi	vii
Daftar Gambar	ix
Daftar Tabel	x
Bab 1. Pendahuluan.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan.....	3
1.4. Manfaat	3
1.5. Batasan Masalah	3
Bab 2. Tinjauan Pustaka	4
2.1. Tanaman Cabai	4
2.2. Internet Of Things.....	4
2.3. ESP32	5
2.4. Sensor DHT22	6
2.5. Sensor Kelembaban YL-69(<i>Soil Moisture Sensor</i>)	7
2.6. Modul Relay 5VDC.....	8
2.7 Modul RTC DS3231	9
2.7. Solenoid Valve 12VDC (Kran Air Elektrik)	10
2.8. Logika <i>Fuzzy</i>	11
2.9. Arduino IDE.....	12
2.10. <i>Blynk</i>	13
2.11. <i>Smartphone</i>	13
Bab 3. Metodologi Penelitian	14

3.1. Perancangan	15
3.1.1. Perancangan Mekanikal.....	17
3.1.2. Perancangan Sistem Elektrikal.....	18
3.1.3. Perancangan sistem kerja	20
3.1.4. Perancangan sistem <i>Fuzzy</i>	21
3.1.5. Perancangan sistem komunikasi ESP32 dan aplikasi <i>Blynk</i>	24
3.1.6. Perancangan <i>GUI</i> pada aplikasi <i>Blynk</i>	25
3.2. Alat dan Bahan	26
3.3. Pengujian	27
3.3.1. Pengujian sensor suhu DHT22 dan sensor kelembaban YL-69 ...	27
3.3.2. Pengujian Sistem <i>Fuzzy</i>	27
3.3.3. Pengujian Sistem Penyiraman Otomatis.....	27
3.3.4. Pengujian Sistem IoT pada aplikasi <i>Blynk</i>	27
Bab 4. Hasil dan Pembahasan.....	28
4.1. Implementasi alat pada tanaman cabai.....	28
4.2. Hasil pengujian sensor DHT22 dan <i>Soil Moisture YL-69</i>	29
4.3. Hasil pengujian <i>Fuzzy Mamdani</i>	31
4.4. Hasil Pengujian Sistem Penyiraman.....	33
4.5. Hasil pengujian sistem IoT menggunakan aplikasi <i>Blynk</i>	41
Bab 5. Kesimpulan dan Saran	45
5.1. Kesimpulan	45
5.2. Saran.....	45
Daftar Pustaka	47
Biodata	49
Lampiran.....	50
Lampiran A (Rangkaian Elektrikal)	50
Lampiran B (Pemrograman utama ESP32)	51
Lampiran C (Pemrograman <i>Fuzzy</i>).....	53
Lampiran D (Pemrograman Penyiraman)	57

Daftar Gambar

Gambar 1. Ilustrasi <i>Internet Of Things</i>	4
Gambar 2. ESP32 <i>Board Pinout</i>	5
Gambar 3. Sensor DHT22 dan konfigurasi pinout	7
Gambar 4. Sensor Kelembaban YL-69	8
Gambar 5. Modul Relay 12VDC 2 Channel	9
Gambar 6. Modul RTC DS3231	9
Gambar 7. Kran Air Elektrik 12VDC.....	10
Gambar 8. Tampilan pada <i>software</i> Arduino IDE	12
Gambar 9. Ilustrasi Aplikasi <i>Blynk</i>	13
Gambar 10. Flowchart Penelitian	14
Gambar 11. Flowchart tahapan perancangan	15
Gambar 12. Diagram Blok Sistem	16
Gambar 13. Drawing Mekanikal <i>Enclosure</i> alat	17
Gambar 14. Rangkaian Elektrikal.....	18
Gambar 15. <i>Flowchart</i> sistem kerja alat.....	20
Gambar 16. Grafik himpunan variabel suhu udara.....	22
Gambar 17. Grafik himpunan variabel kelembaban tanah.....	22
Gambar 18. Grafik himpunan variabel durasi penyiraman.....	23
Gambar 19. Topologi komunikasi ESP32 dan <i>Blynk</i>	24
Gambar 20. GUI sistem pemantauan dan penyiraman <i>Blynk</i>	25
Gambar 21. Implementasi alat pada tanaman cabai	28
Gambar 22. Peletakan sensor DHT22 dan juga Sensor YL-69.....	29
Gambar 23. Hasil pengujian <i>fuzzy</i> pada ESP32	31
Gambar 24. Hasil pengujian <i>fuzzy</i> pada aplikasi Matlab	31
Gambar 25. Hasil Kelembaban tanah sistem penyiraman otomatis.....	34
Gambar 26. Grafik perubahan kelembaban setelah penyiraman aktif.....	35
Gambar 27. Grafik perubahan kelembaban tanah	36
Gambar 28. Grafik perbandingan kondisi kelembaban tanah	38
Gambar 29. Halaman awal sistem IoT pada aplikasi <i>Blynk</i>	41

Daftar Tabel

Tabel 1. Spesifikasi ESP32 (ESP32 Series Datasheet v4.3.pdf,2023)	6
Tabel 2. Tingkat <i>Soil Moisture</i> YL-69 di ADC ESP32	8
Tabel 3. Spesifikasi Solenoid Valve	10
Tabel 4. Konfigurasi Pin Sensor DHT22 dengan ESP32	18
Tabel 5. Konfigurasi Pin Sensor YL-69 dengan ESP32	19
Tabel 6. Konfigurasi Pin RTC DS3231 dengan ESP32	19
Tabel 7. Konfigurasi Pin Modul Relay dengan ESP32	19
Tabel 8. Fungsi keanggotaan <i>fuzzy</i> input sensor suhu	21
Tabel 9. Fungsi keanggotaan <i>fuzzy</i> input sensor kelembaban	22
Tabel 10. Fungsi keanggotaan <i>fuzzy</i> output durasi penyiraman	22
Tabel 11. <i>Rule Base</i>	23
Tabel 12. Alat Bahan dan Estimasi biaya	26
Tabel 13. Pengujian Sensor DHT22	29
Tabel 14. Pengujian Sensor YL-69	30
Tabel 15. Pengujian sistem <i>fuzzy</i> Mamdani	31
Tabel 16. Pengaturan debit air melalui kran manual	33
Tabel 17. Hasil pengujian sistem penyiraman otomatis	33
Tabel 18. Hasil pengujian pergerakan kelembaban tanah	36
Tabel 19. Hasil pengujian perbandingan metode penyiraman	37
Tabel 20. Hasil tampilan <i>Grafical User Interface</i> pada aplikasi <i>Blynk</i>	42
Tabel 21. Pengujian hasil pembacaan nilai sensor pada aplikasi <i>Blynk</i>	43
Tabel 22. Pengujian Penyiraman manual	43
Tabel 23. Pengujian Aksesibilitas Sistem IoT aplikasi <i>Blynk</i>	44

Bab 1. Pendahuluan

1.1. Latar Belakang

Tanaman merupakan salah satu makhluk hidup yang sangat bermanfaat bagi kehidupan manusia maupun makhluk hidup lainnya hingga saat ini. Salah satu tanaman yang banyak digunakan oleh manusia adalah tanaman cabai atau *Capscum Annuum*. Tanaman cabai sering digunakan untuk keperluan dapur, seperti memasak dan lainnya. Dalam budidaya tanaman cabai, perlu dilakukan beberapa hal untuk menjaga kualitas dan tumbuh kembang tanaman tersebut.

Kualitas dan tumbuh kembang tanaman cabai dipengaruhi oleh beberapa hal, salah satunya adalah proses penyiraman[1]. Penyiraman berfungsi untuk memenuhi kebutuhan air serta menjaga kelembaban tanah pada tanaman cabai sehingga dapat menjaga tanaman tersebut agar tumbuh dengan subur dan mendapatkan hasil yang baik. Saat ini proses penyiraman tanaman cabai banyak dilakukan dengan cara konvensional yaitu dengan bantuan tangan manusia. Hal ini sedikit kurang efektif karena manusia tersebut terkadang tidak dapat melakukan penyiraman secara berkala karena kondisi tertentu. Selain itu, penyiraman yang dilakukan oleh manusia dapat menghabiskan banyak waktu, pengeluaran energi cukup besar dan penggunaan air menjadi boros[2]. Oleh karena itu, diperlukan sebuah sistem yang baru untuk mengatasi masalah penyiraman manual.

Beberapa penelitian telah dilakukan untuk mengatasi masalah tersebut, seperti penelitian yang dilakukan oleh Rahmat dkk pada tahun 2019 dengan membuat sebuah alat penyiraman tanaman otomatis berbasis mikrokontroler arduino pada tanaman hias yopi[3]. Penelitian ini dilakukan dengan menggunakan suhu udara dan kelembaban tanah sebagai parameter penyiraman. Namun, sistem penyiraman masih kurang efektif karena alat penyiram bekerja pada satu kondisi saja, dimana saat kondisi parameter mencapai batas tertentu alat penyiram akan aktif. Hal ini tidak cocok diterapkan pada tanaman cabai karena tanaman cabai memiliki kebutuhan air yang berbeda sesuai dengan kondisi kelembaban tanahnya untuk kondisi tanah kering dibutuhkan penyiraman ± 750 ml air, tanah lembab ± 350 ml air, dan tanah basah tidak menyiram[4].

Kemudian, penelitian dilakukan oleh Jeki dkk pada tahun 2021 yang membuat sebuah sistem penyiraman otomatis pada tanaman cabai menggunakan metode *fuzzy*[5]. Pada penelitian ini digunakan sebuah metode yang bernama *fuzzy* untuk mengklasifikasikan parameter berupa suhu dan kelembaban tanah sehingga dapat menghasilkan durasi penyiraman yang bervariasi. Metode penyiraman ini terbilang cukup efektif karena dapat membagi durasi penyiraman pada tiga kondisi sehingga debit air penyiraman dapat terkontrol. Namun, penyiraman tersebut masih belum terjadwal sehingga dalam rentang waktu berdekatan alat

penyiraman dapat aktif, sedangkan penyiraman tanaman cabai yang dianjurkan adalah pada pagi hari (sekitar pukul 07:00) dan sore hari (sekitar pukul 16:00)[4].

Oleh karena itu, pada penelitian ini akan dibuat sebuah sistem penyiraman otomatis pada tanaman cabai dengan waktu penyiraman yang terjadwal serta menggunakan metode *fuzzy* dalam mengklasifikasikan kondisi parameter penyiraman. Metode *fuzzy* digunakan karena penerapan metode ini cukup sederhana, dan juga metode *fuzzy* tidak membutuhkan komputasi yang luas ataupun rumit sehingga cocok untuk diaplikasikan pada sistem yang menggunakan mikrokontroler. Penggunaan metode *fuzzy* pada penelitian ini bertujuan agar sistem dapat menghasilkan durasi penyiraman yang berbeda, sehingga debit air yang diterima oleh tanaman cabai serta kelembaban tanah menjadi lebih terkontrol. Parameter yang akan digunakan adalah sensor DHT22 sebagai sensor suhu udara dan juga sensor YL-69 sebagai sensor kelembaban tanah. Penyiraman secara terjadwal dilakukan dengan menambahkan modul RTC DS3231 sehingga alat akan bekerja sesuai dengan waktu yang telah diatur.

Seiring dengan perkembangan teknologi yang semakin maju. Berbagai teknologi diciptakan untuk mempermudah manusia, salah satu wujud teknologi tersebut adalah adanya sistem yang bernama *Internet Of Things* atau IoT. Pada suatu sistem IoT manusia dapat melakukan pemantauan dan pengontrolan suatu alat dengan melalui jaringan internet dari jarak yang cukup jauh sehingga dapat mempermudah manusia[6]. Dengan memperhatikan keuntungan dari sistem IoT tersebut, maka sistem IoT dapat diterapkan pada alat yang akan dibuat.

Sistem IoT pada penelitian ini digunakan untuk mengatasi masalah sulitnya melakukan proses *monitoring* kondisi lingkungan seperti suhu, kelembaban udara dan kelembaban tanah. Sistem IoT juga dapat digunakan untuk melakukan penyiraman manual pada tanaman cabai. Dengan adanya sistem IoT, manusia dapat melakukan *monitoring* dan pengontrolan tanpa harus berada di lokasi tanaman, melainkan dapat dilakukan dari jarak yang cukup jauh dengan bantuan jaringan internet. Untuk mengimplementasikan sistem IoT pada sistem ini maka akan digunakan mikrokontroler ESP32.

ESP32 digunakan sebagai mikrokontroler utama alat. Perangkat ini berfungsi untuk melakukan *processing* dari data sensor DHT22 dan *Soil Moisture* dan mengirim perintah kepada modul relay untuk menghidupkan output berupa solenoid valve yang berfungsi untuk mengeluarkan air. ESP32 digunakan karena memiliki kemampuan untuk terhubung ke jaringan internet sehingga dapat digunakan untuk sistem berbasis IoT seperti pada penelitian ini. Keuntungan lainnya adalah jumlah memori yang lebih besar dan juga kecepatan *clockspeed* yang lebih baik dari ESP8266. Untuk memudahkan manusia dalam pengguna sistem, maka akan digunakan sebuah aplikasi bernama *Blynk*. *Blynk* digunakan sebagai *interface* antara manusia dan sistem. Aplikasi *Blynk* akan diinstall pada perangkat *smartphone*.

1.2. Rumusan Masalah

Adapun rumusan masalah yang dapat dikemukakan dari latar belakang di atas adalah sebagai berikut:

1. Bagaimana cara mempermudah manusia dalam mengatasi masalah penyiraman yang masih dilakukan secara manual pada tanaman cabai?
2. Bagaimana penerapan dan efektifitas metode *fuzzy* dalam mengendalikan kelembaban tanah pada sistem penyiraman otomatis?
3. Bagaimana cara memantau kondisi suhu udara dan kelembaban tanah pada tanaman cabai menggunakan perangkat *Smartphone*?

1.3. Tujuan

Adapun tujuan dilakukannya penelitian ini adalah sebagai berikut:

1. Membuat alat yang dapat menyiram tanaman cabai secara otomatis dan waktu yang terjadwal menggunakan metode *fuzzy* dan mikrokontroler ESP32.
2. Melakukan pengujian dan analisa efektifitas penerapan metode *fuzzy* yang dirancang untuk mengatur durasi penyiraman dan juga dalam mengendalikan kelembaban tanah.
3. Mengintegrasikan alat dengan sistem IoT , sehingga dapat dilakukan pemantauan kondisi lingkungan sekitar tanaman cabai dan penyiraman otomatis maupun manual melalui perangkat *smartphone*.

1.4. Manfaat

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Mempermudah tugas manusia dalam melakukan penyiraman tanaman cabai.
2. Menjaga kualitas dan tumbuh kembang tanaman cabai dengan adanya sistem penyiraman otomatis dan terjadwal yang dapat memenuhi dan mengendalikan kebutuhan air tanaman tersebut secara teratur.
3. Mempermudah manusia dalam melakukan pemantauan kondisi suhu dan kelembaban tanah pada tanaman cabai menggunakan aplikasi *Blynk* pada *smartphone* melalui jaringan internet.

1.5. Batasan Masalah

Adapun pembatasan ruang lingkup dari penelitian ini adalah sebagai berikut:

1. Sistem menggunakan Sensor DHT22 sebagai sensor suhu dan *Soil Moisture Sensor YL-69* sebagai sensor kelembaban tanah dan mikrokontroler yang digunakan adalah ESP32.
2. Penelitian dilakukan pada tanaman cabai dalam polybag dengan media tanam tanah hitam dan serbuk sabut kelapa.
3. Sistem pemantauan tanaman cabai berbasis IoT menggunakan aplikasi *Blynk* pada *smartphone*.

Bab 2. Tinjauan Pustaka

2.1. Tanaman Cabai

Tanaman cabai merupakan tanaman yang banyak berkembang di negara beriklim tropis, ini disebabkan karena tanaman cabai hanya dapat tumbuh pada suhu tropis. Tanaman ini memiliki banyak fungsi untuk tubuh manusia, karena mengandung mineral dan juga vitamin C. Cabai banyak digunakan untuk penyedap makanan atau pemberi rasa pada makanan karena rasanya yang pedas. Tanaman cabai dapat tumbuh pada ketinggian maksimal 1,400 m di atas permukaan laut (m dpl) dengan suhu optimum 25–27 °C pada siang hari dan 18–20 °C pada malam hari[7]. Sedangkan tingkat kelembaban tanah yang baik bagi tanaman cabai adalah sekitar 60%-80%[4]. Pada penelitian ini tanaman cabai digunakan sebagai objek penelitian dari alat yang akan dibuat. Tanaman cabai yang akan digunakan adalah tanaman yang dibudidayakan menggunakan polybag.

2.2. Internet Of Things

Internet Of Things atau IoT merupakan sebuah konsep masa kini yang memungkinkan sebuah objek dapat terhubung dengan jaringan internet[8]. IoT adalah konsep komputasi yang menggambarkan masa depan di mana setiap objek fisik terhubung ke Internet dan dapat mengidentifikasi dirinya di antara perangkat lain.. Sistem kerja IoT cukup sederhana, yaitu setiap objek/benda yang akan diintegrasikan dengan IoT harus memiliki IP Address sehingga dapat terhubung dengan jaringan internet. Penerapan sistem *Internet Of Things* diterapkan di beberapa aspek seperti pada Sistem Pakan Ayam Otomatis berbasis *Internet Of Things*[9]. Pada proyek ini sistem IoT digunakan sebagai sistem pemantauan kondisi lingkungan disekitar tanaman cabai dan juga penyiraman manual melalui *Smartphone*.

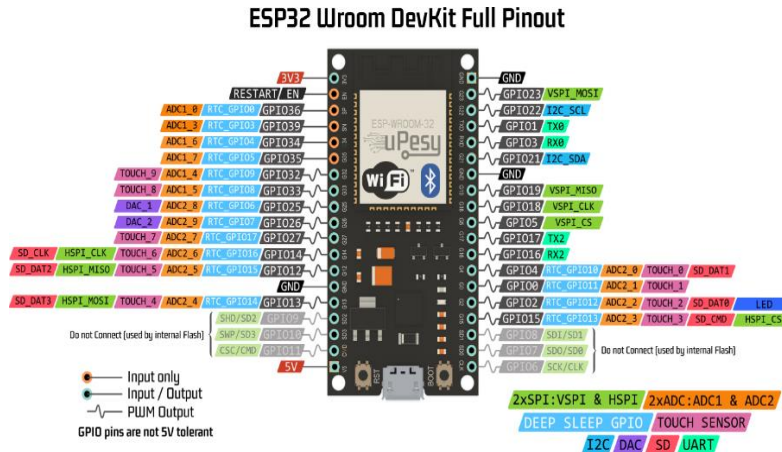


Gambar 1. Ilustrasi *Internet Of Things*

Sumber: (<https://www.tokopedia.com/>)

2.3. ESP32

ESP32 merupakan perangkat mikrokontroler fabrikasi *Espressif System*. Perangkat ini merupakan penerus dari ESP8266. ESP32 sering digunakan untuk proyek yang berhubungan dengan *Internet Of Things*. Ini dikarenakan ESP32 memiliki kemampuan untuk terhubung dengan jaringan internet maupun juga Bluetooth. ESP32 menggunakan *interface* USB-to-UART, sehingga dapat diprogram menggunakan aplikasi atau *software* Arduino IDE dengan menggunakan kabel micro-USB yang biasa digunakan untuk recharge *smartphone*[10]. Penggunaan ESP32 pada proyek ini didasari oleh kapasitas memori yang lebih besar dan juga kecepatan processor yang lebih baik jika dibandingkan dengan pendahulunya ESP8266, serta memiliki versi WiFi dan Bluetooth yang lebih baru. Mikrokontroler ini memiliki 36 digital/I/O Pin(16 diantaranya adalah PWM), 18 analog input , 3 pin UART, 4 pin SPI, dan 2 pin I2C, spesifikasi lengkap dari mikrokontroler ini dapat dilihat pada Tabel 1. Penggunaan perangkat ESP32 sering dilakukan untuk proyek IoT, seperti Sistem Kontrol tanaman hidroponik berbasis IoT menggunakan ESP32[11]. Pada proyek ini ESP32 digunakan sebagai mikrokontroler utama sistem yang berfungsi untuk menerima data sensor, melakukan proses klasifikasi dan juga mengirim sinyal output. ESP32 juga akan berkomunikasi dengan perangkat Smartphone sebagai *interface* sistem melalui jaringan internet.



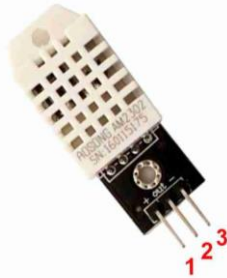
Tabel 1. Spesifikasi ESP32 (ESP32 Series Datasheet v4.3.pdf,2023)

Keterangan	Detail
Microcontroller	Xtensa Dual Core 32-bit LX6 microprocessor
Flash Memory	4MB
SRAM	520MB
EEPROM	Not Available
CLOCK Speed	Upto 240 MHz
Operating Voltage	3.3V DC
Input Voltage	3.3V DC
Current consumption	80 mA – 90mA
DC Current per I/O Pin	40 mA
DC Current For 3.3V Pin	50 mA
Digital IO Pins	36
Analog Input Pins	Up to 18
UARTs	3
SPI	4
I2C	2
CAN	Yes
PWM	16
Wi-Fi	Yes,802.11 b/g/n
Bluetooth	Yes, Bluetooth v4.2 BR/EDR and BLE specification

2.4. Sensor DHT22

Sensor DHT22 merupakan sensor digital untuk kelembaban dan suhu relatif. DHT22 merupakan chip tunggal kelembaban serta multisensor yang tersusun dari modul yang telah dikalibrasi keluaran digital[12]. DHT22 menggunakan kapasitor dan thermistor untuk mengukur udara sekitar. Pada proyek ini DHT22 digunakan untuk mengukur nilai Suhu udara dan juga kelembaban udara disekitar tanaman cabai dan akan mengirim nilai ke perangkat ESP32. Alasan penggunaan DHT22 pada penelitian ini dibandingkan sensor suhu lainnya adalah karena tingkat akurasi DHT22 yang lebih baik dengan galat relatif pengukuran suhu sebesar 4% dan

pengukuran kelembaban sebesar 18%. Sedangkan sensor sejenis nya seperti DHT11 memiliki galat yang lebih besar dengan pengukuran suhu sebesar 1%-7% dan pengukuran kelembaban sebesar 11%-35%[13]. Bentuk fisik dan juga konfigurasi pin out dari DHT22 dapat dilihat pada gambar 3 di bawah ini.



PIN NUMBER	DESCRIPTION
1	VCC (+5V)
2	SIGNAL
3	Ground (0V)

Gambar 3. Sensor DHT22 dan konfigurasi pinout

Sumber: (<https://www.mahirelektro.com/>)

2.5. Sensor Kelembaban YL-69(Soil Moisture Sensor)

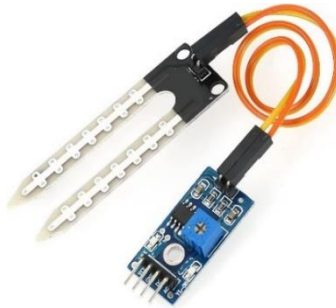
Sensor Kelembaban adalah sebuah sensor analog yang memiliki fungsi untuk mengukur nilai kelembaban tanah. Sensor YL-69 memiliki probe berbentuk dua buah konduktor yang digunakan untuk mengaliri arus listrik melalui tanah dan kemudian sensor membaca nilai resistansi untuk mengukur tingkat kelembaban tanah. Sensor YL-69 memiliki spesifikasi tegangan input sebesar 3.3VDC – 5VDC. Pada proyek ini sensor YL-69 digunakan untuk mendeteksi kelembaban yang ada pada tanah yang digunakan untuk menanam tanaman cabai pada polybag. Dikarenakan miktonkontroller yang digunakan adalah ESP32 yang memiliki pin ADC dengan resolusi 12 bit, maka pembacaan nilai resistansi atau tegangan yang terdeteksi sensor YL-69 pada ADC ESP32 berkisar pada range 0-4095[13]. Sehingga, untuk untuk mendapatkan persentase dari sensor kelembaban, sensor akan mengirim nilai dalam bentuk persentase dengan rumus(1) berikut ini:

$$\%Soil\ Moisture = 100 - \frac{ADC}{4095} \times 100 \quad (1)$$

Tanaman pekarangan atau taman biasanya akan diklasifikasikan dengan menggunakan tiga jenis klasifikasi kondisi tanah yaitu kering, lembab, dan basah[13]. Klasifikasi ini dapat dilihat pada tabel 2 dibawah ini.

Tabel 2. Tingkat Soil Moisture YL-69 di ADC ESP32

ADC	PERCENTAGE	SOIL CONDITION
2525-4095	50-0	KERING
1400-2524	70-51	LEMBAB
570-1399	100-71	BASAH



Gambar 4. Sensor Kelembaban YL-69

Sumber: (<https://www.amazon.in/>)

2.6. Modul Relay 5VDC

Modul Relay merupakan relay berbentuk mini yang memiliki fungsi sama seperti Relay. Modul Relay adalah saklar (*switch*) yang bekerja melalui bantuan aliran listrik dan merupakan komponen elektromagnetic karena terdiri dari dua bagian yaitu coil dan kontak saklar. Prinsip kerja modul relay adalah ketika salah satu kontak dialiri arus listrik maka akan menghidupkan saklar. Modul relay yang akan digunakan pada penelitian ini adalah modul relay 5Vdc, modul ini dapat beroperasi dengan tegangan input 5VDC dan untuk tegangan input setiap pin IN atau input relay menggunakan 5VDC yang dapat dihubungkan langsung dengan output pin ESP32. Yang menjadi kelebihan dari perangkat ini adalah kemampuan kerja relay yang besar yaitu dapat mengendalikan tegangan tinggi 30VDC dan 250VAC. Modul relay ini juga memiliki kontak NO (*Normally Open*) dan NC (*Normally Close*) layaknya Relay biasa[14]. Pada penelitian ini modul relay digunakan untuk mengendalikan hidup mati solenoid valve atau kran air elektrik yang berfungsi untuk menyiram tanaman dan akan dihubungkan dengan

mikrokontroler ESP32. Bentuk fisik dari Modul Relay dapat dilihat pada gambar 5 di bawah ini.



Gambar 5. Modul Relay 12VDC 2 Channel

Sumber: (<https://www.tokopedia.com/>)

2.7 Modul RTC DS3231

RTC atau *Real Time Clock* merupakan modul IC yang berfungsi sebagai penunjuk waktu digital dan fitur pengukur suhu dalam satu modul. Modul ini dapat bekerja dengan menggunakan dua *interface*, yaitu I2C SDA dan juga SCL. Modul RTC yang akan digunakan pada penelitian ini adalah Modul RTC DS3231. Modul RTC DS3231 mampu menyimpan data-data waktu seperti detik, menit, jam, tanggal, bulan, hari dalam seminggu, dan tahun valid hingga 2100. Selain itu, Modul RTC DS3231 sudah tersedia dengan baterai CR2032 3V yang berfungsi sebagai *back up* daya dari RTC apabila daya utama mati, modul ini bekerja dengan tegangan input 3VDC[22]. Pada penelitian ini, modul RTC DS3231 digunakan sebagai penunjuk dan pengatur waktu dari sistem penyiraman terjadwal.



Gambar 6. Modul RTC DS3231

Sumber: (<https://www.lazada.co.id/>)

2.7. Solenoid Valve 12VDC (Kran Air Elektrik)

Solenoid valve merupakan kran air elektrik yang memiliki fungsi untuk mengalirkan air, sama seperti kran air pada umumnya. Namun, yang membedakan dengan kran air pada umumnya adalah solenoid valve ini dikendalikan secara otomatis menggunakan aliran arus listrik, arus listrik tersebut akan menghidupkan coil dan coil akan membuka valve dan air akan mengalir melalui saluran yang tersedia. Input pipa solenoid valve akan terhubung langsung dengan sumber air atau kran air dan output atau sisi lainnya diarahkan ke tanaman cabai. Pada penelitian ini, *solenoid valve* digunakan sebagai kran air untuk melakukan penyiraman pada tanaman cabai, pada bagian ujung yang menuju tanaman cabai akan dipasang sebuah *sprinkle* sehingga air akan menyebar ketika menyiram.



Gambar 7. Kran Air Elektrik 12VDC

Sumber: (<https://www.tokopedia.com/>)

Solenoid Valve yang akan digunakan pada proyek ini adalah *Electric Solenoid Valve* 12VDC dengan diameter pipa atau saluran ½ Inchi. Kondisi awal kran adalah NC sehingga akan dapat mengalir air jika dialiri arus listrik. Spesifikasi dari kran air elektrik yang akan digunakan dapat dilihat pada tabel 3 di bawah ini.

Tabel 3. Spesifikasi Solenoid Valve

Operation Mode	NC (<i>Normally Closed</i>)
Voltage/Current	12VDC / 0.6A
Inlet Size	1/2 Inch
Pressure	0.02 – 0.8 Mpa
Max Temperature	100°C

2.8. Logika Fuzzy

Fuzzy dalam konsep bahasa dapat dipahami sebagai kabur atau samar yang artinya suatu nilai memiliki kemungkinan untuk bernilai benar atau salah secara bersamaan[16]. Konsep Logika *Fuzzy* dikembangkan oleh Prof.Lotfi Zaedah dari Universitas California pada tahun 1965, beliau memodifikasi teori himpunan dimana setiap anggotanya memiliki derajat keanggotaan yang bernilai kontinu antara 0 dan 1[17]. Penerapan logika *fuzzy* biasanya dilakukan untuk menyelesaikan sebuah masalah yang memiliki unsur ketidakpastian sehingga memerlukan sebuah pemrosesan untuk mendapatkan sebuah hasil.

Pada penelitian ini metode *fuzzy* digunakan untuk mengklasifikasikan input data berupa sensor untuk menentukan keluaran atau output berupa durasi penyiraman. Metode *fuzzy* digunakan karena pengaplikasian metode ini cukup sederhana namun dapat memperoleh hasil yang cukup baik. Beberapa hal juga menjadi alasan penggunaan metode *fuzzy* seperti[18]:

- a. Konsep matematis penalaran dari *fuzzy* cukup sederhana dan mudah dimengerti.
- b. Metode *fuzzy* sangat fleksibel.
- c. Logika *fuzzy* mampu memodelkan fungsi-fungsi non linier yang cukup kompleks.
- d. Logika *fuzzy* didasarkan pada bahasa alami.
- e. Mampu bekerja sama dengan teknik-teknik kendali secara konvensional.

Terdapat beberapa metode *fuzzy* yang umumnya digunakan seperti Mamdani, Sugeno, dan Tsukamoto. Metode *fuzzy* yang akan digunakan pada penelitian ini adalah metode *fuzzy* Mamdani. Penggunaan metode *fuzzy* mamdani pada penelitian ini didasari oleh kelebihan sistem Mamdani yang memiliki basis aturan yang lebih intuitif dan lebih mudah dipahami secara sederhana[19]. Metode *fuzzy* Mamdani juga mampu menghasilkan sebuah output berbentuk himpunan *fuzzy* yang hasilnya sesuai dengan perancangan derajat keanggotaan input dan juga *rule base* sehingga pemanfaatan untuk menentukan durasi penyiraman berdasarkan kondisi parameter akan lebih maksimal.

Dalam menerapkan logika *fuzzy* biasanya terdapat langkah langkah yang perlu dilakukan sebagai berikut[20]:

1. Fuzzifikasi

Merupakan proses yang dilakukan untuk mengubah nilai tegas kedalam fungsi keanggotaan *fuzzy*. Fungsi keanggotaan tersebut dilakukan dengan menentukan himpunan *fuzzy* pada setiap variabel input.

2. Rule Base

Pada tahapan ini dilakukan penentuan aturan relasi yang digambarkan dalam bentuk interfensi '*IF-THEN*' yang akan digunakan untuk menyatakan kondisi sesuatu dari kondisi yang sedang terjadi.

3. *Interfensi*

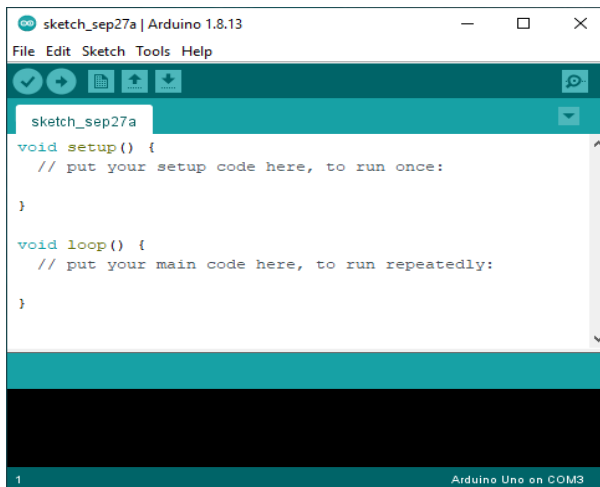
Setelah dibuatnya aturan atau rules, langkah selanjutnya ialah menghitung derajat keanggotaan atau *membership function*(μ), dan mencari α -predikat dan nilai *crisp* pada setiap aturan yang dibuat.

4. *Defuzzifikasi*

Tahapan *Defuzzifikasi* adalah tahapan akhir sebelum menentukan output yang diinginkan. Pada tahap ini dilakukan proses mengubah nilai berupa nilai *fuzzy* menjadi nilai tegas atau *crisp* menggunakan metode yang ada. Pada metode *fuzzy* Mamdani metode *defuzzifikasi* yang digunakan adalah metode *centroid*.

2.9. Arduino IDE

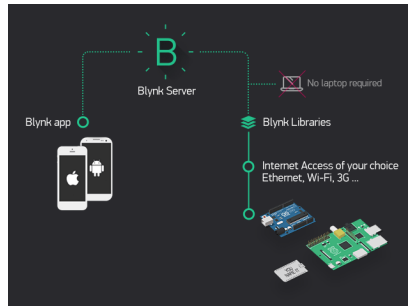
Arduino IDE (*Integrated Development Environment*) merupakan sebuah aplikasi atau software yang digunakan untuk menulis program, meng-*compile* program menjadi bentuk biner dan mengunggah kedalam memori mikrokontroler[21]. Arduino IDE beroperasi dengan menggunakan bahasa pemrograman C++ yang telah disederhanakan. Software ini sering digunakan untuk melakukan pemrograman pada perangkat Arduino. Pada awalnya Arduino IDE digunakan sebagai software *processing* sebelum kemudian dikembangkan menjadi Arduino IDE khusus untuk Arduino. Namun selain Arduino, Arduino IDE juga dapat digunakan untuk melakukan pemrograman pada perangkat mikrokontroler lainnya seperti ESP32. Pada proyek ini Arduino IDE digunakan untuk melakukan pemrograman pada ESP32 untuk menerima data sensor, memproses data sensor, dan fungsi lainnya sesuai rancangan.



Gambar 8. Tampilan pada software Arduino IDE

2.10. Blynk

Blynk merupakan salah satu aplikasi mobile yang memungkinkan penggunaannya untuk melakukan kontrol pada perangkat Mikrokontroler seperti Arduino, ESP32, Raspberry Pi, dan sebagainya[22]. Terdapat beberapa metode yang dapat digunakan mulai dari menggunakan kabel hingga nirkabel. Namun yang paling populer saat ini adalah penggunaan *Blynk* dengan melalui jaringan internet. Aplikasi ini sangat sering digunakan untuk berbagai proyek baik sederhana maupun kompleks yang berkaitan dengan *Internet Of Things* atau IoT. Seperti digunakan dalam Perancangan sistem *monitoring smart bins* berbasis IoT menggunakan aplikasi *Blynk*[23]. Pada aplikasi *Blynk* dapat dibuat sebuah tampilan interface berupa tombol, LED, tampilan data analog, hingga tampilan gambar. Pada proyek ini aplikasi *Blynk* digunakan sebagai *interface* manusia untuk *memonitoring* nilai suhu dan kelembaban tanah pada tanaman cabai. Aplikasi ini akan diinstal pada perangkat *smartphone*.



Gambar 9. Ilustrasi Aplikasi *Blynk*

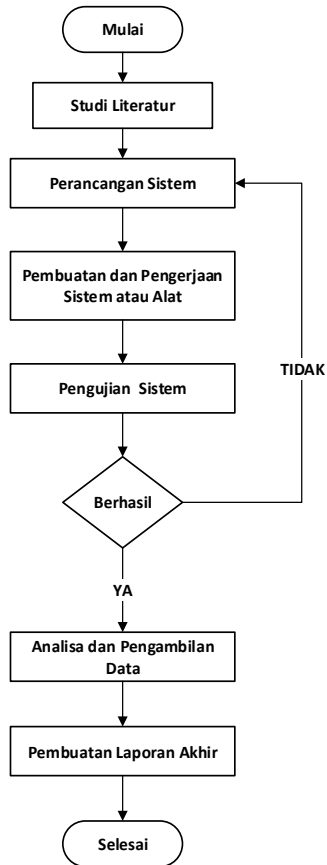
Sumber: (<https://docs.blynk.cc/>)

2.11. Smartphone

Smartphone merupakan elemen yang sangat penting bagi kehidupan masyarakat saat ini karena digunakan sebagai media komunikasi dan informasi. Berbagai kalangan mulai dari anak-anak hingga orang tua memiliki perangkat ini, kemudahan dan fungsionalitas yang bisa didapat dari perangkat ini menjadi salah satu alasan ramainya penggunaannya. Dengan menggunakan *Smartphone* kegiatan komunikasi dengan orang jauh dapat dilakukan dengan mudah melalui media sosial ataupun internet. *Smartphone* yang berkembang saat ini memiliki beberapa sistem operasi seperti Android, iOS, dan lainnya. Pada proyek ini *Smartphone* digunakan sebagai perangkat *interface* untuk *memonitoring* kondisi lingkungan disekitar tanaman cabai dan untuk melakukan proses penyiraman secara manual.

Bab 3. Metodologi Penelitian

Metode penelitian perlu dibuat agar penelitian menjadi lebih terstruktur sehingga akan mendapatkan hasil yang sesuai harapan. Metode penelitian yang dilakukan pada penelitian kali ini dapat dilihat pada gambar 10 dibawah ini.

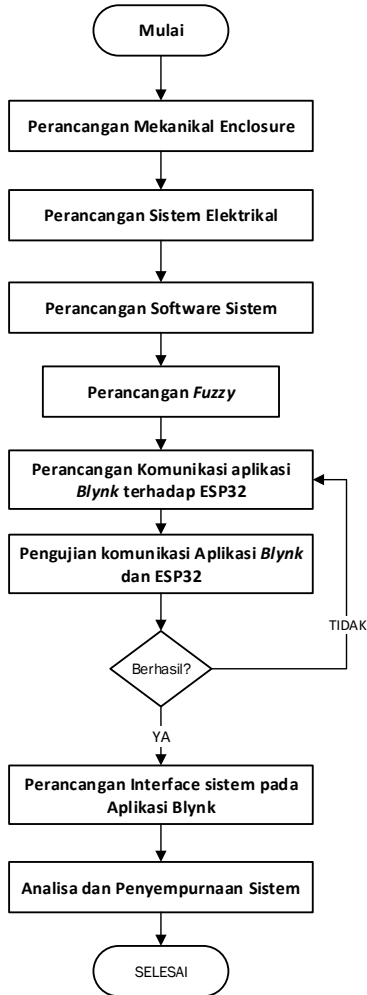


Gambar 10. Flowchart Penelitian

Tahapan pertama yang dilakukan adalah studi literatur, pada tahapan ini dilakukan literasi dan pengkajian dari berbagai sumber mengenai prinsip prinsip sistem yang akan dibuat, seperti penggunaan sensor dan aplikasi pada metode *fuzzy*. Kemudian dilanjutkan dengan tahapan perancangan sistem.

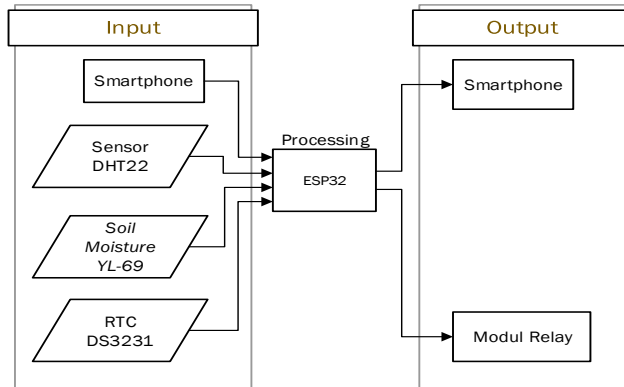
3.1. Perancangan

Tahapan Perancangan sistem perlu dilakukan untuk membuat gambaran proses pengerjaan sistem yang akan dibuat dan menentukan sistem yang akan dibuat agar sesuai dengan tujuan alat. Tahapan perancangan pada pembuatan alat dan penelitian ini dapat dilihat pada gambar 11 di bawah ini.



Gambar 11. Flowchart tahapan perancangan

Dalam proses perancangan, diperlukan adanya diagram blok sistem. Diagram blok sistem ini dibuat untuk menentukan komponen serta sistem yang diperlukan untuk membuat alat. Diagram blok sistem pemantauan dan penyiraman otomatis pada tanaman cabai menggunakan sistem IoT berbasis *Smartphone* dapat dilihat pada gambar 12 dibawah ini.

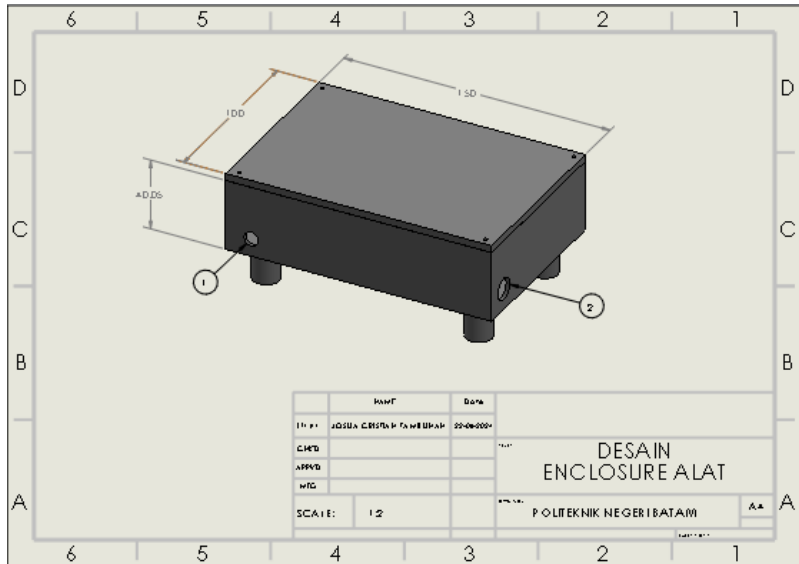


Gambar 12. Diagram Blok Sistem

Pada gambar diagram blok sistem diatas, dapat dilihat bahwa terdapat tiga unsur yang menjadi faktor penting dalam membuat alat. Tiga unsur tersebut adalah Input yang berupa perangkat sensor, modul RTC DS3231 dan juga *smartphone*, Processing yang berupa perangkat ESP32, dan juga Output yang berupa *smartphone* dan juga Modul Relay. Perangkat *smartphone* pada bagian Input digunakan untuk melakukan penyiraman manual pada tanaman cabai, perangkat *smartphone* akan diinstall aplikasi *Blynk*. *Smartphone* tersebut akan mengirim data kepada ESP32 menggunakan jaringan internet melalui aplikasi *Blynk* tersebut. Bagian input lainnya adalah Sensor DHT22, sensor ini digunakan untuk mendeteksi kondisi suhu udara dan kelembaban. Kemudian ada *Soil Moisture* YL-69 yang digunakan untuk mendeteksi kelembaban tanah. Sensor DHT22 dan *Soil Moisture* YL-69 akan terhubung dengan ESP32 menggunakan koneksi kabel atau *wiring*. Input tersebut akan dilakukan processing pada ESP32, proses processing dilakukan dengan menggunakan aplikasi Arduino IDE, Processing yang dilakukan akan menggunakan metode *fuzzy*. Pada bagian Output terdapat perangkat *smartphone* yang sama dengan yang ada di bagian Input. pada bagian output *smartphone* akan menampilkan *monitoring* suhu, kelembaban, dan juga kelembaban tanah yang dikirim oleh ESP32 melalui jaringan internet. Perangkat kedua yaitu Modul Relay, modul relay ini akan digunakan untuk menghidupkan solenoid valve sehingga proses penyiraman dapat terjadi. Modul relay dihubungkan dengan ESP32 menggunakan koneksi kabel.

3.1.1. Perancangan Mekanikal

Tahapan perancangan mekanikal dilakukan untuk membuat *enclosure* atau *cover* dari alat yang dibuat. Rancangan mekanikal dari *cover* dapat dilihat pada gambar 13 di bawah ini.



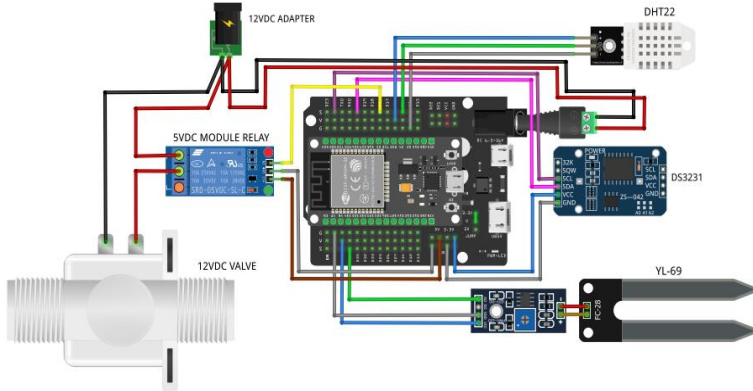
Gambar 13. Drawing Mekanikal *Enclosure* alat

- Keterangan :**
1. Socket Output Solenoid Valve
 2. Socket Power Supply

Enclosure ini dibuat untuk melindungi rangkaian elektrikal yang akan digunakan dalam sistem seperti perangkat ESP32, modul relay, modul RTC DS3231 dan lain-lain. Sedangkan perangkat perangkat seperti sensor DHT22 dan sensor YL-69 akan diletakkan diluar *enclosure* karena perangkat tersebut digunakan untuk mendeteksi lingkungan luar. Bahan yang akan digunakan sebagai bahan dasar *enclosure* ini adalah plastik *rubber*. Ukuran yang akan digunakan pada *enclosure* ini adalah 150mm × 100mm × 40mm untuk masing-masing p × l × t, dan tinggi penyangga 15mm.

3.1.2. Perancangan Sistem Elektrikal

Tahapan perancangan elektrikal dilakukan untuk membuat skematik koneksi maupun hubungan antara setiap komponen elektrikal yang akan digunakan. Tahapan perancangan dilakukan dengan menggunakan aplikasi *fritzing*. Perancangan elektrikal pada penelitian ini dapat dilihat pada gambar 14 di bawah ini.



Gambar 14. Rangkaian Elektrikal

Tahapan ini dilakukan untuk menentukan koneksi antar komponen yang diperlukan untuk membuat alat. Catu daya yang digunakan memiliki tegangan 12Vdc, sedangkan tegangan input yang diperlukan untuk ESP32 dan juga sensor yang digunakan adalah 3.3V/5V. Oleh karena itu, digunakan sebuah *expansion board* ESP32 yang langsung dipasang dengan papan ESP32 secara *plugin*, sehingga baik ESP32 dan juga sensor masing-masing dapat menerima tegangan 3.3V/5V. 12Vdc akan digunakan sebagai input tegangan kran air elektrik atau *solenoid valve*. Dalam menghubungkan setiap input sensor dan juga output modul relay dengan perangkat kontrol ESP32 akan mengikuti konfigurasi pin. Konfigurasi tersebut dapat dilihat pada tabel di bawah ini.

Tabel 4. Konfigurasi Pin Sensor DHT22 dengan ESP32

Sensor DHT22	ESP32
VCC	3.3V
GND	GND
Out	GPIO4

Tabel 5. Konfigurasi Pin Sensor YL-69 dengan ESP32

Sensor YL-69	ESP32
VCC	3.3v
GND	GND
A0	GPIO34

Tabel 6. Konfigurasi Pin RTC DS3231 dengan ESP32

RTC DS3231	ESP32
VCC	3.3V
GND	GND
SDA	GPIO21
SCL	GPIO22

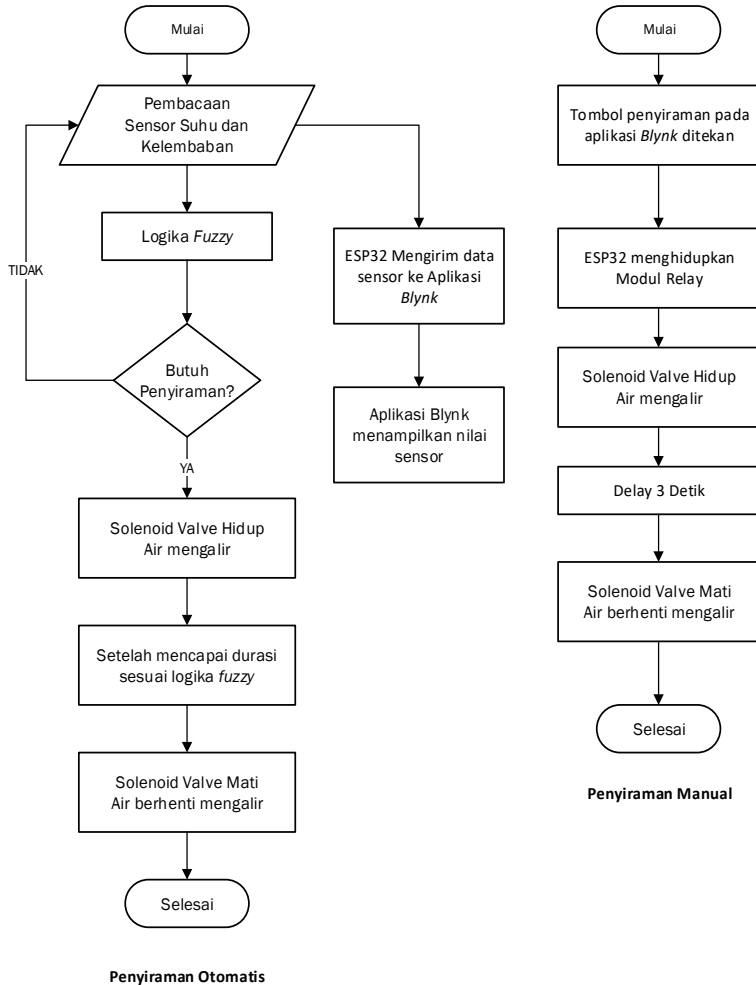
Tabel 7. Konfigurasi Pin Modul Relay dengan ESP32

Modul Relay	ESP32
VCC	3.3V
GND	GND
IN1	GPIO5

Dengan menggunakan konfigurasi pin di atas sensor DHT22 dan sensor YL-69 serta RTC DS3231 masing-masing mendapatkan input VCC sebesar 3.3VDC dari ESP32 *expansion board*. Sedangkan modul relay mendapatkan input untuk VCC sebesar 5VDC.

3.1.3. Perancangan sistem kerja

Tahapan perancangan ini dilakukan untuk membuat alur kerja dari sistem yang akan dibuat. Alur kerja sistem ini berfungsi sebagai referensi saat akan melakukan pemrograman pada perangkat processing yaitu ESP32.



Gambar 15. Flowchart sistem kerja alat

Sistem kerja penyiraman terdiri dari dua bagian, yaitu penyiraman otomatis dan juga penyiraman manual yang dilakukan melalui sistem IoT menggunakan aplikasi *Blynk*. Hal yang membedakan adalah proses penyiraman manual dapat melakukan bypass penyiraman sehingga dapat dilakukan secara manual, dengan dua syarat yaitu waktu tidak menunjukkan jam penyiraman terjadwal dan kelembaban tanah kurang harus dari 45% serta durasi pada penyiraman manual adalah konstan yaitu diatur dalam waktu 3 detik saja.

Sistem penyiraman otomatis bekerja secara berkala dengan waktu tertentu yaitu pagi hari (pukul 07:10) dan juga sore hari (pukul 16:10). Proses awal adalah sensor suhu DHT22 mendeteksi suhu udara dan sensor YL-69 mendeteksi kelembaban tanah yang ada pada pot tanaman cabai, kemudian data sensor tersebut akan dikirim ke ESP32. Kemudian ESP32 akan mengirimkan data sensor ke aplikasi *Blynk* disaat bersamaan dilakukan pemrosesan menggunakan metode *fuzzy* yang telah ditanamkan pada mikrokontroler tersebut. Setelah jeda waktu selama 1 menit terjadi kondisi yang sama, *Fuzzy* akan memutuskan apakah tanaman membutuhkan penyiraman atau tidak. Jika diputuskan perlu dilakukan penyiraman maka ESP32 akan mengirim sinyal ke modul relay sehingga valve solenoid atau kran air elektrik akan terbuka dan ESP32 akan mengirimkan status ke aplikasi *Blynk*. Penyiraman aktif dengan durasi yang telah ditentukan sebelumnya oleh logika *fuzzy*, yang diakhiri dengan solenoid kembali tertutup.

3.1.4. Perancangan sistem *Fuzzy*

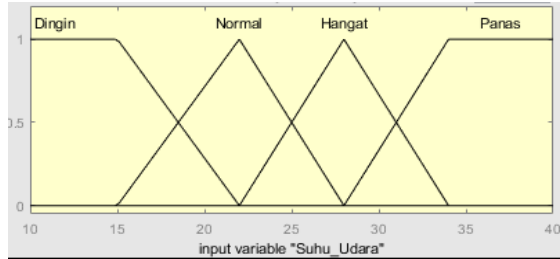
Setelah dilakukan perancangan sistem elektrikal dan juga perancangan sistem alat untuk mengetahui cara atau alur kerja sistem, selanjutnya adalah melakukan rancangan *fuzzy*. Perancangan *fuzzy* dilakukan untuk mengklasifikasikan atau mengolah data inputan berupa sensor sehingga dapat mengeluarkan output sesuai dengan diagram alir sistem. Metode *fuzzy* tersebut akan ditanamkan pada mikrokontroler ESP32 menggunakan pemrograman Arduino IDE. Tahap perancangan *fuzzy* dilakukan dengan beberapa tahap berikut ini.

A. *Fuzzifikasi*

Tahapan pertama pada perancangan *fuzzy* adalah *fuzzifikasi*. Proses *fuzzifikasi* merupakan proses pembentukan fungsi keanggotaan dari variabel parameter yang akan digunakan yaitu sensor suhu DHT22 dan sensor YL-69 sebagai input dan juga durasi penyiraman sebagai output.

Tabel 8. Fungsi keanggotaan *fuzzy* input sensor suhu

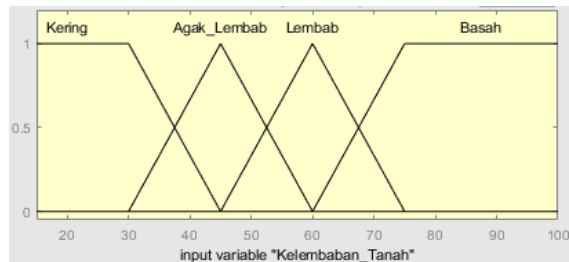
No	Derajat Suhu	Kondisi
1	0°C - 22°C	Dingin
2	15°C - 28°C	Normal
3	22°C - 34°C	Hangat
4	28°C - 40°C	Panas



Gambar 16. Grafik himpunan variabel suhu udara

Tabel 9. Fungsi keanggotaan fuzzy input sensor kelembaban

No	Persentase Kelembaban	Kondisi
1	0% - 45%	Kering
2	30% - 60%	Agak Lembab
3	45% - 75%	Lembab
4	60% - 100%	Basah

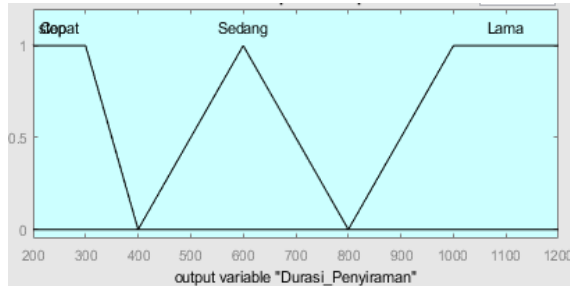


Gambar 17. Grafik himpunan variabel kelembaban tanah

Pada sensor suhu DHT22 derajat keanggotaan dibagi menjadi 4 kelas yaitu Dingin, Normal, Hangat, dan Panas dengan nilai dari masing-masing variabel yang dapat dilihat pada Tabel 8. Sedangkan parameter kedua yaitu kelembaban tanah, fungsi keanggotaan dibagi menjadi 4 kelas yaitu Kering, Agak Lembab, Lembab, dan Basah dengan nilai dari masing-masing variabel yang dapat dilihat pada Tabel 9.

Tabel 10. Fungsi keanggotaan fuzzy output durasi penyiraman

No	Derajat Keanggotaan	Durasi
1	0	Tidak menyiram
2	2 – 4	Cepat
3	4 – 8	Sedang
4	8 – 12	Lama



Gambar 18. Grafik himpunan variabel durasi penyiraman

Pada fungsi keanggotaan output berupa durasi penyiraman, didefinisikan dalam tiga kriteria yaitu cepat, sedang, dan lama. Untuk durasi yang diberikan tersebut diatur masing-masing 2-4 detik, 4-8 detik, dan 8-12 detik.

B. Rule Base

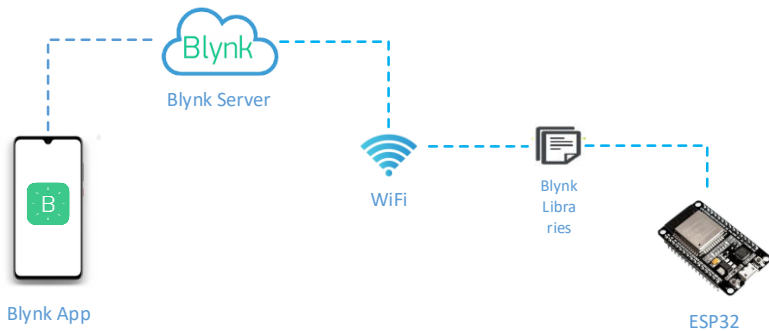
Pada tahap ini fungsi keanggotaan yang telah didefinisikan dan diklasifikasikan akan dibuat sebuah ketentuan atau aturan untuk menghasilkan output berupa durasi penyiraman dari parameter input suhu dan kelembaban tanah menggunakan operator *AND*. *Rule Base* pada penelitian ini dapat dilihat pada tabel 11 di bawah ini.

Tabel 11. Rule Base

No	Suhu	Kelembaban tanah	Durasi
1	Dingin	Kering	Cepat
2	Dingin	Agak Lembab	Cepat
3	Dingin	Lembab	Cepat
4	Dingin	Basah	Tidak Menyiram
5	Normal	Kering	Sedang
6	Normal	Agak Lembab	Cepat
7	Normal	Lembab	Cepat
8	Normal	Basah	Tidak menyiram
9	Hangat	Kering	Lama
10	Hangat	Agak Lembab	Sedang
11	Hangat	Lembab	Cepat
12	Hangat	Basah	Tidak menyiram
13	Panas	Kering	Lama
14	Panas	Agak Lembab	Sedang
15	Panas	Lembab	Sedang
16	Panas	Basah	Tidak Menyiram

3.1.5. Perancangan sistem komunikasi ESP32 dan aplikasi *Blynk*

Tahapan selanjutnya yang dilakukan adalah perancangan dan pembangunan sistem komunikasi antara ESP32 dengan aplikasi *Blynk* yang akan menjadi aplikasi dari sistem *Internet Of Things* yang akan diterapkan. ESP32 dan juga *Blynk* terhubung dengan menggunakan komunikasi *WiFi*. Perancangan ini melibatkan perangkat keras *smartphone* yang sudah terinstal aplikasi *Blynk* dan juga melalui pemrograman software Arduino IDE. Diagram perancangan sistem komunikasi ESP32 dan aplikasi *Blynk* dapat dilihat pada gambar 16 di bawah ini.



Gambar 19. Topologi komunikasi ESP32 dan *Blynk*

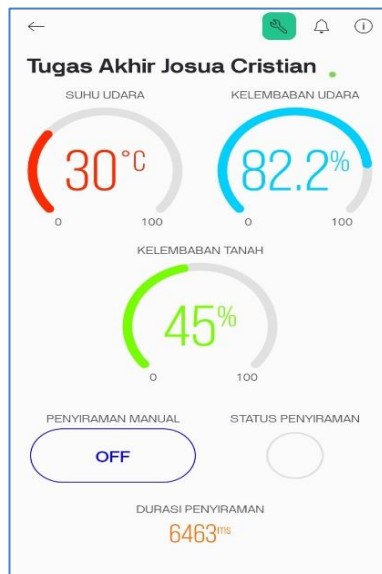
Perancangan komunikasi ini melibatkan dua bagian, yaitu pada *smartphone* yang sudah terinstal aplikasi *Blynk* dan juga pada ESP32 melalui pemrograman Arduino IDE. Langkah pertama yang dilakukan setelah melakukan penginstalan aplikasi *Blynk* adalah membuat sebuah akun *Blynk* pada web. Setelah itu dapat dibuat sebuah project baru pada aplikasi dengan memilih komunikasi WiFi dengan menggunakan widget tombol dan juga sebuah LED pada aplikasi. Setelah itu dilakukan pemrograman pada Arduino IDE untuk memasang *library* yang diperlukan seperti dibawah ini.

```
#include <WiFi.h>
#include <BlynkSimpleEsp32.h>
```

Selain menggunakan library di atas perlu juga dilakukan pemrograman pada Arduino IDE untuk menghubungkan ESP32 dengan server *Blynk*, pemrograman tersebut dilakukan dengan memasukkan *authentication code* yang diberikan oleh server pada program. Setelah itu, dilakukan pengaturan pada aplikasi *Blynk* untuk virtual pin yang akan digunakan seperti V0 untuk tombol dan V1 untuk LED. Dengan alamat pin tersebut dilakukan pemrograman pada Arduino IDE agar tombol yang ada pada aplikasi *Blynk* dapat menghidupkan LED pada ESP32 dan mengirim sinyal kembali ke LED yang ada di aplikasi. Jika hal ini berhasil maka sistem komunikasi antara aplikasi dan juga ESP32 sudah berhasil.

3.1.6. Perancangan GUI pada aplikasi *Blynk*

Setelah komunikasi ESP32 dan juga aplikasi *Blynk* dapat dilakukan, selanjutnya adalah melakukan tahapan perancangan GUI atau *Grafical User Interface* sistem *monitoring* dan kontrol tanaman cabai pada aplikasi. GUI akan terdiri dari beberapa widget seperti tombol, LED dan juga *Value Display*. Widget tersebut akan diberikan alamat yang sesuai pada saat konfigurasi di aplikasi *Blynk*. Dengan menggunakan alamat yang telah terkonfigurasi tersebut selanjutnya adalah mengintegrasikan dengan perangkat ESP32. Proses integrasi ini dilakukan dengan pemrograman menggunakan aplikasi Arduino IDE. Pemrograman dilakukan agar setiap widget pada aplikasi *Blynk* dapat berfungsi dengan baik. GUI yang akan digunakan pada penelitian ini dapat dilihat pada gambar 17 di bawah ini.



Gambar 20. GUI sistem pemantauan dan penyiraman *Blynk*

GUI aplikasi *Blynk* menampilkan beberapa menu yang digunakan untuk *memonitoring* suhu udara dan kelembaban udara, dan kelembaban tanah. Pada GUI tersebut juga memiliki notifikasi sistem penyiraman apakah sedang *ON/OFF*, dan juga terdapat fitur kontrol penyiraman manual yang dapat digunakan untuk *mem-bypass* penyiraman tanaman cabai secara manual sesuai dengan kriteria yang telah ditentukan. Pada GUI tersebut juga terdapat widget yang menampilkan durasi penyiraman ketika sistem penyiraman aktif.

3.2. Alat dan Bahan

Tabel 12. Alat Bahan dan Estimasi biaya

No.	Alat/bahan	Harga Satuan (Rp.)	Jumlah	Total (Rp.)	Keterangan
1	ESP32 Dev-Kit 1	100.000	1	100.000	Dana Pribadi
2	Sensor DHT22 AM2302	50.000	1	50.000	Dana Pribadi
3	<i>Soil Moisture YL-69 FC-28 YL-38</i>	10.000	1	10.000	Dana Pribadi
4	Modul RTC DS3231	29.000	1	29.000	Dana Pribadi
5	Modul Relay 5VDC 1 chanel	10.000	1	10.000	Dana Pribadi
6	12VDC Solenoid Valve ½ Inch	50.000	1	50.000	Dana Pribadi
7	12VDC Power Adapter 1A	25.000	1	25.000	Dana Pribadi
8	Kabel Jumper	500	40	20.000	Dana Pribadi
	Total			294.000	

3.3. Pengujian

3.3.1. Pengujian sensor suhu DHT22 dan sensor kelembaban YL-69

Pengujian dilakukan selama 7 kali dengan selang waktu 2 jam untuk melihat akurasi pembacaan sensor pada serial monitor Arduino IDE dan juga aktual menggunakan *Thermometer* untuk DHT22 dan juga kondisi aktual untuk sensor YL-69. Hal ini perlu dilakukan untuk menguji keakuratan sensor sebagai input metode *fuzzy*. Pengujian dari sensor yang pertama adalah pengujian Sensor DHT22 dalam membaca kondisi suhu udara. Pengujian dilakukan dengan membandingkan hasil pembacaan suhu pada DHT22 dengan pembacaan *Thermometer*. Pengujian sensor yang kedua adalah pengujian sensor YL-69 yang digunakan untuk mendeteksi kelembaban tanah. Pengujian dilakukan dengan membandingkan hasil pembacaan sensor YL-69 dengan kondisi aktual pada tanah.

3.3.2. Pengujian Sistem *Fuzzy*

Tahapan pengujian selanjutnya adalah pengujian dari sistem *fuzzy* pada mikrokontroler ESP32. Pengujian dilakukan untuk mengetahui efektifitas dari sebuah sistem *fuzzy* yang ditanamkan pada ESP32 melalui pemrograman pada Arduino IDE. Pengujian dilakukan dengan membandingkan hasil keluaran dari proses *fuzzy* yang ditampilkan pada Arduino IDE dengan hasil pemrosesan *fuzzy* menggunakan aplikasi Matlab. Pengujian dilakukan dengan menggunakan sampel dua buah input dan satu buah output baik pada ESP32 melalui Arduino IDE dan juga Matlab dengan nilai parameter yang sama.

3.3.3. Pengujian Sistem Penyiraman Otomatis

Tahapan pengujian ini dilakukan dengan menguji alat penyiraman secara langsung pada tanaman cabai. Pengujian ini dilakukan untuk melihat hasil sistem penyiraman dalam mengendalikan durasi penyiraman berdasarkan kondisi lingkungan yang dijadikan parameter input. Kemudian dilakukan juga pengujian terhadap hasil perubahan kelembaban setelah penyiraman aktif dengan memperhatikan tingkat kelembaban tanah hingga dalam kondisi stabil dan juga pengujian perbandingan hasil kelembaban dengan penyiraman secara manual.

3.3.4. Pengujian Sistem IoT pada aplikasi *Blynk*

Pengujian selanjutnya adalah pengujian sistem IoT pada aplikasi *Blynk* yang digunakan untuk memantau suhu, kelembaban udara, dan kelembaban tanah serta fungsionalitas aplikasi *Blynk* dalam melakukan penyiraman manual. Keberhasilan pengujian sistem dilihat dari fungsionalitas fitur dan juga jarak aksesibilitas antara alat dan juga *smartphone* dalam menampilkan hasil pemantauan dengan baik dan juga keberhasilan penyiraman manual.

Bab 4. Hasil dan Pembahasan

Data hasil penelitian merupakan hasil pengujian dari seluruh alat dan sistem yang telah dibuat dan diimplementasikan terhadap objek penelitian sesuai dengan tujuan dilakukannya penelitian ini. Data hasil penelitian diambil dengan beberapa metode seperti *sampling* dan juga perbandingan terhadap aspek eksternal..

4.1. Implementasi alat pada tanaman cabai

Hasil implementasi sistem pada objek merupakan pengaplikasian alat yang telah dirancang secara langsung terhadap objek penelitian yaitu tanaman cabai. Tanaman cabai yang digunakan merupakan tanaman cabai yang sudah berusia 2 bulan 3 minggu saat pengujian dan ditanam di dalam sebuah *polybag* hitam berukuran tinggi : 18cm dan diameter *polybag* : 14cm. Media tanam yang digunakan adalah campuran tanah gambut hitam dan serbuk sabut kelapa.



Gambar 21. Implementasi alat pada tanaman cabai

Sistem kontrol dari alat berada di dalam box berwarna hitam yang di dalamnya terdapat rangkaian elektrikal seperti ESP32, RTC DS3231, dan juga Relay. Sedangkan untuk input berupa sensor DHT22 diletakkan disekitar tanaman cabai yaitu di atas tanah tempat cabai tertanam dengan menggunakan bantuan sebuah batang besi untuk menahan sensor. Sensor YL-69 ditancapkan kedalam tanah dengan jarak 10-13 cm dari permukaan atas tanah, hal ini bertujuan untuk mendeteksi kelembaban tanah yang ada di sekitar akar tanaman bukan sekedar

berada di atas saja. Untuk *solenoid valve* atau kran air elektrik dihubungkan dengan rangkaian menggunakan *socket* sehingga memudahkan peletakan dari kran air. Kran air dapat dipindahkan sesuai kebutuhan untuk memenuhi jarak antara *inlet* selang pengairan dan juga *outlet* selang. Dari hasil implementasi, setiap perangkat yang telah dirancang dapat berfungsi dengan baik.

4.2. Hasil pengujian sensor DHT22 dan *Soil Moisture YL-69*

Sensor DHT22 diletakkan di atas tanaman dengan bantuan sebuah batang aluminium untuk menjangkau daerah sekitar tanaman dan juga sensor YL-69 yang ditancapkan langsung pada bagian samping polybag dengan jarak 10-13 cm dari permukaan atas tanah yang telah diaplikasikan seperti pada gambar 22.



Gambar 22. Peletakan sensor DHT22 dan juga Sensor YL-69

1. Pengujian Sensor DHT22

Tabel di bawah ini menunjukkan hasil pengujian sensor DHT22 dengan perbandingan menggunakan *thermometer HTC-2*. Pengujian dilakukan dengan mengambil sampel data sebanyak 6 kali dalam selang waktu 2 jam pada pagi hari hingga sore hari.

Tabel 13. Pengujian Sensor DHT22

No	Waktu	Sensor DHT22 (°C)	<i>Thermometer</i> (°C)	Error (%)
1	08:00	28.6°C	28.8	0.69%
2	10:00	30.0°C	30.2	0.66%
3	12:00	32.0°C	32.3	0.92%
4	14:00	31.2°C	30.9	0.97%
5	16:00	29.8°C	30	0.66%
6	18:00	28.2°C	28.5	1.05%
Rata-rata error				0.82%

Pada hasil pengujian sensor DHT22 dengan perbandingan menggunakan *Thermometer* mendapatkan hasil error sebesar 0.82%. Perhitungan pada tingkatan error dilakukan dengan menggunakan rumus(2) berikut ini.

$$\%error = \frac{Thermometer - DHT22}{Thermometer} \times 100\% \quad (2)$$

Hasil pengujian dari sensor DHT22 di atas menunjukkan bahwa sensor DHT22 memiliki tingkat akurasi yang masih cukup baik dalam mendeteksi derajat suhu udara jika dibandingkan dengan *thermometer HTC-2*. Dengan begitu penerepan sensor DHT22 dalam mendeteksi suhu udara yang akan digunakan dalam parameter input sistem *fuzzy* nantinya akan mendapatkan hasil yang baik.

2. Pengujian Sensor YL-69

Pada pengujian sensor YL-69 sebagai pendeteksi kelembaban tanah, sebelumnya telah dilakukan kalibrasi untuk menentukan persentase kelembaban dari nilai ADC yang dihasilkan oleh sensor ketika mendeteksi kelembaban tanah. Perhitungan dari persentase tersebut dapat dilihat pada rumus(1). Pada saat pengujian dihasilkan data bahwa pembacaan nilai ADC sensor pada ESP32 menunjukkan angka sekitar 800 saat kondisi tanah sangat basah, sehingga pada proses *mapping* diputuskan bahwa persentase kelembaban maksimal yang bernilai 100% dari pembacaan ADC berada pada angka 800, dan sangat kering pada nilai 4095 dengan persentase 0%. Sebagai perbandingan, digunakan sebuah alat ukur kelembaban tanah atau *soil meter 3in1* yang memiliki range 1-10 untuk menguji nilai kelembaban aktual pada tanah.

Tabel 14. Pengujian Sensor YL-69

No	Kondisi Aktual Tanah	Pembacaan ADC	Persentase Sensor YL-69 (%)	Pengukuran <i>Soil Meter</i> (%)	Error(%)
1	Kering	3227	27%	26%	3.8%
2	Kering	2866	38%	36%	5.5%
3	Lembab	2260	59%	61%	3.2%
4	Lembab	1968	66%	69%	4.3%
5	Basah	1695	73%	75%	2.6%
6	Basah	1220	85%	83%	2.4%
Rata-rata error					3.63%

Dari tabel di atas dapat dilihat hasil pengujian sensor YL-69 setelah dilakukan kalibrasi/*mapping* dan juga membandingkan hasil pembacaan sensor dengan alat ukur *soil meter 3in1*. Dengan rata-rata error sebesar 3.63% dapat disimpulkan bahwa sensor YL-69 dapat mendeteksi kelembaban tanah dari kering hingga basah dengan akurasi yang masih cukup baik untuk digunakan sebagai parameter input kelembaban tanah pada alat yang dirancang.

4.3. Hasil pengujian *Fuzzy Mamdani*

Hasil pengujian sistem *fuzzy Mamdani* ini merupakan hasil pengujian dari sistem atau metode *fuzzy* yang digunakan dalam membuat keputusan durasi penyiraman dari parameter suhu dan kelembaban, dimana metode *fuzzy* ini telah diterapkan langsung dengan alat yang telah dirancang. Hasil pengujian merupakan perbandingan durasi penyiraman yang dihasilkan oleh sistem melalui ESP32 dengan hasil durasi penyiraman yang ditampilkan menggunakan aplikasi Matlab.

```

Temperature : 30.10
Soil Moisture : 3581
Soil Moisture Percentage : 37
Durasi Penyiraman: 83.97
    
```

Gambar 23. Hasil pengujian *fuzzy* pada ESP32



Gambar 24. Hasil pengujian *fuzzy* pada aplikasi Matlab

Hasil pengujian dari sistem *fuzzy* pada sistem ESP32 dan aplikasi Matlab dapat dilihat pada tabel 15 di bawah ini.

Tabel 15. Pengujian sistem *fuzzy Mamdani*

No	Suhu Udara(°C)	Kelembaban(%)	ESP32	Matlab	Error (%)
1	30.1	37	84.0	84.3	0.35%
2	30.0	64	44.3	44.4	0.22%
3	28.4	31	100.9	101.0	0.09%
4	30.4	72	46.4	46.6	0.42%
5	29.4	57	41.8	42.0	0.47%
6	30.2	62	44.7	44.9	0.44%
7	31.0	55	48.9	48.9	0.00%
8	32.2	30	103.7	104.0	0.28%
Rata-rata error					0.28%

Dari data yang dilampirkan pada tabel 15 di atas, hasil pengujian metode *fuzzy* yang dilakukan sebanyak 8 kali menunjukkan adanya perbedaan nilai yang dihasilkan oleh sistem *fuzzy* ESP32 dengan aplikasi Matlab. Dimana durasi yang dihasilkan oleh aplikasi Matlab cenderung lebih besar dari sistem ESP32. Hal ini dapat terjadi karena sebuah sistem atau metode *fuzzy* yang diterapkan pada mikrokontroler ESP32 menggunakan sebuah *library*, dimana *library* tersebut memiliki tingkat akurasi dan juga efektifitas pengolahan yang berbeda dengan perhitungan matematika secara manual melalui aplikasi Matlab. Kemudian perbedaan nilai juga diakibatkan karena pada aplikasi Matlab terdapat pembulatan sebuah nilai baik itu input maupun output sehingga output durasi yang dihasilkan sedikit berbeda dengan yang ada di serial monitor ESP32.

Perbedaan nilai yang dihasilkan juga dikarenakan metode *fuzzy* yang digunakan pada penelitian ini adalah metode *fuzzy* Mamdani, dimana hasil keluaran atau *output* dari *fuzzy* berupa himpunan *fuzzy* sehingga angka-angka akan lebih bervariasi dan juga dapat berupa bilangan desimal. Sedangkan jika dibandingkan dengan metode *fuzzy* Sugeno, output yang dihasilkan tidak bervariasi karena hasil keluaran metode Sugeno berupa bilangan konstanta atau tetap untuk setiap *linguistik* nya. Meskipun terdapat beberapa perbedaan, pada pengujian yang dilakukan juga terdapat hasil durasi dengan nilai yang sama antara ESP32 dengan aplikasi Matlab. Ini menunjukkan bahwa pada suatu kondisi atau nilai, pengolahan hasil metode *fuzzy* pada ESP2 dan aplikasi Matlab dapat menghasilkan output yang sama.

Secara keseluruhan hasil pengujian pada metode *fuzzy* menunjukkan bahwa hasil perbandingan metode *fuzzy* pada ESP32 dengan perhitungan manual pada aplikasi Matlab memiliki persentase error sebesar 0.28% dengan data pengujian sebanyak 8 kali. Dari hasil pengujian ini, metode *fuzzy* yang diterapkan pada sistem penyiraman otomatis untuk menentukan durasi penyiraman dapat disimpulkan sudah cukup efektif dan dapat menghasilkan tingkatan akurasi yang cukup baik dalam menentukan durasi penyiraman.

4.4. Hasil Pengujian Sistem Penyiraman

Pengujian sistem penyiraman dari alat yang dibuat dilakukan selama 7 hari. Sebelum dilakukan pengujian, terlebih dahulu dilakukan pengaturan pada jumlah atau debit air yang mengalir setiap satuan waktunya dengan memperhatikan luas dari pot, sehingga nantinya jumlah air yang keluar pada saat penyiraman aktif dapat mencukupi kebutuhan air dari tanaman cabai. Pengaturan debit air dilakukan dengan menambahkan sebuah kran air manual di antara sumber air dan input solenoid valve.

Tabel 16. Pengaturan debit air melalui kran manual

No	Durasi	Jumlah air (mL)
1	2 detik	±100mL
2	4 detik	±200mL
3	6 detik	±300mL
4	8 detik	± 400mL
5	10 detik	± 500mL

1. Hasil Pengujian Alat Penyiraman Otomatis

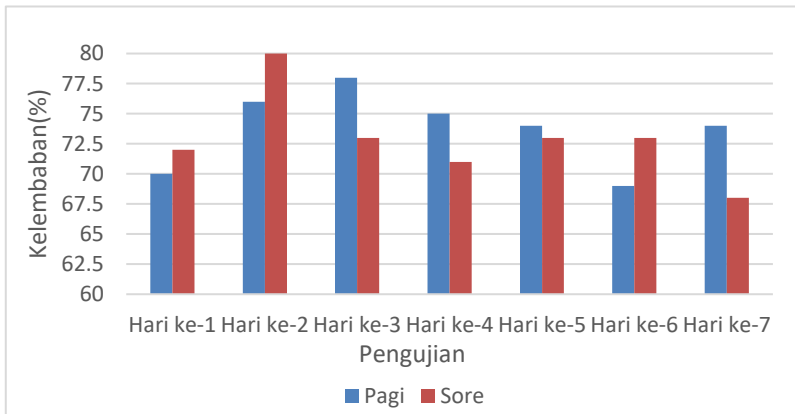
Pengujian penyiraman dilakukan dengan memperhatikan hasil dari alat yang dibuat dalam mengendalikan durasi penyiraman ketika aktif hingga mati. Hasil dari pengujian sistem penyiraman otomatis dapat dilihat pada tabel 17 di bawah ini.

Tabel 17. Hasil pengujian sistem penyiraman otomatis

No	Hari	Waktu	Suhu (°C)	Kelembaban (%)	Durasi (detik)	Hasil Kelembaban (%)
1	Hari ke-1	Pagi (07 :10)	29.9°C	46%	5.4s	70%
		Sore (16 : 10)	30.6°C	65%	4.6s	72%
2	Hari ke-2	Pagi (07 :10)	30°C	51%	5.1s	76%
		Sore (16 : 10)	28°C	40%	7.6s	80%
3	Hari ke-3	Pagi (07 :10)	29.3°C	71%	2.6s	78%
		Sore (16 : 10)	30°C	54%	4.6s	73%
4	Hari ke-4	Pagi (07 :10)	28°C	66%	2.8s	75%
		Sore (16 : 10)	31.4°C	46%	6.9s	71%
5	Hari ke-5	Pagi (07 :10)	30.4°C	63%	4.4s	74%
		Sore (16 : 10)	31.1°C	62%	4.8s	73%
6	Hari ke-6	Pagi (07 :10)	29°C	60%	3.9s	69%
		Sore (16 : 10)	28.4°C	62%	2.9s	73%
7	Hari ke-7	Pagi (07 :10)	28.6°C	68%	2.8s	74%
		Sore (16 : 10)	31°C	55%	4.9s	68%
Rata Rata					4.52s	73.2%

Dari tabel 17 di atas merupakan hasil pengujian sistem penyiraman otomatis pada alat yang dirancang. Pengujian dilakukan sebanyak 7 kali, dengan tujuan untuk menguji penyiraman terjadwal yang dirancang dalam periode hari yang berbeda dan juga melihat beberapa kondisi dari kelembaban tanah serta suhu udara yang ada pada tanaman cabai dan bagaimana durasi penyiraman yang dihasilkan, serta mengetahui pengaruh penyiraman yang dihasilkan terhadap kelembaban tanah dalam periode waktu yang telah ditentukan yaitu pagi hari menuju sore hari dan juga sore hari menuju pagi hari.

Hasil pengujian menunjukkan sistem penyiraman telah bekerja sesuai dengan jadwal yang ditentukan yaitu pagi hari pukul 07:10 dan sore hari pukul 16:10 selama 7 kali pengujian. Hal ini menunjukkan bahwa sistem telah berhasil dalam mengendalikan penyiraman terjadwal. Rata-rata durasi penyiraman yang dihasilkan dari sistem adalah sebesar 4.52 detik. Sehingga dapat disimpulkan bahwa sistem penyiraman otomatis yang dirancang sudah berfungsi dengan baik dalam mengendalikan durasi penyiraman sesuai dengan rancangan.

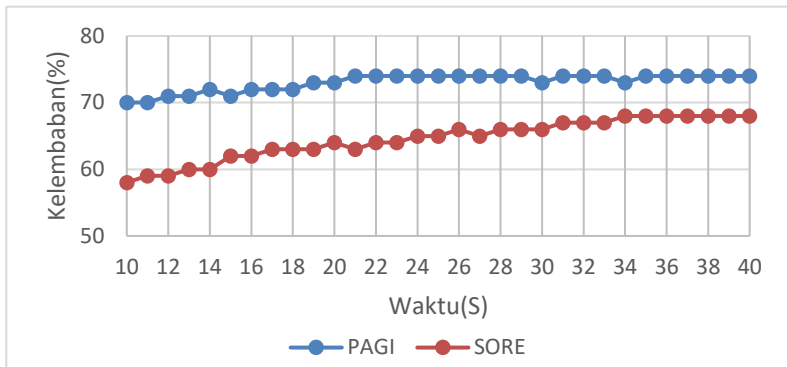


Gambar 25. Hasil Kelembaban tanah sistem penyiraman otomatis

Kemudian, diambil juga data hasil kelembaban tanah setelah sistem penyiraman aktif. Data hasil kelembaban diambil dengan memperhatikan perubahan kelembaban tanah pada sistem *monitoring* aplikasi *Blynk* maupun serial monitor hingga akhirnya mencapai nilai yang stabil. Gambar 25 di atas menunjukkan grafik kelembaban yang dihasilkan, dapat dilihat bahwa kelembaban tanah yang dihasilkan berada pada kisaran 68% hingga 78%, namun pada pengujian ke-2 terdapat peningkatan kelembaban yang besar hingga 80%, hal ini dikarenakan kondisi kelembaban awal yang kering dan durasi yang dihasilkan cukup lama sehingga air yang mengalir cukup banyak dan membuat kondisi tanah menjadi basah.

Secara keseluruhan, hasil kelembaban tanah setelah dilakukan penyiraman mendapatkan rata-rata 73.2% . Sehingga dari data pengujian hasil kelembaban ini dapat disimpulkan bahwa penerapan metode *fuzzy* dalam mengendalikan durasi penyiraman sudah berfungsi dengan cukup baik untuk mengendalikan kelembaban tanah yang sesuai pada tanaman cabai. metode *fuzzy* yang diterapkan mampu bekerja dengan baik dalam mengendalikan durasi penyiraman dengan penyesuaian dari parameter input, serta *rule base* yang telah dirancang mampu menghasilkan durasi penyiraman yang sesuai untuk menjaga kelembaban tanah

Perubahan kelembaban tanah setelah dilakukan penyiraman membutuhkan waktu sehingga mencapai nilai kelembaban yang stabil. Grafik perubahan kelembaban terhadap waktu dapat dilihat pada gambar 25 di bawah ini.



Gambar 26. Grafik perubahan kelembaban setelah penyiraman aktif

Sampel pengujian perubahan kelembaban yang dilakukan pada hari ke-7 menunjukkan bahwa selang waktu yang dibutuhkan tanah untuk mencapai kelembaban normal setelah dilakukan penyiraman adalah setelah sekitar 22 detik pada pengujian pagi hari dengan kelembaban terlihat stabil di angka 73%-74% dan sekitar 34 detik pada pengujian di sore hari dengan kelembaban stabil pada 67%-68%. Hasil ini menunjukkan bahwa nilai kelembaban awal tanah sebelum penyiraman dan juga selisih perubahan kelembaban tanah berpengaruh kepada waktu atau kecepatan perubahan kelembaban tanah itu sendiri.

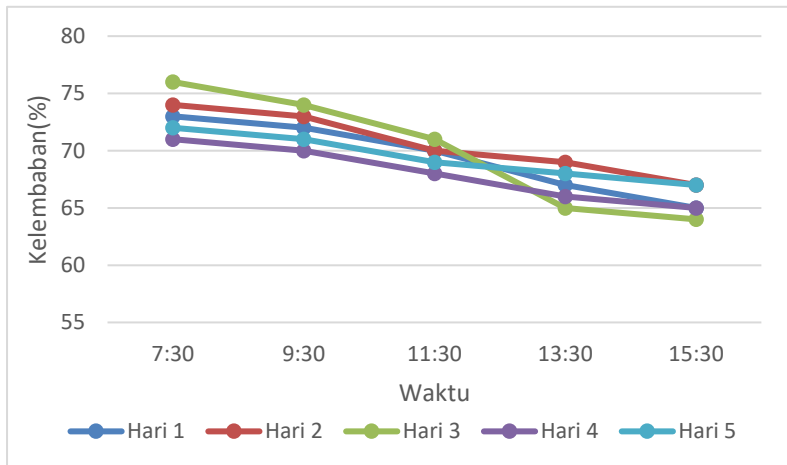
2. Hasil Pengujian Kondisi Kelembaban Tanah

Pada pengujian sistem penyiraman diambil pula data pengujian mengenai kondisi kelembaban tanah yang dihasilkan setelah penyiraman aktif hingga nantinya aktif kembali. Pengujian dilakukan selama 5 hari dengan melihat persentase kelembaban tanah secara aktual menggunakan *soil meter* setiap selang 2 jam setelah penyiraman aktif.

Tabel 18. Hasil pengujian pergerakan kelembaban tanah

Hari ke-	Penyiraman Pagi (07:10)			Kelembaban Tanah <i>Soil Meter</i> (%)					Penyiraman Sore (16:10)		
	Suhu Udara	Kelembaban Tanah	Durasi	07:30	09:30	11:30	13:30	15:30	Suhu Udara	Kelembaban Tanah	Durasi
1	30.6	60	4.7s	73	72	70	67	65	29.8	63	4.3s
2	29.3	64	4.1s	74	73	70	69	67	30	65	4.4s
3	30.4	65	4.4s	76	74	71	65	64	31.3	61	4.9s
4	29.9	60	4.1s	71	70	68	66	65	30.2	62	4.5s
5	28.2	63	3.3s	72	71	69	68	67	29.6	65	4.2s

Hasil pengujian yang ditampilkan pada tabel di atas menunjukkan bahwa kelembaban tanah mengalami perubahan dengan nilai kelembaban yang menurun jika dilihat setiap 2 jam sekali. Namun perubahan yang terjadi tidak terlalu signifikan jika dilihat dari pukul 07:30 sesaat setelah sistem penyiraman pertama aktif hingga pukul 15:30 sebelum sistem penyiraman kedua aktif. Pada pengujian yang dilakukan selama 5 hari berturut-turut, hasil pengujian menunjukkan kelembaban tanah yang dihasilkan masih berada pada range yang mencukupi untuk kebutuhan tanaman cabai yaitu sekitar 64% hingga 76%. Perubahan kelembaban tanah yang terjadi dapat dilihat pada gambar 27 di bawah ini



Gambar 27. Grafik perubahan kelembaban tanah

3. Hasil pengujian perbandingan sistem penyiraman

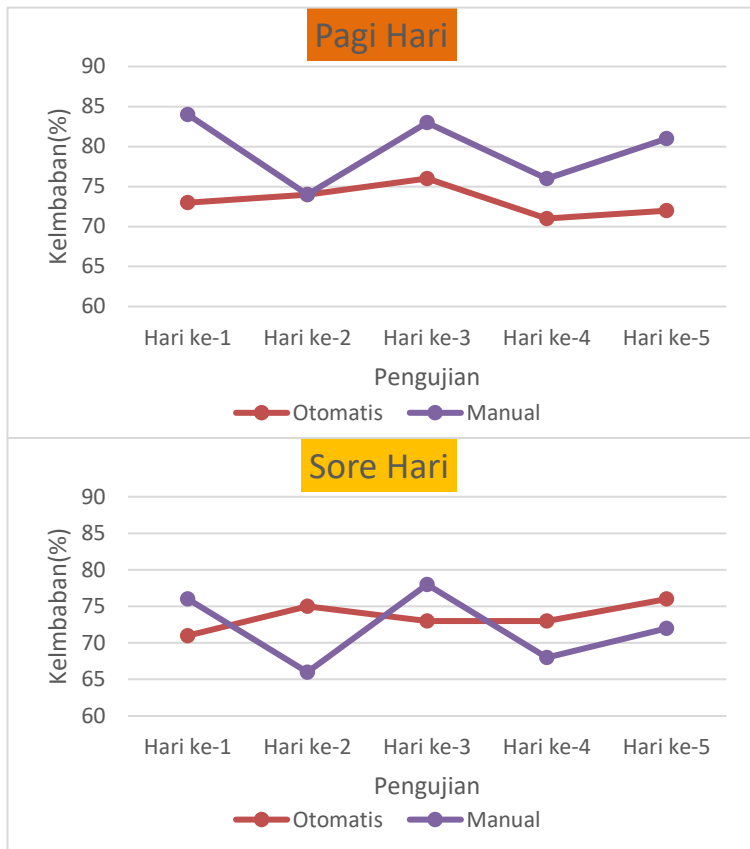
Pada saat yang bersamaan dengan pengujian pergerakan nilai kelembaban tanah, dilakukan juga pengujian perbandingan penyiraman otomatis dan penyiraman manual pada tanaman cabai. Pengujian dilakukan pada sistem penyiraman otomatis yang dibandingkan dengan penyiraman manual. Pengujian dilakukan dengan membandingkan hasil kelembaban tanah yang dihasilkan oleh penyiraman otomatis dengan penyiraman manual pada tanaman cabai yang ditanam di polybag yang berbeda dengan ukuran polybag dan volume tanah yang sama, penyiraman manual juga dilakukan pada waktu yang sama dengan sistem penyiraman otomatis yaitu pukul 07:10 dan pukul 16:10. Penyiraman manual dilakukan dengan menyiram tanaman cabai dengan volume air yang sekiranya cukup dengan melihat kondisi tanah tanaman cabai secara langsung tanpa memperhatikan tingkat kelembabannya. Kemudian setelah penyiraman otomatis aktif dan juga dilakukan penyiraman manual, dilakukan pengukuran hasil kelembaban tanah pada tanaman cabai menggunakan alat ukur *soil meter* setelah 2 menit untuk mendapatkan nilai kelembaban tanah saat air sudah menyerap sempurna kedalam tanah kedua polybag. Hasil pengujian perbandingan penyiraman otomatis dan penyiraman manual dapat dilihat pada tabel.

Tabel 19. Hasil pengujian perbandingan metode penyiraman

Hari ke-	Waktu	Penyiraman Otomatis				Penyiraman Manual	
		Suhu (°C)	Kelembaban (%)	Durasi (s)	Hasil Kelembaban (%)	Volume Air	Hasil Kelembaban (%)
1	07:10	30.6	60	4.7s	73	±450mL	84
	16:10	29.8	63	4.3s	71	±150mL	76
2	07:10	29.3	64	4.1s	74	±200mL	74
	16:10	30	65	4.4s	75	±100mL	66
3	07:10	30	65	4.4s	76	±350mL	83
	16:10	31.3	61	4.9s	73	±200mL	78
4	07:10	29.6	60	4.1s	71	±300mL	76
	16:10	30.2	62	4.5s	73	±150mL	68
5	07:10	28.2	63	3.3s	72	±250mL	81
	16:10	29.6	65	4.2s	76	±200mL	72
Rata-rata					73.4		75.6

Dari hasil pengujian di atas menunjukkan bahwa terdapat perbedaan yang cukup signifikan dari kelembaban tanah yang dihasilkan oleh sistem penyiraman otomatis yang dikembangkan dibandingkan dengan metode penyiraman manual oleh manusia. Jika ditampilkan kedalam bentuk grafik hasil pengujian dibagi

menjadi kondisi pagi hari dan sore hari seperti yang ditampilkan pada gambar 28 di bawah ini.



Gambar 28. Grafik perbandingan kondisi kelembaban tanah

Dari grafik di atas, pada pengujian yang dilakukan di pagi hari menunjukkan bahwa sistem penyiraman otomatis terlihat dapat mengendalikan kelembaban tanah selama 5 hari dengan cukup stabil di angka 71% - 76%. Namun pada metode penyiraman manual terdapat ketidakstabilan kelembaban tanah yang dihasilkan, hingga pada satu titik yaitu pada penyiraman hari ke-1 tingkat kelembaban terlalu tinggi yaitu 84%, hal ini terjadi karena volume air yang diberikan kepada tanah terkadang tidak sesuai dengan kondisi kelembaban awal tanah tersebut. Hal yang sama terjadi pada pengujian di sore hari, sistem penyiraman otomatis dapat

mengendalikan kelembaban tanah setiap sore nya selama 5 hari dengan kisaran nilai kelembaban 71% - 76%. Berbanding terbalik dengan penyiraman manual yang mengalami perubahan naik-turun yang kurang stabil. Secara keseluruhan pengujian sistem penyiraman otomatis mampu menghasilkan kelembaban tanah dengan rata-rata 73.4 % sedangkan penyiraman manual menghasilkan nilai rata-rata 75.6%. Dengan hasil tersebut dapat disimpulkan bahwa penyiraman otomatis lebih efektif dalam mengendalikan kelembaban tanah setiap harinya.

Kemudian dengan menggunakan data pengujian perbandingan sistem penyiraman otomatis dan penyiraman manual, dilakukan analisa dengan menggunakan standar deviasi. Hal ini dilakukan untuk mengetahui tingkat konsistensi penyebaran data kelembaban yang dihasilkan terhadap nilai rata-rata pada sistem penyiraman yang dihasilkan. Standar deviasi merupakan akar pangkat dua dari variansi data. Semakin tinggi nilai standar deviasi maka jarak persebaran data semakin jauh dengan nilai rata-rata, begitu juga sebaliknya. Rumus untuk menghitung standar deviasi dapat dilihat pada rumus(3) berikut ini.

$$\sigma = \sqrt{\frac{\sum(x_i - \tilde{x})^2}{n}} \quad (3)$$

Keterangan: σ : *Standar Deviasi*

x_i : Nilai x ke- i

\tilde{x} : Nilai rata-rata

n : Jumlah data

Dengan menggunakan rumus(3) di atas, diperoleh standar deviasi untuk penyiraman sistem otomatis sebagai berikut.

$$\begin{aligned} \sigma &= \sqrt{\frac{0.16 + 5.76 + 0.36 + 2.56 + 6.76 + 0.16 + 5.76 + 0.16 + 1.96 + 6.76}{10}} \\ \sigma &= \sqrt{\frac{30.4}{10}} \\ \sigma &= \sqrt{3.4} \\ \sigma &= 1.74 \end{aligned}$$

Sedangkan pada sistem penyiraman manual, diperoleh nilai standar deviasi sebagai berikut ini

$$\sigma = \sqrt{\frac{67.24 + 0.04 + 3.24 + 96.04 + 51.84 + 4.84 + 0.04 + 60.84 + 27.04 + 14.44}{10}}$$

$$\sigma = \sqrt{\frac{325.6}{10}}$$

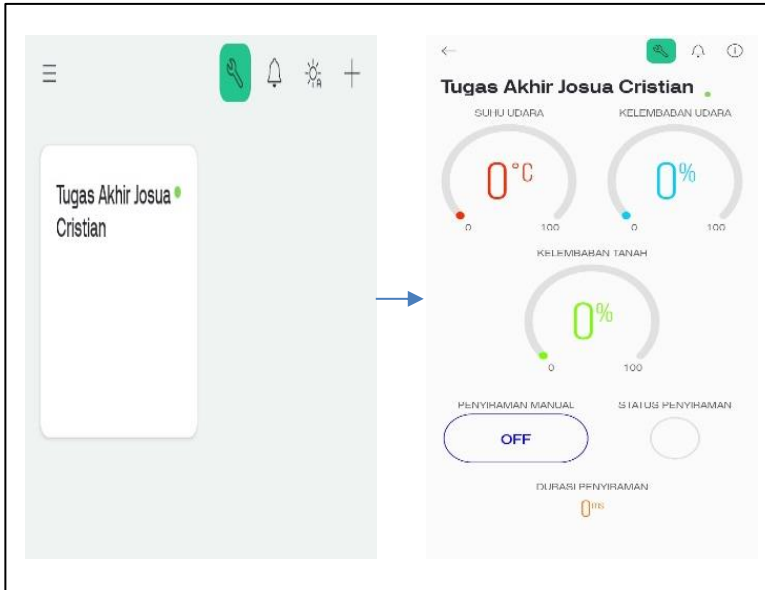
$$\sigma = \sqrt{32.56}$$

$$\sigma = 5.70$$

Nilai standar deviasi dari kedua metode yang digunakan diperoleh bahwa sistem penyiraman otomatis dari alat yang dirancang mendapatkan nilai standar deviasi 1.74 dan sistem penyiraman manual mendapatkan hasil standar deviasi sebesar 5.70, selisih standar deviasi yang diperoleh 3.96. Dari hasil standar deviasi ini dapat dilihat bahwa sistem penyiraman otomatis memiliki tingkat penyebaran data yang lebih baik daripada sistem penyiraman manual yang berarti sistem penyiraman otomatis lebih efektif dalam menjaga kestabilan hasil kelembaban tanah dibandingkan dengan penyiraman manual oleh manusia.

4.5. Hasil pengujian sistem IoT menggunakan aplikasi *Blynk*

Sistem pemantauan atau *monitoring* yang dirancang untuk memantau kondisi lingkungan dan juga sistem penyiraman pada tanaman cabai menggunakan aplikasi *Blynk* pada *smartphone*.


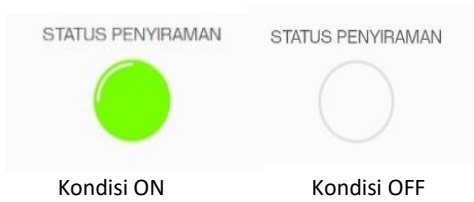
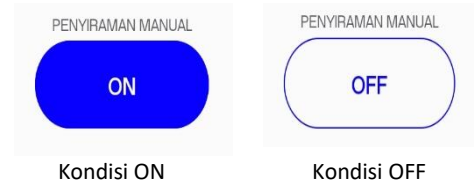



Gambar 29. Halaman awal sistem IoT pada aplikasi *Blynk*

Pada sistem *monitoring*, *value* dari masing-masing sensor akan ter-*update* setiap detik dan ditampilkan pada aplikasi *Blynk* sesuai dengan kondisi lingkungan. Ketika sistem penyiraman sedang berlangsung, indikator dari sistem penyiraman pada aplikasi akan aktif dan akan ditampilkan juga nilai durasi penyiraman yang terjadi. Sedangkan fitur penyiraman manual berfungsi untuk menghidupkan kran air secara manual, tetapi pada dua kondisi fitur ini tidak dapat digunakan yaitu, ketika suhu udara berada di bawah 29°C dan kelembaban tanah berada di atas 45%, serta pada saat jam menunjukkan waktu penyiraman terjadwal akan aktif yaitu pukul 07:10 dan 16:10.

Pengujian dilakukan pada fungsionalitas dari setiap fitur yang telah dirancang pada aplikasi *Blynk*. Hasil pengujian sistem IoT dan tampilan dari *Graphical User Interface* dari setiap fitur dapat dilihat pada tabel 20 di bawah ini. Dengan hasil pengujian yang menunjukkan bahwa setiap fitur yang ada pada aplikasi *Blynk* dapat berfungsi dengan baik.

Tabel 20. Hasil tampilan *Graphical User Interface* pada aplikasi *Blynk*

No	Fitur	Tampilan GUI
1	Pemantauan Suhu dan Kelembaban Udara	
2	Pemantauan Kelembaban Tanah	
3	Notifikasi penyiraman	
4.	Penyiraman manual	
5	Durasi Penyiraman	

Hasil pengujian dari tampilan setiap fitur pada aplikasi *blynk* menunjukkan bahwa setiap fitur dapat berfungsi dengan baik dan dapat ditampilkan pada aplikasi. Selanjutnya dilakukan pengujian terhadap hasil pembacaan dari nilai sensor suhu udara dan kelembaban tanah pada aplikasi *blynk* dengan membandingkan hasil pembacaan yang ditampilkan pada serial monitor. Pengujian dilakukan dengan melihat hasil pembacaan nilai sensor pada serial monitor dan aplikasi *blynk* pada waktu yang sama. Hasil pengujian yang dilakukan dapat dilihat pada tabel 21.

Tabel 21. Pengujian hasil pembacaan nilai sensor pada aplikasi *Blynk*

No.	Serial Monitor			Aplikasi <i>Blynk</i>		
	Suhu udara	Kelembaban udara	Kelembaban tanah	Suhu udara	Kelembaban udara	Kelembaban tanah
1	29.6°C	86.0%	13%	29.6°C	86.0%	13%
2	28.9°C	88.0%	45%	28.9°C	88.0%	45%
3	29.2°C	92.2%	34%	29.2°C	92.2%	34%
4	31.1°C	54.3%	56%	31.1°C	54.3%	56%
5	30.9°C	60.2%	46%	30.9°C	60.2%	46%

Pengujian hasil pembacaan nilai sensor pada aplikasi *Blynk* yang dibandingkan dengan pembacaan nilai sensor pada serial monitor arduino IDE menunjukkan hasil yang sangat baik. Dimana pada pengujian yang dilakukan sebanyak 5 kali, tidak ada perbedaan hasil pembacaan nilai sensor yang ditampilkan oleh aplikasi *Blynk*. Sehingga dapat disimpulkan bahwa aplikasi *Blynk* dapat berfungsi dengan baik dalam menampilkan nilai pembacaan sensor.

Kemudian dilakukan juga pengujian pada fungsi penyiraman manual, pengujian dilakukan untuk melihat apakah sistem penyiraman manual sudah berfungsi sesuai rancangan saat tombol ditekan. Hasil pengujian pada sistem penyiraman manual melalui aplikasi *blynk* dapat dilihat pada tabel di bawah ini.

Tabel 22. Pengujian Penyiraman manual

No	Waktu	Suhu Udara	Kelembaban Tanah	Kondisi Penyiraman Saat Tombol Ditekan
1	07 : 00	28.2°C	40%	Tidak Aktif
2	09 : 00	30.2°C	74%	Tidak Aktif
3	11 : 00	30.8°C	65%	Tidak Aktif
4	13 : 00	29.8°C	55%	Tidak Aktif
5	15 : 00	29.5°C	42%	Aktif
6	17 : 00	29/5°C	70%	Tidak Aktif

Dari hasil pengujian sistem penyiraman manual di atas, dapat dilihat bahwa ketika waktu menunjukkan pukul 07:00 sistem penyiraman tidak aktif meskipun kelembaban tanah bernilai 40%. Sistem penyiraman tidak aktif ketika kelembaban tanah bernilai 74% , 63% , dan juga 55% meskipun bukan dalam periode penyiraman terjadwal. Sistem penyiraman hanya aktif ketika pengujian pada pukul 15:00 dimana kondisi kelembaban tanah 42% dan suhu udara 29.5°C.

Hasil pengujian ini dapat disimpulkan bahwa sistem penyiraman manual sudah berfungsi sesuai dengan rancangan awal dengan setiap syarat atau kondisi aktifnya penyiraman manual.

Tabel 23. Pengujian Aksesibilitas Sistem IoT aplikasi *Blynk*

No	Percobaan	Jarak alat dari sumber internet						
		5m	10m	15m	20m	25m	30m	40m
1	Pemantauan Suhu Udara	✓	✓	✓	✓	✓	✗	✗
2	Pemantauan Kelembaban udara	✓	✓	✓	✓	✓	✗	✗
3	Pemantauan Kelembaban tanah	✓	✓	✓	✓	✓	✗	✗
4	Notifikasi penyiraman	✓	✓	✓	✓	✗	✗	✗
5	Penyiraman manual	✓	✓	✓	✓	✗	✗	✗

Dari tabel 21 di atas , tanda centang (✓) menandakan sistem IoT dapat berfungsi dengan baik, sedangkan tanda silang (✗) berarti sistem IoT mengalami masalah koneksi sehingga tidak dapat berfungsi dengan baik.

Sistem IoT berbasis aplikasi *Blynk* kemudian diuji dengan meletakkan alat hingga berada dalam jarak 40m dari sumber Internet. Hasil pengujian aksesibilitas ini menunjukkan bahwa fungsionalitas dari sistem IoT dapat bekerja dengan baik pada sistem pemantauan baik itu pemantauan suhu dan kelembaban udara, serta kelembaban tanah saat alat diletakkan pada jarak 25m dari sumber internet. Namun, sistem IoT menunjukkan adanya kegagalan pada fungsionalitas sistem penyiraman manual dan notifikasi pada jarak >20m.

Jarak aksesibilitas sistem IoT ini juga dipengaruhi oleh tipe dan *range* jangkauan dari *WiFi* yang digunakan. Pada penelitian ini, sumber internet yang digunakan adalah hotspot dari perangkat *smartphone*. Pengujian ini dilakukan pada aksesibilitas alat penyiraman dengan sistem IoT, sedangkan untuk aplikasi *Blynk* sistem IoT dapat diakses dari manapun dan kapanpun melalui aplikasi yang sudah teregistrasi pada program.

Bab 5. Kesimpulan dan Saran

5.1. Kesimpulan

Setelah perancangan dan juga pengujian terhadap hasil rancangan dari penelitian Sistem Penyiraman Pintar dan Pemantauan Menggunakan Sistem IoT Berbasis *Smartphone* pada Tanaman Cabai selesai, maka secara keseluruhan dapat diambil kesimpulan sebagai berikut:

1. Alat penyiraman yang dirancang mampu menghasilkan sistem penyiraman otomatis pada tanaman cabai dengan dua parameter input yaitu suhu udara dan juga kelembaban tanah. Sistem dapat beroperasi sesuai dengan jadwal yang ditentukan yaitu pagi hari dan sore hari, dan dengan menggunakan metode *fuzzy Mamdani* mampu menghasilkan durasi penyiraman yang berbeda.
2. Hasil pengujian metode *fuzzy Mamdani* pada ESP32 yang dibandingkan dengan menggunakan aplikasi Matlab menunjukkan *error* sebesar 0.28%. Pada sistem penyiraman metode *fuzzy* mampu menghasilkan durasi penyiraman dengan rata-rata 4.52s dan mampu menghasilkan kelembaban tanah setiap harinya dengan rata-rata 73.2% melalui pembacaan sensor YL-69. Dengan hasil pengujian ini, maka dapat diambil kesimpulan bahwa metode *fuzzy Mamdani* yang digunakan sudah cukup efektif dalam mengendalikan durasi penyiraman dan juga menjaga kelembaban tanah. Hasil pengujian juga menunjukkan bahwa sistem penyiraman otomatis yang dirancang mampu menjaga kestabilan kelembaban tanah yang dihasilkan lebih baik jika dibandingkan dengan metode penyiraman manual.
3. Sistem IoT yang telah dirancang mampu digunakan untuk memantau kondisi lingkungan seperti suhu dan kelembaban udara, dan juga kelembaban tanah melalui aplikasi *Blynk* pada *smartphone*. Sistem IoT juga dapat digunakan untuk melakukan penyiraman manual dan juga mengetahui indikasi dari kondisi sistem penyiraman serta durasi penyiraman yang terjadi. Sistem IoT dapat beroperasi dengan baik pada jarak maksimal sekitar 20m dari sumber internet menggunakan *Hotspot*.

5.2. Saran

Dalam pengerjaan Tugas Akhir ini terdapat beberapa kekurangan yang masih perlu diperhatikan, sehingga pada penelitian selanjutnya kekurangan tersebut dapat dikembangkan. Berdasarkan hal tersebut, penulis memberikan saran sebagai berikut:

1. Penelitian selanjutnya diharapkan dapat menyempurnakan sistem penyiraman dengan menambahkan parameter lainnya seperti pH tanah. Hal

ini dapat meningkatkan efektifitas dari sistem penyiraman pada tanamn cabai.

2. Penelitian selanjutnya dapat menggunakan sensor kelembaban tanah yang lebih akurat dan efisien terhadap perubahan kondisi tanah dan cuaca.
3. Sistem IoT yang digunakan pada penelitian ini masih menggunakan jaringan WiFi/*hotspot* dimana jarak aksesibilitasnya masih kurang baik. Peneliti selanjutnya diharapkan dapat menggunakan jaringan nirkabel dengan jangkauan yang lebih baik ataupun dapat menggunakan perangkat WiFi portabel.

Daftar Pustaka

- [1] Gozali, Muhammad Syafei, et al. "Multimeter Portabel Pengukur Nutrisi Tanah Pertanian untuk Kelompok Tani Maju Mandiri di Pulau Setokok." *Jurnal Pengabdian kepada Masyarakat Politeknik Negeri Batam* 5.1 (2023): 84-96.
- [2] Putri, Astriana Rahma, Suroso Suroso, and Nasron Nasron. "Perancangan Alat Penyiram Tanaman Otomatis pada Miniatur Greenhouse Berbasis IOT." *Prosiding SENIATI 5.2* (2019): 155-159.
- [3] Tullah, Rahmat, Sutarman Sutarman, and Agus Hendra Setyawan. "Sistem penyiraman tanaman otomatis berbasis mikrokontroler arduino uno pada toko tanaman hias yopi." *Jurnal Sisfotek Global* 9.1 (2019).
- [4] Hendri, Muhammad, Sucipto Sucipto, and Rachmat Wahid Saleh Insani. *SISTEM PENYIRAMAN OTOMATIS PADA TANAMAN CABAI RAWIT MENGGUNAKAN METODE FUZZY SUGENO*. Diss. Universitas Muhammadiyah Pontianak, 2022.
- [5] Sugandi, Budi, and Jeki Armentaria. "Sistem Penyiraman Tanaman Otomatis Menggunakan Metode Logika Fuzzy." *Journal of Applied Electrical Engineering* 5.1 (2021): 5-8.
- [6] Ariani, Fenty, et al. "Sistem *monitoring* suhu dan pencahayaan berbasis internet of thing (iot) untuk penetasan telur ayam." *Expert J. Manaj. Sist. Inf. dan Teknol* 10.2 . 2020.
- [7] Airlangga, Rahmadani Pramanindita, and Shandra Amarillis. "Pengaruh Cekaman Kering terhadap Respon Pertumbuhan Cabai Merah pada Fase Vegetatif." *Buletin Agrohorti* 11.2 (2023): 297-306.
- [8] Hermawan, Rian, and Triani Yulastari. "Hermawan, Rian MACHINE LEARNING MONITORING HAMA TANAMAN BIBIT ANGGREK BERBASIS IOT MENGGUNAKAN METODE KNN PADA PLATFROM BLYNK." *Jurnal Teknologi Informasi dan Komunikasi* 15.1 (2022): 29-45
- [9] Surahman, Ade, et al. "Sistem Pakan Ayam Otomatis Berbasis Internet Of Things." *Jurnal Teknologi Dan Sistem Tertanam* 2.1 (2021): 13-20
- [10] Surya Aulia, Rahman. *RANCANG BANGUN ALAT KONTROL SUHU AIR OTOMATIS PADA SISTEM TUMBUHAN AQUAPONIK BERBASIS NODE MCU DENGAN KONSEP INTERNET OF THINGS (IOT)*. Diss. Universitas Diponegoro, 2022.
- [11] Hidayat, Muh Adrian Juniarta, and Ahmad Zuli Amrullah. "Sistem Kontrol Dan *Monitoring* Tanaman Hidroponik Berbasis Internet Of Things (Iot) Menggunakan NodeMCU ESP32." *Jurnal Sainstekom* 12.1 (2022): 23-32.
- [12] Kinnasih, Ika Wahyu, and Dzulkifli Dzulkifli. "RANCANG BANGUN ALAT PENGONTROL SUHU DAN KELEMBAPAN PADA TEMPAT PENETASAN TELUR

- MENGGUNAKAN SENSOR DHT22 DAN MOTOR SWING BERBASIS IoT." *Inovasi Fisika Indonesia* 11.3 (2022): 57-72.
- [13] Satya, Trias Prima, et al. "Analisis Akurasi Sistem sensor DHT22 berbasis Arduino terhadap Thermohyrometer Standar." *Jurnal Fisika dan aplikasinya* 16.1 (2020): 40-45.
- [14] Suryanto, Suryanto. *RANCANG BANGUN SISTEM OTOMATIS DAN MONITORING JARAK JAUH ENGINEERING WORK STATION DI PLTU VIA WEB*. Diss. Universitas Muhammadiyah Ponorogo, 2020.
- [15] R. Setiawan, "Rancang Bangun Alat Pemberi Pakan Ikan Menggunakan Mikrokontroler," *J. ICTEE*, vol. 1, no. 1, pp. 51–54, 2020.
- [16] Ariyadi, Anggra Dinanda, Hani Zulfia Zahro, and Joseph Dedy Irawan. "PROTOTYPE PENERAPAN SMART BUILDING BERBASIS INTERNET OF THING." *JATI (Jurnal Mahasiswa Teknik Informatika)* 7.1 (2023): 797-804.
- [17] KHUSAENI, MUHAMMAD FAJAR. *PREDIKSI PERHITUNGAN JUMLAH PRODUKSI SABUK BONCENG DENGAN METODE FUZZY MAMDANI (Studi Kasus: Home Industry Yatra)*. Diss. UNIVERSITAS MUHAMMADIYAH MAGELANG, 2021.
- [18] M GILANG, DEFRIZA. *RANCANG BANGUN SISTEM PENGUKURAN pH, SUHU DAN KELEMBAPAN TANAH MENGGUNAKAN METODE FUZZY LOGIC BERBASIS IoT (INTERNET OF THINGS)*. 2023.
- [19] Laoli Kristiani Mei, Mei. *PROTOTYPE MONITORING SUHU DAN KELEMBAPAN PADA RUANGAN DENGAN MIKROKONTROLER ESP32 MENGGUNAKAN METODE FUZZY MAMDANI BERBASIS IOT (Internet of Things)*. Diss. Universitas Nasional, 2023.
- [20] Setiyawan, Dio. "Fuzzy Inference System Metode Tsukamoto untuk Penentuan Program Studi Fakultas Sains dan Teknologi di Universitas Muhammadiyah Kalimantan Timur." (2022).
- [21] Sitorus, Jimmi Hendrik P., and Reagan Surbakti Saragih. "Perancangan Pengontrol Lampu Rumah Miniatur Dengan Menggunakan Micro Controler Arduino Berbasis Android." *Jurnal Bisantara Informatika* 4.1 (2020): 11-11.
- [22] Basri, Muh Arga, Sofyan Sofyan, and Kurniawati Naim. "Rancang Bangun Electronic Load Control Generator Pembangkit Listrik Tenaga Mikro Hidro Berbasis Mikrokontroler dan IoT." *Seminar Nasional Teknik Elektro dan Informatika (SNTEI)*. 2021.
- [23] Juwariyah, Tatik, Luh Krisnawati, and Sri Sulasminingsih. "Perancangan sistem *monitoring* terpadu Smart Bins berbasis IoT menggunakan aplikasi *Blynk*." *Jurnal Informatika dan Rekayasa Elektronik* 3.2 (2020): 91-99.

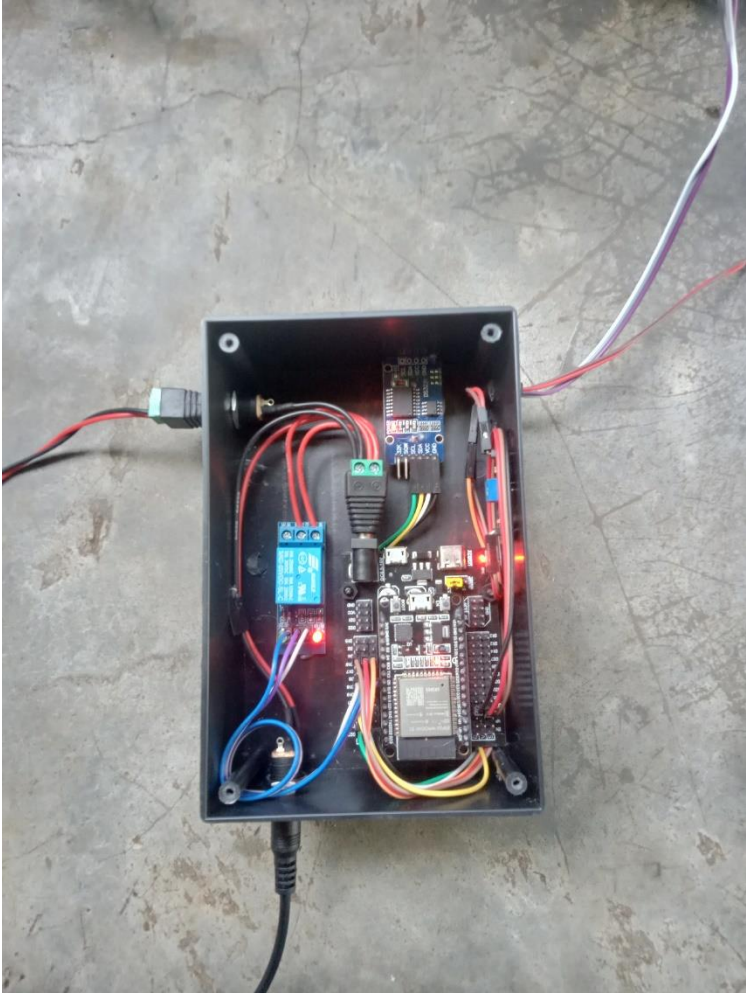
Biodata



Nama : Josua Cristian Tambunan
TTL : Batam / 22 Juni 2001
Agama : Kristen Protestan
Alamat : Tembesi Lestari No.18 RT006/RW003
Tembesi, Sagulung
Email : josuacristian99@gmail.com
Riwayat Pendidikan : - SMA/SMK : SMK Negeri 1 Batam
- SMP : SMP Negeri 37 Batam

Lampiran

Lampiran A (Rangkaian Elektrikal)



Lampiran B (Pemrograman utama ESP32)

```
TA_JOSUA | Arduino 1.8.13
File Edit Sketch Tools Help

//Blynk authentication code
#define BLYNK_TEMPLATE_ID "TMPLvBWOcFYA"
#define BLYNK_TEMPLATE_NAME "Tugas Akhir Josea Cristian"
#define BLYNK_AUTH_TOKEN "xeFX0NurrqPCW8s4YhllcFcjFOVWYa"

#include <WiFi.h> // Library ESP32
#include <BlynkSimpleEsp32.h> // Library Aplikasi Blynk
#include <DHT.h> // Library DHT22
#include <Dmwy.h> // Library tambahan esp32
#include "RTClib.h" // Date and time functions using a DS3231 RTC connected via I2C and Wire lib
#include <Fuzzy.h> // Library fuzzy

char auth[] = BLYNK_AUTH_TOKEN;
char ssid[] = "beheue"; // Nama WiFi yang digunakan
char pass[] = "joesu3300q"; // Password WiFi

#define BLYNK_PRINT Serial
#define DHTPIN 4 // Connect DHT sensor to GPIO 4 (D4) in ESP32
#define SOIL_MOISTURE_PIN 34 // Connect soil moisture sensor to GPIO 34 (ADC1_CH6) in ESP32
#define DHTTYPE DHT22 // Initialize DHT type
#define RELAY_PIN 5 // Connect Relay Pin to GPIO5 (D5)

RTC_DS3231 rtc;
BlynkTimer timer;
DHT dht(DHTPIN, DHTTYPE);

int output_fuzzy = 0;

int pinValue;

// First condition
const int OnHour1 = 7; //SET TIME TO ON SYSTEM (24 HOUR FORMAT)
const int OnMin1 = 10;
const int OnSec1 = 10;
// Second condition
const int OnHour2 = 16; //SET TIME TO ON SYSTEM (24 HOUR FORMAT)
const int OnMin2 = 10;
const int OnSec2 = 10;

void checkBlynkStatus() //called every 3 seconds by SimpleTimer
{
  bool isConnected = Blynk.connected();
  if (isConnected == false) {
    Serial.print("\nBlynk Not Connected ");
  }
  if (isConnected == true) {
    Serial.println("\nBlynk Connected");
  }
}

void sendSensor() //Mempirim data sensor ke aplikasi Blynk
{
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit
  int soilmoisture = analogRead(SOIL_MOISTURE_PIN); // Read from the soil moisture sensor
```

```

int soilmoisturepercentage = map(soilmoisture, 000, 4095, 100, 0); //make a moisture percentage

Serial.print("\n Temperature : ");
Serial.print(temperature);
Serial.print("\t Humidity : ");
Serial.print(humidity);
Serial.print("\n\n Soil : ");
Serial.print(soilmoisture);
Serial.print("\t Soil Percentage : ");
Serial.print(soilmoisturepercentage);
Serial.print("\n");

Blynk.virtualWrite(V0, soilmoisturepercentage); //mengirim data Kelembaban tanah ke aplikasi Blynk
Blynk.virtualWrite(V1, humidity); //mengirim data Humidity ke aplikasi Blynk
Blynk.virtualWrite(V2, temperature); //mengirim data Suhu udara ke aplikasi Blynk
//Blynk.virtualWrite(V2, input1); //mengirim data Suhu udara ke aplikasi Blynk
}

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

  Blynk.begin(auth, ssid, pass);

  pinMode(RELAY_PIN, OUTPUT);
  digitalWrite(RELAY_PIN, HIGH);

  dht.begin();
  rtc.begin();

  //checkBlynkStatus();
  timer.setInterval(2000L, checkBlynkStatus); // check if Blynk server is connected every 2 seconds
  timer.setInterval(5000L, sendSensor); // Setup a function to be called every second

  fuzzy_system();
}

void blynk()
{
  Blynk.run();
  timer.run();
}

void loop()
{
  blynk();

  DateTime now = rtc.now() + TimeSpan(0,0,0,39);
  waktu();

  //if (pinValue == 1)
  if (now.hour() == OnHour1 && now.minute() == OnMin1 && now.second() == OnSec1 || now.hour() == OnHour2 && now.minute() == OnMin2 && now.second() == OnSec2)
  {
    Serial.print("\n\n Sistem Penyiraman Aktif");
    fuzzy_sett();

    //menampilkanFuzzy();
    if (output_fuzzy != 0)
    {
      penyiraman();
    }
    Blynk.virtualWrite(V4, 0);
  }
}

```

Dina Saang

Lampiran C (Pemrograman Fuzzy)

```
TA_050UA - Fuzzy.ino | Arduino 1.8.13
File Edit Sketch Tools Help

TA_050UA 05/21/18 Fuzzy Panyaman

// Fuzzy
Fuzzy *fuzzy = new Fuzzy();

// FuzzyInputSuhuUdara
FuzzySet *Dingin = new FuzzySet(0, 8, 15, 22);
//FuzzySet *Sejuk = new FuzzySet(10, 15, 20, 20);
FuzzySet *Normal = new FuzzySet(15, 22, 22, 23);
FuzzySet *Hangat = new FuzzySet(22, 25, 26, 34);
FuzzySet *Panas = new FuzzySet(28, 34, 40, 40);

// FuzzyInputKelembaban
FuzzySet *Kering = new FuzzySet(0, 10, 30, 45);
FuzzySet *AgakLembab = new FuzzySet(35, 45, 45, 60);
FuzzySet *Lembab = new FuzzySet(50, 60, 60, 75);
FuzzySet *Basah = new FuzzySet(65, 75, 100, 100);

// FuzzyOutputDurasi
FuzzySet *TidakMenyiram = new FuzzySet(0, 0, 0, 0);
FuzzySet *Cepat = new FuzzySet(20, 25, 35, 40);
FuzzySet *Sedang = new FuzzySet(40, 60, 60, 80);
FuzzySet *Lama = new FuzzySet(80, 100, 120, 120);

void fuzzy_system()
{
  // FuzzyInput

  FuzzyInput *suhu_udara = new FuzzyInput(1);
  suhu_udara->addFuzzySet(Dingin);
  //suhu_udara->addFuzzySet(Sejuk);
  suhu_udara->addFuzzySet(Normal);
  suhu_udara->addFuzzySet(Hangat);
  suhu_udara->addFuzzySet(Panas);
  fuzzy->addFuzzyInput(suhu_udara);

  // FuzzyInput
  FuzzyInput *kelembaban_tanah = new FuzzyInput(2);
  kelembaban_tanah->addFuzzySet(Kering);
  kelembaban_tanah->addFuzzySet(AgakLembab);
  kelembaban_tanah->addFuzzySet(Lembab);
  kelembaban_tanah->addFuzzySet(Basah);
  fuzzy->addFuzzyInput(kelembaban_tanah);

  // FuzzyOutput
  FuzzyOutput *durasi = new FuzzyOutput(1);
  durasi->addFuzzySet(TidakMenyiram);
  durasi->addFuzzySet(Cepat);
  durasi->addFuzzySet(Sedang);
  durasi->addFuzzySet(Lama);
  fuzzy->addFuzzyOutput(durasi);

  // FuzzyRule
  FuzzyRuleAntecedent *ifsuhuDingin_KelembabanKering = new FuzzyRuleAntecedent();
  ifsuhuDingin_KelembabanKering->joinWithAND(Dingin, Kering);

  FuzzyRuleConsequent *thendurasi_cepat_1 = new FuzzyRuleConsequent();
  thendurasi_cepat_1->addOutput(Cepat);

  FuzzyRule *fuzzyRule1 = new FuzzyRule(1, ifsuhuDingin_KelembabanKering, thendurasi_cepat_1);
  fuzzy->addFuzzyRule(fuzzyRule1);

  //FuzzyRule2
  FuzzyRuleAntecedent *ifsuhuDingin_KelembabanAgakLembab = new FuzzyRuleAntecedent();
  ifsuhuDingin_KelembabanAgakLembab->joinWithAND(Dingin, AgakLembab);

  FuzzyRuleConsequent *thendurasi_cepat_2 = new FuzzyRuleConsequent();
  thendurasi_cepat_2->addOutput(Cepat);

  FuzzyRule *fuzzyRule2 = new FuzzyRule(2, ifsuhuDingin_KelembabanAgakLembab, thendurasi_cepat_2);
  fuzzy->addFuzzyRule(fuzzyRule2);

  // FuzzyRule3
  FuzzyRuleAntecedent *ifsuhuDingin_KelembabanLembab = new FuzzyRuleAntecedent();
  ifsuhuDingin_KelembabanLembab->joinWithAND(Dingin, Lembab);

  FuzzyRuleConsequent *thendurasi_cepat_3 = new FuzzyRuleConsequent();
  thendurasi_cepat_3->addOutput(Cepat);
```

```

FuzzyRule 'fuzzyRule3 = new FuzzyRule(3, ifSuhuDingin_KelembabanLembab, thenDurasi_cepat_3);
fuzzy->addFuzzyRule(fuzzyRule3);

// FuzzyRule4
FuzzyRuleAntecedent 'ifSuhuDingin_KelembabanBasah = new FuzzyRuleAntecedent();
ifSuhuDingin_KelembabanBasah->joinWithAND(Dingin, Basah);
FuzzyRuleConsequent 'thentidak_menyiram_1 = new FuzzyRuleConsequent();
thentidak_menyiram_1->addOutput(TidakMenyiram);

FuzzyRule 'fuzzyRule4 = new FuzzyRule(4, ifSuhuDingin_KelembabanBasah, thentidak_menyiram_1);
fuzzy->addFuzzyRule(fuzzyRule4);

// FuzzyRule5
FuzzyRuleAntecedent 'ifSuhuNormal_KelembabanKering = new FuzzyRuleAntecedent();
ifSuhuNormal_KelembabanKering->joinWithAND(Normal, Kering);
FuzzyRuleConsequent 'thendurasi_sedang_1 = new FuzzyRuleConsequent();
thendurasi_sedang_1->addOutput(Sedang);
FuzzyRule 'fuzzyRule5 = new FuzzyRule(5, ifSuhuNormal_KelembabanKering, thendurasi_sedang_1);
fuzzy->addFuzzyRule(fuzzyRule5);

//FuzzyRule6
FuzzyRuleAntecedent 'ifSuhuNormal_KelembabanAgakLembab = new FuzzyRuleAntecedent();
ifSuhuNormal_KelembabanAgakLembab->joinWithAND(Normal, AgakLembab);
FuzzyRuleConsequent 'thendurasi_cepat_4 = new FuzzyRuleConsequent();
thendurasi_cepat_4->addOutput(Cepat);

FuzzyRule 'fuzzyRule6 = new FuzzyRule(6, ifSuhuNormal_KelembabanAgakLembab, thendurasi_cepat_4);
fuzzy->addFuzzyRule(fuzzyRule6);

// FuzzyRule7
FuzzyRuleAntecedent 'ifSuhuNormal_KelembabanLembab = new FuzzyRuleAntecedent();
ifSuhuNormal_KelembabanLembab->joinWithAND(Normal, Lembab);
FuzzyRuleConsequent 'thendurasi_cepat_5 = new FuzzyRuleConsequent();
thendurasi_cepat_5->addOutput(Cepat);

FuzzyRule 'fuzzyRule7 = new FuzzyRule(7, ifSuhuNormal_KelembabanLembab, thendurasi_cepat_5);
fuzzy->addFuzzyRule(fuzzyRule7);

// fuzzyRule8
FuzzyRuleAntecedent 'ifSuhuNormal_KelembabanBasah = new FuzzyRuleAntecedent();
ifSuhuNormal_KelembabanBasah->joinWithAND(Normal, Basah);
FuzzyRuleConsequent 'thentidak_menyiram_2 = new FuzzyRuleConsequent();
thentidak_menyiram_2->addOutput(Cepat);
FuzzyRule 'fuzzyRule8 = new FuzzyRule(8, ifSuhuNormal_KelembabanBasah, thentidak_menyiram_2);
fuzzy->addFuzzyRule(fuzzyRule8);

// FuzzyRule9
FuzzyRuleAntecedent 'ifSuhuHangat_KelembabanKering = new FuzzyRuleAntecedent();
ifSuhuHangat_KelembabanKering->joinWithAND(Hangat, Kering);
FuzzyRuleConsequent 'thendurasi_lama_1 = new FuzzyRuleConsequent();
thendurasi_lama_1->addOutput(Lama);
FuzzyRule 'fuzzyRule9 = new FuzzyRule(9, ifSuhuHangat_KelembabanKering, thendurasi_lama_1);
fuzzy->addFuzzyRule(fuzzyRule9);

//FuzzyRule10
FuzzyRuleAntecedent 'ifSuhuHangat_KelembabanAgakLembab = new FuzzyRuleAntecedent();
ifSuhuHangat_KelembabanAgakLembab->joinWithAND(Hangat, AgakLembab);
FuzzyRuleConsequent 'thendurasi_sedang_2 = new FuzzyRuleConsequent();
thendurasi_sedang_2->addOutput(Sedang);
FuzzyRule 'fuzzyRule10 = new FuzzyRule(10, ifSuhuHangat_KelembabanAgakLembab, thendurasi_sedang_2);
fuzzy->addFuzzyRule(fuzzyRule10);

// FuzzyRule11
FuzzyRuleAntecedent 'ifSuhuHangat_KelembabanLembab = new FuzzyRuleAntecedent();
ifSuhuHangat_KelembabanLembab->joinWithAND(Hangat, Lembab);
FuzzyRuleConsequent 'thendurasi_cepat_6 = new FuzzyRuleConsequent();

```

```

thendurasi_cepat_6->addOutput(Cepat);

FuzzyRule *fuzzyRule11 = new FuzzyRule(11, ifSubuhHangat_KelembabanLembab, thendurasi_cepat_6);
fuzzy->addFuzzyRule(fuzzyRule11);

// FuzzyRule12
FuzzyRuleAntecedent *ifSubuhHangat_KelembabanBasah = new FuzzyRuleAntecedent();
ifSubuhHangat_KelembabanBasah->joinWithAND(Hangat, Basah);

FuzzyRuleConsequent *thentidak_menyiram_3 = new FuzzyRuleConsequent();
thentidak_menyiram_3->addOutput(TidakMenyiram);

FuzzyRule *fuzzyRule12 = new FuzzyRule(12, ifSubuhHangat_KelembabanBasah, thentidak_menyiram_3);
fuzzy->addFuzzyRule(fuzzyRule12);

// FuzzyRule13
FuzzyRuleAntecedent *ifSubuhPanas_KelembabanKering = new FuzzyRuleAntecedent();
ifSubuhPanas_KelembabanKering->joinWithAND(Panas, Kering);

FuzzyRuleConsequent *thendurasi_lama_2 = new FuzzyRuleConsequent();
thendurasi_lama_2->addOutput(Lama);

FuzzyRule *fuzzyRule13 = new FuzzyRule(13, ifSubuhPanas_KelembabanKering, thendurasi_lama_2);
fuzzy->addFuzzyRule(fuzzyRule13);

//FuzzyRule14
FuzzyRuleAntecedent *ifSubuhPanas_KelembabanApakLembab = new FuzzyRuleAntecedent();
ifSubuhPanas_KelembabanApakLembab->joinWithAND(Panas, ApakLembab);

FuzzyRuleConsequent *thendurasi_sedang_3 = new FuzzyRuleConsequent();
thendurasi_sedang_3->addOutput(Sedang);

FuzzyRule *fuzzyRule14 = new FuzzyRule(14, ifSubuhPanas_KelembabanApakLembab, thendurasi_sedang_3);
fuzzy->addFuzzyRule(fuzzyRule14);

// FuzzyRule15
FuzzyRuleAntecedent *ifSubuhPanas_KelembabanLembab = new FuzzyRuleAntecedent();
ifSubuhPanas_KelembabanLembab->joinWithAND(Panas, Lembab);

FuzzyRuleConsequent *thendurasi_sedang_4 = new FuzzyRuleConsequent();
thendurasi_sedang_4->addOutput(Sedang);

FuzzyRule *fuzzyRule15 = new FuzzyRule(15, ifSubuhPanas_KelembabanLembab, thendurasi_sedang_4);
fuzzy->addFuzzyRule(fuzzyRule15);

// FuzzyRule16
FuzzyRuleAntecedent *ifSubuhPanas_KelembabanBasah = new FuzzyRuleAntecedent();
ifSubuhPanas_KelembabanBasah->joinWithAND(Panas, Basah);

FuzzyRuleConsequent *thentidak_menyiram_4 = new FuzzyRuleConsequent();
thentidak_menyiram_4->addOutput(TidakMenyiram);

FuzzyRule *fuzzyRule16 = new FuzzyRule(16, ifSubuhPanas_KelembabanBasah, thentidak_menyiram_4);
fuzzy->addFuzzyRule(fuzzyRule16);
}

void fuzzy_sect()
{
// get variable inputs
float temperature = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit
int soilmoisture = analogRead(SOIL_MOISTURE_PIN); // Read from the soil moisture sensor
int soilmoisturepercentage = map(soilmoisture, 800, 4095, 100, 0); //make a moisture percentage
//float input1 = random(28, 33);

Serial.print("\n\t\t\t Temperature : ");
Serial.print(temperature);
//Serial.print(input1);
Serial.print("\n\t\t\t Soil Moisture : ");
Serial.print(soilmoisture);
Serial.print("\n\t\t\t Soil Moisture Percentage : ");
Serial.print(soilmoisturepercentage);

int Kelembaban = soilmoisturepercentage;
float suhu_udara = temperature;

fuzzy->setInput(1, suhu_udara);

```


Lampiran D (Pemrograman Penyiraman)

```
TA_JOSUA - Penyiraman | Arduino 1.8.13
File Edit Sketch Tools Help

TA_JOSUA DB3216 F4m Penyiraman

const int batassuhu = 30;
const int batasmisture = 40;

void penyiraman()
{
  int selangwaktu = (output_fuzzy * 100);
  digitalWrite(RELAY_PIN, LOW);
  Serial.print("\n\n Relay Aktif");
  Blynk.virtualWrite(V3, 1);
  Blynk.virtualWrite(V4, selangwaktu);
  delay(selangwaktu);
  digitalWrite(RELAY_PIN, HIGH);
  Serial.print("\n\n Relay Mati");
  Blynk.virtualWrite(V3, 0);
}

BLYNK_WRITE(V23)
{
  pinValue = param.asInt(); // assigning incoming value from pin V23 to a variable

  DateTime now = rtc.now() + TimeSpan(0,0,0,30);

  float temperature = dht.readTemperature();
  int soilmoisture = analogRead(SOIL_MOISTURE_PIN); // Read from the soil moisture sensor
  int soilmoisturepercentage = map(soilmoisture, 800, 4095, 100, 0);

  // process received value

  if(pinValue == 1 && temperature >= batassuhu && soilmoisturepercentage <= batasmisture && soilmoisturepercentage != 0)
  {
    if(now.hour() != OnHour1 && now.minute() != OnMin1 || now.hour() != OnHour2 && now.minute() != OnMin2)
    {
      Serial.print("\n Penyiraman Hidup");
      digitalWrite(RELAY_PIN, LOW);
      Blynk.virtualWrite(V3, 1);
      Blynk.virtualWrite(V4, 3000);
      delay(3000);
      Serial.print("\n Penyiraman Mati");
      digitalWrite(RELAY_PIN, HIGH);
      Blynk.virtualWrite(V3, 0);
    }
  }
}
```