

SKRIPSI_IOT_UP2

by Yusa Edu

Submission date: 29-Aug-2025 12:30PM (UTC+0100)

Submission ID: 2725088864

File name: SKRIPSI_IOT_UP2-2.docx (8.35M)

Word count: 24374

Character count: 160405

¹⁸
**IMPLEMENTASI INTERNET OF THINGS UNTUK
PENYIRAMAN DAN PENGKABUTAN OTOMATIS PADA
TANAMAN MENGGUNAKAN ALGORITMA NAÏVE BAYES**
²
(Studi Kasus di Avicenna Greenhouse)

SKRIPPSI

**Karya Tulis sebagai syarat memperoleh
Gelar Sarjana Komputer dari Fakultas Teknologi Informasi
Universitas Bale Bandung**

Disusun oleh:

**ADAM SETIADI
NPM.301210013**



**PROGRAM STRATA 1
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAR BALE BANDUNG
BANDUNG
2025**

LEMBAR PERSETUJUAN PEMBIMBING

¹⁸
**IMPLEMENTASI INTERNET OF THINGS UNTUK PENYIRAMAN DAN
PENGKABUTAN OTOMATIS PADA TANAMAN MENGGUNAKAN
ALGORITMA NAÏVE BAYES (Studi Kasus di Avicenna Greenhouse)**

Disusun Oleh:
ADAM SETIADI
NPM. 301210013

²
Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA

**FAKULTAS TENOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2025

Disetujui Oleh:

Pembimbing Utama

Pembimbing Pendamping

⁴
Yusuf Muhamam, S.Kom., M.Kom.
NIK. 04104820003

Yaya Suharya, S.Kom, M.T.
NIK. 01043170007

LEMBAR PERSETUJUAN PENGUJI

**18
IMPLEMENTASI INTERNET OF THINGS UNTUK PENYIRAMAN DAN
PENGKABUTAN OTOMATIS PADA TANAMAN MENGGUNAKAN
ALGORITMA NAÏVE BAYES (Studi Kasus di Avicenna Greenhouse)**

Disusun Oleh:
ADAM SETIADI
NPM. 301210013

4
Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TENOLOGI INFORMASI
UNIVERSITAS BALE BANDUNG**

Baleendah, Agustus 2025

Disetujui Oleh:

Pengaji 1

Pengaji 2

Sutiyono, S.T., M.Kom.
NIK. 01043180002

Mohammad Bayu Anggara, S.Kom., M.Kom.
NIK. 01043170007

LEMBAR PERSETUJUAN PROGRAM STUDI

**18
IMPLEMENTASI INTERNET OF THINGS UNTUK PENYIRAMAN DAN
PENGKABUTAN OTOMATIS PADA TANAMAN MENGGUNAKAN
ALGORITMA NAÏVE BAYES (Studi Kasus di Avicenna Greenhouse)**

Disusun Oleh:

ADAM SETIADI

NPM. 301210013

4
Telah diterima dan disetujui untuk memenuhi persyaratan mencapai gelar

SARJANA KOMPUTER

Pada

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS BALE BANDUNG

Baleendah, Agustus 2025

Disetujui Oleh:

Mengetahui,

Mengesahkan

Dekan

Ketua Program Studi

Yudi Herdiana, S.T., M.T.
NIK. 04104808008

Yusuf Muharam, S.Kom., M.Kom.
NIK. 04104820003

HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini, saya:

Nama : Adam Setiadi
NPM : 301210013
Jurusan : Teknik Informatika
Fakultas : Teknologi Informasi
Judul : ¹⁸ Implementasi *Internet Of Things* Untuk Penyiraman Dan Pengkabutan Otomatis Pada Tanaman Menggunakan Algoritma *Naive Bayes* (Studi Kasus di Avicenna Greenhouse)

⁹⁴ Dengan ini penulis menyatakan bahwa skripsi ini sepenuhnya adalah hasil dari penelitian, pemikiran, dan penyusunan yang dilakukan secara mandiri oleh penulis, baik dalam bentuk naskah laporan maupun pengembangan aplikasi yang termasuk dalam karya ini. Segala bentuk karya atau referensi yang digunakan dari pihak lain telah dilengkapi dengan penulisan sumber secara jelas dan tepat sesuai dengan aturan penulisan ilmiah yang berlaku.

ABSTRAK

¹¹³ Perkembangan teknologi *Internet of Things (IOT)* memberikan dampak yang signifikan di berbagai bidang, termasuk sektor pertanian. Salah satu implementasi yang menonjol adalah otomatisasi sistem penyiraman dan pengkabutan tanaman, khususnya di lingkungan *Greenhouse*. Kondisi ini menuntut pengelolaan lingkungan yang konsisten dan berkelanjutan agar pertumbuhan tanaman tetap optimal dan hasil pertanian yang dihasilkan berkualitas serta efisien.

²⁰ Penelitian ini memiliki bertujuan merancang dan menerapkan sistem otomatis untuk penyiraman dan pengkabutan tanaman cabai berbasis *IOT*, dengan algoritma *Naïve bayes* sebagai metode prediksi dalam pengambilan keputusan. Sistem menggunakan ¹⁰ sensor suhu udara, sensor kelembapan udara, dan sensor kelembapan tanah untuk memonitoring kondisi lingkungan secara langsung. Data dari sensor diolah oleh algoritma untuk menentukan apakah perlu dilakukan penyiraman atau pengkabutan, sesuai hasil ¹⁵³ klasifikasi kondisi. Sistem ini memiliki fitur tambahan seperti notifikasi melalui antarmuka web yang dapat memantau dan mengontrol sistem dari jarak yang jauh. Dengan klasifikasi cerdas berbasis data nyata, sistem dapat mengambil keputusan secara otomatis dan akurat tanpa campur tangan manual. Penerapan sistem ini diharapkan bisa menekan pemborosan air, mengurangi kerja manual, serta menjaga kestabilan iklim mikro di *Greenhouse* guna menunjang pertumbuhan tanaman secara optimal dan berkelanjutan.

Hasil dari pengujian menunjukkan bahwa sistem bisa berjalan secara otomatis, cepat merespons perubahan kondisi lingkungan, dan dikontrol dengan mudah melalui aplikasi berbasis web. Penerapan sistem ini di Avicenna *Greenhouse* menunjukkan adanya peningkatan efisiensi penggunaan air, pengurangan intervensi manual, serta perbaikan stabilitas suhu dan kelembapan di dalam *Greenhouse*. Sistem yang dibangun tidak hanya meningkatkan efektivitas penyiraman dan pengkabutan tanaman, tetapi juga mendukung praktik pertanian presisi yang hemat sumber daya dan dapat diterapkan oleh petani di *Greenhouse*.

Kata Kunci: *Internet of Things, Naïve bayes, Pengkabutan, Penyiraman.*

**87
ABSTRACT**

The development of Internet of Things (IOT) technology has a significant impact in various fields, including the agricultural sector. One of the prominent implementations is the automation of plant watering and fogging systems, especially in the Greenhouse environment. This condition requires consistent and sustainable environmental management so that plant growth remains optimal and agricultural products produced are quality and efficient.

¹⁴² *This study aims to design and implement an automated system for watering and hauling chili plants based on IOT, with Naïve bayes algorithm as a predictive method in decision making. The system uses air temperature sensors, air humidity sensors, and soil moisture sensors to monitor environmental conditions directly and continuously. The data from the sensors is processed by an algorithm to determine whether watering or fogging is necessary, according to the results of the classification of conditions. The system has additional features such as notifications through a web interface that can monitor and control the system from a distance. With intelligent classification based on real data, the system can make decisions automatically and accurately without manual intervention. The implementation of this system is expected to reduce water waste, reduce manual labor, and maintain a stable microclimate in the Greenhouse to support optimal and sustainable plant growth.*

¹⁷⁴ *The test results show that the system can run automatically, quickly respond to changing environmental conditions, and be easily controlled through a web-based application. The application of this system in Avicenna Greenhouse shows an increase in water use efficiency, reduction of manual intervention, and improvement of temperature and humidity stability in the Greenhouse. The built system not only increases the effectiveness of watering and Hilling crops, but also supports precision farming practices that are resource-efficient and can be implemented by farmers in greenhouses.*

Keywords: Internet of Things, Misting, Naïve bayes, Watering.

KATA PENGANTAR

Segala pujian penulis sampaikan kepada Allah SWT atas anugerah, kasih sayang, dan bimbingan-Nya. Penulis bisa menyelesaikan laporan skripsi yang berjudul “¹⁸Implementasi Internet of Things untuk Penyiraman dan Pengkabutan Otomatis pada Tanaman Menggunakan Algoritma Naive Bayes (Studi Kasus di Avicenna Greenhouse)”. Shalawat dan salam selalu disampaikan kepada Nabi Muhammad, misalnya, orang -orang dengan ajarannya adalah panduan dalam kehidupan sehari -hari.¹⁶⁶

Dengan penuh rasa hormat, penulis ingin menyampaikan terima kasih yang mendalam kepada kedua orang tua tercinta atas doa, kasih sayang, serta dukungan yang tiada henti. Ucapan terima kasih juga penulis tujuhan kepada pihak-pihak yang telah memberikan bantuan dan bimbingan selama proses penyusunan skripsi ini, di antaranya:

1. Bapak Yudi Herdiana, S.T., M.T., selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.¹⁵
2. Bapak Yusuf Muhamram, S.Kom., M.Kom., selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung Sekaligus dosen pembimbing utama.¹⁰⁵
3. Bapak Yaya Suharya, S.Kom, M.Kom., selaku pembimbing pendamping.¹⁹⁴
4. Segenap Bapak/Ibu Dosen di Fakultas Teknologi Informasi.
5. Kepada seluruh pihak yang sudah support kepada penulis untuk menyelesaikan usulan penelitian ini.⁴⁷

Penulis menyadari bahwa laporan skripsi ini masih memiliki banyak kekurangan. Oleh karena itu, saran dan kritik yang bersifat membangun sangat penulis harapkan demi penyempurnaan di masa mendatang. Semoga laporan ini dapat diterima serta memberikan manfaat bagi berbagai pihak.

⁶⁹
Bandung, Mei 2025

Penulis

DAFTAR ISI

ABSTRAK	v
ABSTRACT	19
KATA PENGANTAR	vi
DAFTAR ISI.....	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian	5
1.5 Metodologi Penelitian.....	5
1.6 Sistematika Penulisan	6
BAB II TINJAUAN PUSTAKA.....	8
2.1 Landasan Teori.....	8
2.2 Dasar Teori.....	13
2.2.1 97 <i>Internet of Things</i>	13
2.2.2 Penyiraman Tanaman.....	14
2.2.3 Pengkabutan Tanaman	15
2.2.4 <i>Machine Learning</i>	17
2.2.5 Algoritma <i>Naïve bayes</i>	18
2.2.6 <i>ESP32</i>	19
2.2.7 <i>Soil moisture Sensor</i>	21
2.2.8 <i>DHT11</i>	22
2.2.9 <i>Nozzle Sprayer</i>	23
2.2.10 <i>Relay</i>	24
2.2.11 <i>Unified Modeling language</i>	25
2.2.12 <i>Website</i>	28
2.2.13 <i>Hosting</i>	29

2.2.14	Metode <i>Agile</i>	29
2.2.15	<i>GitHub</i>	31
2.2.16	<i>Arduino</i>	32
2.2.17	<i>Arduino IDE</i>	33
2.2.18	<i>MQTT</i>	34
2.2.19	<i>MySQL</i>	35
2.2.20	<i>Figma</i>	36
2.2.21	<i>Python</i>	37
2.2.22	<i>Flask</i>	39
2.2.23	<i>Draw.io</i>	40
2.2.24	<i>Visual Studio Code</i>	41
2.2.25	<i>Fritzing</i>	41
2.2.26	<i>HTML</i>	42
2.2.27	<i>CSS</i>	43
2.2.28	<i>JavaScript</i>	44
2.2.29	<i>HiveMQ</i>	46
2.2.30	<i>WhatsApp-web.js</i>	46
2.2.31	<i>Black Box Testing</i>	47
2.2.32	<i>Cross Validation</i>	47
2.2.33	<i>Accuracy Score</i>	48
2.2.34	<i>K-Fold Cross Validation</i>	48
BAB III METODOLOGI PENELITIAN		49
3.1	<i>Kerangka Pikir</i>	49
3.2	Deskripsi	50
3.2.1	Identifikasi Masalah	50
3.2.2	Pengumpulan Data	50
3.2.3	Perencanaan.....	50
3.2.4	Perancangan	53
3.2.5	Pembangunan.....	54
3.2.6	Pengujian.....	55
3.2.7	Penerapan	55
3.2.8	Peluncuran.....	56
3.2.9	Pembuatan Laporan.....	56
BAB IV ANALISIS DAN PERANCANGAN		57

4.1	Analisis.....	57
4.1.1	Analisis Masalah	57
4.1.2	Analisis <i>Hardware</i>	57
4.1.3	Analisis <i>Software</i>	58
4.1.4	Analisis Pengguna.....	59
4.1.5	User Interface.....	60
4.1.6	Fitur - fitur.....	60
4.1.7	Analisis Data.....	61
4.1.8	Analisis Biaya.....	62
4.2	Perancangan	64
4.2.1	Blok <i>Diagram</i> Alat dan Sistem	64
4.2.2	Diagram Alur Data.....	66
4.2.3	Denah <i>Greenhouse</i>.....	68
4.2.4	³⁵ <i>Unified Modeling Language (UML)</i>	70
4.2.5	Struktur Tabel.....	80
4.2.6	Perancangan <i>Wiring Diagram</i>.....	82
4.2.7	⁴⁶ Desain <i>Antarmuka</i>.....	86
BAB V IMPLEMENTASI DAN PENGUJIAN		91
5.1	Implementasi	91
5.1.1	Listing Program.....	91
5.1.2	Implementasi Sistem	102
5.1.3	Spesifikasi Sistem	102
5.1.4	⁸² Instalasi Sistem	103
5.1.5	Menjalankan Sistem	105
5.2	Pengujian.....	110
⁴⁵ BAB VI KESIMPULAN.....		118
6.1	Kesimpulan	118
6.2	Saran.....	118
DAFTAR PUSTAKA		120
LAMPIRAN.....		126

DAFTAR TABEL

Tabel 2. 1 Acuan Jurnal Penelitian	8
Tabel 2. 2 Pengujian Kelembapan Tanah	14
Tabel 2. 3 Pengujian DHT11.....	15
⁶⁰ Tabel 2. 4 <i>Use Case Diagram</i>	25
Tabel 2. 5 <i>Activity Diagram</i>	26
Tabel 2. 6 <i>Class Diagram</i>	27
Tabel 3. 1 Perangkat Keras	52
Tabel 3. 2 Perangkat <i>Internet of Things</i>	52
Tabel 3. 3 Perangkat Lunak	52
² Tabel 4. 1 Analisis Biaya	63
¹⁸⁶ Tabel 4. 2 Tabel diagram konsep	65
Tabel 4. 3 Deskripsi Diagram Alur Alat dan Sistem	66
Tabel 4. 4 Deskripsi Denah <i>Greenhouse</i>	69
Tabel 4. 5 Deskripsi Aktor	71
⁷⁷ Tabel 4. 6 Tabel data sensor.....	80
Tabel 4. 7 Tabel jadwal penyiraman.....	81
Tabel 4. 8 Tabel riwayat aksi	81
Tabel 4. 9 tabel nomor_hp	81
Tabel 4. 10 Deskripsi <i>Wiring Diagram</i>	85
⁵ Tabel 5. 1 Spesifikasi perangkat keras.....	103
Tabel 5. 2 Spesifikasi perangkat lunak	103
Tabel 5. 3 Pengujian alat mikrokontroller.....	110
Tabel 5. 4 Pengujian <i>Dashboard Monitoring</i>	111
Tabel 5. 5 Hasil Pengujian <i>Cross Validation</i> Penyiraman.....	115
Tabel 5. 6 Hasil Pengujian <i>Cross Validation</i> Pengkabutan	115
³ Tabel 5. 7 Pengujian Sensor Kelembapan Tanah.....	115
Tabel 5. 8 Pengujian Sensor Kelembapan Suhu dan Udara.....	116

DAFTAR GAMBAR

Gambar 2. 1 Konsep <i>Internet of Things</i>	13
Gambar 2. 2 Skema <i>Artificial Intelligence</i> dan <i>Machine Learning</i>	17
Gambar 2. 3 <i>ESP32</i>	20
Gambar 2. 4 <i>Soil Mastuire Sensor</i>	21
Gambar 2. 5 <i>DHT11</i>	22
Gambar 2. 6 <i>Nozzle Sprayer</i>	23
Gambar 2. 7 <i>Relay</i>	24
Gambar 2. 8 Arsitektur <i>Website</i>	28
Gambar 2. 9 Metode <i>Agile</i>	30
Gambar 2. 10 <i>GitHub</i>	31
Gambar 2. 11 Bahasa Pemrograman <i>Arduino</i>	32
Gambar 2. 12 <i>Arduino IDE</i>	34
Gambar 2. 13 Arsitektur <i>MQTT</i>	34
Gambar 2. 14 Arsitektur <i>MySQL</i>	35
Gambar 2. 15 Tampilan <i>Figma</i>	36
Gambar 2. 16 Arsitektur <i>Python</i>	38
Gambar 2. 17 Arsitektur <i>Flask</i>	39
Gambar 2. 18 <i>Draw.io</i>	40
Gambar 2. 19 <i>Visual Studio Code</i>	41
Gambar 2. 20 <i>Fritzing</i>	42
Gambar 2. 21 Arsitektur <i>HTML</i>	43
Gambar 2. 22 Arsitektur <i>CSS</i>	44
Gambar 2. 23 Arsitektur <i>Javascript</i>	45
Gambar 2. 24 Tampilan <i>HiveMQ</i>	46
Gambar 2. 25 <i>WhatsApp-web.js</i>	47
Gambar 3. 1 Kerangka Pikir.....	49
Gambar 4. 1 Blok diagram alat dan sistem	64
Gambar 4. 2 Diagram alur alat dan sistem.....	66
Gambar 4. 3 Denah <i>Greenhouse</i>	69
Gambar 4. 4 <i>Use Case Diagram</i>	71

Gambar 4. 5 <i>Activity Login</i>	72
Gambar 4. 6 <i>Activity Dashboard</i>	73
Gambar 4. 7 <i>Activity Dashboard</i> Data Sensor	74
Gambar 4. 8 <i>Activity WhatsApp</i>	75
Gambar 4. 9 <i>Activity</i> Kontrol Alat	76
Gambar 4. 10 <i>Activity Logout</i>	78
Gambar 4. 11 <i>Class Diagram</i>	79
Gambar 4. 12 Skema Rangkaian Kelembapan Tanah	82
Gambar 4. 13 Implementasi <i>Wiring</i> Sensor Kelembapan Tanah	82
Gambar 4. 14 Skema Rangkaian Kelembapan Udara dan Suhu	83
Gambar 4. 15 Implementasi <i>Wiring</i> Sensor Kelembapan Udara dan Suhu	83
Gambar 4. 16 Skema Rangkaian Keseluruhan Sensor	84
Gambar 4. 17 <i>Wiring Diagram</i>	85
Gambar 4. 18 <i>User Interface Login</i>	86
Gambar 4. 19 <i>User Interface Dashboard</i>	87
Gambar 4. 20 <i>User Interface</i> Data Sensor	88
Gambar 4. 21 <i>User Interface</i> Kontrol Alat	89
Gambar 4. 22 <i>User Interface WhatsApp</i>	90
Gambar 5. 1 Halaman <i>Login</i>	105
Gambar 5. 2 Halaman <i>Dashboard</i>	105
Gambar 5. 3 Gambar Halaman Data Sensor	106
Gambar 5. 4 Gambar Halaman Kontrol Alat	107
Gambar 5. 5 Halaman <i>WhatsApp</i>	108
Gambar 5. 6 Notifikasi <i>WhatsApp</i>	109

108
DAFTAR LAMPIRAN

Lampiran 1 Hasil Wawancara.....	126
Lampiran 2 Dokumentasi Wawancara.....	128
Lampiran 3 <i>TOR</i>	129
Lampiran 4 Instalasi Alat Di <i>Greenhouse</i>	130
Lampiran 5 Listing Program	134

PENDAHULUAN**1.1 Latar Belakang**

Pada jaman digital saat ini, teknologi *Internet of Things (IOT)* sudah banyak diterapkan ²⁰³ dalam berbagai bidang, termasuk pertanian. Salah satu implementasi *IOT* yang semakin berkembang adalah sistem penyiraman dan pengkabutan otomatis yang memiliki tujuan untuk meningkatkan efisiensi penggunaan air. Dengan memanfaatkan sensor dan algoritma *Machine Learning*, sistem ini mampu menyesuaikan kondisi penyiraman dan pengkabutan secara *real-time*. Serta menerapkan algoritma *Naïve Bayes* ke dalam sistem otomatisasi penyiraman tanaman dengan menggunakan data sensor dapat secara signifikan meningkatkan efisiensi dan efektivitas dalam pengelolaan penggunaan air (Alamsyah et al., 2024).

Penelitian ini dilaksanakan di Avicenna *Greenhouse*, yang berlokasi di Kp. Padarek Rt.03 Rw.02, ¹⁷¹ Desa Drawati, Kecamatan Paseh, Kabupaten Bandung, Provinsi Jawa Barat. Avicenna *Greenhouse* merupakan sebuah rumah kaca yang difokuskan pada budidaya tanaman cabai dan telah berdiri sejak tahun 2024. Pengelolaan tanaman cabai di Avicenna *Greenhouse* saat ini, sistem penyiraman harus menyambungkan kabel pompa secara manual agar air dapat mengalir ke tanaman. Penyiraman tanaman dilakukan satu hingga maksimal dua kali dalam sehari, biasanya pada pukul 7-8 pagi dan 4-5 sore. Budidaya tanaman cabai dilakukan dalam sekitar 165 *polybag* yang disusun di atas lahan berukuran kurang lebih 10 x 12 meter. Masa panen tanaman cabai di Avicenna *Greenhouse* memerlukan waktu sekitar empat bulan sejak masa tanam.

Meskipun telah menerapkan sistem irigasi otomatis sederhana, proses penyiraman di Avicenna *Greenhouse* masih belum sepenuhnya praktis karena pengguna tetap harus menghubungkan kabel pompa secara manual setiap kali ingin menyiram tanaman. Hal ini membuat pengelolaan penyiraman menjadi kurang efisien dan masih bergantung pada kehadiran manusia. Selain itu, Avicenna *Greenhouse* belum memiliki sistem pengkabutan air untuk menjaga

suhu dan kelembapan udara serta di dalam ruangan. Cara kerja yang masih mengandalkan aktivitas manual ini berisiko menyebabkan ketidakpastian dalam perawatan tanaman dan berpotensi mempengaruhi hasil panen. Berdasarkan kondisi tersebut, diperlukan pengembangan sistem otomatis dengan *Internet of Things* yang dapat mengatur penyiraman dan pengkabutan tanaman secara otomatis, agar pengelolaan tanaman di Avicenna *Greenhouse* menjadi lebih efektif dan efisien.

Beberapa penelitian terdahulu yang penelitiannya memiliki pembahasan yang sama ² sudah dilakukan oleh peneliti sebelumnya, seperti yang sudah dilakukan oleh (Alamsyah et al., 2024), Mengungkapkan bahwa di Indonesia, masih banyak petani yang menggunakan cara manual sehingga proses penyiraman belum efisien. Penelitian Oleh (M. Iqbal Hasani & Sri Wulandari, 2023), Mengungkapkan bahwa penyiraman pada tanaman biasanya dilakukan dengan cara manual oleh pekerja memakai peralatan dasar yaitu gayung, selang, dan ember yang membutuhkan waktu serta tenaga yang cukup besar. Penelitian Oleh (Muhammad Rusdi et al., 2023), mengungkapkan bahwa permasalahan mitra yaitu masih menggunakan cara konvensional dalam proses pemeliharaan tanaman. Proses penyiraman yang berlebihan dapat menyebabkan adanya pembusukan pada tanaman anggrek dan tidak adanya teknologi untuk memantau kelembapan dan suhu pada area budidaya. Permasalahan ini memperlihatkan dibutuhkannya solusi teknologi yang mampu meningkatkan efisiensi dan efektivitas dalam kegiatan budidaya tanaman. Setelah dilakukan perbandingan dengan penelitian-penelitian sebelumnya, penelitian yang dilakukan penulis memiliki beberapa kelebihan dan kebaruan. Sistem yang dikembangkan tidak hanya mengotomatisasi proses penyiraman dan pengkabutan, tetapi juga menggunakan algoritma *Naïve bayes* sebagai dasar pengambilan data sensor untuk melakukan prediksi berdasarkan sensor suhu, kelembapan udara, dan kelembapan tanah secara *real-time*. Selain itu, sistem ini dikembangkan menggunakan *MQTT* untuk komunikasi data antar perangkat, serta *framework Flask* untuk pengembangan aplikasi dan juga mengintegrasikan dengan *WhatsApp* untuk mengirimkan notifikasi kepada pengguna. Sistem ini dibangun dengan metode pengembangan *Agile*, dimana metode ini memiliki penyesuaian dinamis dalam setiap tahap pengembangan dan

jika terdapat perubahan pada tahapan akan fleksibel tidak harus mengulang ke tahapan awal. Algoritma *Naive Bayes* dalam penelitian ini karena dianggap relevan untuk proses klasifikasi data yang diperoleh dari sensor. Penerapannya dapat mendukung pengambilan keputusan, khususnya dalam menentukan kapan penyiraman dan pengkabutan perlu dilakukan. Oleh karena itu, penelitian ini berfokus pada penerapan *Internet of Things* yang mampu mengotomatisasi proses penyiraman dan pengkabutan tanaman di dalam *Greenhouse*.¹⁹⁹

Solusi yang ditawarkan dalam penelitian ini adalah mengimplementasikan *Internet of Things* yang dapat mengotomatiskan proses penyiraman dan pengkabutan di Avicenna *Greenhouse* melalui pengembangan aplikasi dengan penerapan algoritma *Naïve bayes* sebagai prediksi suhu ruangan, kelembapan tanah pada tanaman, kelembapan udara pada ruangan. Sistem ini dikembangkan menggunakan metode *Agile* yang terdiri dari perencanaan, perancangan, pembangunan, pengujian, penerapan, peluncuran. Dalam proses perancangannya, aplikasi ini menggunakan sensor untuk suhu, kelembapan udara dan kelembapan tanah.⁹ Data pada sensor tersebut dapat dilakukan prediksi untuk penyiraman dan pengkabutan pada tanaman supaya efektif menggunakan algoritma *Naïve bayes*. Aplikasi dikembangkan menggunakan bahasa pemrograman *Arduino*, *Flask Python*. Hasil pada penelitian yang dibangun berupa sebuah aplikasi web yang menyediakan *monitoring* suhu ruangan, kelembapan udara pada ruangan, kelembapan tanah pada pot tanaman, mengendalikan penyiraman dan pengkabutan secara otomatis. Manfaat yang diharapkan adalah peningkatan efisiensi penggunaan air, stabilitas kondisi lingkungan dalam *Greenhouse*, serta peningkatan produktivitas dan kualitas pertumbuhan tanaman. Dengan demikian penelitian ini di harapkan dapat memberikan kemudahan untuk pemilik dalam melakukan penyiraman dan pengkabutan otomatis di *Greenhouse*.³⁵ Kesimpulannya, penelitian ini memiliki tujuan untuk merancang dan membangun aplikasi berbasis *Internet of Things* yang dapat membantu pemilik atau petani dalam melakukan penyiraman dan pengkabutan secara otomatis, dengan judul “IMPLEMENTASI INTERNET OF THINGS UNTUK PENYIRAMAN DAN PENGKABUTAN OTOMATIS PADA TANAMAN MENGGUNAKAN ALGORITMA NAÏVE BAYES (Studi Kasus Di Avicenna Greenhouse)”.²⁰⁰

46 1.2 Rumusan Masalah

Dengan mengacu pada pembahasan latar belakang, maka dapat disimpulkan bahwa fokus masalah penelitian ini adalah sebagai berikut:

1. Bagaimana cara implementasi teknologi *IOT* untuk penyiraman dan pengkabutan otomatis?
2. Bagaimana penerapan algoritma *Naïve bayes* dapat membantu dalam pengambilan keputusan otomatis berdasarkan data sensor?
3. Bagaimana membangun aplikasi berbasis web yang dapat membantu pengguna memonitoring kondisi pada lingkungan *Greenhouse* secara *real-time* serta mengendalikan sistem penyiraman dan pengkabutan secara otomatis?

107 1.3 Batasan Masalah

Beberapa ruang lingkup permasalahan yang menjadi pembatas dalam penelitian ini adalah sebagai berikut.:

1. Penelitian difokuskan pada pengembangan aplikasi penyiraman dan pengkabutan otomatis berbasis *IOT* untuk tanaman cabai di *Avicenna Greenhouse*.
2. Aplikasi yang dikembangkan memanfaatkan sensor kelembapan tanah, suhu, dan kelembapan udara sebagai parameter utama dalam proses penyiraman dan pengkabutan.
3. Aplikasi ini menggunakan algoritma *Naïve bayes* dalam pengambilan keputusan untuk menentukan kapan penyiraman dan pengkabutan dilakukan berdasarkan data sensor sebagai notifikasi.
4. Pengembangan aplikasi mencakup integrasi dengan platform berbasis web untuk *monitoring* dan pengendalian sistem secara *real-time*, tetapi tidak mencakup fitur lanjutan seperti rekomendasi pemupukan atau analisis pertumbuhan tanaman.
5. Pada penelitian yang dilaksanakan menggunakan metode *Agile* dalam pengembangan aplikasi supaya dapat meningkatkan fleksibilitas dalam implementasi dan evaluasi.

185 1.4 Tujuan Penelitian

Dengan merujuk pada **rumusan masalah dan batasan penelitian yang telah ditetapkan**, maka tujuan penelitian ini dapat dirumuskan sebagai berikut:

1. Menerapkan teknologi *IOT* untuk penyiraman dan pengkabutan otomatis pada *Greenhouse*.
2. Menerapkan algoritma *Naïve bayes* untuk membantu proses pengambilan keputusan berdasarkan data sensor.
3. Merancang dan membangun aplikasi *monitoring* berbasis web yang dapat membantu pengguna memonitoring kondisi *Greenhouse* secara *real-time* dan mengendalikan sistem penyiraman dan pengkabutan secara otomatis.

1.5 Metodologi Penelitian

10 1.5.1 Metode Pengumpulan Data

Agar data yang diperoleh dapat akurat dan sesuai, maka penelitian ini memanfaatkan metode pengumpulan data sebagai berikut:

a. Observasi

Pada **metode** observasi dilaksanakan dengan cara melakukan **pengamatan langsung** terhadap kondisi serta aktivitas di lokasi penelitian, yaitu Avicenna *Greenhouse* tempat tanaman cabai dibudidayakan. Observasi dilakukan untuk memahami proses penyiraman dan pengkabutan yang saat ini berjalan, serta kendala-kendala yang dihadapi dalam pengelolaan irigasi.

b. Wawancara

Wawancara dilaksanakan bersama pihak yang terlibat langsung pada pengelolaan *Greenhouse*, seperti pemilik *Greenhouse* atau petani.

c. Studi Pustaka

Kegiatan studi pustaka dilaksanakan melalui pengumpulan serta **kajian literatur** dari beragam sumber, seperti buku, jurnal, artikel ilmiah dan dokumen yang relevan dengan topik *IoT*, penyiraman otomatis, serta pengkabutan, serta algoritma *Naïve bayes* dan sumber lain yang bersangkutan dengan topik penelitian. Melalui studi pustaka, peneliti memperoleh landasan teoritis dan pemahaman terhadap teknologi dan metode yang relevan.

² 2. Metode Pengembangan Sistem

Dalam penelitian ini, digunakan metode *Agile* sebagai **model** pembangunan perangkat lunak. *Agile* mengedepankan proses siklus berulang dan peningkatan bertahap dalam *Software Development Life Cycle (SDLC)*. Tujuannya untuk menciptakan perangkat lunak yang dapat berkembang secara fleksibel dan menyesuaikan diri dengan perubahan kebutuhan di tengah proses pengembangan. Dalam metode *Agile*, seluruh proses pengembangan dibagi menjadi bagian-bagian kecil yang disebut iterasi atau *sprint*. Setiap *sprint* berlangsung dalam waktu yang relatif singkat dan mencakup beberapa tahapan penting, yaitu perencanaan, perancangan, pembangunan, pengujian, penerapan, peluncuran.

1.6 Sistematika Penulisan

Penyusunan laporan ¹⁴⁹ skripsi ini terbagi menjadi enam bab, masing-masing dilengkapi dengan sejumlah ⁴² sub bab. Urutan pembahasannya adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini memuat uraian mengenai latar belakang masalah, rumusan masalah, batasan penelitian, tujuan penelitian, metode yang digunakan, serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bab ini menyajikan landasan teori dan konsep-konsep ^{yang} mendukung penelitian. Sumber kajian diperoleh dari buku, jurnal, maupun situs web yang relevan.

²¹ BAB III: METODOLOGI PENELITIAN

Bab ini memaparkan metode penelitian yang diterapkan tahapan-tahapan yang dilakukan penulis selama proses penelitian di Avicenna *Greenhouse*.

¹²³ BAB IV: ANALISIS DAN PERANCANGAN

Bab ini membahas hasil analisis, perancangan perangkat keras, perangkat lunak, basis data, serta antarmuka. Selain itu, juga dijelaskan rancangan sistem perangkat lunak yang akan dibangun.

BAB V IMPLEMENTASI DAN PENGUJIAN

Pada bab ini memiliki uraian mengenai proses penerapan aplikasi, penjelasan setiap tahapan pengembangan, tampilan fitur-fitur yang telah dibuat, serta cara kerja aplikasi melalui pengujian yang dilakukan.

**¹²⁷
BAB VI PENUTUP**

Bab terakhir berisi kesimpulan dari penelitian yang dilaksanakan serta saran untuk pengembangan dan penerapan Internet of Things ² pada penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori yaitu merupakan kumpulan materi yang berhubungan dengan penelitian yang akan dilaksanakan. Dalam penyusunan penelitian, merujuk pada penelitian sebelumnya sebagai referensi. Tujuan dari hal ini adalah untuk membandingkan kesamaan serta perbedaan pada ³⁷ penelitian yang sedang dilakukan dengan penelitian sebelumnya atau terdahulu.

Tabel 2. 1 Acuan Jurnal Penelitian

No	Jurnal Penelitian	Masalah	Metode	Kesimpulan
1	<p>Judul: Sistem ¹⁶ Di ¹⁶ Indonesia, ¹⁶ Penyiraman Tanaman Otomatis Menggunakan Logika Fuzzy Dengan Teknologi Internet of Things Berbasis ESP8266 Dan Aplikasi Blynk</p> <p>Penulis: Ridho Alamsyah, Eddy Ratna Mufidah</p>	<p>banyak petani masih mengandalkan metode manual sehingga proses penyiraman belum optimal.</p>	<p>Penelitian ini menggunakan metode studi literatur, perancangan perangkat, logika fuzzy yaitu dalam penelitian ini dihasilkan keputusan menggunakan fuzzy logic yang terdiri dari beberapa tahap yaitu fuzzifikasi, mesin inferensi, dan defuzzifikasi. Pada output yang dihasilkan adalah keputusan akhir apakah pompa air otomatis akan</p>	<p>Penelitian ini memanfaatkan ⁴⁹ sensor kelembapan tanah untuk mendekripsi tingkat kelembapan. Dalam media tanam secara otomatis. Fokus utama dari sistem yang dikembangkan adalah otomasi penyiraman tanaman tanpa melibatkan aspek penerapan lainnya. Sistem ini juga terintegrasi dengan aplikasi</p>

	Ryansyah, Andari Yasinta Permana. Tahun: 2024		menyalah atau tidak dan pengujian perangkat.	<i>Blynk</i> yang berfungsi sebagai antarmuka untuk memantau serta mengendalikan proses penyiraman secara <i>real-time</i> melalui perangkat <i>mobile</i> .
2	Judul: 89 Implementasi Internet of Things (IOT) Pada Sistem Otomatisasi Penyiraman Tanaman Berbasis Mobile Penulis: M. Iqbal Hasani, Sri Wulandari. Tahun: 2023	Kegiatan penyiraman tanaman biasanya masih dikerjakan secara manual menggunakan peralatan sederhana, misalnya selang, ember, atau gayung, yang tentunya memerlukan banyak tenaga dan waktu	Penelitian ini mengimplementasikan algoritma <i>Naïve bayes</i> dimanfaatkan untuk menganalisis data yang diperoleh dari sensor dan menentukan langkah yang perlu diambil oleh sistem. Lalu melibatkan penerapan metodologi ¹¹¹ <i>System Development Life Cycle</i> , yang terdiri dari beberapa tahap, yaitu Analisis Kebutuhan, Desain Implementasi, Sistem, Pengujian, Pemeliharaan dan Peluncuran.	Penelitian ini memanfaatkan sensor kelembaban tanah untuk mendeteksi tingkat kelembaban, serta sensor suhu untuk memantau suhu sekitar tanaman. Penelitian ini berfokus pada otomasi penyiraman tanaman tanpa mempertimbangkan aspek <i>IOT</i> lainnya. Aplikasi <i>Blynk</i> digunakan untuk memonitoring dan mengendalikan proses penyiraman secara <i>real time</i> . Selain

				itu, penerapan algoritma <i>Naïve bayes</i> terbukti mampu mendukung pengambilan keputusan berdasarkan data sensor secara akurat.
3	Judul: 13 Implementasi Teknologi Penyiraman Sistem Pengkabutan Otomatisdan Monitoring Pintar Berbasis Tenaga Surya untuk Tempat Budidaya Tanaman Anggrek UD fairus Di Kabupaten Merauke. Penulis: Muhamad Rusdi, Muriani, Rivaldo Pasca Corpputy,	Permasalahan mitra yaitu masih menggunakan cara konvensional dalam proses pemeliharaan tanaman anggrek dari tahapan penyiraman tanaman anggrek secara langsung menggunakan selang hingga pemberian pupuk. Proses penyiraman yang berlebihan dapat menyebabkan adanya pembusukan pada tanaman anggrek dan tidak adanya teknologi yang memonitoring	Metode pelaksanaan yang digunakan pada pengabdian ini terdiri dari beberapa tahapan yaitu: diskusi dan observasi, Sosialisasi Program kelembaban tanah dan tingkat kelembaban lingkungan serta teknologi, <i>workshop</i> perancangan teknologi, evaluasi pengoperasian dan pemeliharaan teknologi, evaluasi dan capaian luaran kegiatan	Penelitian ini memanfaatkan sensor untuk mendeteksi kelembaban tanah sekitar tanaman. Sistem yang dikembangkan berfokus pada penyiraman dan pengkabutan tanaman secara efisien. Selain itu, aplikasi <i>Blynk</i> digunakan sebagai antarmuka pemantauan dan pengendalian

13 Mardiyasa Putra Yoga, Grace Christin Aditya Ronsumbre, Diah Bayu Titisari. Tahun: 2023	kelembapan dan suhu pada area budidaya. Selain itu, meningkatnya biaya operasional pemeliharaan tanaman anggrek yang disebabkan penggunaan mesin penyiraman berbahan bakar bensin.	sistem secara <i>real-time</i> , sehingga memungkinkan pengguna untuk mengatur penyiraman dan pengkabutan secara jarak jauh melalui perangkat <i>mobile</i> .
--	--	---

Beberapa penelitian terdahulu menunjukkan potensi besar penerapan *Internet of Things (IOT)* dalam otomasi penyiraman dan pengkabutan tanaman. Penelitian oleh (Alamsyah et al., 2024) membangun sistem penyiraman otomatis mengandalkan *ESP8266* dan aplikasi *Blynk*. Sistem tersebut menggunakan sensor kelembapan tanah untuk mengukur tingkat kelembapan media tanam dan mengaktifkan penyiraman, yang dapat dipantau secara *real-time* melalui perangkat *mobile*. Penelitian oleh (M. Iqbal Hasani & Sri Wulandari, 2023) juga memanfaatkan sensor kelembapan tanah dan suhu, dengan dukungan algoritma *Naïve bayes* untuk mendukung pengambilan keputusan penyiraman berdasarkan data sensor. Sistem ini diintegrasikan dengan aplikasi *Blynk* untuk pemantauan dan kendali secara jarak jauh. Sementara itu, penelitian oleh (Muhamad Rusdi et al., 2023) mengembangkan sistem penyiraman dan pengkabutan otomatis berbasis tenaga surya yang diterapkan pada budidaya anggrek. Sistem ini menggabungkan sensor kelembapan dan suhu serta aplikasi *Blynk* sebagai *interface* untuk memantau dan mengontrol penyiraman dan pengkabutan secara efisien.

Ketiga penelitian tersebut memiliki kesamaan dalam penggunaan teknologi *IOT* untuk otomatisasi proses penyiraman tanaman, serta penggunaan sensor

kelembapan dan suhu sebagai indikator utama dalam pengambilan keputusan. Namun, tidak sama dengan penelitian-penelitian tersebut, penelitian ini tidak hanya mengotomatiskan proses penyiraman, tetapi juga mengintegrasikan sistem pengkabutan tanaman dalam satu platform yang saling terhubung. Selain itu, sistem yang dikembangkan dalam penelitian ini menggunakan antarmuka berbasis web mandiri yang dibangun dengan *framework Flask*, bukan mengandalkan aplikasi pihak ketiga, sehingga memberikan fleksibilitas dan kontrol penuh terhadap fungsionalitas sistem. Penggunaan protokol *MQTT* dan integrasi notifikasi *WhatsApp* juga menjadi pembeda utama, yang tidak ditemukan pada tiga penelitian terdahulu.

Namun, terdapat beberapa keterbatasan pada studi sebelumnya. Pertama, fokus sebagian besar penelitian hanya terbatas pada proses penyiraman, tanpa mengintegrasikan sistem pengkabutan sebagai salah satu elemen penting dalam pengelolaan suhu dan kelembapan udara pada *Greenhouse*. Kedua, sistem yang dikembangkan sepenuhnya bergantung pada aplikasi pihak ketiga seperti *Blynk*, sehingga menimbulkan ketergantungan eksternal dan keterbatasan dalam fleksibilitas desain sistem. Ketiga, tidak ada integrasi terhadap notifikasi instan yang langsung terhubung ke pengguna seperti melalui *WhatsApp*, sehingga interaksi pengguna terhadap sistem cenderung pasif.

Penelitian ini menawarkan kelebihan dan kebaruan dengan merancang dan membangun sistem otomatisasi penyiraman dan pengkabutan tanaman cabai berbasis web tanpa ketergantungan pada aplikasi pihak ketiga. Sistem dikembangkan menggunakan *framework Flask (Python)* yang terintegrasi langsung dengan perangkat *IOT* melalui protokol *MQTT* untuk komunikasi data secara *real-time*. Algoritma *Naïve bayes* diterapkan sebagai sistem pengambilan keputusan berbasis data sensor suhu, kelembapan tanah dan udara untuk prediksi penyiraman dan pengkabutan. Selain itu, sistem ini juga menambahkan notifikasi langsung melalui aplikasi *WhatsApp* untuk memberikan informasi hasil prediksi dari *navie bayes* dan status aksi sistem kepada pengguna. Proses pengembangan sistem dengan menggunakan metode *Agile*, memungkinkan penyesuaian dinamis dalam setiap tahap pengembangan. Dengan memilih pendekatan ini, pada penelitian memberikan kontribusi inovatif terhadap sistem otomasi pertanian

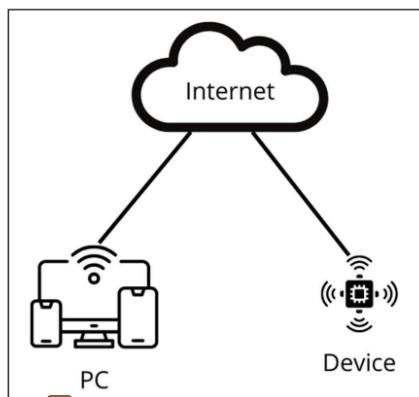
berbasis *IOT* yang lebih efisien, fleksibel, dan responsif terhadap kebutuhan pengguna.

192 2.2 Dasar Teori

Adapun teori-teori yang ada sebagai acuan dalam implementasi *Internet of Things* yaitu sebagai berikut:

1 2.2.1 *Internet of Things*

Internet of Things yaitu merupakan keterampilan perangkat yang dapat terhubung, bertukar data dan berkomunikasi melalui jaringan internet. *Internet of Things* merupakan teknologi yang dapat melakukan komunikasi, kontrol, kerja sama dengan berbagai macam perangkat keras, data di jaringan internet. (Ifa Susuek Anselmus Talli et al., 2023).



67 Gambar 2.1 Konsep *Internet of Things*

Sumber: (Kelasplc, 2023)

Gambar 2.1 menunjukkan hubungan antara perangkat *PC* dan *device* (perangkat dengan sensor) yang terkoneksi ke intemet. Perangkat-perangkat tersebut mengirimkan dan menerima data melalui jaringan intemet, sehingga

memungkinkan terjadinya integrasi sistem, otomatisasi proses, dan pengambilan keputusan berbasis data secara *real-time*.

2.2.2 Penyiraman Tanaman

Penyiraman pada tanaman berarti memberikan air secara langsung kepada tanaman. Tujuannya yaitu untuk memasok air dan nutrisi yang dibutuhkan akar supaya diserap dan disalurkan ke seluruh bagian dalam pada tanaman. Menyiram tanaman dapat dilakukan dengan berbagai cara, seperti melakukan penuangan air di dekat perakaran, mengguyur pot dengan air, hingga menggunakan sistem irigasi otomatis (Kurniawan, 2023).

Tabel 2. 2 Pengujian Kelembapan Tanah

Sumber: (Ifa Susuek Anselmus Talli et al., 2023)

No	Kondisi Tanah	Sensor Soil moisture	Alat Ukur
1	Tanah dicampur pupuk	66%	Basah
2	Tanah dicampur pasir	40%	Sangat Kering
3	Tanah Biasa (Liat)	62%	Basah
4	Tanah Campur Kopi	70%	Basah
5	Tanah Campur Garam	61%	Basah
6	Tanah Disiram Air Teh	60%	Basah
7	Tanah Disiram Air Sabun	65%	Basah
8	Tanah ditaburkan kapur	66%	Basah
9	Tanah ditaburkan micin	65%	Basah
10	Tanah ditaburkan sekam	64%	Basah

Berdasarkan data dari tabel Pengujian Kelembapan Tanah, nilai kelembapan tanah bervariasi tergantung pada jenis dan perlakuan terhadap media tanam. Dari sepuluh sampel pengujian yang dilakukan, sebagian besar menunjukkan nilai kelembapan antara 60% hingga 70%, yang dikategorikan dalam kondisi "basah" menurut alat ukur kelembapan.

Untuk tanaman cabai, kelembapan tanah yang ideal berada dalam kisaran 60% hingga 80%, sebagaimana dijelaskan dalam penelitian oleh (S Nursuwars & Sujana, 2018). Kisaran ini penting karena sistem perakaran tanaman cabai memerlukan kelembapan yang cukup untuk menyerap air dan nutrisi secara optimal, tetapi tidak terlalu jenuh agar tidak menyebabkan pembusukan akar. Kelembapan di bawah 60% umumnya menunjukkan kondisi yang terlalu kering, seperti pada tanah berpasir dengan kelembapan 40%, yang dinyatakan sebagai "sangat kering". Tanah dengan kondisi ini berisiko menghambat pertumbuhan tanaman karena suplai air tidak mencukupi untuk proses fotosintesis dan penyerapan nutrisi. Sebaliknya, menjaga kelembapan tanah dalam kisaran 60% hingga 70% seperti pada tanah berpupuk, tanah liat, atau tanah jenis lainnya yang dicampurkan dengan bahan organik (kopi, sekam) dapat memberikan kondisi yang stabil dan mendukung pertumbuhan tanaman cabai secara optimal.

2.2.3 Pengkabutan Tanaman

Misting atau pengkabutan merupakan salah satu teknik penting dalam merawat tanaman, terutama pada tanaman *hortikultura* dan tanaman hias yang sensitif terhadap perubahan suhu dan kelembapan. Teknik ini dilakukan dengan menyemprotkan air berupa butiran sangat halus ke udara di area sekitar tanaman. Proses ini memanfaatkan prinsip pendinginan *evaporatif*, yaitu saat butiran air menguap, panas dari udara sekitar diserap sehingga menyebabkan penurunan suhu lokal secara efektif. (Kurniawan, 2023).

Tabel 2. 3 Pengujian DHT11

Sumber: (Fahmi et al., 2022)

No	Waktu	Suhu	Kelembapan Udara
1	2022-08-02T07:58:49+00:00	22,9°C	77%
2	2022-08-02T07:59:19+00:00	22,4 °C	78%
3	2022-08-02T07:00:22+00:00	26,8 °C	71%
4	2022-08-02T07:00:44+00:00	30,4 °C	64%

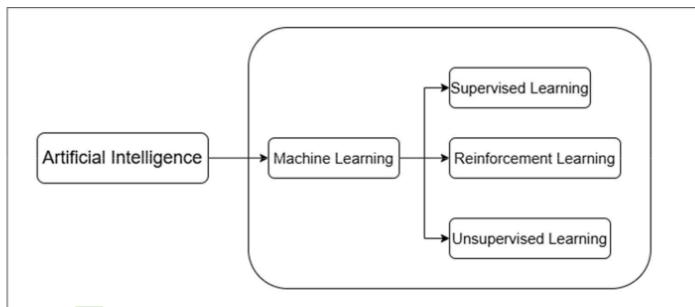
5	2022-08-02T07:01:31+00:00	23,9 °C	71%
6	2022-08-02T07:01:53+00:00	21,3 °C	92%
7	2022-08-02T07:02:17+00:00	23,4 °C	77%
8	2022-08-02T07:02:33+00:00	27,1 °C	71%
9	2022-08-02T07:02:49+00:00	27,8 °C	70%
10	2022-08-02T07:02:52+00:00	27,2 °C	70%
11	2022-08-02T07:03:10+00:00	27,2 °C	70%
12	2022-08-02T07:03:25+00:00	19,3 °C	96%
13	2022-08-02T07:03:50+00:00	23,3 °C	67%

Berdasarkan hasil uji coba terhadap sensor *DHT11* yang dilakukan di Greenhouse budidaya tanaman cabai, diperoleh data suhu dan kelembapan udara yang bervariasi. Hasil pengujian sebagaimana tercantum pada Tabel 2.3 menunjukkan bahwa sensor mampu mendeteksi suhu dalam rentang 19,3°C hingga 30,4°C dan kelembapan udara antara 64% hingga 96%. Nilai-nilai ini memperlihatkan dinamika kondisi mikroklimat dalam Greenhouse selama periode pengamatan. Salah satu titik pembacaan menunjukkan suhu sebesar 27,3°C dan kelembapan udara 71%, yang dapat dikategorikan sebagai kondisi optimal untuk pertumbuhan tanaman cabai. Data ini juga menjadi acuan dalam menilai

keandalan sensor dalam membaca kondisi lingkungan. Sensor *DHT11* cukup layak digunakan untuk mendukung pengaktifan sistem pengkabutan berdasarkan data suhu dan kelembapan udara. Pengkabutan akan menyala secara otomatis ketika suhu melebihi 27°C dan kelembapan udara turun di bawah 71%. Hal ini karena pada kondisi tersebut, udara menjadi lebih kering dan tanaman mengeluarkan lebih banyak air melalui daunnya. Dengan adanya sistem pengkabutan, kelembapan udara di sekitar tanaman bisa dijaga, dan suhu di dalam Greenhouse bisa menjadi lebih sejuk secara alami. Dengan begitu, tanaman tetap berada dalam kondisi yang baik untuk tumbuh. Data dari sensor digunakan sebagai dasar untuk menghidupkan pengkabutan secara otomatis jika suhu dan kelembapan sudah mencapai batas yang bisa memengaruhi pertumbuhan tanaman cabai.

⁷⁴ 2.2.4 Machine Learning

Machine Learning (ML) merupakan bagian dari kecerdasan buatan yang dapat membuat sistem belajar dari data dan membuat keputusan atau prediksi tanpa diprogram secara rinci. ML adalah area penelitian yang berkaitan dengan pengembangan dan evaluasi algoritma agar komputer dapat belajar. Samuel menyatakan bahwa *Machine Learning* terdiri atas algoritma yang bersifat umum yang mampu menghasilkan informasi menarik atau berguna ⁵⁸ dari kumpulan data tanpa perlu menulis kode yang khusus. (Ibnu Daqiqil Id, 2021).



¹³⁰ Gambar 2. 2 Skema *Artificial Intelligence* dan *Machine Learning*

Sumber: (Roihan et al., 2019)

Gambar di atas menjelaskan bahwa *Machine Learning* merupakan salah satu bagian dari *Artificial Intelligence*. Di dalam *Machine Learning*, terdapat tiga pendekatan utama, yaitu ¹⁸⁷ *Unsupervised Learning*, *Supervised Learning* dan *Reinforcement Learning*. *Supervised Learning* dengan menggunakan dataset berlabel, di mana model dilatih untuk memetakan *input* ke *output* berdasarkan data yang telah diketahui hasilnya. *Unsupervised Learning* bekerja dengan tanpa label dan bertujuan menemukan pola tersembunyi dalam data. Sedangkan *Reinforcement Learning* menggunakan pendekatan *trial-and-error* dengan sistem penghargaan untuk menemukan strategi optimal dalam pengambilan keputusan. Pada penelitian ini, metode *Supervised Learning* digunakan karena data yang digunakan berupa dataset dengan label yang telah ditentukan sebelumnya, sehingga model dapat dilatih untuk melakukan klasifikasi atau prediksi

berdasarkan data tersebut secara akurat. Ketiga pendekatan tersebut mempunyai kelebihan masing-masing tergantung dari karakteristik data dan tujuan analisis yang diinginkan. Dengan memilih pendekatan yang tepat, proses pembelajaran mesin memberikan hasil yang lebih optimal terhadap kebutuhan penelitian.

2.2.5 Algoritma *Naïve bayes*¹²⁹

Naïve Bayes merupakan sebuah algoritma klasifikasi dalam *Machine Learning* yang didasarkan *Theorema Bayes*, yaitu sebuah model probabilistik yang digunakan untuk memprediksi kelas dari suatu data berdasarkan probabilitas sebelumnya (*prior probability*) dan informasi baru (*posterior probability*). Meskipun teori ini pertama kali dikenalkan oleh Thomas Bayes pada abad ke-18, penerapannya dalam bidang ilmu komputer dan pembelajaran mesin menjadi sangat signifikan dalam beberapa dekade terakhir. Algoritma ini bekerja dengan mengasumsikan bahwa fitur-fitur pada dataset saling bebas secara kondisi (*naive assumption*), yang menjadikannya sangat sederhana namun efektif untuk berbagai tugas klasifikasi seperti *spam filtering*, *sentiment analysis*, dan diagnosis medis (Ibnu Daqiqil Id, 2021). Pada dasarnya Algoritma *Naïve Bayes* memiliki beberapa varian yang digunakan sesuai dengan tipe data yang dianalisis:

- a. *Multinomial Naïve Bayes*: Digunakan untuk data diskrit berupa jumlah frekuensi kemunculan fitur tertentu. Umumnya digunakan untuk klasifikasi teks, seperti *document classification* atau *spam filtering*.
- b. *Bernoulli Naïve Bayes*: Cocok untuk data biner, yaitu fitur yang hanya memiliki dua nilai: 0 dan 1.
- c. *Gaussian Naïve Bayes*: Digunakan untuk data numerik atau *kontinu* yang diasumsikan mengikuti distribusi *Gaussian* (normal).

Pada penelitian ini, varian yang digunakan adalah ²² *Gaussian Naïve bayes*, yang merupakan salah satu jenis *Naïve bayes*. Prinsip dasarnya serupa dengan *Naïve bayes*, tetapi pada *Naïve bayes* kita tidak bisa menghitung probabilitas untuk data yang bersifat *continues*. Oleh karena itu kita menggunakan distribusi *gaussian* untuk menghitung probabilitas dengan asumsi data terdistribusi normal. Adapun *pdf* (*probability density function*) dari *gaussian distribution* adalah:

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Sumber: (Ibnu Daqiqil Id, 2021)

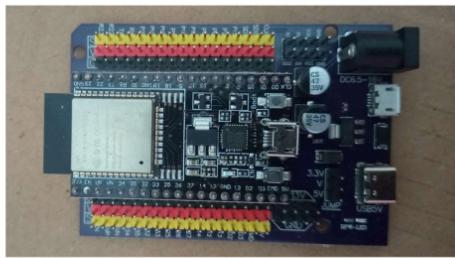
Rumus di atas digunakan untuk menghitung probabilitas suatu nilai fitur x_i yang termasuk dalam kelas y . Di mana μ_y adalah rata-rata (*mean*) dan σ_y^2 adalah varians dari fitur tersebut untuk kelas y . Fungsi ini menggambarkan seberapa besar kemungkinan nilai x_i muncul dalam distribusi *Gaussian* kelas tertentu.

Metode *Gaussian Naïve bayes* cocok diterapkan pada data yang bersifat *kontinu* dan mendekati distribusi normal. Pada penelitian ini, metode ini digunakan untuk mengklasifikasikan kondisi lingkungan di dalam *Greenhouse* berdasarkan tiga parameter sensor utama, yaitu suhu, kelembapan udara, dan kelembapan tanah. Setiap parameter sensor diklasifikasikan untuk menentukan apakah kondisi tersebut memerlukan penyiraman, pengkabutan, atau tidak perlu tindakan.

Sebagai contoh, sistem mengumpulkan data dari sensor secara *real-time*, lalu menghitung nilai probabilitas dari setiap parameter sensor terhadap kelas tindakan menggunakan rumus di atas. Setelah diperoleh nilai *mean* dan *varians* untuk masing-masing kelas tindakan berdasarkan data pelatihan, sistem dapat menentukan kelas mana yang memiliki probabilitas terbesar untuk kondisi saat itu. Misalnya, jika suhu tinggi dan kelembapan rendah, hasil klasifikasi kemungkinan besar akan menunjuk pada tindakan pengkabutan.

2.2.6 *ESP32*

ESP32 merupakan sebuah nama alat mikrokontroler yang dibuat oleh ¹⁶⁴ perusahaan teknologi yang ada di Shanghai, Tiongkok, yaitu *Espressif Systems*. Mikrokontroler ini merupakan penerus dari seri sebelumnya, yaitu *ESP8266*, dengan peningkatan performa dan fitur yang lebih lengkap. *ESP32* menawarkan solusi jaringan *WiFi* dan *Bluetooth* yang mandiri sebagai jembatan antara sistem mikrokontroler dengan jaringan nirkabel (Kusumah & Pradana, 2019).



Gambar 2. 3 *ESP32*

Sumber: (Penulis, 2025)

Pada gambar di atas, *ESP32* terpasang pada sebuah *development board* yang telah dilengkapi dengan berbagai pin *header* untuk kemudahan koneksi ke komponen eksternal. Modul *ESP32* terletak di sisi kiri atas papan (ditandai dengan chip berwarna logam), yang merupakan inti dari sistem. *Board* ini menyediakan koneksi ke pin digital dan analog yang ditandai dengan label seperti D0, D1, dan seterusnya, serta port daya seperti 3.3V dan 5V untuk menyuplai arus ke sensor. Modul juga lengkap dengan port Micro USB dan USB Type-C untuk pemrograman dan komunikasi data, serta input catu daya eksternal (DC 6.5–16V) yang memungkinkan perangkat bekerja secara independen tanpa koneksi komputer. Keberadaan komponen tambahan seperti regulator tegangan, kapasitor, dan LED indikator juga berperan penting dalam memastikan kestabilan sistem.

Dalam penelitian ini, *ESP32* berfungsi menjadi pusat kendali sistem *IOT* yang bertugas menerima data ³⁶ dari sensor suhu, kelembaban udara, dan kelembaban tanah. Data yang diterima akan diolah secara lokal, kemudian *ESP32* mengambil keputusan untuk mengaktifkan aktuator seperti pompa air (untuk penyiraman) atau misting (pengkabutan), serta mengirimkan data ke *broker MQTT* melalui jaringan *WiFi*.

Dalam penelitian ini, *ESP32* dipasang menggunakan *expansion shield plate* untuk memudahkan koneksi pin dan pemasangan sensor secara modular. Penggunaan *shield* juga membuat sistem lebih rapi dan stabil dalam pengembangan. Beberapa pin seperti GPIO26 dan GPIO27 digunakan untuk menghubungkan sensor dan aktuator secara langsung ke papan ekspansi.

⁵ 2.2.7 Soil moisture Sensor

Soil moisture Sensor yaitu alat yang digunakan untuk mengukur kelembaban tanah. Alat ini cukup mudah digunakan, namun sangat cocok untuk mengawasi taman atau kadar air pada tanaman di halaman. Sensor memiliki dua *probe* yang berfungsi untuk mengalirkan listrik melalui tanah, lalu mengukur resistansinya untuk menentukan tingkat kelembaban. Lebih masif air dalam tanah membuatnya lebih baik dalam menyalurkan listrik (resistansi rendah), sedangkan tanah yang kering cenderung sulit untuk menyalurkan listrik (resistansi tinggi) (Eka Candra & Maulana Universitas Putera Batam, 2019).



Gambar 2. 4 Soil Mastuire Sensor

Sumber: (Penulis, 2025)

Pada gambar di atas, terlihat bahwa sensor memiliki dua buah batang *probe* berbahan logam yang dilapisi pelindung, dengan masing-masing *probe* tersambung ke bagian rangkaian elektronik di bagian atas. Tulisan dalam karakter Mandarin yang tertera di papan menunjukkan bahwa sensor ini merupakan versi produksi dari pabrikan Cina. *Probe* tersebut didesain untuk ditancapkan langsung ke dalam tanah, dan sensor akan mendeteksi nilai resistansi berdasarkan kadar air yang menyentuh kedua elektroda. Sensor ini terhubung ke mikrokontroler seperti *ESP32* melalui pin analog. Pada praktiknya, nilai kelembaban dari sensor ini akan dibaca oleh mikrokontroler dan diubah menjadi data persentase kelembaban tanah. Data tersebut selanjutnya dapat digunakan untuk mengambil keputusan, seperti mengaktifkan sistem penyiraman otomatis ketika kelembaban tanah turun di bawah ambang batas yang telah ditentukan.

2.2.8 ¹⁷³ **DHT11**

Sensor *DHT11* yaitu modul yang berfungsi untuk mengukur suhu dan kelembapan udara di lingkungan. Alat ini termasuk dalam kategori perangkat resistif seperti sensor suhu *NTC*. Cara *DHT11* beroperasi adalah dengan mengamati perubahan resistansi yang terjadi akibat suhu dan kelembapan udara, lalu mengirimkan informasi dalam bentuk sinyal digital berupa pulsa rendah dan tinggi ke alat mikrokontroler yang menunjukkan nilai suhu dan kelembapan udara. (Ifa Susuek Anselmus Talli et al., 2023).



Gambar 2. 5 *DHT11*

Sumber: (Penulis, 2025)

Pada gambar di atas terlihat modul *DHT11* yang memiliki bentuk fisik khas dengan bagian sensor berwarna biru dan kisi-kisi di bagian depan sebagai ventilasi untuk sirkulasi udara. Sensor ini dilengkapi dengan tiga pin konektor di sisi bawahnya yang digunakan untuk sambungan ke catu daya, *ground*, dan sinyal data. Bagian belakang sensor terhubung dengan kabel yang telah dilapisi pelindung hitam agar lebih aman dan rapi dalam instalasi. Modul *DHT11* ini biasa digunakan pada sistem monitoring berbasis mikrokontroler seperti *ESP32* untuk membaca kondisi suhu dan kelembapan lingkungan secara *real-time*. Sensor *DHT11* memiliki kelebihan berupa konsumsi daya rendah, ukuran kecil, dan kemudahan dalam integrasi ke berbagai proyek elektronik, terutama di bidang pertanian, sistem otomatisasi rumah, serta pemantauan lingkungan berbasis *IOT*. Nilai suhu dan kelembapan yang dihasilkan sensor dapat ditampilkan pada layar atau dikirim ke *server* atau aplikasi pemantauan jarak jauh melalui koneksi *WiFi*.

2.2.9 Nozzle Sprayer

Nozzle Sprayer merupakan komponen dari alat penyemprot yang terdiri dari pipa yang menghubungkan tangki dengan bagian ujung alat semprot. *Nozzle* ini memiliki peran untuk mengubah aliran cairan menjadi semprotan dengan pola khusus dan tekanan yang sesuai dengan kebutuhan. Bentuk *nozzle* dapat bervariasi, seperti konus, kipas datar, dan kerucut penuh. Masing-masing bentuk *nozzle* menghasilkan pola semprotan yang berbeda dan sesuai untuk berbagai jenis kebutuhan. (Solahart Handal, 2025).



Gambar 2. 6 *Nozzle Sprayer*

Sumber: (Penulis, 2025)

Pada gambar di atas, terlihat *nozzle* dengan ujung berwarna jingga yang dapat disesuaikan, memungkinkan pengguna untuk mengatur intensitas atau pola semprotan, seperti kabut halus atau semprotan terarah. Komponen utama dari *nozzle* ini terdiri dari dua bagian: saluran utama berwarna hitam sebagai penghubung ke selang distribusi air atau cairan, dan bagian ujung *nozzle* yang berfungsi untuk mengatomisasi cairan menjadi partikel halus.

Desain ulir pada bagian jingga memungkinkan pengaturan manual tekanan dan *volume* semprotan, menjadikannya sangat ideal untuk sistem irigasi tetes atau penyemprotan otomatis berbasis sensor. *Nozzle* ini sangat umum digunakan dalam sistem pertanian modern, khususnya untuk penyemprotan pestisida, nutrisi cair, atau air dalam skala kecil dan terkontrol. Selain memberikan efisiensi penggunaan cairan, *nozzle* ini juga membantu menghindari pemborosan dan memastikan distribusi merata ke tanaman.

2.2.10 ⁸¹ Relay

Modul *Relay* merupakan sebuah saklar elektromagnetik yang berfungsi untuk memutuskan dan menyambungkan aliran listrik. Cara kerja *Relay* secara umum mirip dengan kontaktor magnet, yaitu bergantung pada gaya magnet yang dihasilkan oleh lilitan kawat kumparan. (Effendi et al., 2022).



Gambar 2. 7 *Relay*

Sumber: (Penulis, 2025)

Pada gambar tersebut menunjukkan sebuah modul *Relay 2 Channel*. Dua kabel merah dan hitam terhubung pada terminal beban, yang biasanya digunakan untuk mengontrol perangkat eksternal seperti pompa air. Modul ini memiliki *optocoupler* dan *transistor* sebagai penguat sinyal dari mikrokontroler, serta *jumper* pengatur mode aktif. *Optocoupler* (atau *optoisolator*) berfungsi untuk menghubungkan dua rangkaian listrik tanpa kontak langsung, melainkan menggunakan cahaya sebagai media isolasi. *Transistor* bekerja sebagai saklar elektronik yang memungkinkan *Relay* aktif hanya saat sinyal dari mikrokontroler sesuai. Modul *Relay* ini umumnya membutuhkan tegangan kerja 5V atau 3.3V, tergantung spesifikasinya, dan memiliki indikator LED untuk menunjukkan status aktif (*ON*) atau tidak aktif (*OFF*) pada masing-masing *Channel*.

Dalam penelitian ini, *Relay* digunakan sebagai aktuator untuk mengontrol proses penyiraman dan pengkabutan otomatis di *Greenhouse* tanaman cabai. Perintah aktivasi *Relay* dikendalikan secara otomatis oleh sistem. Dengan penggunaan *Relay* ini, sistem dapat menghidupkan atau mematikan alat secara fisik, seperti pompa, sesuai kebutuhan tanaman secara cerdas dan efisien.

2.2.11 ⁸⁴ *Unified Modeling language*

Merupakan suatu bahasa visual yang berfungsi untuk memodelkan dan merancang aplikasi dengan cara yang jelas dan teratur. Melalui *UML*, pengembang dapat membuat beragam diagram yang menggambarkan cara kerja suatu sistem, pihak yang terlibat, serta aliran data yang berjalan di dalamnya (Irhan Hisyam Dwi Nugroho, 2024).

Berikut adalah diagram-diagram dalam *UML*:

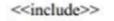
³³ 1. *Use Case Diagram*

Use case diagram merupakan bagian dari jenis lainnya pada diagram *UML* (*Unified Modelling Language*) untuk menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case* bisa mendeskripsikan tipe interaksi antara pengguna dengan sistemnya (Irhan Hisyam Dwi Nugroho, 2024).

Daftar simbol pada *use case diagram* dapat dilihat pada tabel berikut:

Tabel 2. 4 *Use Case Diagram*

Sumber: (Irhan Hisyam Dwi Nugroho, 2024)

No	Simbol	Keterangan
1	Aktor/Pengguna 	Perwakilan dari peran orang, sistem yang lain, atau alat komunikasi dengan <i>use case</i>
2	<i>Use Case</i> 	Merupakan abstraksi dan hubungan antara sistem dan aktor
3	<i>Association/ Asosiasi</i> 	Merupakan Abstraksi dari koneksi antara aktor dengan <i>use case</i>
4	<i>Generalisasi</i> 	Menampilkan keahlian aktor agar dapat terlibat dalam contoh penggunaan
5	<i>Include</i> 	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
6	<i>Extend</i>	Menunjukkan bahwa suatu <i>use case</i> merupakan

		tambahan kegunaan dari skenario lain jika suatu keadaan dipenuhi
--	---	--

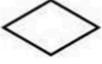
2. Activity Diagram

Merupakan salah satu jenis diagram UML yang bermanfaat untuk merancang model dari berbagai prosedur dalam sebuah sistem, urutan dari prosedur digambarkan dalam arah ⁸⁸ vertikal. Diagram ini adalah pengembangan dari use case dan menunjukkan jalur **aktivitas** yang ditampilkan sebagai serangkaian menu atau proses bisnis yang terdapat dalam sistem tersebut (Ucy Sugiarti, 2024). ¹⁶²

Daftar simbol pada *Activity diagram* sebagai berikut:

Tabel 2. 5 *Activity Diagram*

Sumber: (Ucy Sugiarti, 2024)

No	Simbol	Keterangan
1		Sebuah diagram aktivitas dimulai dengan sebuah kondisi awal.
2		Kegiatan yang dilakukan oleh sistem, umumnya diawali dengan kata kerja.
3		Percabangan di mana terdapat beberapa opsi kegiatan yang tersedia.
4		Penggabungan di mana sejumlah kegiatan yang berbeda disatukan menjadi kesatuan.
5		Status akhir yang dihasilkan oleh sistem, sebuah diagram aktivitas berfungsi memiliki status akhir.
6		Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

3. Class Diagram

Class Diagram merupakan suatu **diagram** yang digunakan untuk merancang sistem berorientasi objek untuk memvisualisasikan struktur kelas, atribut, metode, serta hubungan antar kelas dalam sistem (Ayoni Sulthon, 2023).

Daftar simbol pada *Class Diagram* sebagai berikut:

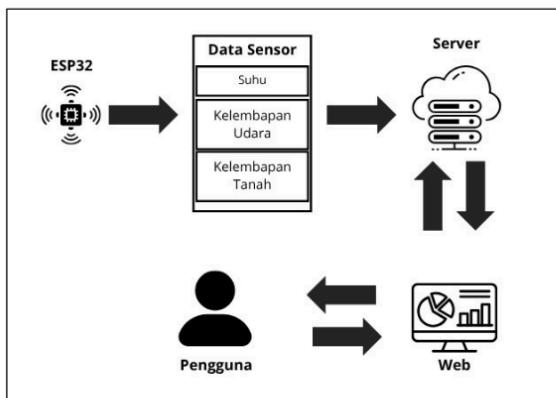
Tabel 2. 6 *Class Diagram*

Sumber: (Ayoni Sulthon, 2023)

No	Simbol	Keterangan
1	<i>Generalization</i> 	Hubungan yang menunjukkan bagaimana objek anak memiliki berbagai tingkah laku dan struktur data yang terhubung dengan objek induk di atasnya
2 ²⁷	<i>Nary Association</i> 	Usaha untuk menghindari keterkaitan dengan lebih dari dua benda.
3	<i>Class</i> 	Kumpulan dari benda-benda yang memiliki atribut dan tindakan yang serupa.
4	<i>Realization</i> 	Tindakan yang benar-benar dilakukan oleh suatu benda.
5	<i>Collaboration</i> 	Penjelasan mengenai rangkaian tindakan yang diperlihatkan oleh sistem yang memproduksi sebuah hasil yang dapat diukur untuk seorang pelaku.
6	<i>Dependency</i> 	Keterkaitan dimana perubahan yang terjadi pada satu unsur otonom berdampak pada unsur yang bergantung pada unsur yang tidak otonom.
7	<i>Association</i> 	Apa yang menghubungkan antara objek satu dengan objek lainnya.

2.2.12 Website

Situs *web* yakni himpunan dari laman-laman situs yang ada pada salah satu domain atau *subdomain* yang ada pada *World Wide Web (WWW)* di internet. Situs *web* bisa berisi gambar, video, teks dan elemen interaktif lainnya yang dirancang untuk menyampaikan informasi, menyediakan layanan, atau menjalankan fungsi tertentu kepada pengunjungnya. (Iftitah Nurul Laily, 2022).



Gambar 2. 8 Arsitektur *Website*

Sumber: (Wardani et al., 2021)

Gambar di atas menunjukkan alur kerja sistem berbasis *IOT* yang terhubung dengan *Website*. Sensor suhu, kelembapan udara, dan kelembapan tanah akan mengirimkan data ke mikrokontroler *ESP32*. Data tersebut kemudian dikirim ke server melalui jaringan internet. Server memproses data menggunakan algoritma *Naïve bayes* untuk menentukan apakah sistem penyiraman atau pengkabutan perlu dilakukan dan pengguna dapat memantau kondisi *Greenhouse* secara *real-time*.

Selain sebagai media *monitoring*, *Website* ini juga dapat membuat pengguna untuk mengendalikan sistem secara manual dari jarak jauh, seperti mengaktifkan atau mematikan penyiraman dan pengkabutan. Dengan demikian, *Website* menjadi bagian penting dalam sistem karena menghubungkan pengguna dengan perangkat *IOT* dan *server* secara interaktif dan efisien.

2.2.13 *Hosting*

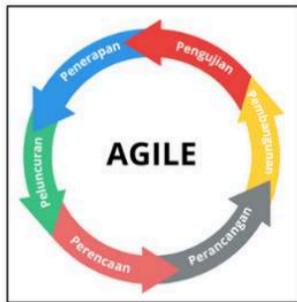
Hosting atau web *Hosting* adalah layanan penyimpanan yang digunakan untuk menampung seluruh kebutuhan sebuah *Website*, seperti *file*, gambar, video, *script*, *email*, aplikasi, dan *database*, agar *Website* tersebut dapat diakses secara *online* melalui internet. Tidak dengan layanan web *Hosting*, sebuah *Website* tidak bisa diakses oleh pengguna internet (Ariata C., 2024).

Ada beberapa jenis web *Hosting* yang bisa dipilih sesuai kebutuhan, antara lain:

- a. *Shared Hosting*: Berbagi sumber daya *server* dengan beberapa *Website* lain. Cocok untuk pemula dan situs dengan lalu lintas rendah.
- b. *VPS Hosting (Virtual Private Server)*: Memberikan lingkungan *server virtual* dengan sumber daya terdedikasi, meskipun masih berbagi fisik *server* dengan pengguna lain.
- c. *Dedicated Hosting*: Menyediakan satu *server* fisik secara eksklusif untuk satu *Website*. Cocok untuk situs dengan lalu lintas tinggi dan kebutuhan kustomisasi penuh.
- d. *Cloud Hosting*: Mengelola sumber daya dari jaringan *server* yang saling terhubung, menawarkan fleksibilitas dan skalabilitas yang tinggi.

2.2.14 *Metode Agile*

Metodologi *Agile Software Development Life Cycle (SDLC)* adalah pendekatan modern yang lebih fleksibel disandingkan metode tradisional yang linier dan berurutan. *Agile* mengutamakan iterasi berkelanjutan dalam siklus pendek yang disebut *sprint*, sehingga memungkinkan penyesuaian cepat terhadap perubahan selama proses pengembangan. Dengan cara ini, tim dapat fokus pada pengembangan fitur yang berfungsi dan dapat diuji setiap siklusnya. Secara teori, *Agile* menekankan kolaborasi tim, komunikasi terbuka, dan pengiriman produk secara bertahap yang dapat diuji dan diperbaiki terus-menerus. Pendekatan ini mempercepat *feedback*, meningkatkan kualitas perangkat lunak, serta mengurangi risiko kegagalan proyek. *Agile* juga mendorong keterlibatan pengguna secara aktif agar hasil akhir lebih sesuai dengan kebutuhan (AgileTech Vietnam, 2025).



Gambar 2. 9 Metode *Agile*

Sumber: (AgileTech Vietnam, 2025)

Berikut adalah tahap-tahap pengembangan perangkat lunak yang diterapkan dalam metode *Agile*:

1. Perencanaan

Tahap ini dilakukan untuk mengidentifikasi kebutuhan sistem serta menyusun tujuan yang ingin dicapai pada periode pengembangan tertentu yang disebut *sprint*.

2. Perancangan

Setelah kebutuhan dan tujuan ditentukan, tahap selanjutnya adalah merancang struktur sistem dan antarmuka pengguna.

3. Pembangunan

Pada tahap ini, sistem atau fitur mulai dibangun berdasarkan perancangan yang telah dibuat. Proses pengembangan dilakukan secara bertahap sesuai dengan durasi *sprint* yang telah ditetapkan.

4. Pengujian

Setelah proses pengembangan selesai, Perangkat lunak yang dibuat diuji untuk memastikan fungsinya bekerja dengan baik dan sesuai dengan keperluan.

5. Penerapan

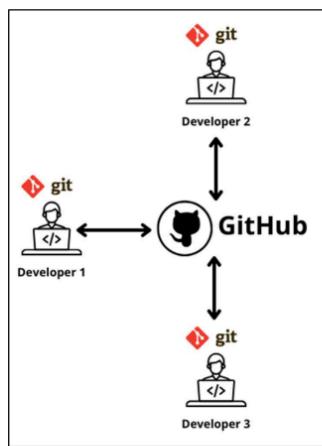
Jika perangkat lunak telah dinyatakan layak, maka hasil pengembangan akan diterapkan pada lingkungan produksi agar dapat digunakan oleh pengguna akhir.

6. Peluncuran

Setelah perangkat lunak melewati seluruh tahapan, lalu jika tidak ada perubahan pada tahapan sebelumnya maka sistem siap untuk diluncurkan.

2.2.15 GitHub

GitHub adalah platform berbasis *cloud* yang digunakan secara luas oleh pengembang perangkat lunak untuk mengelola proyek secara kolaboratif. Platform ini mendukung sistem kendali versi *Git* yang mengizinkan pemakai melacak perubahan kode, bekerja dalam *GitHub* juga menyediakan berbagai fitur seperti *repository*, *branch*, *pull request*, dan *issue tracker* untuk mempermudah proses pengembangan tim (*GitHub*, 2025).



Gambar 2. 10 GitHub

Sumber: (AgileTech Vietnam, 2025)

Gambar di atas menggambarkan alur kerja kolaboratif dalam pengembangan perangkat lunak menggunakan *Git* dan *GitHub*. Setiap pengembang (*Developer 1*, *Developer 2*, dan *Developer 3*) melakukan perubahan kode secara lokal menggunakan *Git*, kemudian mendorong (*push*) atau menarik (*pull*) perubahan tersebut dari repository pusat yang disimpan di *GitHub*. Dengan pendekatan ini,

setiap perubahan kode dapat dilacak, dikelola, dan digabungkan (*merge*) dengan lebih mudah, sehingga mendukung pengembangan proyek secara terorganisir dan efisien.

Dalam konteks penelitian ini, *GitHub* digunakan sebagai media penyimpanan terpusat untuk mendokumentasikan perancangan dan menyimpan kode pengembangan sistem *IOT* penyiraman dan pengkabutan otomatis tanaman cabai. Pendekatan ini tidak hanya mendukung keteraturan pengembangan, tetapi juga mempermudah pengawasan, pemeliharaan, dan pembaruan sistem secara berkala, terutama saat melibatkan lebih dari satu pengembang atau saat sistem berkembang di masa mendatang.

2.2.16 Arduino

Arduino ditemukan oleh Hernando Barragan pada tahun 2003, pada saat ia membangun sistem *wiring*. *Arduino* memanfaatkan bahasa pemrogramannya sendiri, yang mirip dengan C++. Terdapat tiga bagian penting pada bahasa pemrograman *Arduino* yaitu *value*, *function* dan *structure*. *Function* memungkinkan dalam mengendalikan *board*. Dengan memakai *function* operasi matematika, analisis data dan tugas lainnya bisa dilakukan. *Value* berfungsi mewakili variabel, konstanta dan tipe data yang digunakan seperti *array*, *boolean*, *char*, *float*, dan lainnya. (Togi, 2021).

```

1 // Header
2 #include <library.h>
3
4 // Deklarasi Variabel Global
5
6 // Setup Function
7 void setup() {
8     // Inisialisasi pin, variabel, atau kondisi awal
9 }
10
11 // Loop Function
12 void loop() {
13     // Kode yang akan dijalankan secara berulang
14 }
15

```

Gambar 2. 11 Bahasa Pemrograman *Arduino*

Sumber: (Elga Aris Prastyo, 2025)

Berikut ini adalah deskripsi dari setiap bagian yang terdapat pada bagian penyusun program di atas:

1. Header

Bagian ini digunakan untuk menyertakan *library* yang digunakan dalam program. *Library* merupakan kumpulan fungsi yang sudah dibuat sebelumnya untuk memudahkan pemrograman, seperti mengakses sensor atau modul tertentu.

2. Deklarasi Variabel Global

Pada bagian ini, variabel yang akan diterapkan di semua bagian program baik di dalam *setup()* maupun *loop()* dideklarasikan. Variabel global memungkinkan penyimpanan data yang dapat diakses dari berbagai fungsi. Hal ini berguna untuk menyimpan nilai yang harus tetap tersedia selama program berjalan.

3. Setup Function

Fungsi *setup()* merupakan fungsi istimewa yang hanya dijalankan sekali ketika program *Arduino* pertama kali berjalan. Fungsi ini dipakai untuk melakukan inisialisasi, contohnya mengatur pin, menginisialisasi variabel, atau menyiapkan kondisi semula.

4. Loop Function

Fungsi *loop()* merupakan bagian inti dari program *Arduino* yang berjalan terus-menerus setelah fungsi *setup()* selesai. Di dalam fungsi ini, perintah-perintah yang ingin dijalankan secara berulang diletakkan, seperti pembacaan sensor, pengendalian aktuator, atau logika pemrosesan data.

61 2.2.17 *Arduino IDE*

Arduino IDE (Integrated Development Environment) software resmi yang biasanya digunakan untuk mengedit, menulis, dan mengunggah program ke papan mikrokontroler. IDE ini mempunyai interface yang simple dan mudah digunakan.¹⁹¹ Terdapat dua versi utama *Arduino IDE* saat ini, yaitu *Arduino IDE* 1.x dan *Arduino IDE* 2.x. Versi 2.x merupakan pengembangan besar dari versi sebelumnya, dengan peningkatan kecepatan, antarmuka pengguna yang lebih modern, serta dukungan fitur lanjutan seperti *auto-completion* navigasi cepat, dan debugger bawaan. (Liam Aljundi, 2024).

```

1 // Header
2 #include <library.h>
3
4 // Deklarasi Variabel Global
5
6 // Setup Function
7 void setup() {
8     // Inisialisasi pin, variabel, atau kondisi awal
9 }
10
11 // Loop Function
12 void loop() {
13     // Kode yang akan dijalankan secara berulang
14 }
15

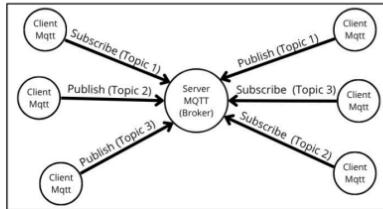
```

Gambar 2. 12 *Arduino IDE*

Sumber: (Penulis, 2025)

147 2.2.18 MQTT

MQTT (Message Queuing Telemetry Transport) merupakan protokol *subscribe-publish* yang dibuat hanya untuk mengirimkan *message* antara perangkat *IOT* dengan *overhead* yang rendah dan konsumsi energi yang efisien. Dalam era *Internet of Things (IOT)* yang cepat berkembang, di mana jutaan perangkat terhubung dan bertukar data, dibutuhkan protokol komunikasi yang ringan dan efisien (Indobot Academy, 2023).

Gambar 2. 13 Arsitektur *MQTT*

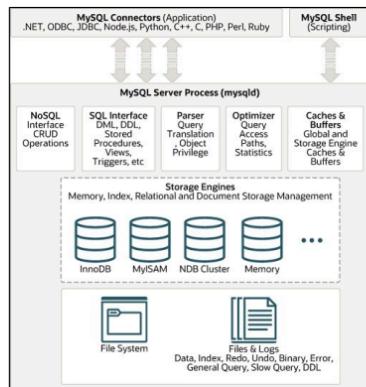
Sumber: (Valerie Lampkin, 2012)

Gambar di atas menggambarkan konsep dan cara kerja protokol *MQTT (Message Queuing Telemetry Transport)* yang berbasis arsitektur *publish-subscribe*. Dalam sistem ini, terdapat dua komponen utama, yaitu *client MQTT*

dan *server MQTT (broker)*. Setiap *client* dapat berperan sebagai *publisher* yang mengirimkan pesan ke suatu *topic*, atau sebagai *subscriber* yang menerima pesan dari *topic* tertentu yang telah didaftarkan. *Broker MQTT* bertindak sebagai perantara yang menerima pesan dari *publisher* dan meneruskannya ke semua *client* yang melakukan *subscribe* pada *topic* yang relevan. Sebagai contoh pada gambar, terdapat *client* yang melakukan *publish* pada *Topic 1*, dan *broker* akan mengirimkan pesan tersebut kepada *client* lain yang telah *subscribe* ke *Topic 1*. Mekanisme ini memastikan komunikasi yang efisien dan terdistribusi dengan baik antar perangkat *IOT* tanpa perlu mengetahui alamat satu sama lain. Dalam konteks penelitian ini, *MQTT* digunakan untuk menghubungkan perangkat sensor dan aktuator di *Greenhouse* dengan *server* dan antarmuka web.

⁹⁶ 2.2.19 MySQL

MySQL merupakan sebuah sistem pengelolaan basis data yang berlandaskan pada *SQL (Structured Query Language)*, memberikan kemampuan bagi pengguna untuk menyimpan, mengatur, dan mengambil data. Sebagai sebuah perangkat lunak yang bersifat sumber terbuka, *MySQL* memberikan kebebasan kepada para pengembang untuk melakukan modifikasi sesuai kebutuhan (Yazid Yusuf, 2024).



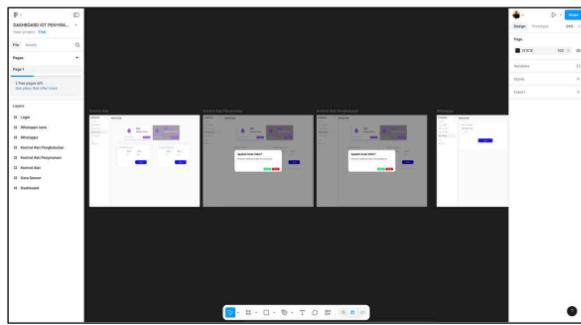
Gambar 2. 14 Arsitektur *MySQL*

Sumber: (MySQLTutorial.org, 2008)

Gambar di atas menjelaskan arsitektur dari *MySQL* menyediakan *MySQL Connectors* yang support berbagai bahasa program seperti *Python*, *JavaScript* dan sebagainya, memungkinkan integrasi lintas platform. *MySQL Server Process* (*mysqld*) menjadi komponen inti yang mengelola berbagai fungsi utama seperti antarmuka *SQL* (*DML*, *DDL*, prosedur tersimpan, *view*, *trigger*), *parser* untuk penerjemahan *query*, serta *optimizer* untuk menentukan jalur akses terbaik ke data. Selain itu, terdapat *caches* & *buffers* yang bertugas mempercepat pemrosesan data melalui penyimpanan sementara. Di bagian bawah, arsitektur ini menunjukkan beragam *storage engine* seperti InnoDB, MyISAM, NDB Cluster, dan *Memory*, yang berperan dalam manajemen penyimpanan data. Semua data dan *log* terkait aktivitas basis data disimpan dalam file sistem, termasuk data transaksi, *redo log*, dan *error log*. Dalam konteks penelitian ini, *MySQL* digunakan sebagai media penyimpanan data sensor suhu, kelembapan tanah, dan kelembapan udara yang diperoleh dari perangkat *IOT* di *Greenhouse*.

2.2.20 *Figma*

Figma adalah rangkaian produk yang memungkinkan Anda membuat, berbagi, dan menguji desain, presentasi, dan papan tulis. Anda bisa bekerja dengan orang lain secara *real-time*, mengakses fitur-fitur canggih dalam *Mode Dev*, dan berintegrasi dengan alat populer seperti *GitHub* (Figma, 2025).



Gambar 2. 15 Tampilan *Figma*

Sumber: (Penulis, 2025)

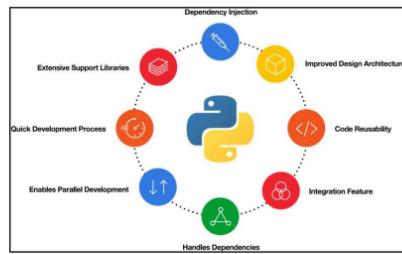
Figma merupakan platform desain antarmuka berbasis *cloud* yang mendukung kolaborasi secara *real-time*. Aplikasi ini memungkinkan tim desain dan pengembang bekerja bersama dalam satu ruang kerja digital, tanpa perlu menginstal perangkat lunak tambahan. Sebagaimana ditampilkan pada Gambar, antarmuka *Figma* terdiri dari panel *layer* di sisi kiri, area kerja utama di tengah, dan panel properti di sebelah kanan. Tampilan tersebut memperlihatkan fitur *horizontal layer scrolling* yang membantu pengguna menavigasi *layer* yang kompleks secara lebih mudah dan efisien.

Keunggulan utama *Figma* yang membuatnya populer dalam dunia desain digital antara lain:

- a. *Real-time functionality*: Setiap perubahan yang dilakukan akan langsung disimpan otomatis dan dapat dilihat oleh seluruh anggota tim secara serentak, memungkinkan kolaborasi yang dinamis.
- b. *Integrated prototyping*: *Figma* menyediakan fitur untuk membuat prototipe interaktif yang dapat langsung diuji pada perangkat target, tanpa perlu berpindah ke aplikasi lain.
- c. *Design library*: Fitur ini memungkinkan pengguna untuk membuat dan menggunakan kembali komponen, gaya, serta aset desain dalam berbagai proyek, sehingga meningkatkan konsistensi desain.
- d. *Easy sharing*: Projek yang dibuat di *Figma* dapat dengan mudah dibagikan melalui tautan kepada anggota tim (Nadifa Padantya Raihanah, 2023).

¹² 2.2.21 Python

Python merupakan bahasa pemrograman yang berorientasi objek, diinterpretasikan. Data bawaan tingkat tinggi struktur, pengikatan dinamis, dikombinasikan dengan pengetikan dinamis dan membuatnya sangat menarik untuk pengembangan aplikasi secara cepat, serta untuk digunakan sebagai bahasa skrip untuk menghubungkan komponen yang ada bersama. Sintaks *Python* yang sederhana dan mudah dipelajari menekankan keterbacaan dan karenanya mengurangi biaya pemeliharaan program. *Python* mendukung modul dan paket, yang mendorong program modularitas dan penggunaan kembali kode. (Python.org, 2025).



Gambar 2. 16 Arsitektur Python

Sumber: (Web Idea Solution, 2025)

Pada gambar di atas, *Python* ditampilkan sebagai inti dari berbagai fitur utama yang membuatnya unggul dan relevan dalam pengembangan sistem modern, termasuk untuk penelitian ini. Di antaranya adalah:

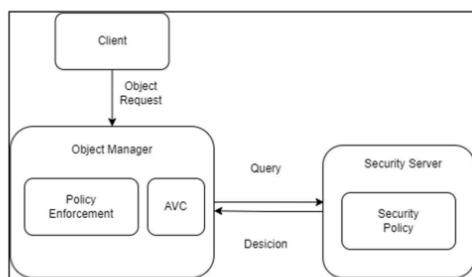
- Dependency Injection*: Mempermudah pengelolaan dependensi antar komponen, sehingga lebih fleksibel dan mudah diuji.
- Improved Design Architecture*: Mendukung pengembangan arsitektur sistem yang bersih dan terstruktur, termasuk pada sistem *IOT* berbasis sensor.
- Code Reusability*: *Python* memungkinkan penggunaan ulang kode secara luas melalui modul dan pustaka.
- Integration Feature*: *Python* sangat mudah diintegrasikan dengan berbagai sistem, protokol (seperti *MQTT*), dan *database*.
- Handles Dependencies*: Mengelola ketergantungan pustaka dengan mudah menggunakan tools seperti *pip* atau *venv*.
- Enables Parallel Development*: Mendukung pengembangan paralel antar tim melalui modul-modul terpisah.
- Quick Development Process*: Cepat dalam proses pengembangan karena sintaksnya ringkas dan komunitasnya luas.
- Extensive Support Libraries*: *Python* memiliki pustaka pendukung yang sangat lengkap.

Dalam konteks penelitian ini, *Python* menjadi komponen utama dalam membangun sistem *IOT* otomatis untuk penyiraman dan pengkabutan tanaman cabai. *Python* digunakan dalam proses integrasi data sensor (suhu, kelembapan

udara, dan tanah), penerapan algoritma *Naïve bayes* untuk pengambilan keputusan otomatis, serta pengembangan aplikasi web.

2.2.22 *Flask*

Flask adalah kerangka kerja aplikasi web yang ringan dan sesuai dengan standar *Web Server Gateway Interface (WSGI)*. Kerangka kerja ini dirancang untuk memudahkan pengembangan aplikasi secara cepat dan sederhana, serta memiliki fleksibilitas untuk dikembangkan menjadi aplikasi yang lebih kompleks (Pallets, 2025).



Gambar 2. 17 Arsitektur *Flask*

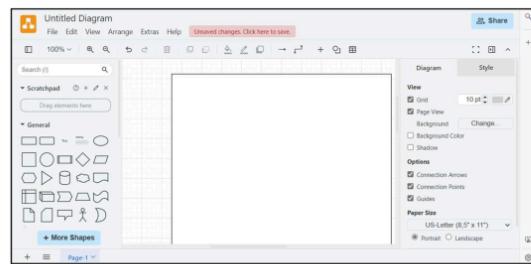
Sumber: (Ding et al., 2012)

Gambar di atas menggambarkan konsep kerja *Flask* dalam konteks keamanan dan otorisasi berbasis kebijakan. Saat seorang *client* mengirimkan permintaan akses terhadap objek, permintaan ini dikelola oleh *Object Manager* yang terdiri dari dua komponen utama: *Policy Enforcement* (untuk menegakkan kebijakan) dan *AVC* (*Access Vector Cache*), yang menangani kontrol akses. *Object Manager* kemudian mengajukan *query* ke *Security Server*, yang berisi komponen *Security Policy*. Berdasarkan kebijakan ini, *server* memberikan keputusan (*decision*) apakah permintaan diizinkan atau ditolak. Dalam konteks penelitian ini, *Flask* digunakan sebagai kerangka kerja utama untuk membangun aplikasi *monitoring* berbasis web. Aplikasi ini memungkinkan pengguna untuk melihat kondisi *real-time* dari *Greenhouse* melalui data sensor seperti suhu,

kelembaban udara, dan kelembaban tanah. *Flask* juga mengatur kontrol akses dan komunikasi antara antarmuka pengguna dengan *backend*, termasuk dalam proses pengambilan keputusan otomatis menggunakan algoritma *Naïve bayes*.

2.2.23 Draw.io

Draw.io adalah platform penggambaran grafik, *flowcart*, *chartnetwork*, diagram dan lainnya. *Draw.io*, selain itu, menawarkan kapabilitas untuk membuat diagram langsung melalui web yang bekerja sama dengan *Google Drive* dan *Dropbox* untuk menyimpan proyeknya. *Draw.io* ditemukan atau didirikan pada tahun 2000 oleh Gaudenz Alderdi Norpathapton (Walid & Susanto, 2024).



Gambar 2. 18 *Draw.io*

Sumber: (Penulis, 2025)

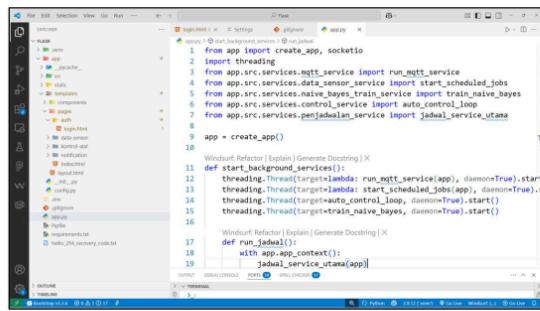
Platform *Draw.io* mendukung kolaborasi berbasis *cloud* karena terintegrasi dengan layanan penyimpanan seperti *Google Drive* dan *Dropbox*, sehingga memungkinkan penyimpanan dan akses proyek secara *online* secara *real-time*. Dalam konteks perancangan sistem *IOT*, kemampuan ini sangat bermanfaat untuk mendukung kerja tim lintas perangkat dan lokasi yang membutuhkan akses cepat terhadap diagram rancangan. *Draw.io* sangat berguna dalam merancang sistem karena menawarkan kemudahan penggunaan, antarmuka yang intuitif, serta kemampuan ekspor dalam berbagai format seperti PNG, PDF, dan XML yang mempermudah dokumentasi dan pelaporan.

Dengan fitur yang fleksibel dan lengkap tersebut, *Draw.io* dipilih sebagai alat bantu utama dalam merancang sistem *IOT* penyiraman dan pengabutan

otomatis tanaman cabai, karena mampu menggambarkan struktur, alur data, dan integrasi antar komponen dengan jelas dan efisien, sekaligus mendukung proses pengembangan secara terstruktur dan terdokumentasi.

82 2.2.24 Visual Studio Code

Merupakan sebuah kode editor yang dirancang ringkas. Editor ini mendukung beragam operasi pengembangan, seperti *debugging*, menjalankan tugas, serta kendali versi. Tujuannya adalah menyediakan perangkat yang dibutuhkan pengembang untuk siklus penulisan kode dan *debugging* yang efisien. Sementara itu, alur kerja yang lebih rumit diserahkan pada *IDE* yang lebih kaya fitur. (Microsoft, 2025).



```

1 from app import create_app, socketio
2 import threading
3 from app.src.services.mqtt_service import run_mqtt_service
4 from app.src.services.data_sensor_service import start_scheduled_jobs
5 from app.src.services.naive_bayes_train_service import train_naive_bayes
6 from app.src.services.control_service import auto_control_loop
7 from app.src.services.pengadalan_service import jadwal_service_utama
8
9 app = create_app()
10
11 def start_background_services():
12     threading.Thread(target=lambda: run_mqtt_service(app), daemon=True).start()
13     threading.Thread(target=lambda: start_scheduled_jobs(app), daemon=True).start()
14     threading.Thread(target=auto_control_loop, daemon=True).start()
15     threading.Thread(target=train_naive_bayes, daemon=True).start()
16
17 def run_jadwal():
18     with app.app_context():
19         jadwal_service_utama(app)

```

Gambar 2. 19 *Visual Studio Code*

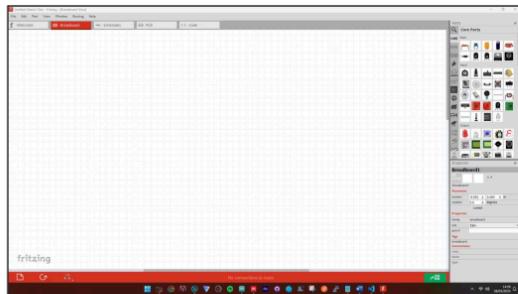
Sumber: (Penulis,2025)

Gambar di atas menampilkan antarmuka *VS Code* yang digunakan dalam menulis kode program untuk projek pembuatan aplikasi *IOT* untuk sistem penyiraman dan pengkabutan otomatis tanaman cabai di *Greenhouse*.

2.2.25 Fritzing

Fritzing adalah inisiatif perangkat keras sumber terbuka yang membuat elektronik dapat diakses sebagai materi kreatif bagi siapa saja. Menawarkan alat perangkat lunak, layanan dalam pemrosesan dan *Arduino* dan situs web

komunitas, menciptakan lingkungan inovatif yang membuat penggunanya mendokumentasikan tahapan pembuatan purwarupa mereka. (Fritzing, 2025).



Gambar 2. 20 *Fritzing*

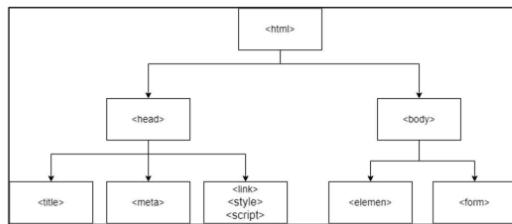
Sumber: (Penulis, 2025)

Dalam penelitian ini, *Fritzing* digunakan untuk merancang diagram *wiring* sistem *IOT* otomatis untuk penyiraman dan pengkabutan tanaman cabai di *Greenhouse*. Diagram *wiring* ini mencakup koneksi antara sensor suhu, kelembaban udara, dan kelembaban tanah dengan mikrokontroler, serta pengaturan aktuator seperti pompa dan *sprayer*. Dengan tampilan intuitif dan fitur visualisasi koneksi kabel yang jelas, *Fritzing* memudahkan dalam mendokumentasikan jalur koneksi secara akurat sebelum diimplementasikan secara fisik.

Fitur seperti *Breadboard View*, *Schematic View*, dan *PCB View* mendukung proses desain mulai dari prototipe awal hingga ke perancangan sirkuit cetak (*PCB*), menjadikan *Fritzing* alat penting dalam pengembangan sistem elektronik berbasis *IOT* yang terstruktur dan dapat direproduksi.

2.2.26 *HTML*

HTML (*HyperText Markup Language*) adalah bahasa yang digunakan untuk membuat struktur dasar halaman web. *HTML* terdiri dari elemen-elemen seperti tag, atribut, dan konten yang membentuk tampilan dan susunan informasi di web (Publikasi et al., 2025).

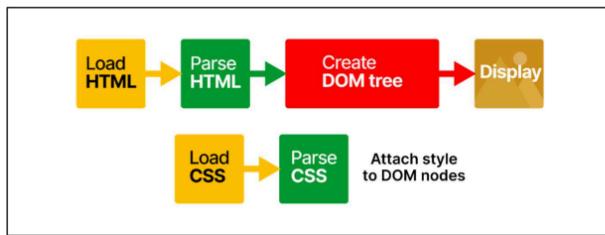
Gambar 2. 21 Arsitektur *HTML*

Sumber: (Ade Roni, 2025)

Gambar di atas menunjukkan struktur hierarki dasar dokumen *HTML*. Elemen *<html>* berfungsi sebagai elemen utama yang membungkus seluruh isi dokumen. Di dalamnya terdapat dua bagian utama, yaitu *<head>* dan *<body>*. Bagian *<head>* berisi elemen-elemen yang bersifat metadata dan tidak langsung ditampilkan di halaman web, seperti *<title>* (judul halaman), *<meta>* (informasi *metadata* seperti *charset* dan deskripsi), serta elemen-elemen tambahan seperti *<link>* (menghubungkan file eksternal seperti *CSS*), *<style>* (gaya *CSS* internal), dan *<script>* (kode *JavaScript*). Sementara itu, bagian *<body>* memuat konten utama yang akan ditampilkan di halaman web, seperti elemen teks, gambar, dan form interaktif. Gambar ini menggambarkan bahwa struktur *HTML* bekerja secara hierarkis dan terstruktur, dimulai dari tag utama hingga elemen-elemen turunan yang membentuk antarmuka pengguna serta logika interaksi dalam sebuah halaman web. Struktur ini penting untuk memastikan halaman web dapat dibaca dan ditampilkan dengan benar oleh browser.

2.2.27 CSS

³⁸ Berfungsi untuk menata dan memperindah pada tampilan web. *CSS* dapat membuat pengembang web mengontrol gaya visual dari elemen-elemen *HTML* seperti warna latar belakang, ukuran dan jenis *font*, spasi antar elemen, serta tata letak keseluruhan halaman. Dengan *CSS*, desain antarmuka web menjadi lebih konsisten, menarik, dan responsif terhadap berbagai ukuran layar perangkat (Publikasi et al., 2025).



Gambar 2. 22 Arsitektur CSS

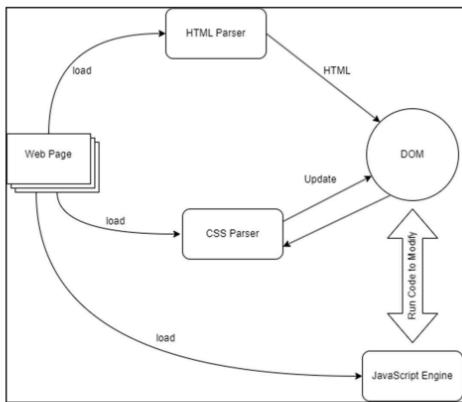
Sumber: (Mozilla Foundation., 2025)

Gambar di atas menjelaskan alur kerja browser saat merender halaman web. Proses diawali dengan pemuatan (*load*) dan penguraian (*parse*) file *HTML* untuk membentuk struktur *DOM* (*Document Object Model*) *tree*. Secara paralel, browser juga memuat dan mengurai file *CSS* untuk membentuk *CSSOM* (*CSS Object Model*). Kombinasi antara *DOM* dan *CSSOM* ini kemudian digunakan oleh browser untuk merender dan menampilkan antarmuka halaman web secara visual kepada pengguna.

Dalam konteks penelitian ini, yaitu merancang dan membangun aplikasi *IOT* untuk penyiraman dan pengkabutan otomatis tanaman cabai di *Greenhouse*, *CSS* berperan penting dalam membentuk tampilan antarmuka dari aplikasi monitoring berbasis web. Aplikasi ini memungkinkan pengguna memantau data sensor suhu, ¹⁸³ kelembapan tanah dan udara secara *real-time*, serta mengontrol sistem penyiraman dan pengkabutan. *CSS* memastikan bahwa informasi yang disajikan dapat ditampilkan secara responsif, mudah dipahami, dan menarik secara visual baik di perangkat *desktop* maupun *mobile*. Dengan tampilan yang baik, pengguna dapat mengambil keputusan dengan cepat dan efisien berdasarkan data lingkungan yang disajikan.

2.2.28 JavaScript

JavaScript adalah bahasa skrip lintas platform berorientasi objek yang digunakan untuk membuat halaman web interaktif misalnya, memiliki animasi, tombol yang dapat diklik, menu *popup*, dll (Mozilla Foundation, 2025).



Gambar 2. 23 Arsitektur *JavaScript*

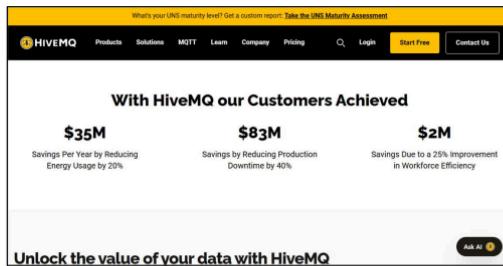
Sumber: (Rifqi Mulyawan Digital, 2025)

Gambar di atas menunjukkan bagaimana *JavaScript* bekerja dalam proses rendering halaman web di browser. Setelah halaman web dimuat, konten *HTML* dikirim ke *HTML Parser* untuk membentuk struktur *DOM* (*Document Object Model*). Sementara itu, file *CSS* juga dimuat dan diproses oleh *CSS Parser* untuk menentukan tampilan elemen-elemen pada *DOM*. Di sisi lain, *JavaScript Engine* memuat skrip *JavaScript* yang dapat berjalan secara asinkron dan digunakan untuk memanipulasi atau memperbarui struktur *DOM* secara dinamis. Proses ini memungkinkan interaksi waktu nyata, seperti pembaruan nilai sensor atau tampilan kondisi tanpa perlu *reload* halaman.

Dalam konteks penelitian ini, *JavaScript* sangat penting dalam pengembangan aplikasi *monitoring* berbasis web untuk sistem penyiraman dan pengkabutan otomatis tanaman cabai di *Greenhouse*. Melalui *JavaScript*, antarmuka aplikasi dapat menampilkan data sensor suhu, kelembapan udara, dan tanah secara ¹⁵⁵ *real-time*, serta memungkinkan pengguna mengontrol sistem dari jarak jauh. *JavaScript* juga bekerja sama dengan *CSS* dan *HTML* untuk memberikan respons visual instan saat terjadi perubahan kondisi lingkungan, serta menjalankan logika interaktif seperti aktivasi penyiraman berbasis hasil prediksi dari algoritma *Naïve bayes*. Dengan begitu, *JavaScript* mendukung efisiensi dan kepraktisan dalam pengawasan dan pengendalian *Greenhouse* secara otomatis.

2.2.29 *HiveMQ*

HiveMQ adalah platform *MQTT* kelas dunia yang siap digunakan di lingkungan perusahaan. Platform ini dibuat dengan tujuan mentransfer data secara cepat dan efektif, dari perangkat-perangkat *IOT* ke sistem di perusahaan. antara perangkat *IOT* dan sistem perusahaan. *HiveMQ* sepenuhnya mendukung protokol *MQTT*, memungkinkan komunikasi dua arah secara instan dan *real-time*, sehingga mendukung otomatisasi, analitik, dan pemrosesan data berbasis *AI*.



Gambar 2. 24 Tampilan *HiveMQ*

Sumber: (Penulis, 2025)

HiveMQ dipilih sebagai *broker MQTT* dalam sistem ini karena menawarkan komunikasi data yang cepat, ringan, dan andal antara perangkat *IOT* (seperti *ESP32*) dengan *server*. *HiveMQ* dirancang khusus untuk mendukung komunikasi *real-time* pada skala industri, serta kompatibel sepenuhnya dengan protokol *MQTT*. Platform ini memungkinkan pertukaran data dua arah secara instan, sehingga sangat sesuai untuk sistem otomatisasi seperti penyiraman dan pengabutan tanaman yang memerlukan respon cepat berdasarkan data sensor.

2.2.30 *WhatsApp-web.js*

WhatsApp-web.js adalah sebuah *library Node.js* yang memungkinkan pengembang untuk mengintegrasikan dan mengontrol *WhatsApp* melalui *WhatsApp Web* secara otomatis. *Library* ini bekerja dengan cara meluncurkan aplikasi *WhatsApp Web* di dalam browser yang dikendalikan oleh *Puppeteer*,

sebuah pustaka pengendali *headless browser*. Melalui pendekatan ini, *WhatsApp-web.js* dapat mengakses berbagai fungsi internal *WhatsApp Web*, sehingga memungkinkan pengguna untuk mengirim dan menerima pesan, membaca status, mengelola grup, dan fitur lainnya, hampir sama seperti menggunakan *WhatsApp Web* secara manual. Karena *library* ini memanfaatkan antarmuka resmi *WhatsApp Web*, risiko pemblokiran menjadi lebih rendah dibandingkan pendekatan tidak resmi lainnya.



Gambar 2. 25 *WhatsApp-web.js*

Sumber: (Pengguna, 2025)

2.2.31 *Black Box Testing*

Biasanya dikenal sebagai *functional testing* atau *specification-based testing*, adalah sebuah metode pengujian yang berfokus pada pengujian fungsi sistem tanpa perlu mengetahui struktur internal ataupun kode programnya (Nur Ichsanudin et al., 2022).

2.2.32 ²⁶ *Cross Validation*

Cross Validation merupakan teknik untuk memvalidasi keakuratan sebuah model yang dibangun berdasarkan data set tertentu. Data yang digunakan dalam proses pembentukan model disebut sebagai data latih atau training dan data yang digunakan untuk memvalidasi model disebut sebagai data *testing* (Ayu Made Supartini et al., 2017).

2.2.33 Accuracy Score

Accuracy Score merupakan metrik memiliki fungsi untuk menilai ketepatan sebuah model dalam melakukan prediksi. Metrik ini menghitung persentase jumlah prediksi yang benar dibandingkan dengan total keseluruhan data yang diuji. Dengan kata lain, *accuracy* menunjukkan seberapa sering model memberikan hasil yang sesuai dengan kenyataan (Hidayat et al., 2021).

2.2.34 K-Fold Cross Validation

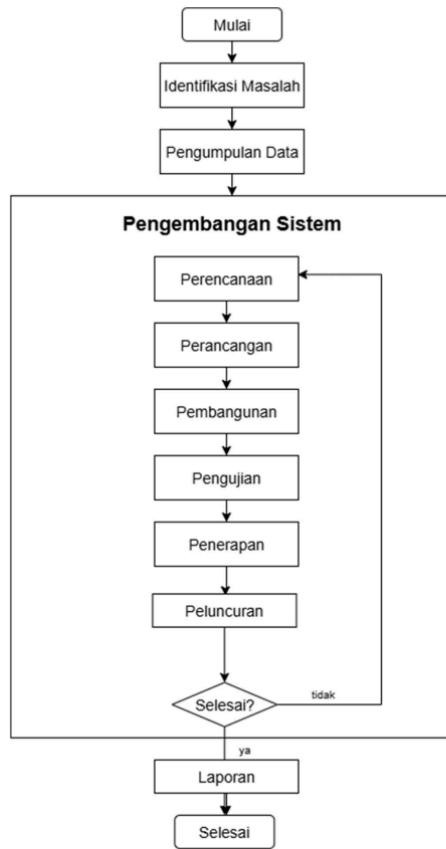
7 *K-Fold CV* merupakan himpunan data yang diberikan lalu dibagi menjadi beberapa bagian sejumlah K bagian yang disebut *Fold* di mana setiap *Fold* digunakan sebagai set pengujian di beberapa titik. Mari kita ambil skenario validasi silang *5-Fold* ($K = 5$). Kumpulan data dibagi menjadi *5 Fold*. Pada iterasi pertama, *Fold* pertama berfungsi untuk menguji model dan bagian lainnya digunakan untuk melatih model. Pada iterasi kedua, *Fold* kedua berfungsi untuk set pengujian sementara bagian lainnya berfungsi sebagai set pelatihan. Proses ini diulangi sampai setiap *Fold* dari *5 Fold* telah digunakan sebagai set pengujian (Peryanto et al., 2020).⁹⁰

BAB III

METODOLOGI PENELITIAN

3.1 Kerangka Pikir

Adalah alur pemikiran yang dibangun berdasarkan aktivitas penyusunnya.
Berikut adalah rangkaian berpikir yang menguraikan langkah-langkah yang akan ditempuh dalam penelitian ini:



Gambar 3. 1 Kerangka Pikir

3.2 Deskripsi

Penelitian ini mengadopsi metodologi *Agile* dalam proses pembangunan sistem. Berikut adalah urutan langkah-langkah penelitian yang ditempuh:

3.2.1 Identifikasi Masalah

Merupakan usaha untuk menguraikan permasalahan. Identifikasi ini dilakukan sebagai langkah awal riset. Dimulai dengan meminta izin kepada pemilik Avicenna *Greenhouse* untuk melakukan penelitian kemudian melaksanakan pengamatan langsung terhadap objek penelitian.

3.2.2 ¹⁹⁶ Pengumpulan Data

Supaya memperoleh data yang akurat dan sesuai, penulis mengaplikasikan beragam teknik pengumpulan yang diperlukan dalam implementasi alat *Internet of Things*, penyusunan proposal, dan penyusunan laporan, yaitu sebagai berikut:

1. Observasi

Penulis melakukan pengamatan langsung di Avicenna *Greenhouse* untuk memahami kondisi dan kebutuhan penyiraman tanaman secara manual.

2. Wawancara

Penulis melakukan sesi tanya jawab dengan pemilik Avicenna *Greenhouse* guna memperoleh informasi terkait metode penyiraman yang digunakan, kendala dalam penyiraman dan pengkabutan tanaman, serta harapan terhadap sistem penyiraman otomatis berbasis *Internet of Things*.

3. Studi Pustaka

Penulis mengumpulkan serta menelaah data dari berbagai jurnal ilmiah, buku, laman web dan sumber lainnya yang sesuai dengan topik *Internet of Things*, sistem penyiraman otomatis, dan pengkabutan pada tanaman.

3.2.3 Perencanaan

Setelah melakukan pengumpulan data, penulis melakukan tahap perencanaan sistem. Tahapan ini bertujuan untuk menetapkan kebutuhan sistem serta

merancang alur pengembangan yang akan dilaksanakan dalam beberapa iterasi (*sprint*). Perencanaan meliputi identifikasi kebutuhan sistem, baik dari sisi fungsional maupun *non-fungsional*, agar proses implementasi dapat berjalan sesuai dengan tujuan penelitian.

1. Analisis kebutuhan fungsional

Pada analisis kebutuhan fungsi yaitu merupakan proses mengidentifikasi fitur utama dapat diterapkan oleh sistem berdasarkan fungsionalitas yang diharapkan.

Ada juga kebutuhan fungsional sistem yang akan dibangun adalah sebagai berikut:

a. Autentikasi Pengguna

- 1) Sistem menyediakan *form Login* untuk *admin*.
- 2) Sistem memverifikasi kredensial pengguna sebelum memberikan akses ke halaman utama.

b. Dashboard Admin

- 1) Menampilkan data dari sensor suhu, kelembapan tanah, dan kelembapan udara.
- 2) Menampilkan status perangkat penyiram dan pengkabut.

c. Integrasi IOT dan Proses Pengambilan Keputusan

- 1) Sistem menggunakan algoritma *Naive Bayes* untuk menentukan kapan melakukan penyiraman dan pengkabutan.

d. Pengendalian Perangkat

- 1) Sistem menerima data dari *ESP32* melalui protokol *MQTT*.
- 2) Sistem mengirim perintah ke *ESP32* untuk mengontrol perangkat.

e. Notifikasi dan Monitoring

- 1) *Admin* dapat mengaktifkan/menonaktifkan penyiraman dan pengkabutan secara manual melalui notifikasi via *WhatsApp*.

136 2. Kebutuhan Non-Fungsional

Analisis kebutuhan non-fungsional mencakup spesifikasi teknis dan kualitas sistem yang memengaruhi performa dan keandalan sistem secara keseluruhan. Kebutuhan ini tidak secara langsung berkaitan dengan fungsi sistem, namun sangat penting untuk menjamin bahwa sistem berjalan dengan efisien dan aman.

Adapun kebutuhan non-fungsional dalam sistem ini meliputi:

⁴⁴
1) Perangkat Keras

Tabel 3. 1 Perangkat Keras

<i>Processor</i>	Intel Dual-Core 3.10GHz
<i>Storage Capacity</i>	500 GB
<i>RAM Size</i>	4 GB
Monitor Resolution	Resolusi 1280 x 800px

¹⁷⁵
2) Perangkat *Internet of Things*

Tabel 3. 2 Perangkat *Internet of Things*

Mikrokontroler	¹¹⁹ <i>ESP32</i>
Sensor Suhu dan Kelembapan Udara	<i>DHT11</i>
Sensor Kelembapan Tanah	<i>Soil moisture Sensor</i>
Pompa Air	<i>Mini water pump DC 3–6V & Sanyo</i>
Pengkabutan	<i>Nozzle Sprayer</i>
<i>Relay Module</i>	2 Channel Relay 5V
Power Supply & Adaptor	Adapter 5V/2A
Selang	Selang plastik PE (<i>Polyethylene</i>)
Internet	<i>Hostpot HP</i>
Wadah Alat	<i>Box Casing</i>

3) Perangkat Lunak

Tabel 3. 3 Perangkat Lunak

Sistem Operasi	Window 11 / Ubuntu 22.04
Bahasa Pemrograman	<i>Python, Arduino</i>
<i>Backend dan Frontend</i>	<i>Flask</i>
<i>Database</i>	<i>MySQL</i>
<i>Broker MQTT</i>	<i>HiveMQ</i>
Editor Kode	<i>Visual Studio Code, Arduino IDE</i>
Diagram & Wireframe	<i>Draw.io</i>

<i>User Interface</i>	<i>Figma</i>
<i>Wiring</i>	<i>Fritzing</i>
<i>Laporan</i>	<i>Microsoft Word</i>
<i>WhatsApp</i>	<i>WhatsApp-web.js</i>

3.2.4 Perancangan

1. Perancangan Diagram

a. *Use Case Diagram*

Diagram ini memvisualisasikan keterhubungan antara aktor dan fungsionalitas sistem. Aktor dalam sistem ini adalah *Admin*, yang memiliki hak akses penuh terhadap pengelolaan data dan pengendalian perangkat.

- 1) Melakukan *Login* ke *Dashboard*
- 2) Melihat data sensor secara *real-time*
- 3) Melihat hasil prediksi sistem (penyiraman/pengkabutan)
- 4) Mengaktifkan/nonaktifkan perangkat secara manual
- 5) Menerima notifikasi hasil prediksi dan mengirim link untuk akses *Dashboard* melalui *WhatsApp*.

b. *Activity Diagram*

Diagram aktivitas menggambarkan alur proses sistem mulai dari pengambilan data sensor hingga pengiriman hasil prediksi. Adapun alur aktivitas yang digambarkan meliputi:

- 1) Aktivitas *Login admin*
- 2) Aktivitas membaca data dari sensor
- 3) Aktivitas memproses data dengan algoritma *Naive Bayes*
- 4) Aktivitas mengirim hasil prediksi ke *WhatsApp*
- 5) Aktivitas untuk kontrol penyiraman dan pengkabutan

c. *Wiring Diagram*

Wiring Diagram (skema rangkaian) menunjukkan koneksi fisik antar komponen perangkat keras seperti sensor, *ESP32*, dan aktuator. Beberapa komponen seperti:

- 1) *ESP32* (mikrokontroler)
- 2) Sensor DHT11 (untuk suhu dan kelembapan udara)
- 3) Sensor kelembapan tanah
- 4) *Nozzle* kabut dan kipas
- 5) *Breadboard* dan kabel jumper

d. *Class Diagram*

Perancangan sistem menggunakan *Class Diagram* untuk menggambarkan struktur *kelas* dalam sistem serta hubungan antar *kelas*. *Class Diagram* membantu dalam proses implementasi sistem berbasis objek, terutama dalam pengembangan *backend* dan integrasi dengan sensor *IOT*. Berikut adalah penjelasan dari masing-masing *kelas*:

- 1) Data Sensor
 - 2) Riwayat Aksi
 - 3) Jadwal Penyiraman
 - 4) Hasil Prediksi
 2. Perancangan Antarmuka Pengguna
- Pada tahapan ini, desain antarmuka aplikasi dibuat menggunakan aplikasi *Figma* dengan desain antarmuka yang sederhana dan tentunya memperhatikan *user experience*.

3.2.5 Pembangunan

Tahap ini melibatkan proses penerjemahan seluruh desain sistem menjadi kode program sesuai dengan spesifikasi yang telah dirancang. Sistem yang dibangun dengan memanfaatkan bahasa pemrograman *Python* dengan *framework Flask* sebagai web *framework*.

Komunikasi antara perangkat *IOT* dan *server* dilakukan menggunakan protokol *MQTT* untuk memastikan pengiriman data sensor dan perintah kontrol secara efisien dan *real-time*.

Pada sisi klien, aplikasi antarmuka pengguna dikembangkan dengan *Flask* untuk menampilkan data sensor dan kontrol perangkat. Sistem ini juga menggunakan *MySQL* sebagai basis data untuk menyimpan data *historis* sensor dan *log* kontrol perangkat.

3.2.6 Pengujian

Tahap pengujian ddilaksanakan guna memastikan seluruh fitur dalam sistem berjalan sesuai dengan fungsi dan bebas dari kesalahan logika maupun bug yang signifikan. Pengujian dilakukan menggunakan metode *black box testing*, yaitu dengan menguji setiap fitur sistem berdasarkan input dan output tanpa melihat langsung struktur kode program.⁵⁹⁸³

Selain pengujian fungsional, pengujian juga dilakukan terhadap model *Machine Learning Naive Bayes* yang digunakan untuk memprediksi kebutuhan penyiraman dan pengkabutan. Evaluasi dilakukan dengan menghitung akurasi model terhadap data latih menggunakan *accuracy_score*. Hasil akurasi menunjukkan bahwa model mampu mempelajari pola data suhu udara, kelembaban tanah dan udara dengan baik.

Untuk menguji keandalan model secara umum, dilakukan juga pengujian menggunakan teknik *cross-validation*, yaitu dengan memisahkan dataset ke dalam beberapa subset dan menguji serta melatih model secara bergiliran pada setiap bagian data. Teknik ini membantu mengukur stabilitas dan kemampuan generalisasi model terhadap data baru. Hasil pengujian menunjukkan bahwa model memiliki performa yang konsisten dan dapat digunakan secara efektif dalam sistem monitoring otomatis ini.¹⁴¹

3.2.7 Penerapan

Tahap penerapan adalah proses penempatan sistem ke lingkungan sebenarnya, yaitu dengan melakukan *deploy* aplikasi ke *server* dan menghubungkannya dengan perangkat *IOT* di lapangan. Pada tahap ini, sistem yang telah dikembangkan mulai dijalankan dan dikonfigurasikan agar dapat berfungsi sesuai dengan kondisi operasional.

Proses penerapan mencakup pengunggahan kode program ke *server*, pengaturan koneksi *MQTT*, integrasi dengan basis data, serta pengujian awal untuk memastikan bahwa sistem dapat berjalan dengan baik. Penerapan dilakukan secara bertahap sesuai dengan pendekatan metode *Agile*, di mana setiap bagian sistem yang telah selesai dapat langsung diuji di lapangan.

3.2.8 Peluncuran

Setelah sistem dinyatakan layak berdasarkan hasil evaluasi, maka dilakukan tahap peluncuran. Tahap ini menandai bahwa sistem telah siap digunakan secara penuh oleh pengguna di lapangan.

Peluncuran dilakukan dengan memberikan akses ke seluruh fitur sistem, menyosialisasikan cara penggunaannya, serta Memverifikasi bahwa setiap bagian dari *hardware* dan *software* dapat beroperasi dengan benar.

3.2.9 Pembuatan Laporan

Tahapan terakhir adalah penyusunan laporan hasil penelitian yang menguraikan secara detail tahapan-tahapan yang dijalankan dalam penelitian dan hasil yang di peroleh. Laporan ini bertujuan untuk secara sistematis menyampaikan informasi kepada pembaca tentang isi dari penelitian yang telah dilakukan.¹⁷⁷

BAB IV

ANALISIS DAN PERANCANGAN

4.1 Analisis

Analisis dilakukan sebagai langkah awal penelitian untuk mengidentifikasi kebutuhan-kebutuhan yang diperlukan dalam merancang sistem. Berikut adalah analisis yang telah penulis lakukan untuk membantu dalam perancangan dan pembangunan sistem penyiraman dan pengkabutan ⁹⁵ otomatis pada tanaman berbasis *Internet of Things (IOT)*. Analisis ini mencakup pembahasan mengenai analisis masalah, analisis perangkat keras dan perangkat lunak.

4.1.1 Analisis Masalah

Langkah pertama pada tahap ini adalah mengidentifikasi permasalahan yang terjadi di lingkungan Avicenna *Greenhouse* terkait efisiensi dan efektivitas dalam proses penyiraman dan pengkabutan tanaman. Proses dilakukan dengan mengamati kondisi lapangan, di mana kegiatan penyiraman dan pengkabutan masih dilakukan secara manual tanpa memperhatikan parameter lingkungan secara *real-time*, seperti suhu, kelembapan udara, dan kelembapan tanah. Hal ini menyebabkan ketidak tepatan waktu dalam pemberian air atau kabut, yang bisa berdampak pada kualitas pertumbuhan pada tanaman. Dan Selain itu, tidak adanya sistem monitoring dan kontrol jarak jauh menyulitkan pengguna untuk melakukan pengawasan dan pengambilan keputusan secara cepat.

4.1.2 Analisis Hardware

Untuk membangun sistem penyiraman dan pengkabutan otomatis pada tanaman di *Greenhouse*, diperlukan perangkat keras yang dapat mendeteksi kondisi lingkungan, mengontrol aktuator, serta mengirimkan data secara *real-time* ke *server*. Komponen-komponen ini saling terintegrasi melalui jaringan internet dan dikendalikan oleh sistem berbasis web. Berikut adalah perangkat keras yang digunakan dalam sistem:

- 1) ²⁴ Sensor *DHT11*, berfungsi untuk membaca suhu dan kelembapan udara di dalam *Greenhouse* secara berkala.
- 2) *Soil moisture Sensor*, berfungsi untuk mengukur kelembapan tanah guna menentukan tingkat kebutuhan air pada tanaman.
- 3) *Mini Water Pump DC 3–6V & Sanyo*, digunakan untuk mengalirkan air ke tanaman saat sistem memutuskan untuk melakukan penyiraman.
- 4) *Nozzle Sprayer*, digunakan untuk menghasilkan kabut air.
- 5) ²⁰ *Relay Module 2 Channel 5V*, berfungsi sebagai saklar elektronik untuk mengontrol on/off pompa air dan sprayer secara otomatis.
- 6) Adapter 5V/2A, menyediakan catu daya utama bagi mikrokontroler dan komponen lainnya.
- 7) Selang Plastik PE (*Polyethylene*), digunakan untuk menyalurkan air dari pompa ke tanaman atau *Nozzle Sprayer*.
- 8) *Hotspot HP*, digunakan sebagai sumber koneksi internet agar *ESP32* dapat terhubung dengan *server* dan mengirim data secara real-time.
- 9) *Box Casing*, berfungsi sebagai wadah pelindung untuk seluruh rangkaian elektronik agar aman dari air dan gangguan luar.

4.1.3 Analisis Software

Untuk mendukung pembangunan aplikasi penyiraman dan pengabutan berbasis ¹⁷⁸ *Internet of Things (IOT)*, diperlukan sejumlah *software yang* yang membantu dalam proses pemrograman, pengujian, pengambilan keputusan, dan pengelolaan data. Berikut ini ³ Adalah beberapa *software yang digunakan* dalam sistem:

- 1) *Arduino IDE* berfungsi untuk memprogram mikrokontroler *ESP32* supaya dapat membaca data sensor, mengaktifkan *Relay*, serta mengirim dan menerima data melalui koneksi *WiFi*.
- 2) *Python* digunakan sebagai bahasa pemrograman untuk mengimplementasikan algoritma *Naive Bayes*.
- 3) *Flask framework backend* berbasis *Python* yang digunakan untuk membangun aplikasi web dan integrasi antara *server* dengan perangkat *IOT*.

- 4) *MySQL* sistem manajemen basis data relasional memiliki fungsi untuk menyimpan data sensor, hasil prediksi, jadwal penyiraman, dan riwayat aksi.
- 5) *MQTT* protokol komunikasi yang berfungsi untuk melakukan pertukaran data secara *real-time* antara *ESP32* dan *server*.
- 6) *WhatsApp* digunakan untuk mengirim notifikasi kepada pengguna terkait status penyiraman atau pengkabutan, baik otomatis maupun manual. Integrasi dilakukan menggunakan *library WhatsApp-web.js*, yang memungkinkan pengiriman pesan melalui antarmuka *WhatsApp Web* dengan bantuan *Puppeteer*. Pendekatan ini memungkinkan notifikasi *real-time* tanpa perlu akses *API* resmi *WhatsApp*.
- 7) *Visual Studio Code (VS Code)* sebagai editor kode utama yang berfungsi dalam pengembangan program *backend*, algoritma, dan antarmuka.
- 8) *Fritzing* alat bantu visual untuk merancang dan mendokumentasikan rangkaian perangkat keras *IOT*.
- 9) *Draw.io* digunakan untuk membuat berbagai diagram seperti *UML* yang mendukung dokumentasi sistem.
- 10) *Figma* platform desain antarmuka pengguna (*UI*) yang digunakan untuk merancang tampilan halaman *Dashboard* sistem monitoring dan kontrol.

4.1.4 Analisis Pengguna

Analisis pengguna dalam sistem penyiraman dan pengkabutan otomatis ini berfokus pada siapa saja yang akan menggunakan sistem serta bagaimana interaksinya terhadap fitur-fitur yang tersedia. Pengguna utama sistem adalah pengelola *Greenhouse* yang bertanggung jawab dalam pemantauan dan perawatan tanaman. Pengguna dapat mengakses *Dashboard* untuk melihat data sensor secara *real-time*, mengatur jadwal penyiraman dan pengkabutan, serta memantau riwayat aktivitas yang dilakukan sistem secara otomatis maupun manual. Selain itu, pengguna juga akan menerima notifikasi melalui *WhatsApp* sebagai bentuk pemberitahuan terkait aksi yang telah dilakukan oleh sistem. Meskipun sistem ini bersifat otomatis, pengguna tetap memiliki kendali penuh untuk menyesuaikan pengaturan sesuai kebutuhan. Oleh karena itu, antarmuka sistem dirancang

dengan tampilan yang sederhana, informatif, dan mudah dioperasikan guna mendukung kenyamanan, keandalan, serta efektivitas dalam merawat tanaman di *Greenhouse* secara modern dan efisien.

4.1.5 *User Interface*

User Interface pada sistem penyiraman dan pengkabutan otomatis ini memiliki dua bagian utama, yaitu antarmuka perangkat fisik dan antarmuka digital berbasis web. Pembagian ini bertujuan untuk memfasilitasi interaksi pengguna secara langsung melalui perangkat *IOT* di lapangan, serta mendukung pengelolaan dan pemantauan sistem secara jarak jauh melalui *Dashboard* web yang responsif dan informatif.

Antarmuka web memungkinkan pengguna (*admin/pengelola Greenhouse*) untuk mengelola sistem secara menyeluruh melalui *Dashboard* Web. Fitur-fitur utama yang disediakan dalam tampilan web antara lain:

- 72
- a. *User Interface* Halaman Login
 - b. *User Interface* Halaman Dashboard Utama
 - c. *User Interface* Halaman Data Sensor
 - d. *User Interface* Halaman Kontrol alat
 - e. *User Interface* Halaman WhatsApp

4.1.6 Fitur - fitur

Fitur-fitur yang tersedia dalam sistem penyiraman dan pengkabutan otomatis ini dirancang untuk memberikan kemudahan kepada pengguna dalam memantau kondisi lingkungan tanaman serta mengontrol penyiraman dan pengkabutan secara efisien. Berikut adalah fitur-fitur yang disediakan dalam sistem:

- 3
- a. Menampilkan data suhu, kelembapan udara, dan kelembapan tanah secara *real-time*. Data ini ditampilkan dalam bentuk komponen *card* pada *Dashboard* untuk memudahkan pemahaman pengguna terhadap kondisi tanaman.
 - b. Sistem secara otomatis memproses data yang diperoleh dari sensor menggunakan algoritma *Naive Bayes* untuk menentukan apakah perlu

dilakukan penyiraman, pengkabutan, atau tidak melakukan tindakan sama sekali. Setiap kali algoritma dijalankan, hasil prediksi beserta data sensornya akan disimpan secara otomatis ke dalam *database* sebagai *log* historis. Selain itu, sistem juga mempunyai pilihan untuk menghapus data prediksi yang sudah tersimpan, sehingga pengguna juga dapat mengelola riwayat data sesuai kebutuhan.¹⁸⁹

- c. Pengguna dapat mengaktifkan atau menonaktifkan penyiraman dan pengkabutan secara manual melalui *Dashboard* jika ingin melakukan kontrol secara langsung.
- d. Pengguna dapat mengatur jadwal penyiraman atau pengkabutan pada waktu tertentu secara berulang setiap hari.
- e. memungkinkan sistem untuk secara otomatis memberhentikan proses penyiraman atau pengkabutan pada saat nilai sensor telah mencapai batasan tertentu yang telah ditentukan.
- f. Sistem mencatat setiap tindakan penyiraman dan pengkabutan, baik yang dilakukan secara otomatis maupun manual. Riwayat ini dapat ditinjau untuk evaluasi dan dokumentasi.
- g. Sistem mengirimkan pesan otomatis ke *WhatsApp* pengguna setiap kali terjadi penyiraman atau pengkabutan, lengkap dengan informasi waktu dan jenis aksi yang dilakukan.
- h. Admin dapat menambahkan dan menghapus nomor *WhatsApp* pengguna melalui halaman pengelolaan *WhatsApp*.

4.1.7 Analisis Data

Untuk mendukung perancangan sistem penyiraman dan pengkabutan otomatis berbasis *IOT* dan web, penulis memerlukan beberapa jenis data sebagai bahan analisis dan dasar pengembangan sistem. Data-data ini diperoleh dari hasil pembacaan sensor, proses prediksi menggunakan algoritma *Naive Bayes*, serta interaksi pengguna terhadap sistem melalui *Dashboard* dan notifikasi *WhatsApp*. Berikut dibawah ini adalah jenis data yang digunakan dan dianalisis dalam penelitian, yaitu sebagai berikut:

a. Data Sensor

Data ini berasal dari pembacaan sensor suhu dan kelembapan udara, serta sensor kelembapan tanah.¹⁷⁰ Data ini dikumpulkan secara berkala dan digunakan sebagai input utama dalam proses pengambilan keputusan otomatis oleh sistem. Analisis terhadap data ini dilakukan untuk memahami kondisi lingkungan di *Greenhouse* serta menguji konsistensi pembacaan sensor.

b. Data Prediksi Sistem

Data ini dihasilkan dari pemrosesan algoritma *Naive Bayes* yang menentukan status tindakan: penyiraman, pengkabutan, atau tidak melakukan aksi. Setiap hasil prediksi disimpan ke dalam *database* untuk dianalisis lebih lanjut, baik dalam konteks validasi algoritma maupun evaluasi akurasi sistem terhadap kondisi sebenarnya.

c. Data Interaksi Pengguna

Data ini mencakup aktivitas pengguna melalui *Dashboard*, seperti pengaturan jadwal, kontrol manual, serta penghapusan data prediksi. Selain itu, interaksi berupa tanggapan terhadap notifikasi *WhatsApp* juga menjadi bagian dari data yang dianalisis untuk menilai sejauh mana sistem memberikan kemudahan dan kontrol bagi pengguna.

d. Data Kinerja Sistem

Data ini meliputi waktu respons sistem dalam membaca sensor, menjalankan prediksi, menyalakan atau mematikan perangkat (pompa atau *sprayer*), serta kecepatan pengiriman notifikasi ke *WhatsApp*. Data ini dianalisis untuk mengevaluasi kestabilan dan efisiensi sistem secara keseluruhan, termasuk keterandalan koneksi antara *ESP32* dan *server*.

4.1.8 Analisis Biaya

Untuk support pembangunan aplikasi penyiraman dan pengkabutan otomatis berbasis *Internet of Things (IOT)* dengan metode *agile* dan algoritma *Naive Bayes*, penulis melakukan analisis terhadap kebutuhan biaya yang mencakup pengadaan perangkat keras, perangkat lunak, serta komponen pendukung lainnya yang digunakan dalam proses pengembangan dan implementasi sistem.

Tabel 4. 1 Analisis Biaya

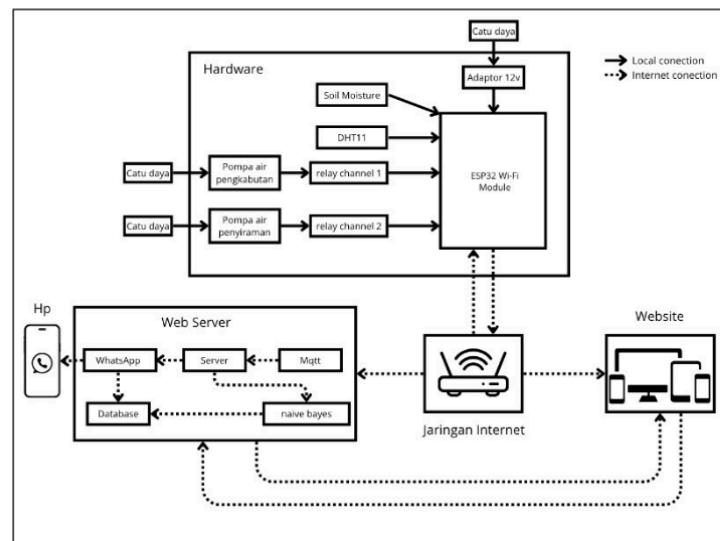
No	Jenis Kebutuhan	Biaya
1	Biaya ATK	Rp. 2.000.000
2	DHT11	Rp. 10.000
3	2 Channel Relay	Rp. 23.000
4	Kabel USB	Rp. 15.000
5	Kabel AWG 24 3 roll (Hitam, Hijau, Merah)	Rp. 80.000
6	Tang Kabel	Rp. 34.000
7	Heatshrink Tube Isolasi bakar	Rp. 13.000
8	Adaptor DC 12V	Rp. 8.000
9	Expansion Board ESP32 38 pin	Rp. 29.000
10	ESP32 DOIT 38 pin	Rp. 66.500
11	Soil moisture sensor	Rp. 9.000
12	Dupont Connector Female 100pcs	Rp. 11.000
13	Selang PE 6 MM 20 meter	Rp. 39.000
14	Mist Nozzle Sprayer 15pcs	Rp. 21.000
15	Stop kontak steker saklar on off 2pcs	Rp. 23.000
16	Avometer Digital	Rp. 43.000
17	Box Componen	Rp. 17.000
18	Baut sekrup 20pc	Rp. 3.000
19	Pompa air sanyo	Rp. 200.000
20	Pompa air aquarium	Rp. 80.000
21	Kabel Listrik hitam 7mm 4 meter	Rp. 20.000
22	Alat set Bor	Rp. 362.000
23	Lem tembak + glue gun	Rp. 63.000
24	Set solder Listrik + timah solder	Rp. 140.000
25	Token Listrik	Rp. 50.000
26	Kuota Internet	Rp. 100.000
27	Hosting VPS 4 core 4 gb ram 100gb ssd 3 bln	Rp. 600.000
28	Transportasi Kendaraan	Rp. 500.000
29	Pembuatan Aplikasi	Rp. 5.000.0000
31	Selang Air	Rp. 50.000
32	Pipa paralon	Rp. 100.000
Total Jumlah		Rp 9.809.500

4.2 Perancangan

Sebelum memasuki tahap pembuatan sistem penyiraman dan pengabutan otomatis berbasis *IOT*, diperlukan perancangan sistem yang terencana dan menyeluruh. Tahapan ini mencakup pembuatan blok diagram untuk menggambarkan alur kerja sistem penyiraman dan pengabutan secara otomatis. Selanjutnya, dilakukan perancangan rangkaian perangkat keras menggunakan aplikasi *Fritzing* untuk memvisualisasikan koneksi antar komponen seperti sensor, mikrokontroler, pompa, dan *sprayer*. Pada sisi perangkat lunak, sistem dirancang dalam bentuk *Dashboard* web, yang dimodelkan dengan *Unified Modeling Language (UML)* melalui penyusunan *Activity*, *Use Case* dan *Class Diagram* sebagai acuan dalam pengembangan fungsionalitas dan alur sistem.

4.2.1 Blok Diagram Alat dan Sistem

Perancangan ini menunjukkan blok diagram konsep yang menggambarkan integrasi antara perangkat keras dan perangkat lunak.¹²⁸



Gambar 4. 1 Blok diagram alat dan sistem

Penjelasan mengenai gambar diagram di atas dapat secara terinci diuraikan dalam sebuah tabel deskripsi yang telah disediakan:

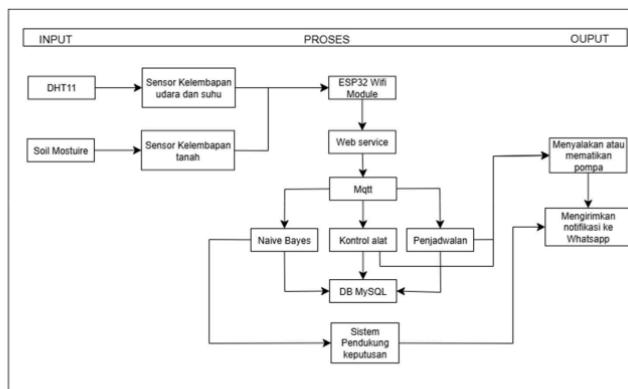
Tabel 4. 2 Tabel diagram konsep

Lingkup	Deskripsi
<i>Hardware</i>	Perangkat keras terdiri dari <i>ESP32 WiFi Modul</i> sebagai pusat kendali, yang terhubung dengan sensor <i>DHT11</i> dan <i>Soil moisture</i> (kelembapan tanah). Modul <i>ESP32</i> juga mengontrol dua <i>Relay channel</i> , masing-masing terhubung ke pompa air penyiraman dan pompa pengkabutan, dengan catu daya terpisah.
Jaringan Internet	Modul <i>ESP32</i> terhubung ke jaringan internet melalui koneksi jaringan internet, memungkinkan pertukaran data antara perangkat fisik dan sistem <i>backend (web server serta website)</i> . Komunikasi menggunakan protokol <i>MQTT</i> dan jalur internet digunakan untuk proses pengiriman dan penerimaan data sensor serta kontrol pompa.
<i>Web server</i>	<i>Web server</i> berfungsi sebagai pusat logika sistem. Data dari sensor yang dikirim melalui <i>MQTT</i> diolah oleh <i>server backend</i> , yang memanfaatkan algoritma <i>Naive Bayes</i> untuk melakukan prediksi tindakan (siram, kabut, atau tidak). Seluruh data disimpan dalam <i>database</i> untuk kebutuhan histori dan pengambilan keputusan. Selain itu, <i>server</i> juga mengirimkan notifikasi otomatis ke <i>WhatsApp</i> pengguna setiap kali sistem melakukan penyiraman atau pengkabutan, baik secara otomatis maupun manual.
<i>Website</i>	<i>Website</i> menampilkan data sensor, histori aksi, dan menyediakan kontrol manual untuk penyiraman dan

	pengkabutan. Komunikasi dilakukan secara <i>real-time</i> melalui integrasi dengan sistem <i>MQTT</i> dan <i>database</i> .
--	---

4.2.2 Diagram Alur Data

Perancangan ini menggambarkan alur data penyiraman dan pengkabutan otomatis pada tanaman, mulai dari proses pembacaan data sensor oleh *ESP32*, hingga pengambilan keputusan penyiraman atau pengkabutan menggunakan algoritma *Naive Bayes* di *web server* dan pengguna juga menerima notifikasi status aksi secara *real-time* melalui *WhatsApp*.



Gambar 4. 2 Diagram alur alat dan sistem

Penjelasan mengenai gambar diagram di atas dapat secara terinci diuraikan dalam sebuah tabel deskripsi yang telah disediakan:

Tabel 4. 3 Deskripsi Diagram Alur Alat dan Sistem

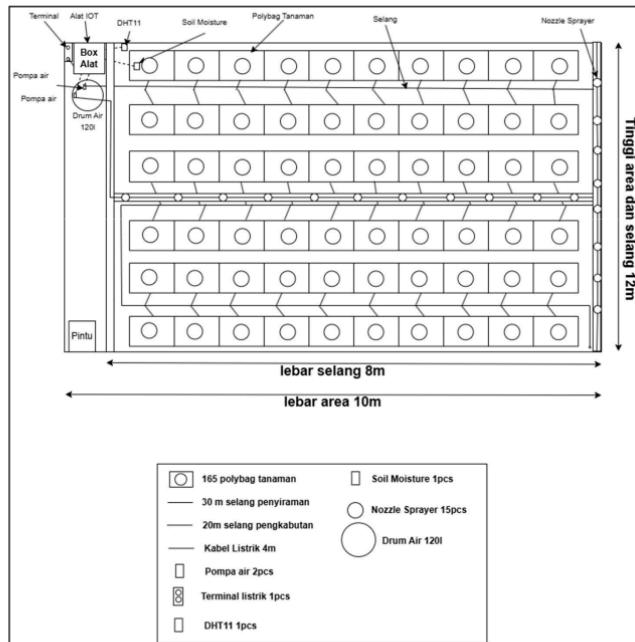
Komponen	Kategori	Deskripsi
DHT11	Input	Sensor yang digunakan untuk mengukur suhu dan kelembapan udara di lingkungan <i>Greenhouse</i> .

<i>Soil moisture</i>	<i>Input</i>	Sensor yang digunakan untuk mendekripsi tingkat kelembapan tanah guna menentukan kebutuhan penyiraman.
Sensor Kelembapan & Suhu	Proses	Data sensor yang dibaca dari <i>DHT11</i> .
Sensor Kelembapan Tanah	Proses	Data sensor yang dibaca dari <i>Soil moisture</i>
<i>ESP32 WiFi Modul</i>	Proses	Mikrokontroler yang berfungsi mengirim data sensor ke <i>web service</i> dan menerima instruksi penyiraman atau pengkabutan.
<i>Web Service</i>	Proses	<i>Server backend</i> yang mengelola komunikasi data dari <i>ESP32</i> dan mendistribusikannya ke berbagai modul logika sistem.
<i>MQTT</i>	Proses	Protokol komunikasi ringan yang digunakan untuk mengirim data sensor dari <i>ESP32</i> ke <i>server</i> secara <i>real-time</i> dan mengirim instruksi kembali ke perangkat.
<i>Naive Bayes</i>	Proses	Algoritma klasifikasi yang menganalisis data sensor untuk memutuskan tindakan penyiraman, pengkabutan, atau tidak bertindak.
Kontrol Alat	Proses	Modul yang mengeksekusi hasil prediksi untuk mengontrol pompa penyiraman atau pengkabutan sesuai hasil keputusan.
Penjadwalan	Proses	Modul yang memungkinkan

		sistem melakukan penyiraman atau pengkabutan otomatis sesuai waktu yang telah ditentukan.
<i>DB MySQL</i>	Proses	Basis data yang menyimpan semua nomor <i>WhatsApp</i> , riwayat aksi, jadwal, dan hasil prediksi untuk keperluan analisis dan dokumentasi.
Sistem Pendukung Keputusan	Proses	Komponen logika tambahan yang memanfaatkan data historis dalam <i>database</i> untuk menyempurnakan proses pengambilan keputusan sistem.
Menyalakan & Mematikan Pompa	<i>Output</i>	Hasil akhir berupa aksi langsung sistem untuk mengaktifkan atau menonaktifkan pompa berdasarkan kontrol manual maupun penjadwalan.
Mengirimkan Notifikasi ke <i>WhatsApp</i>	<i>Output</i>	Sistem mengirimkan pemberitahuan otomatis ke <i>WhatsApp</i> pengguna mengenai status penyiraman atau pengkabutan.

4.2.3 Denah *Greenhouse*

Tata letak sistem penyiraman dan pengkabutan pada penelitian ini dirancang dengan mempertimbangkan efisiensi distribusi air serta kestabilan dalam pembacaan sensor di dalam *Greenhouse*. Penempatan perangkat dilakukan secara strategis agar mendukung kinerja sistem secara optimal, mulai dari pengambilan data, pemrosesan keputusan, hingga eksekusi penyiraman atau pengkabutan.



Gambar 4. 3 Denah Greenhouse

Tata letak sistem penyiraman dan pengkabutan otomatis ini diterapkan pada *Greenhouse* dengan luas area 10meter x 12meter, yang berisi 165 *polybag* tanaman tersusun dalam beberapa baris sejajar. Desain tata letak dirancang untuk memastikan distribusi air dan kabut yang merata, serta memudahkan instalasi dan pemeliharaan sistem secara keseluruhan. Beberapa komponen penting yang ditata secara strategis meliputi:

Tabel 4. 4 Deskripsi Denah Greenhouse

Komponen	Deskripsi
Box Alat & Alat IOT	Diletakkan di sisi kiri atas area <i>Greenhouse</i> dekat dengan terminal listrik, berfungsi sebagai pusat kontrol yang berisi <i>ESP32 WiFi</i> modul, <i>Relay</i> , dan konektor sensor

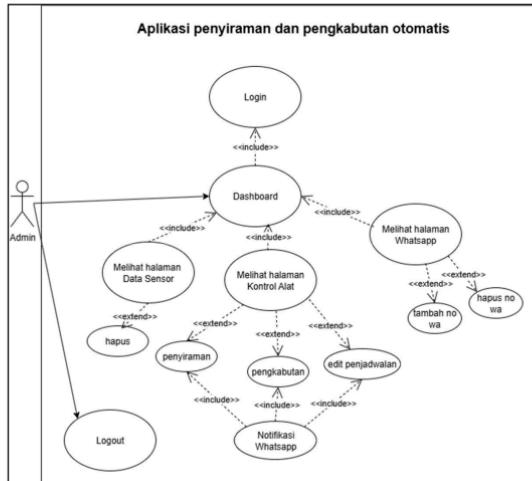
Sensor <i>DHT11</i> & <i>Soil moisture</i>	<i>DHT11</i> diletakkan di dekat box alat untuk mengukur suhu dan kelembapan udara secara sentral. Sensor <i>Soil moisture</i> ditanam pada salah satu baris tanaman untuk mewakili kondisi kelembapan tanah dari area tanam
Pompa Air & Drum 120 Liter	Dua pompa air diposisikan di dekat drum air 120L pada sisi kiri area tanam. Satu pompa digunakan untuk penyiraman, dan satu lagi untuk pengkabutan. Keduanya terhubung ke jaringan pipa melalui selang
Jaringan Selang dan <i>Nozzle</i>	Sistem selang penyiraman sepanjang 30meter dan selang pengkabutan sepanjang 20meter ditata memanjang mengikuti barisan tanaman. 15 <i>nozzle sprayer</i> dipasang merata di titik-titik strategis di atas tanaman untuk menyebarkan air kabut secara efisien dan merata. Lebar penempatan selang adalah 8 meter, dengan panjang bentangan mengikuti tinggi area 12meter
Terminal Listrik & Kabel	Satu terminal listrik diletakkan di bagian pojok atas dekat box alat untuk menyuplai daya ke pompa dan mikrokontroler. Sistem menggunakan 4meter kabel listrik yang ditata dengan rapi mengikuti sisi dalam <i>Greenhouse</i> agar aman dan tidak mengganggu jalur tanam

4.2.4 Unified Modeling Language (UML)

Untuk menggambarkan suatu proses dan struktur sistem secara konseptual, digunakan pendekatan *Unified Modeling Language (UML)*. *UML* berfungsi untuk memodelkan interaksi antara pengguna dan sistem, alur kegiatan dalam sistem. Pada tahap ini ditampilkan beberapa jenis diagram *UML*, yaitu:

8. 1. Use Case Diagram

Use Case Diagram berikut menggambarkan interaksi antara aktor dengan sistem penyiraman dan pengkabutan otomatis berbasis *IOT* di *Greenhouse*. Diagram ini dirancang untuk memperlihatkan fungsi-fungsi utama yang dapat dilakukan oleh masing-masing aktor.



176
Gambar 4. 4 *Use Case Diagram*

Penjelasan mengenai gambar *diagram* di atas dapat secara terinci diuraikan dalam sebuah tabel deskripsi yang telah disediakan:

2
Tabel 4. 5 Deskripsi Aktor

Aktor	Deskripsi
<i>Admin</i>	<i>Admin</i> sebagai <i>pengguna utama</i> sistem <i>yang</i> mempunyai hak akses penuh pada seluruh fitur aplikasi penyiraman dan pengkabutan otomatis. <i>Admin</i> bisa melakukan <i>Login</i> untuk mengakses ke <i>Dashboard</i> , melihat data sensor, serta menghapus data yang tidak diperlukan. <i>Admin</i> juga dapat mengontrol penyiraman dan pengkabutan secara manual, mengatur jadwal otomatisasi alat, dan mengelola daftar nomor <i>WhatsApp</i> yang digunakan untuk menerima notifikasi. Selain itu, <i>admin</i> dapat menambahkan atau menghapus nomor <i>WhatsApp</i> , serta menerima pemberitahuan otomatis dari sistem mengenai status penyiraman atau pengkabutan yang telah dilakukan. Setelah seluruh aktivitas selesai, <i>admin</i> dapat

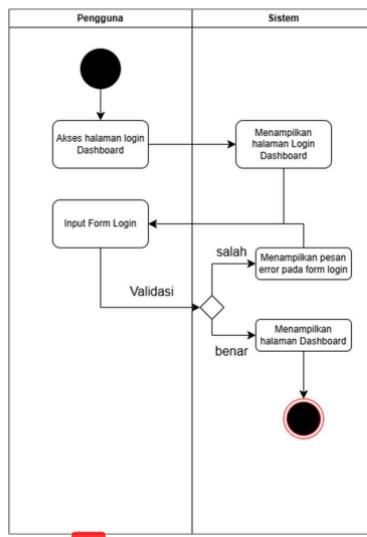
	mengakukan <i>Logout</i> dari sistem.
--	---------------------------------------

135 2. *Activity Diagram*

Activity diagram berfungsi untuk menggambarkan urutan-urutan aktivitas dalam suatu sistem serta alur kerja dari satu proses ke proses lainnya.

a. *Activity Login*

Dalam *Activity Login*, menggambarkan tahapan aktivitas pada saat melakukan *Login*, sesuai dengan yang tergambar pada gambar di bawah ini:



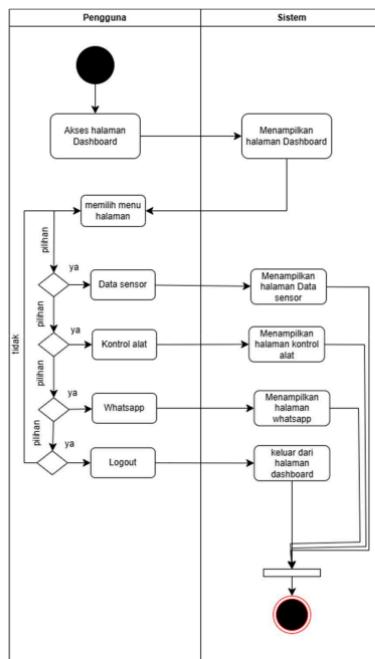
133 Gambar 4. 5 *Activity Login*

14 Gambar tersebut menunjukkan *Activity diagram* yang menjelaskan proses pengguna saat melakukan *Login* ke halaman *Dashboard*. Proses diawali ketika pengguna mengakses halaman *Login* melalui aplikasi. Sistem kemudian menampilkan form *Login* yang memungkinkan pengguna untuk memasukkan informasi autentikasi, seperti *email* atau *username* dan *password*. Setelah pengguna mengisi form dan mengirimkannya, sistem melakukan validasi

terhadap data yang di inputkan. Jika data yang di inputkan salah, sistem akan memunculkan pesan error pada form *Login* dan pengguna diminta untuk mengulang proses input. Namun, jika data valid, sistem akan memunculkan halaman *Dashboard* sebagai tanda bahwa proses *Login* berhasil.²¹

b. Activity Dashboard

Dalam *Activity Dashboard*, menggambarkan tahapan aktivitas pada saat masuk *Dashboard*:



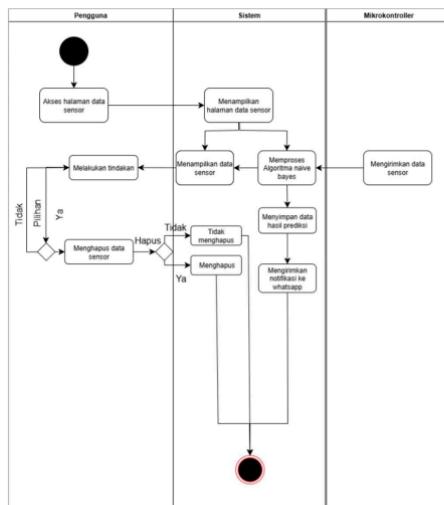
Gambar 4. 6 *Activity Dashboard*

Gambar tersebut menunjukkan *Activity* diagram yang menggambarkan alur aktivitas pengguna saat mengakses dan berinteraksi dengan halaman *Dashboard* sistem. Proses diawali ketika pengguna mengakses halaman *Dashboard*. Sistem kemudian menampilkan halaman *Dashboard* sebagai tampilan awal. Selanjutnya, pengguna dapat memilih berbagai menu halaman yang tersedia. Jika pengguna

memilih menu Data Sensor, maka sistem akan menampilkan halaman Data Sensor. Jika pengguna memilih menu Kontrol Alat, sistem akan menampilkan halaman ¹⁵ kontrol alat. Jika pengguna memilih menu WhatsApp, sistem akan menampilkan halaman WhatsApp. Sementara itu, jika pengguna memilih opsi Logout, maka sistem akan keluar dari halaman Dashboard dan proses berakhir.

c. Activity Dashboard Data Sensor

Dalam *Activity Dashboard* data sensor, menggambarkan tahapan aktivitas pada saat masuk *Dashboard* data sensor, pada gambar di bawah ini:



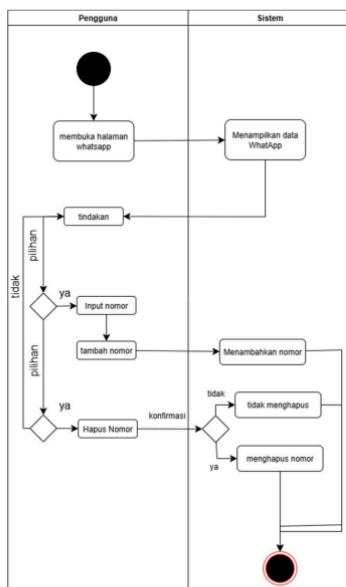
Gambar 4. 7 *Activity Dashboard Data Sensor*

Proses diawali ketika pengguna mengakses halaman data sensor. Sistem kemudian menampilkan halaman yang memiliki data sensor yang telah dikirimkan sebelumnya oleh mikrokontroler. Setelah data ditampilkan, sistem langsung memproses data tersebut menggunakan algoritma *Naive Bayes* untuk menghasilkan prediksi. Hasil dari proses prediksi kemudian disimpan oleh sistem, dan sistem juga dapat mengirimkan notifikasi ke *WhatsApp*. Selanjutnya, pengguna dapat memilih untuk melakukan tindakan, seperti menghapus data sensor. Jika pengguna memilih untuk menghapus data, sistem akan menanyakan

kembali konfirmasi penghapusan. Jika pengguna memilih penghapusan, sistem akan menghapus data tersebut. Jika tidak, maka data tetap disimpan.

d. *Activity WhatsApp*

Dalam *Activity WhatsApp*, menggambarkan tahapan aktivitas pada saat masuk WhatsApp, pada gambar di bawah ini:



Gambar 4. 8 *Activity WhatsApp*

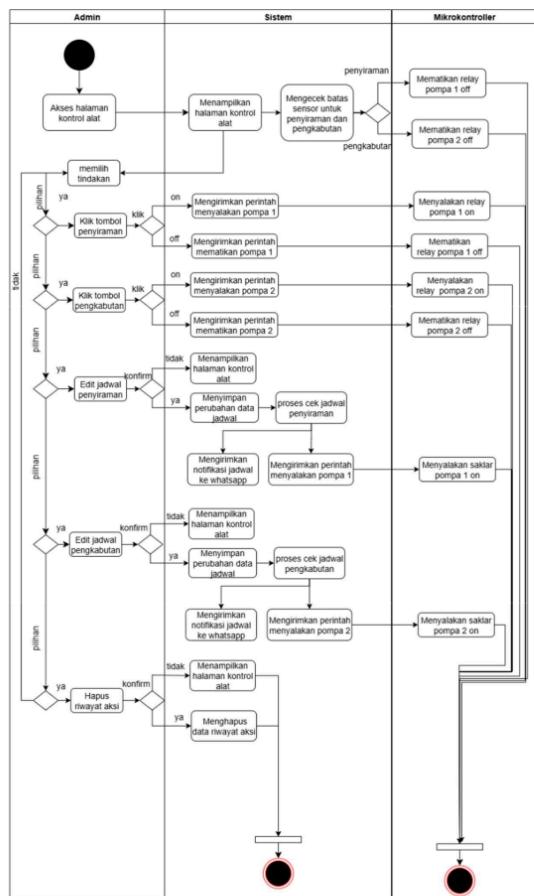
Gambar tersebut menunjukkan *Activity diagram* yang menggambarkan proses interaksi pengguna dengan halaman WhatsApp. Proses diawali ketika pengguna membuka halaman WhatsApp. Setelah itu, sistem akan menampilkan data WhatsApp yang telah tersimpan sebelumnya. Pengguna kemudian dapat melakukan tindakan, seperti menambahkan nomor WhatsApp baru atau menghapus nomor yang sudah ada. Jika pengguna memilih untuk menambahkan nomor, maka pengguna perlu menginput nomor terlebih dahulu dan memilih opsi "tambah nomor". Setelah itu, sistem akan menyimpan nomor tersebut ke dalam database. Selain itu, pengguna juga dapat memilih untuk menghapus nomor

14

WhatsApp. Jika opsi ini dipilih, sistem akan meminta konfirmasi terlebih dahulu. Jika pengguna mengonfirmasi penghapusan, sistem akan menghapus nomor tersebut. Jika tidak, maka sistem tidak akan menghapusnya.

e. Activity Kontrol Alat

Dalam *Activity* kontrol alat, menggambarkan tahapan aktivitas pada saat masuk kontrol alat, pada gambar di bawah ini:



Gambar 4. 9 *Activity* Kontrol Alat

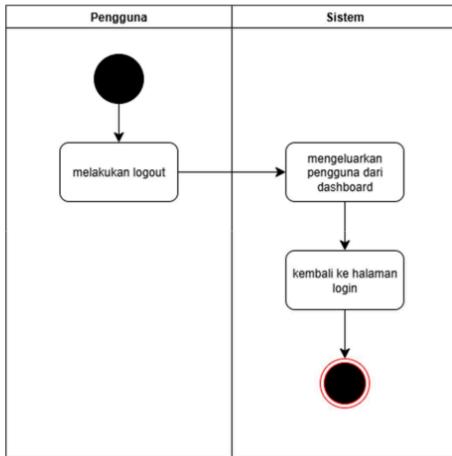
Gambar tersebut menunjukkan ^[121] **Activity diagram** yang menggambarkan alur interaksi antara *admin*, sistem, dan mikrokontroler dalam proses pengendalian alat penyiraman dan pengkabutan. Aktivitas dimulai ketika *admin* mengakses halaman kontrol alat. Setelah itu, sistem akan menampilkan halaman kontrol serta secara otomatis melakukan pengecekan batas sensor untuk menentukan apakah kondisi saat ini harus menonaktifkan proses penyiraman atau pengkabutan. Jika nilai sensor berada dalam batas tertentu, sistem akan mengirimkan perintah kepada mikrokontroler untuk mematikan *Relay* pompa penyiraman dan pengkabutan.

Selanjutnya, *admin* dapat memilih tindakan yang ingin dilakukan melalui halaman kontrol. Apabila *admin* mengklik tombol penyiraman dan memilih mode "on", maka sistem akan mengirimkan perintah untuk menyalakan *Relay* pompa penyiraman ke mikrokontroler. Sebaliknya, jika *admin* memilih mode "off", maka sistem akan memerintahkan mikrokontroler untuk mematikan *Relay* pompa tersebut. Hal serupa juga berlaku pada tombol pengkabutan, di mana sistem akan mengirimkan perintah untuk mengaktifkan atau menonaktifkan *Relay* pompa pengkabutan sesuai dengan perintah yang diberikan oleh *admin*.

Selain itu, *admin* juga memiliki opsi untuk mengedit jadwal penyiraman maupun pengkabutan. Ketika *admin* memilih opsi tersebut, sistem akan menampilkan halaman pengaturan jadwal dan menunggu konfirmasi perubahan dari *admin*. Setelah data jadwal disimpan, sistem akan memproses pengecekan terhadap waktu penyiraman atau pengkabutan yang telah dijadwalkan. Jika sesuai, sistem akan mengirimkan notifikasi melalui *WhatsApp* dan mengirimkan perintah ke mikrokontroler untuk menyalakan saklar pompa sesuai dengan jadwal yang telah ditentukan. *Admin* juga dapat melakukan penghapusan terhadap data riwayat aksi yang tercatat. Jika *admin* mengonfirmasi tindakan tersebut, maka sistem akan menghapus data riwayat aksi dari basis data dan menampilkan Kembali ke halaman kontrol alat sebagai penutup proses. Secara keseluruhan, diagram ini menggambarkan bagaimana *admin* dapat mengontrol perangkat secara manual maupun otomatis, serta bagaimana sistem berperan sebagai penghubung yang mengelola perintah dan respon antara antarmuka pengguna dan perangkat mikrokontroler dalam sistem pengendalian alat untuk penyiraman dan pengkabutan berbasis *Internet of Things*.

f. *Activity Admin Logout*

Dalam *admin Logout*, menggambarkan tahapan aktivitas pada saat melakukan ²
Logout, pada gambar di bawah ini:



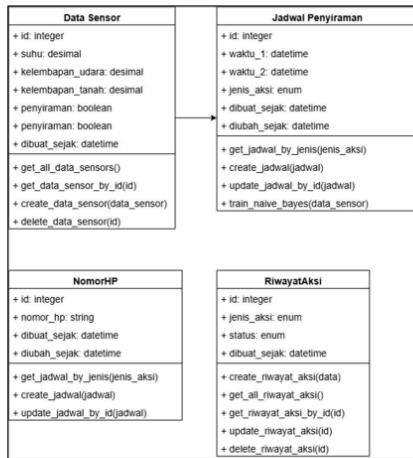
Gambar 4. 10 *Activity Logout*

Gambar tersebut menunjukkan *Activity* diagram yang menggambarkan tahapan aktivitas saat pengguna, dalam hal ini *admin*, melakukan proses logout dari sistem. Proses diawali ketika pengguna memilih opsi logout melalui antarmuka aplikasi. Setelah perintah logout dikirimkan, sistem segera memproses permintaan tersebut dengan mengeluarkan pengguna dari sesi aktif dan menghentikan akses ke halaman *Dashboard*. Setelah pengguna berhasil dikeluarkan dari *Dashboard*, sistem secara otomatis akan mengarahkan pengguna kembali ke halaman *Login*.

Langkah ini menandakan bahwa sesi autentikasi telah berakhir, sehingga untuk mengakses kembali sistem, pengguna perlu melakukan proses *Login* ulang. Diagram ini mencerminkan bagaimana sistem menjaga keamanan akses pengguna dengan memastikan bahwa setiap sesi ditutup secara eksplisit melalui proses *logout* yang *valid*. Hal ini penting untuk mencegah akses tidak sah terhadap data atau fitur sistem jika perangkat digunakan oleh orang lain setelah sesi berakhir.

117 3. *Class Diagram*

Class Diagram berikut menggambarkan struktur sistem dan hubungan antara kelas-kelas yang terlibat dalam sistem penyiraman dan pengkabutan otomatis berbasis *IOT* di *Greenhouse*.



Gambar 4. 11 *Class Diagram*

Gambar tersebut menunjukkan *Class Diagram* yang menggambarkan struktur utama sistem penyiraman dan pengkabutan otomatis berbasis *IOT* yang digunakan di *Greenhouse*. Diagram ini terdiri dari empat kelas utama, yaitu *DataSensor*, *JadwalPenyiraman*, *NomorHP*, dan *RiwayatAksi*, yang masing-masing memiliki atribut dan fungsi spesifik sesuai dengan tanggung jawabnya. Kelas *DataSensor* berfungsi untuk merepresentasikan data hasil pembacaan dari berbagai sensor seperti suhu, kelembapan udara, dan kelembapan tanah. Kelas ini juga mencatat status penyiraman dan pengkabutan serta waktu pencatatan data. Fungsi-fungsi di dalamnya meliputi pengambilan seluruh data sensor, pengambilan data berdasarkan ID, pembuatan data baru, dan penghapusan data sensor.

Kelas *JadwalPenyiraman* digunakan untuk menyimpan dan mengatur jadwal penyiraman maupun pengkabutan. Kelas ini memiliki atribut waktu, jenis aksi

(penyiraman atau pengkabutan), serta informasi waktu pembuatan dan pembaruan data. Terdapat relasi antara kelas ini dengan DataSensor, yang digunakan dalam proses pelatihan algoritma *Naive Bayes* guna menentukan jadwal penyiraman otomatis berdasarkan pola data sensor yang telah direkam sebelumnya. Fungsi pada kelas ini mencakup pengambilan jadwal berdasarkan jenis aksi, pembuatan dan pembaruan jadwal, serta pelatihan model prediksi. Kelas NomorHP menyimpan daftar nomor *WhatsApp* yang digunakan untuk menerima notifikasi sistem. Atribut utama di dalamnya adalah nomor HP, serta waktu pembuatan dan pembaruan data. Kelas ini menyediakan fungsi untuk mengelola data nomor berdasarkan jenis aksi, serta untuk membuat dan memperbarui data nomor. Kelas RiwayatAksi mencatat semua aktivitas atau perintah yang dilakukan pada sistem, baik secara manual oleh *admin* maupun otomatis oleh sistem. Setiap riwayat berisi jenis aksi, status, dan waktu pencatatan. Fungsi di dalamnya memungkinkan sistem untuk mencatat, memperbarui, mengambil, dan menghapus data riwayat.

¹⁰⁴ 4.2.5 Struktur Tabel

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam aplikasi *IOT* sebagai pengukur kinerja proses penyiraman dan pengkabutan di avicenna *Greenhouse* adalah sebagai berikut

- Perancangan tabel data_sensor

³⁷
Tabel 4. 6 Tabel data sensor

Field	Type	Size	Indeks	Deskripsi
id	Integer	10	<i>Primary key</i>	Id primary key tabel data sensor
suhu	Float	-	-	Kolom untuk data sensor suhu
kelembapan udara	Float	-	-	Kolom untuk data sensor udara
kelembapan tanah	Float	-	-	Kolom untuk data sensor udara
penyiraman	Integer	1	-	Kolom untuk data aksi penyiraman
pengkabutan	Integer	1	-	Kolom untuk data aksi pengkabutan
dibuat_sejak	Datetime	-	-	Waktu data sensor dibuat

b. Perancangan tabel jadwal penyiraman

Tabel 4. 7 Tabel jadwal penyiraman

Field	Type	Size	Indeks	Deskripsi
id	Integer	10	Primary key	Id primary key tabel jadwal penyiraman
waktu_1	Time	-	-	Kolom untuk waktu kesatu penyiraman atau pengkabutan
waktu_2	Time	-	-	Kolom untuk waktu kedua penyiraman atau pengkabutan
jenis_aksi	Enum	'penyiram an','pengk abutan'	-	Kolom untuk jenis aksi yaitu penyiraman atau pengkabutan
dibuat_sejak	Datetime	-	-	Waktu dibuatnya data jadwal penyiraman
diubah_sejak	Datetime	-	-	Waktu diubahnya data jadwal penyiraman

c. Perancangan tabel riwayat aksi

³⁷
Tabel 4. 8 Tabel riwayat aksi

Field	Type	Size	Indeks	Deskripsi
id	Integer	10	Primary key	Id primary key tabel riwayat aksi
Jenis_aksi	Enum	'penyiraman','pengkabutan'	-	Kolom untuk jenis aksi yaitu penyiraman atau pengkabutan
status	Enum	'aktif','nonaktif'	-	Kolom untuk status penyiraman atau pengkabutan
dibuat_sejak	Datetime	-	-	Waktu dibuatnya data jadwal riwayat aksi

d. Perancangan tabel nomor_hp

Tabel 4. 9 Tabel nomor_hp

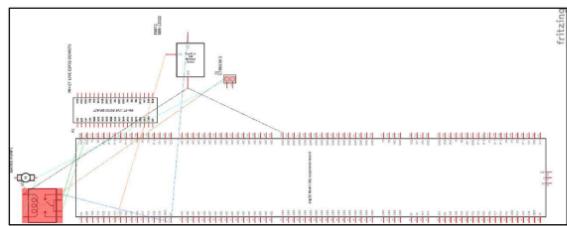
Field	Type	Size	Indeks	Deskripsi
id	Integer	10	Primary key	Id primary key tabel nomor_hp
Nomor_hp	String	15	-	Kolom untuk data nomor WhatsApp
dibuat_sejak	Datetime	-	-	Waktu dibuatnya data nomor_hp
diubah_sejak	Datetime	-	-	Waktu diubahnya data nomor_hp

4.2.6 Perancangan *Wiring Diagram*

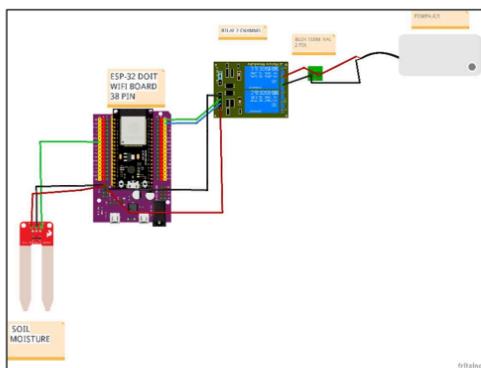
Sebelum sistem dapat berjalan, diperlukan perancangan rangkaian pada perangkat keras supaya seluruh komponen dapat bekerja dengan terintegrasi sesuai dengan fungsi nya masing-masing. Pada tahap ini, dilakukan penyusunan rangkaian elektronik menggunakan aplikasi *Fritzing* untuk memvisualisasi koneksi antar komponen *IOT*.

1. Skema Rangkaian Sensor Kelembapan Tanah

Berikut merupakan skema dan *wiring* rangkaian sensor menunjukkan cara menghubungkan sensor kelembapan tanah ke mikrokontroler *ESP32*.



Gambar 4. 12 Skema Rangkaian Kelembapan Tanah



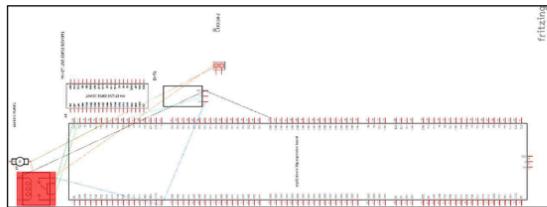
Gambar 4. 13 Implementasi *Wiring* Sensor Kelembapan Tanah

Sensor kelembapan tanah (*soil moisture*) terhubung langsung ke mikrokontroler *ESP32 DOIT 38 pin* menggunakan tiga kabel utama, yaitu kabel

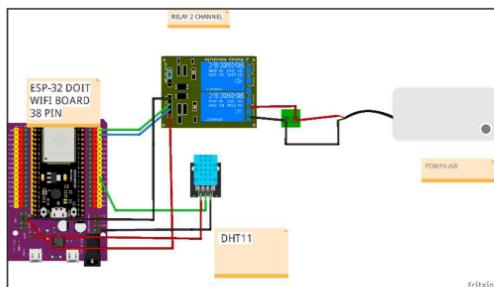
merah (*VCC*) sebagai sumber daya yang disambungkan ke pin 3V3 pada *ESP32*, kabel hitam (*GND*) yang dihubungkan ke pin *GND* untuk ground, dan kabel hijau sebagai kabel data yang terhubung ke pin pada *ESP32* untuk membaca nilai kelembapan tanah. *ESP32* kemudian terhubung ke modul *Relay 1 channel* untuk mengontrol pompa air. Koneksi dari *ESP32* ke *Relay* dilakukan menggunakan kabel merah dari pin 3V3 ke *VCC* pada *Relay*, kabel hitam dari *GND* *ESP32* ke *GND Relay*, serta kabel kontrol biru yang menghubungkan salah satu pin digital *ESP32* ke pin pada *Relay*. Sinyal dari pin digital ini akan mengaktifkan atau menonaktifkan *Relay* sesuai dengan logika yang diprogram. Modul *Relay* selanjutnya mengendalikan pompa air yang dihubungkan melalui terminal blok 2 pin pada *Relay*. Sumber daya eksternal (misalnya adaptor *DC*) mengalir melalui *Relay* ke pompa.

2. Skema Rangkaian Kelembapan udara dan suhu

Berikut merupakan skema dan *wiring* rangkaian sensor menunjukkan cara menghubungkan sensor kelembapan udara dan suhu ke mikrokontroler *ESP32*.



Gambar 4. 14 Skema Rangkaian Kelembapan Udara dan Suhu

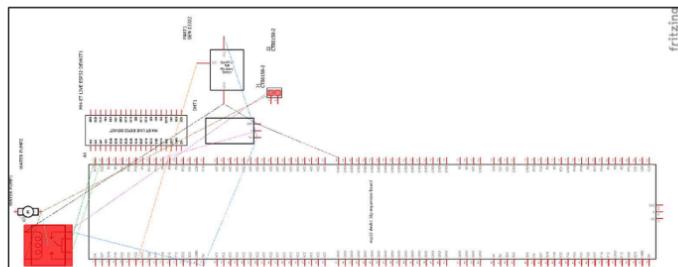


Gambar 4. 15 Implementasi Wiring Sensor Kelembapan Udara dan Suhu

Sensor *DHT11* terhubung ke mikrokontroler *ESP32* menggunakan tiga kabel utama: kabel merah (*VCC*) terhubung ke pin *3V3* pada *ESP32* untuk memberikan daya, kabel hitam (*GND*) ke pin *GND* sebagai ground, dan kabel hijau sebagai jalur data yang mengirimkan informasi suhu dan kelembaban ke pin digital *ESP32*. Data dari sensor ini dapat digunakan sebagai parameter tambahan untuk mengatur kapan pompa air harus diaktifkan atau tidak. *Relay 2 channel* berfungsi sebagai saklar elektronik yang dikendalikan oleh *ESP32*. Modul *Relay* terhubung ke *ESP32* dengan kabel merah dari pin *5V* ke *VCC*, kabel hitam ke *GND*, dan kabel kontrol (biru atau hijau) yang menghubungkan salah satu pin digital *ESP32* ke pin *IN* pada *Relay* untuk mengontrol status *ON* atau *OFF* dari *Relay*. Pompa air terhubung ke salah satu *channel* dari *Relay*. Sumber daya eksternal mengalir melalui *Relay* sebelum menuju ke pompa, sehingga pompa hanya akan menyala ketika *Relay* diaktifkan oleh sinyal dari *ESP32*. Secara keseluruhan, rangkaian ini memungkinkan *ESP32* untuk mengontrol penyiraman tanaman secara otomatis berdasarkan data suhu dan kelembaban udara dari sensor *DHT11*, serta kondisi yang telah diprogram dalam sistem.

3. Skema Rangkaian Keseluruhan alat

Rangkaian ini merupakan sistem penyiraman otomatis yang terdiri dari *ESP32* sebagai otak kendali, sensor untuk membaca suhu, kelembaban udara dan kelembaban tanah serta modul relay 2 *channel* yang mengontrol pompa air. Sensor mengirimkan data ke *ESP32*, lalu *ESP32* mengaktifkan *relay* untuk menyalakan pompa jika kondisi lingkungan memenuhi syarat. Rangkaian ini bekerja secara otomatis berdasarkan parameter yang ditentukan dalam program.



Gambar 4. 16 Skema Rangkaian Keseluruhan Sensor

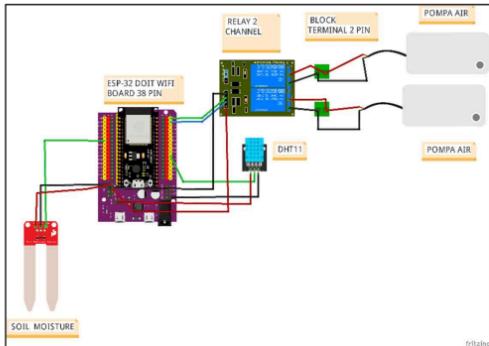
Gambar 4. 17 *Wiring Diagram*

Diagram di atas menggambarkan sistem otomatis berbasis *ESP32* yang mengontrol dua pompa air untuk kebutuhan penyiraman dan pengabutan tanaman. Sistem ini bekerja dengan membaca data dari sensor kelembapan tanah dan sensor suhu serta kelembapan udara. Data dari sensor diproses oleh *ESP32* untuk menentukan apakah pompa penyiram atau pompa pengabut perlu diaktifkan. Modul *relay 2 channel* digunakan sebagai pengendali kedua pompa tersebut. Rangkaian ini dirancang untuk menjalankan proses penyiraman dan pengabutan secara otomatis berdasarkan kondisi lingkungan yang terdeteksi. Komponen yang digunakan dalam rangkaian ini antara lain mikrokontroler *ESP32 DOIT 38 pin* sebagai pusat kendali sistem, sensor kelembapan tanah untuk mendeteksi tingkat air dalam tanah, sensor *DHT11* untuk membaca suhu dan kelembapan udara, modul *relay 2 channel* sebagai saklar otomatis, serta dua unit pompa air yang masing-masing berfungsi untuk penyiraman dan pengabutan tanaman. Komponen yang digunakan dalam rangkaian ini antara lain:

Tabel 4. 10 Deskripsi *Wiring Diagram*

Komponen	Deskripsi
<i>ESP32 DOIT Dev Board (38 pin)</i>	Sebagai pusat kendali sistem, yang menerima data dari sensor dan mengirimkan perintah ke aktuator (<i>Relay</i>)
Sensor <i>Soil moisture</i>	Untuk mengukur tingkat kelembapan tanah.
Sensor <i>DHT11</i>	Berfungsi untuk mengukur suhu dan kelembapan udara.

<i>Relay 2 Channel</i>	<i>Relay</i> ini digunakan untuk menghidupkan atau mematikan dua pompa air, masing-masing untuk sistem penyiraman dan sistem pengkabutan
Blok terminal 2 pin	Digunakan untuk menghubungkan pompa air ke rangkaian <i>Relay</i> dengan aman
Pompa air	Berfungsi sebagai aktuator untuk menyiram tanaman dan melakukan pengkabutan

4.2.7 Desain Antarmuka

Desain antarmuka (*User Interface*) merupakan proses perancangan tampilan sistem yang bertujuan ¹⁶⁹ untuk memberikan kemudahan dan kenyamanan bagi ⁹⁵ pengguna dalam mengoperasikan aplikasi. Pada sistem penyiraman dan pengkabutan otomatis tanaman berbasis *Internet of Things* ini, antarmuka dirancang dalam bentuk *Dashboard* web yang memungkinkan *admin* untuk memantau kondisi lingkungan tanaman, melihat data historis aktivitas penyiraman dan pengkabutan, serta mengatur notifikasi. Berikut merupakan tampilan antarmuka dari sistem yang diimplementasikan di Avicenna *Greenhouse*:

1. Desain tampilan antarmuka *Login*

Berikut merupakan tampilan antarmuka form *Login* untuk sistem penyiraman dan pengkabutan yang akan dirancang:

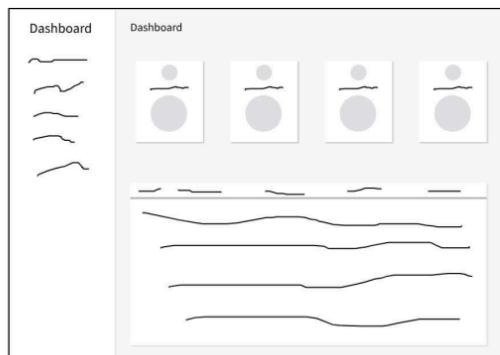


Gambar 4. 18 *User Interface Login*

Gambar di atas menampilkan antarmuka login dari sistem yang terdiri dari dua kolom input, yaitu kolom untuk *username* atau *email* dan kolom untuk *password*. Di bawahnya terdapat tombol “*Log in*” yang berfungsi untuk mengautentikasi pengguna. Setelah data yang dimasukkan sesuai dan *valid*, pengguna akan diarahkan ke halaman *Dashboard* utama, di mana mereka dapat memantau kondisi lingkungan serta mengatur sistem secara keseluruhan. ¹³⁸ Tampilan ini merupakan bagian penting dari sistem keamanan yang memastikan hanya pengguna yang berwenang yang dapat mengakses fitur kontrol *dan* pemantauan.

2. Desain tampilan antarmuka *Dashboard*

Berikut merupakan tampilan antarmuka *Dashboard* untuk sistem penyiraman dan pengkabutan yang akan dirancang:

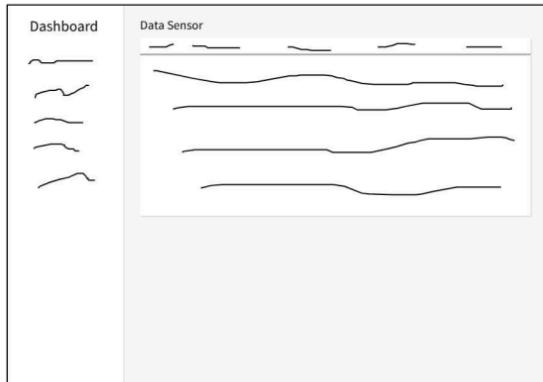


Gambar 4. 19 *User Interface Dashboard*

Antarmuka pengguna pada sistem penyiraman dan pengkabutan otomatis menampilkan *Dashboard* yang dirancang untuk memberikan informasi lingkungan secara menyeluruh. Tampilan ini menyajikan ³ data sensor, seperti kelembapan tanah, suhu, dan kelembapan udara, dalam bentuk visual yang mudah dipahami. Di bagian atas terdapat kartu informasi untuk menampilkan status atau nilai terkini dari masing-masing parameter. Sedangkan di bagian bawah, terdapat tabel atau grafik histori yang menampilkan riwayat aktivitas penyiraman dan pengkabutan.

3. Desain tampilan antarmuka data sensor

Berikut merupakan tampilan antarmuka data sensor untuk sistem penyiraman dan pengkabutan yang akan dirancang:



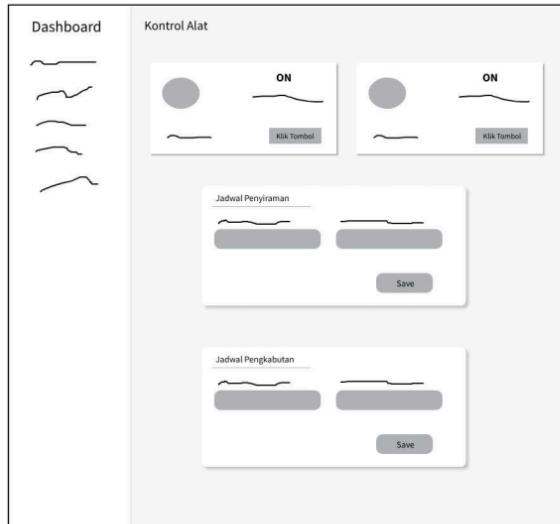
Gambar 4. 20 *User Interface* Data Sensor

Antarmuka data sensor pada sistem penyiraman dan pengkabutan otomatis dirancang untuk menampilkan hasil pembacaan sensor secara *real-time* dengan tampilan yang informatif dan mudah dipahami. ²⁵ Data lingkungan seperti kelembapan tanah, suhu udara, kelembapan udara, dan parameter penting lainnya disajikan dalam bentuk grafik garis atau tabel yang dinamis. Tampilan ini memungkinkan admin untuk memantau perubahan nilai sensor dari waktu ke waktu secara langsung.

Informasi yang ditampilkan menjadi dasar bagi sistem dalam mengambil keputusan secara otomatis menggunakan algoritma *Naïve Bayes*, yang akan menentukan kapan penyiraman atau pengkabutan perlu diaktifkan. Antarmuka ini juga membantu pengguna dalam mengevaluasi kinerja sistem secara keseluruhan, mendeteksi anomali, serta melakukan analisis pola lingkungan berdasarkan histori data. Dengan adanya visualisasi yang jelas, admin dapat lebih mudah memahami kondisi aktual di lapangan dan melakukan penyesuaian parameter sistem jika diperlukan, sehingga mendukung pengelolaan lingkungan tanaman secara lebih efisien.

4. Desain tampilan antarmuka kontrol alat

Berikut merupakan tampilan antarmuka kontrol alat untuk sistem penyiraman dan pengkabutan yang akan dirancang:

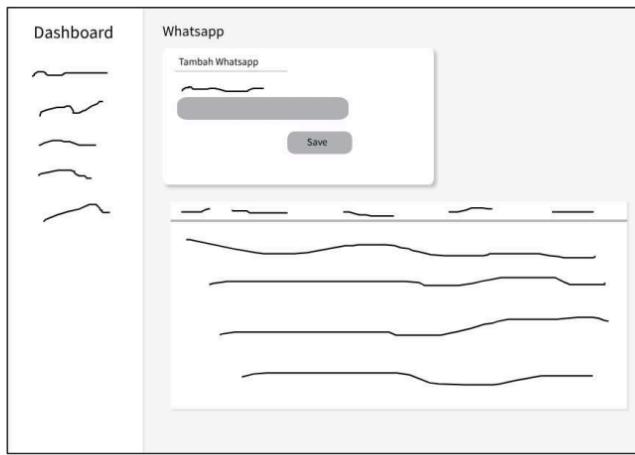


Gambar 4. 21 *User Interface* Kontrol Alat

Antarmuka kontrol alat pada sistem penyiraman dan pengkabutan otomatis dirancang untuk memberikan kemudahan kepada *admin* dalam mengendalikan perangkat secara manual. Tampilan ini menampilkan dua tombol kontrol utama, masing-masing untuk menghidupkan atau mematikan pompa penyiraman dan pengkabutan. Status perangkat ditampilkan secara *real-time* dengan indikator *on/off*. Selain itu, antarmuka ini juga dilengkapi dengan fitur pengaturan batas sensor, yang memungkinkan *admin* mengubah nilai ambang batas suhu, kelembapan udara, atau kelembapan tanah sesuai kebutuhan. Perubahan dapat disimpan dengan menekan tombol *save*.

5. Desain tampilan antarmuka *WhatsApp*

Berikut merupakan tampilan antarmuka *WhatsApp* untuk sistem penyiraman dan pengkabutan yang akan dirancang:



Gambar 4. 22 *User Interface WhatsApp*

Gambar di atas menunjukkan desain wireframe antarmuka fitur *WhatsApp* dalam sistem *IOT* penyiraman dan pengkabutan otomatis. Fitur ini memungkinkan ¹⁵⁸ *admin* untuk menambahkan dan mengelola nomor *WhatsApp* yang akan digunakan sebagai tujuan notifikasi otomatis dari sistem. Pada bagian atas tampilan, terdapat form input dengan label "Tambah WhatsApp", di mana *admin* dapat memasukkan nomor telepon yang ingin ditambahkan. Di bawahnya terdapat tombol "Save" untuk menyimpan nomor tersebut ke dalam sistem. Bagian bawah wireframe menampilkan tabel atau daftar nomor *WhatsApp* yang telah didaftarkan. Setiap entri kemungkinan menampilkan informasi seperti nomor yang terdaftar, status pengiriman pesan, serta waktu pengiriman notifikasi terakhir. Desain antarmuka ini dirancang sederhana dan fungsional agar memudahkan pengguna dalam mengelola daftar kontak yang menerima notifikasi, seperti peringatan batas sensor terlampaui atau status perangkat.

IMPLEMENTASI DAN PENGUJIAN**5.1 Implementasi**

Setelah melaksanakan analisis dan perancangan maka selanjutnya adalah pengimplementasian untuk menjalankan analisis dan perancangan yang sudah dibuat ke dalam bentuk aplikasi.

5.1.1 Listing Program

1. Listing program Kontrol alat

```
def kontrol_penyiraman_service(perintah):  
    kirim_perintah_siram(perintah)  
    created_notification_service(perintah,"penyiraman")  
  
def kontrol_pengkabutan_service(perintah):  
    kirim_perintah_kabut(perintah)  
    created_notification_service(perintah,"pengkabutan")  
  
def created_notification_service(perintah,action=None):  
    data = {  
        "jenis_aksi": action,  
        "status": "aktif" if perintah == "1" else "nonaktif"  
    }  
  
    status = "Aktif" if perintah == "1" else "Mati"  
    waktu = datetime.now().strftime("Tanggal %d-%m-%Y, jam :%H:%M')  
    pesan = f"\ud83c\udcbb Pengkabutan: {status}. {waktu}"  
    notify_sensor_data_Service(pesan, app)  
    create_riwayat_aksi_repository(data)  
    notify_sensor_data_Service(pesan, app)
```

Kode program ini digunakan untuk mengontrol alat penyiraman dan pengkabutan secara otomatis berdasarkan perintah yang diterima. Fungsi *kontrol_penyiraman_service* dan *kontrol_pengkabutan_service* digunakan untuk mengirimkan perintah aktivasi atau non-aktivasi ke alat yang bersangkutan melalui fungsi *kirim_perintah_siram* atau *kirim_perintah_kabut*. Setelah perintah dijalankan, sistem secara otomatis mencatat dan mengirimkan notifikasi melalui fungsi *created_notification_service*. Fungsi ini akan menentukan jenis aksi yang dilakukan, menetapkan status aktif atau nonaktif berdasarkan nilai perintah, dan mencatat waktu pelaksanaan aksi. Notifikasi yang telah dibentuk kemudian dikirim menggunakan *notify_sensor_data_Service*, sekaligus dicatat ke dalam sistem melalui *create_riwayat_aksi_repository* sebagai bagian dari histori aktivitas alat.

2. Listing program *Naïve Bayes*

```
model = None
required_features = ['10Suhu Udara', 'Kelembapan Udara', 'Kelembapan Tanah']
target_columns = ['Penyiraman', 'Pengkabutan']
122def load_data_from_csv(filename='sensor_data.csv'):
    file_path = os.path.join(os.path.dirname(os.path.abspath(__file__)), filename)
76    if not os.path.isfile(file_path):
        print(f"

```

```

print(f'❌ Gagal membaca file CSV: {e}')
return None

def train_naive_bayes(csv_path='sensor_data.csv'):
    global model
    df = load_data_from_csv(csv_path)
    if df is None or len(df) < 5:
        print("⚠ Data terlalu sedikit atau tidak valid.")
        return
    X = df[required_features]
    y = df[target_columns]
    model = MultiOutputClassifier(GaussianNB())
    100
    model.fit(X, y)
    y_pred = model.predict(X)
    acc_penyiraman = accuracy_score(y['Penyiraman'], y_pred[:, 0])
    208
    acc_pengkabutan = accuracy_score(y['Pengkabutan'], y_pred[:, 1])
    print(f'🎯 Akurasi Penyiraman: {acc_penyiraman:.2f}')
    print(f'🎯 Akurasi Pengkabutan: {acc_pengkabutan:.2f}')

```

Program ini digunakan untuk melatih model *Gaussian Naive Bayes* dalam memprediksi kebutuhan penyiraman dan pengkabutan berdasarkan 10 data suhu udara, kelembapan udara, dan kelembapan tanah. Data diambil dari file sensor_data.csv dan dikonversi ke format numerik. Model dilatih menggunakan *MultiOutputClassifier* untuk menghasilkan dua prediksi sekaligus. Setelah pelatihan, program menghitung dan menampilkan akurasi model terhadap data latih untuk masing-masing tindakan.

3. Listing program Penjadwalan

```

scheduler = BackgroundScheduler(timezone=pytz.timezone('Asia/Jakarta'))
# penjadwalan_service.py
scheduler = BackgroundScheduler(timezone=pytz.timezone('Asia/Jakarta'))
app = None # global app instance
def init_app(app_instance):

```

```

global app
app = app_instance

def update_jadwal_service(jadwal_id, waktu_1, waktu_2, jenis):
    try:
        if jadwal_id is None:
            create_jadwal = {
                "waktu_1": waktu_1,
                "waktu_2": waktu_2,
                "jenis_aksi": jenis
            }
            jadwal = create_jadwal_penyiraman_repository(create_jadwal)
            return {"success": True, "jadwal": {
                "id": jadwal.id,
                "waktu_1": jadwal.waktu_1.strftime("%H:%M:%S"),
                "waktu_2": jadwal.waktu_2.strftime("%H:%M:%S"),
                "jenis_aksi": jadwal.jenis_aksi
            }}
        jadwal = update_jadwal_by_id_repository(jadwal_id, waktu_1, waktu_2,
                                                jenis)
        return {"success": True, "jadwal": {
            "id": jadwal.id,
            "waktu_1": jadwal.waktu_1.strftime("%H:%M:%S"),
            "waktu_2": jadwal.waktu_2.strftime("%H:%M:%S"),
            "jenis_aksi": jadwal.jenis_aksi
        }}
    except Exception as e:76
        print("Error update_jadwal:", e)
    return {"success": False}

```

Program ini digunakan untuk mengatur jadwal otomatis penyiraman dan pengkabutan menggunakan dengan zona waktu Indonesia. Fungsi `init_app` digunakan untuk menginisialisasi `instance` aplikasi `Flask` agar dapat digunakan

dalam proses penjadwalan. Fungsi `update_jadwal_service` bertugas untuk membuat atau memperbarui jadwal berdasarkan waktu dan jenis aksi yang ditentukan. Data jadwal kemudian disimpan atau diperbarui melalui fungsi `repository`, dan jika berhasil, informasi jadwal akan dikembalikan dalam format yang telah diformat sesuai waktu lokal. Program ini memungkinkan sistem untuk menjalankan aksi penyiraman atau pengabutan ¹⁸⁴ secara otomatis pada waktu yang telah ditentukan.

4. Listing program Notifikasi

```
def notify_sensor_data_Service(msg, app=None):
    load_dotenv	override=True)
    wa_server_url = os.getenv('WA_SERVER_URL')
    session_id = os.getenv('WA_SESSION_ID')
    if not wa_server_url:
        print("WA_SERVER_URL tidak ditemukan di .env")
        return
    if not session_id:
        print("WA_SESSION_ID tidak ditemukan di .env")
        return
    with app.app_context():

        try:
            numbers = get_all_nomor_hp()
            if not numbers:
                print("Tidak ada nomor WA yang ditemukan di database.")
                return
            for record in numbers:
                phone_number = record.nomor_hp if hasattr(record, 'nomor_hp') else
record['nomor_hp']
                payload = {
                    "number": phone_number,
                    "message": msg,
```

```

    "sessionId": session_id
}

112 headers = {
    "Content-Type": "application/json"
}

try:
    response = requests.post(wa_server_url, json=payload,
headers=headers)

    print(f"✉️ Mengirim pesan ke {phone_number}...")
    print(f"Payload: {payload}")
    response.raise_for_status()
    res_json = response.json()
    if res_json.get("status") == "success":
        print(f"✅ Pesan berhasil dikirim ke {phone_number}")
    else:
        print(f"❌ Gagal kirim ke {phone_number}: {res_json}")
85 except requests.exceptions.RequestException as err:
85     print(f"❌ Error koneksi ke {phone_number}: {err}")

except Exception as e:
    print(f"❌ Error saat mengambil nomor dari database: {e}")

```

Program ini digunakan untuk mengirimkan notifikasi ke *WhatsApp* secara otomatis. Fungsi `notify_sensor_data_Service` akan membaca konfigurasi URL server *WhatsApp* dan *session ID* dari file `.env`. Setelah itu, program mengambil seluruh nomor *WhatsApp* yang tersimpan di *database*, kemudian mengirimkan pesan ke setiap nomor tersebut menggunakan permintaan *HTTP POST* ke server *WhatsApp* yang telah disiapkan. Pesan dikirim dalam format *JSON* berisi nomor penerima, isi pesan, dan *session ID*. Fungsi ini berjalan dalam konteks aplikasi *Flask* untuk memastikan akses database berjalan dengan benar. Jika pengiriman berhasil, sistem akan menampilkan pesan konfirmasi di konsol; jika gagal, pesan kesalahan akan ditampilkan.

5. Listing program *MQTT*

```
# Memuat variabel lingkungan dari file .env
load_dotenv()
# 🔒 Konfigurasi MQTT & Flask API
BROKER = os.environ.get('MQTT_BROKER', "")
PORT = 8883
145 USERNAME = os.environ.get('MQTT_USERNAME')
PASSWORD = os.environ.get('MQTT_PASSWORD')
FLASK_URL = os.environ.get('FLASK_URL')
# MQTT_service.py
app_context = None
TOPICS = [
    ("otomatis/siram/status", 1),
    ("otomatis/kabut/status", 1),
49    ("sensor/kelembapan_tanah", 1),
    ("sensor/suhu_udara", 1),
    ("sensor/kelembapan_udara", 1)
]
# 🔔 Jalankan MQTT
def run_MQTT_service(app_instance):
    global app_context
    app_context = app_instance
    global client
    print(f"🔧 Menghubungkan ke MQTT {BROKER}:{PORT}")
154    client = paho.Client(client_id="otomatisasi_penyiraman",
    protocol=paho.MQTTv5)
    client.username_pw_set(USERNAME, PASSWORD)
103    client.tls_set(ca_certs=certifi.where(),
    tls_version=ssl.PROTOCOL_TLS_CLIENT)
    client.on_connect = on_connect
    client.message_callback_add("sensor/kelembapan_tanah",
```

```

handle_sensor_data)
    client.message_callback_add("sensor/suhu_udara", handle_sensor_data)
    client.message_callback_add("sensor/kelembapan_udara",
handle_sensor_data)
101
try:
    client.connect(BROKER, PORT)
    client.loop_start()
except Exception as e:
    print(f"🔴 Gagal koneksi: {e}")
    socketio.emit("MQTT_error", {"error": str(e)})
    return
try:
    while True:
        now = datetime.now()
        for jam in check_times:
            if now >= next_check_time[jam]:
                print(f"⌚ Menjalankan pengecekan sensor untuk jam ke-{jam}")
                check_sensor_status()
                # Jadwalkan ulang jam ke-jam berikutnya
                next_check_time[jam] += timedelta(days=1)
                time.sleep(60) # cek setiap 1 menit apakah waktunya eksekusi
except KeyboardInterrupt:
    print("🔴 Menutup koneksi MQTT...")
    client.disconnect()
    client.loop_stop()
# 🖥 Perintah manual

```

Program ini berfungsi sebagai penghubung antara *server* dan perangkat *ESP32* menggunakan protokol *MQTT*. Konfigurasi broker *MQTT*, termasuk alamat *server*, *port*, *username*, dan *password*, diambil dari file *.env*. Fungsi *run_MQTT_service* digunakan untuk memulai koneksi ke broker *MQTT* menggunakan *TLS* dan mengatur topik-topik tertentu yang akan didengarkan,

³⁶ seperti data dari sensor suhu, kelembapan udara, dan kelembapan tanah. Program juga menambahkan callback khusus untuk menangani data sensor yang diterima. Setelah koneksi berhasil, sistem masuk ke dalam *loop* yang berjalan terus-menerus untuk memantau waktu tertentu guna menjalankan pengecekan status sensor secara otomatis. Jika koneksi ke *broker* gagal, sistem akan menampilkan pesan kesalahan dan mengirim notifikasi ke antarmuka *web* melalui *socketio*. Program ini memungkinkan komunikasi dua arah antara perangkat *IoT* dan *server*, mendukung pengiriman dan penerimaan data secara *real-time*.

6. Listing program *Arduino ESP32*

```
// === Callback MQTT ===  
51  
void callback(char* topic, byte* payload, unsigned int length) {  
    String message;  
    for (unsigned int i = 0; i < length; i++) {  
        message += (char)payload[i];  
    }  
    if (String(topic) == topic_siram_status) {  
        digitalWrite(RELAY1_PIN, message == "1" ? LOW : HIGH);  
    } else if (String(topic) == topic_kabut_status) {  
        digitalWrite(RELAY2_PIN, message == "1" ? LOW : HIGH);  
    }  
}  
  
// === Koneksi ke WiFi ===  
29  
void setup_WiFi() {  
    delay(10);  
    Serial.println();  
    Serial.print("Menghubungkan ke ");  
    Serial.println(ssid);  
    WiFi.begin(ssid, password);  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500);  
    }
```

```
Serial.print(".");
}

Serial.println("");
Serial.println("WiFi terhubung");
Serial.println("Alamat IP: ");
Serial.println(WiFi.localIP());
}

// === Setup Awal ===

void setup() {
93 Serial.begin(115200);
dht.begin();
pinMode(RELAY1_PIN, OUTPUT);
pinMode(RELAY2_PIN, OUTPUT);
digitalWrite(RELAY1_PIN, HIGH); // Relay non-aktif (aktif LOW)
digitalWrite(RELAY2_PIN, HIGH);
setup_WiFi();
espClient.setInsecure(); // Nonaktifkan verifikasi sertifikat SSL (hanya untuk
pengujian)
144 client.setServer(MQTT_server, MQTT_port);
client.setCallback(callback);
}

// === WIFI ===
23 void reconnect() {
while (!client.connected()) {
Serial.print("Menghubungkan ke MQTT...");
if (client.connect("ESP32Client", MQTT_user, MQTT_password)) {
Serial.println("terhubung");
client.subscribe(topic_siram_status);
client.subscribe(topic_kabut_status);
} else {
Serial.print("gagal, rc=");
Serial.print(client.state());
}
}
}
```

```

Serial.println(" coba lagi dalam 5 detik");
delay(5000);
}
}
}
// === Loop ===
void loop() {
if (!client.connected()) {
reconnect();
}
client.loop();
// Baca sensor
457
float suhu = dht.readTemperature();
float kelembapan = dht.readHumidity();
int nilaiSensorTanah = analogRead(SOIL_MOISTURE_PIN);
// Konversi nilai kelembapan tanah ke persentase
int kelembapanTanah = map(nilaiSensorTanah, DRY_VALUE,
WET_VALUE, 0, 100);
kelembapanTanah = constrain(kelembapanTanah, 0, 100);
// Kirim data ke MQTT
3
client.publish(topic_suhu_udara, String(suhu).c_str(), true);
client.publish(topic_kelembapan_udara, String(kelembapan).c_str(), true);
client.publish(topic_kelembapan_tanah, String(kelembapanTanah).c_str(),
true);
delay(3000); // Kirim data setiap 3 detik
}
}

```

Program ini dijalankan di perangkat *ESP32* untuk membaca data sensor dan berkomunikasi dengan *server* menggunakan protokol *MQTT*. Saat perangkat dinyalakan, *ESP32* akan terhubung ke jaringan *WiFi* dan kemudian menyambung ke broker *MQTT*. Dua topik utama, yaitu *topic_siram_status* dan *topic_kabut_status*, digunakan untuk menerima perintah dari *server*. Ketika pesan

diterima, *ESP32* akan mengaktifkan atau menonaktifkan *Relay* sesuai perintah yang dikirim, yang mengontrol alat penyiraman dan pengabutan. Di dalam fungsi *loop*, *ESP32* secara berkala membaca suhu ¹⁰ dan kelembapan udara dari sensor *DHT*, serta kelembapan tanah dari sensor analog. Nilai-nilai ini kemudian dikirimkan ke *server* melalui *MQTT* setiap 3 detik, menggunakan topik yang telah ditentukan. Program juga dilengkapi mekanisme *reconnect()* untuk memastikan koneksi *MQTT* tetap aktif secara otomatis jika terjadi pemutusan. Dengan demikian, sistem ini memungkinkan komunikasi dua arah yang stabil antara perangkat *IoT* dan *server* untuk keperluan monitoring serta kontrol otomatis.

³² 5.1.2 Implementasi Sistem

Implementasi sistem adalah tahap penerapan sistem yang akan dilakukan jika sistem telah disetujui termasuk program yang telah dibuat pada tahap perancangan sistem agar siap untuk dioperasikan. Adapun waktu dan tempat penerapan sistem yang sudah dibuat sebagai berikut:

1. Waktu dan tempat implementasi

Tempat : Avicenna *Greenhouse*

Alamat : Kp. Padarek Rt. 03 Rw. 02 ⁵⁶ Desa Drawati Kec. Paseh Kab. Bandung, Prov. Jawa Barat

Waktu : 5 Juli 2025

⁵ 5.1.3 Spesifikasi Sistem

Spesifikasi sistem menjelaskan secara rinci kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam pengembangan serta implementasi sistem penyiraman dan pengabutan otomatis berbasis *IoT*. Informasi ini penting untuk memastikan sistem dapat berjalan secara optimal dalam memantau kondisi lingkungan dan mengaktifkan penyiraman atau pengabutan sesuai parameter yang telah ditentukan, seperti kelembapan tanah, suhu udara, dan kelembapan udara. Spesifikasi sistem dibagi menjadi dua bagian utama, yaitu:

1. 78 Spesifikasi Perangkat Keras

Berikut merupakan *hardware* yang digunakan dalam membangun aplikasi:

3 Tabel 5. 1 Spesifikasi perangkat keras

Processor	Core 2
RAM	2 GB
Disk Penyimpanan	50GB
Sensor	<i>DHT11, Soil moisture</i>
Modul Alat	<i>ESP32, Relay 2 Channel, Pompa air.</i>

2. 5 Spesifikasi Perangkat Lunak

Berikut adalah *perangkat lunak* yang digunakan dalam pembangunan *sistem ini* adalah sebagai berikut:

Tabel 5. 2 Spesifikasi perangkat lunak

Sistem Operasi	Linux Ubuntu 20 / Window 10
Database	<i>MySQL</i>
Bahasa Pemrograman	<i>Python, Arduino</i>

5.1.4 Instalasi Sistem

Instalasi sistem ini memuat penjelasan mengenai langkah-langkah yang harus dilakukan untuk instalasi aplikasi dan database di *VPS*, serta konfigurasi perangkat *IOT* menggunakan mikrokontroler *ESP32*.

1. Instalasi Aplikasi

Instalasi aplikasi dilakukan di dua bagian, yaitu di *server VPS* dan di perangkat *ESP32*. Aplikasi *server* dibangun menggunakan *Python (Flask)*, sementara *ESP32* diprogram melalui *Arduino IDE*.

a. Instalasi Aplikasi

1. Mengakses *VPS* melalui *SSH*
2. Mengupdate sistem (*apt update && apt upgrade*)
3. Menginstal *Python* dan *pip*

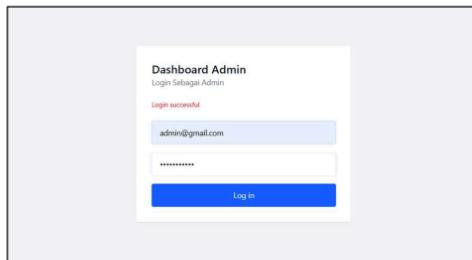
4. Menginstal *virtual environment* (*python3 -m venv venv*)
5. Mengaktifkan *virtual environment*
6. Menginstal *library Python* yang dibutuhkan (*pip install -r requirements.txt*)
7. Mengunggah file aplikasi ke *VPS* (via *Git/SCP*)
8. Masuk ke direktori aplikasi
9. Mengatur *file konfigurasi* (*env/database/MQTT*)
10. Memastikan semua konfigurasi aman.
11. Menjalankan *Flask*
 - b. Instalasi *Database*
 1. Masuk ke *server VPS* menggunakan *SSH* dari komputer lokal.
 2. Melakukan penginstalan *software MySQL* sebagai sistem manajemen basis data (*DBMS*).
 3. Mengakses konsol *MySQL* untuk melakukan konfigurasi awal.
 4. Menyiapkan basis data yang akan digunakan untuk menyimpan data sensor, log aktivitas, dan konfigurasi system.
 5. Menambahkan pengguna baru dengan *username* dan *password* khusus untuk aplikasi *IOT*.
 6. Memberikan hak penuh (*privileges*) kepada user untuk mengelola *database* yang telah dibuat.
 7. Menguji koneksi dari aplikasi ke *database* untuk memastikan konfigurasi berhasil.
 - c. Instalasi Alat *IOT*
 1. Mengunggah program ke *ESP32* melalui *Arduino IDE*
 2. Menyiapkan hotspot/*WiFi* untuk *ESP32*
 3. Menyiapkan *broker MQTT*.
 4. Menguji komunikasi *WIFI* (*subscribe/publish*) dengan *server*
 5. Menempatkan alat dalam *box casing*
 6. Menyambungkan adaptor *ESP32* dan pompa ke terminal listrik
 7. Melakukan uji coba manual (perintah siram/kabut dari *server*)
 8. Menata tata letak posisi alat dan sensor sesuai pada denah *Greenhouse*.
 9. Menyambungkan selang, pompa air dan terminal listrik dengan mikrokontroller

5.1.5 Menjalankan Sistem

Pada bagian ini menjelaskan bagaimana langkah – langkah menjalankan sistem aplikasi:

1. Halaman *Login*

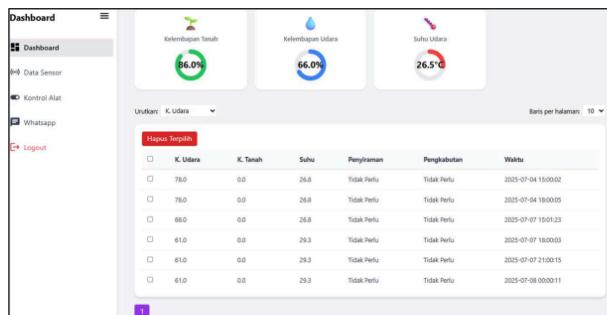
Sebelum mengakses *Dashboard IOT* penyiraman dan pengkabutan, *admin* di ²⁰⁰ haruskan untuk *Login* menggunakan *email* dan *password* yang telah ditetapkan.



Gambar 5. 1 Halaman *Login*

2. Halaman *Dashboard*

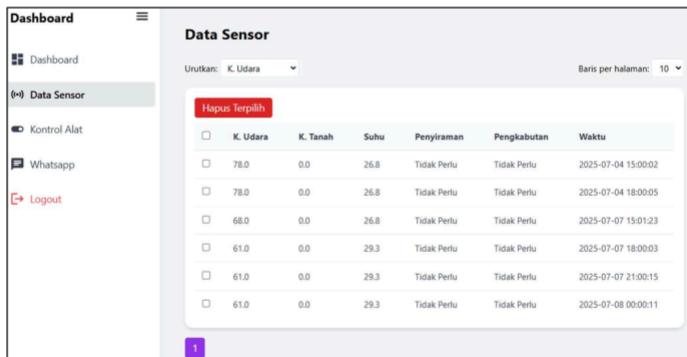
Setelah melakukan *Login*, sistem akan menampilkan halaman awal pada *Dashboard* yang menampilkan data sensor pada *card* komponen dan juga tabel data sensor yang telah dilakukan prediksi *naïve bayes*. Pada tabel dapat melakukan filter data sesuai sensor dan juga dapat menghapus data.



Gambar 5. 2 Halaman *Dashboard*

3. Halaman Data Sensor

Halaman ini menampilkan data hasil pembacaan sensor yang dikirim secara berkala lalu disimpan ke dalam basis data. Sistem dijadwalkan untuk merekam data setiap tiga jam, yaitu pada pukul 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, dan 21:00. Setiap entri data mencakup informasi seperti kelembapan udara, kelembapan tanah, suhu, penyiraman dan pengkabutan serta waktu pengambilan data. Setelah data diterima, sistem secara otomatis menjalankan algoritma *Naive Bayes* untuk melakukan analisis dan klasifikasi terhadap kondisi lingkungan. Hasil dari analisis ini akan menentukan apakah penyiraman dan pengkabutan tanaman diperlukan, dengan hasil keputusan ditampilkan langsung dalam kolom penyiraman dan pengkabutan pada tabel. Sistem menyediakan fitur pengurutan berdasarkan nilai dari masing-masing sensor. Selain itu, terdapat pilihan untuk menentukan jumlah data yang ditampilkan pada tabel. Fitur lainnya adalah kemampuan untuk menghapus data yang tidak lagi diperlukan.



The screenshot shows a web-based application interface. On the left is a sidebar with navigation links: 'Dashboard', 'Data Sensor' (which is currently selected and highlighted in grey), 'Kontrol Alat', 'Whatsapp', and 'Logout'. The main content area has a title 'Data Sensor'. Below it is a table with the following data:

<input type="checkbox"/>	K. Udara	K. Tanah	Suhu	Penyiraman	Pengkabutan	Waktu
<input type="checkbox"/>	78.0	0.0	26.8	Tidak Perlu	Tidak Perlu	2025-07-04 15:00:02
<input type="checkbox"/>	78.0	0.0	26.8	Tidak Perlu	Tidak Perlu	2025-07-04 18:00:05
<input type="checkbox"/>	68.0	0.0	26.8	Tidak Perlu	Tidak Perlu	2025-07-07 15:01:23
<input type="checkbox"/>	61.0	0.0	29.3	Tidak Perlu	Tidak Perlu	2025-07-07 18:00:03
<input type="checkbox"/>	61.0	0.0	29.3	Tidak Perlu	Tidak Perlu	2025-07-07 21:00:15
<input type="checkbox"/>	61.0	0.0	29.3	Tidak Perlu	Tidak Perlu	2025-07-08 00:00:11

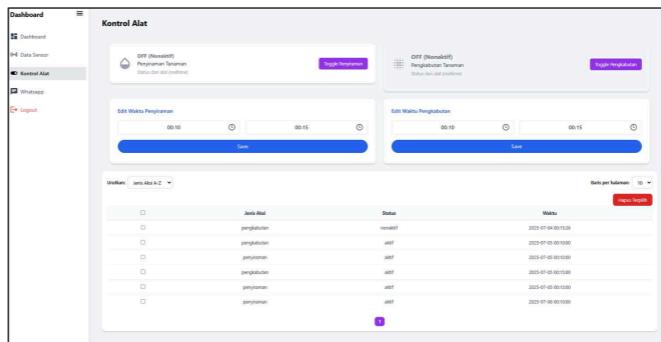
Gambar 5. 3 Gambar Halaman Data Sensor

4. Halaman Kontrol Alat

Halaman ini menyediakan dua mode kontrol, yaitu manual dan otomatis, sehingga pengguna dapat dengan mudah memonitor dan mengendalikan alat sesuai kebutuhan. Di bagian atas halaman, ditampilkan status terkini dari penyiraman dan pengkabutan, apakah dalam kondisi aktif (*ON*) atau nonaktif

(OFF). Pengguna dapat menekan tombol "Toggle penyiraman" dan "Toggle pengkabutan" untuk mengaktifkan atau menonaktifkan alat secara manual. Tombol ini memungkinkan kontrol langsung terhadap perangkat, terutama dalam situasi mendesak atau saat dilakukan uji coba sistem. Terdapat juga fitur pengaturan waktu otomatis untuk penyiraman dan pengkabutan. Pengguna dapat mengatur waktu mulai dan waktu selesai untuk setiap proses, lalu menyimpannya dengan tombol *Save*. Ketika waktu yang ditentukan tercapai, sistem akan mengaktifkan alat secara otomatis sesuai dengan jadwal yang telah diset sebelumnya.

Selain berdasarkan waktu, sistem juga menerapkan logika otomatis berbasis pembacaan sensor untuk menghentikan proses penyiraman dan pengkabutan. Proses penyiraman akan berhenti secara otomatis apabila sensor kelembapan tanah mendeteksi nilai lebih dari 80%. Sementara itu, proses pengkabutan akan dihentikan jika suhu udara turun di bawah 27°C dan kelembapan udara lebih dari 71%. Logika ini memastikan bahwa tanaman tidak mengalami kelebihan air atau kelembapan yang berlebih. Di bagian bawah halaman, tersedia tabel Riwayat Aksi yang mencatat semua aktivitas alat, baik yang dilakukan secara otomatis maupun manual. Setiap data mencakup informasi seperti jenis alat, status (aktif/nonaktif), dan waktu pelaksanaan. Tabel ini dilengkapi dengan fitur filter berdasarkan jenis alat, serta opsi penghapusan data secara selektif melalui tombol *Hapus Terpilih*, setelah pengguna mencantang baris data yang diinginkan.

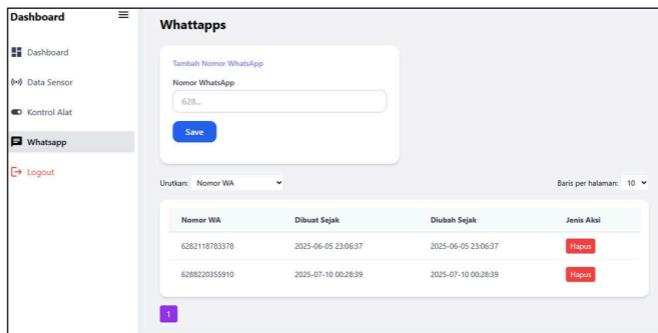


Gambar 5. 4 Gambar Halaman Kontrol Alat

5. Halaman *WhatsApp*

Pada halaman ini, admin dapat menambahkan nomor *WhatsApp* baru dengan mengisi kolom input yang tersedia, kemudian menekan tombol Save. Nomor yang ditambahkan akan langsung masuk ke dalam daftar, dan sistem akan mulai mengirimkan notifikasi ke nomor tersebut secara otomatis. Tabel yang terdapat di bawah form input menampilkan daftar seluruh nomor *WhatsApp* yang telah terdaftar. Informasi yang disajikan meliputi nomor *WhatsApp*, tanggal dan waktu saat nomor tersebut dibuat, serta kapan terakhir kali diperbarui. Setiap baris juga dilengkapi dengan tombol Hapus yang memungkinkan pengguna menghapus nomor tertentu dari daftar jika tidak lagi diperlukan.

Selain itu, sistem menyediakan fitur pengurutan berdasarkan nomor *WhatsApp* dan pengaturan jumlah baris data yang ingin ditampilkan per halaman. Hal ini memudahkan pengguna dalam mencari, mengelola, dan mengorganisir daftar kontak penerima notifikasi.



The screenshot shows a user interface for managing WhatsApp numbers. On the left, there's a sidebar with options: Dashboard, Data Sensor, Kontrol Alat, WhatsApp (which is selected and highlighted in grey), and Logout. The main area has a title 'Whattapps' and a sub-section 'Tambah Nomor WhatsApp'. It contains a text input field labeled 'Nomor WhatsApp' with the value '628...' and a blue 'Save' button. Below this is a table with two rows of data:

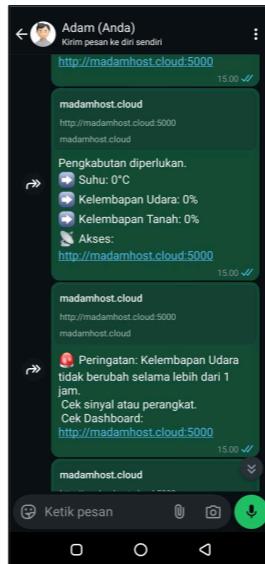
Nomor WA	Dibuat Sejak	Dilubah Sejak	Jenis Aksi
6282118733378	2025-06-05 23:06:37	2025-06-05 23:06:37	Hapus
62885220355910	2025-07-10 00:28:39	2025-07-10 00:28:39	Hapus

Gambar 5. 5 Halaman *WhatsApp*

6. Notifikasi *WhatsApp*

Fitur Notifikasi *WhatsApp* merupakan bagian dari sistem monitoring *IoT* yang berfungsi untuk mengirimkan pemberitahuan secara otomatis kepada pengguna melalui pesan *WhatsApp*. Pesan notifikasi dikirimkan secara otomatis oleh *server* berdasarkan analisis data sensor yang diterima. Terdapat beberapa jenis notifikasi yang dikirim, antara lain:

- a. Pertama, sistem akan mengirimkan peringatan kondisi sensor, misalnya jika suhu udara, kelembapan udara, atau kelembapan tanah tidak berubah selama lebih dari satu jam. Notifikasi ini bertujuan untuk menginformasikan kemungkinan gangguan pada sensor atau koneksi perangkat, sehingga pengguna dapat segera melakukan pengecekan.
- b. Kedua, sistem mengirimkan notifikasi prediksi kebutuhan penyiraman atau pengkabutan berdasarkan hasil perhitungan dari algoritma *Naive Bayes*. Jika kondisi lingkungan menunjukkan bahwa penyiraman atau pengkabutan ³⁶ diperlukan, maka pengguna akan menerima pesan yang menjelaskan nilai suhu, kelembapan udara, dan kelembapan tanah secara detail, serta status akses perangkat.
- c. Ketiga, sistem juga mengirimkan pesan ketika penyiraman atau pengkabutan diaktifkan, baik berdasarkan jadwal otomatis maupun intervensi manual oleh pengguna. Notifikasi ini mencantumkan waktu dan tanggal pengaktifan alat, sehingga pengguna dapat memverifikasi aktivitas.



Gambar 5. 6 Notifikasi *WhatsApp*

5.2 Pengujian

Setelah tahap perancangan dan pembuatan aplikasi selesai, kemudian penulis melakukan pengujian dengan menggunakan metode pengujian *black-box testing*¹⁶⁰ supaya bisa memastikan bahwa fitur utama berfungsi sesuai dengan spesifikasi. Pengujian dilakukan pada alat mikrokontroller dan aplikasi monitoring, di sajikan dalam dua tabel. Hasil pengujian sistem ditampilkan pada tabel berikut:

1. Pengujian Alat Mikrokontroller

Tabel 5. 3 Pengujian alat mikrokontroller

No	Item	Skenario	Hasil yang diharapkan	Hasil Pengujian
1	<i>ESP32</i>	Mengirim dan menerima data melalui <i>MQTT</i> saat terhubung ke WiFi	<i>ESP32</i> berhasil publish dan subscribe ke topik <i>MQTT</i>	Valid
2	<i>DHT11</i>	Membaca suhu dan kelembapan udara	Data suhu dan kelembapan udara tampil di aplikasi dan dapat memicu pengkabutan jika suhu > 27°C dan kelembapan udara < 71%	Valid
3	<i>Soil moisture</i>	Membaca tingkat kelembapan tanah	Data kelembapan tanah tampil di aplikasi dan dapat memicu penyiraman jika < 80%	Valid
4	<i>Relay</i>	Menerima perintah ON/OFF dari <i>ESP32</i> sesuai perintah	<i>Relay</i> aktif dan nonaktif sesuai perintah dari	Valid

		<i>MQTT</i>	aplikasi (otomatis atau manual)	
5	Pompa air penyiraman	Aktif saat perintah penyiraman dikirim dan kelembapan tanah < 80%	Pompa menyala saat kondisi terpenuhi dan berhenti saat kelembapan > 80%	Valid
6	Pompa air pengkabutan	Aktif saat suhu udara > 27°C dan kelembapan < 71% serta menerima perintah pengkabutan	Pompa menyala saat kondisi terpenuhi dan berhenti saat suhu < 27°C dan kelembapan udara > 71%	Valid

2. Pengujian *Dashboard Monitoring*

Tabel 5. 4 Pengujian *Dashboard Monitoring*

No	Item	Skenario	Hasil yang diharapkan	Hasil Pengujian
1	<i>Login Admin</i>	Admin menginputkan <i>email</i> dan <i>password</i> yang valid	Sistem menampilkan halaman <i>Dashboard</i>	Valid
2	Menampilkan data sensor	Sistem menerima data <i>MQTT</i> dari <i>ESP32</i> dan menampilkannya pada <i>card</i> komponen <i>html</i>	Data suhu, kelembapan tanah, dan kelembapan udara tampil secara <i>realtime</i>	Valid
3	Menyimpan data sensor, prediksi	Sistem menyimpan data sensor dan hasil prediksi dari algoritma <i>Naive</i>	Data sensor dan hasil prediksi tersimpan	Valid

	penyiraman dan pengkabutan menggunakan algoritma <i>naïve bayes</i>	<i>Bayes</i> ke database secara otomatis setiap jam (03:00, 06:00, 09:00, 12:00, 15:00, 18:00, 21:00, 00:00)	otomatis setiap jam pada waktu yang telah ditentukan	
4.	Menghapus data sensor dan prediksi pada tabel	Admin menekan tombol hapus pada tabel data sensor	Data yang terpilih akan terhapus dari tabel dan <i>database</i>	valid
5	Mengaktifkan atau mematikan penyiraman melalui tombol penyiraman	Admin menekan tombol penyiraman manual	<i>Relay</i> pompa air aktif/nonaktif, dan status penyiraman dikirim ke mikrokontroller	Valid
6	Mengaktifkan atau mematikan pengkabutan melalui tombol pengkabutan	Admin menekan tombol pengkabutan manual	<i>Relay</i> pompa kabut aktif/nonaktif, dan status pengkabutan dikirim ke mikrokontroller	Valid
7	Mengedit waktu penyiraman	Admin mengatur jadwal penyiraman otomatis	Jadwal penyiraman diperbarui dan disimpan, sistem mengeksekusi sesuai waktu yang diatur	Valid
8	Mengedit	Admin mengatur jadwal	Jadwal	Valid

	waktu pengkabutan	pengkabutan otomatis	pengkabutan diperbarui dan disimpan, sistem mengeksekusi sesuai waktu yang diatur	
9	Menyimpan otomatis Riwayat aksi penyiraman atau pengkabutan	Setiap kali penyiraman/pengkabutan aktif, sistem menyimpan data riwayat ke <i>database</i>	Riwayat tercatat otomatis lengkap dengan waktu, jenis aksi, dan metode pemicunya (otomatis/manual)	Valid
10	Menghapus data Riwayat aksi	Admin menghapus salah satu atau semua riwayat	Riwayat terhapus dari tampilan dan database	Valid
11	Menambahkan nomor untuk notifikasi <i>WhatsApp</i>	Admin memasukkan nomor telepon dan menyimpannya	Nomor berhasil disimpan ke database dan ditampilkan dalam daftar penerima notifikasi	Valid
12	Menghapus nomor untuk notifikasi <i>WhatsApp</i>	Admin menghapus salah satu nomor dari daftar	Nomor tidak lagi muncul dan tidak menerima notifikasi lagi	Valid
13	Menampilkan notifikasi <i>WhatsApp</i>	Sistem mendeteksi kondisi pemicu (prediksi siram/kabut, suhu/kelembapan dan mengirimkan notifikasi	<i>WhatsApp</i> terkirim ke seluruh nomor yang terdaftar sesuai isi pesan	Valid

		ke WhatsApp	yang telah ditentukan	
14	Logout	Admin ³⁰ menekan tombol <i>logout</i>	Sistem keluar dari sesi dan kembali ke halaman Login	Valid

3. Pengujian *Cross Validation* dan Akurasi Model

Pengujian model dilakukan untuk mengevaluasi tingkat akurasi model *Naive Bayes* dalam memprediksi dua variabel target, yaitu penyiraman dan pengkabutan, berdasarkan data sensor ¹⁶⁸ lingkungan yang terdiri dari suhu udara, kelembapan udara, dan tanah. Dataset yang digunakan terdiri dari 202 baris data, masing-masing memiliki lima label, yaitu: Suhu Udara, Kelembapan Udara, Kelembapan Tanah, Penyiraman, dan Pengkabutan.

Metode pengujian yang digunakan adalah *5-Fold Cross Validation*. Teknik ini melakukan pembagian pada ¹⁵⁹ dataset menjadi lima bagian (*Fold*) yang memiliki jumlah data yang seimbang. Pada setiap iterasi, satu *Fold* digunakan sebagai data uji (*testing set*), sedangkan empat *Fold* lainnya digunakan sebagai data latih (*training set*). Proses ini diulang sebanyak lima kali sehingga setiap *Fold* memiliki kesempatan yang sama untuk menjadi data uji. Setelah seluruh iterasi selesai, nilai akurasi dari setiap iterasi dihitung, kemudian dirata-ratakan untuk memperoleh nilai evaluasi akhir dari performa model.

Penggunaan nilai $k = 5$ pada *cross validation* dipilih karena *dataset* yang digunakan cukup besar dan pemilihan *5-Fold* dinilai mampu memberikan estimasi performa model yang seimbang antara kompleksitas perhitungan dan ketelitian hasil evaluasi.

Model yang digunakan dalam pengujian ini adalah *Gaussian Naive Bayes*, yang dibungkus menggunakan *MultiOutputClassifier* karena terdapat dua variabel target (*multilabel*). Evaluasi dilakukan secara terpisah untuk masing-masing target dengan menggunakan metrik *accuracy score*, yaitu metrik yang menghitung seberapa banyak prediksi model yang benar dibandingkan dengan total jumlah data uji.

Berikut adalah hasil pengujian *cross validation* terhadap kedua target:

a. Hasil pengujian *cross validation* penyiraman

Tabel 5. 5 Hasil Pengujian *Cross Validation* Penyiraman

Fold	Akurasi
1	1.0000
2	1.0000
3	1.0000
4	1.0000
5	1.0000
Rata – Rata	1.0000

b. Hasil pengujian *cross validation* pengkabutan

Tabel 5. 6 Hasil Pengujian *Cross Validation* Pengkabutan

Fold	Akurasi
1	1.000
2	0.9512
3	1.000
4	1.000
5	1.000
Rata – rata	0.9902

4. Pengujian Sensor

a. Sensor Kelembapan Tanah

Tabel 5. 7 Pengujian Sensor Kelembapan Tanah

No	Kondisi Tanah	Sensor Soil moisture	Alat Ukur
1	Tanah Berpupuk	69%	Basah
2	Tanah Berpasir	32%	Sangat Kering
3	Tanah Biasa (Liat)	61%	Basah
4	Tanah Campur Kopi	72%	Basah
5	Tanah Campur Garam	58%	Basah
6	Tanah Disiram Air Teh	63%	Basah
7	Tanah Disiram Air Sabun	66%	Basah

8	Tanah ditaburkan kapur	55%	Basah
9	Tanah ditaburkan micin	62%	Basah

Berdasarkan hasil pengujian kelembapan tanah pada berbagai jenis dan perlakuan media tanam, nilai pembacaan sensor *Soil Moisture* berkisar antara 32% hingga 72%. Pada media tanah berpasir, sensor menunjukkan nilai 32% yang dikategorikan sebagai kondisi sangat kering, sesuai dengan sifat tanah pasir yang cepat melepaskan air. Sementara itu, media seperti tanah berpupuk, tanah liat, tanah campur kopi, dan tanah yang disiram air cenderung menghasilkan nilai di atas 60% yang termasuk dalam kategori basah. Beberapa perlakuan lain, seperti penambahan garam dan kapur, menurunkan daya serap air sehingga nilai sensor hanya berada pada kisaran 55–58% yang dikategorikan sebagai lembap. Secara keseluruhan, pengujian ini menunjukkan bahwa sensor *Soil Moisture* dapat mendeteksi variasi kelembapan berdasarkan jenis media tanam dan perlakuan yang diberikan. Nilai sensor berbanding lurus dengan kemampuan tanah dalam menyerap dan menahan air, sehingga alat ini dapat digunakan untuk memantau kondisi kelembapan tanah secara real-time pada berbagai jenis media.

b. Sensor Kelembapan Suhu dan Udara

Tabel 5. 8 Pengujian Sensor Kelembapan Suhu dan Udara

No	Waktu	Suhu	Udara
1	2025-07-02T07:58:49+00	24,1°C	82%
2	2025-07-02T08:00:15+00	23,8 °C	84%
3	2025-07-02T09:05:22+00	27,5 °C	72%
4	2025-07-02T10:30:44+00	30,7 °C	65%
5	2025-07-02T11:15:31+00	29,3 °C	68%
6	2025-07-02T12:20:53+00	31,2 °C	63%
7	2025-07-02T13:45:17+00	32,1 °C	61%
8	2025-07-02T15:02:33+00	30,4 °C	66%
9	2025-07-02T17:18:49+00	28,0 °C	71%
10	2025-07-02T19:25:52+00	25,6 °C	78%

11	2025-07-02T21:40:10+00	23,9 °C	85%
12	2025-07-03T00:05:25+00	21,5 °C	91%
13	2025-07-03T05:10:50+00	22,0 °C	88%

³ Berdasarkan hasil pengujian sensor suhu dan kelembapan udara pada bulan Juli 2025, terlihat adanya pola fluktuasi nilai sesuai dengan waktu pengambilan data. Pada pagi hari (05:00–10:59), suhu berada di kisaran 22,0 °C hingga 30,7 °C dengan kelembapan cukup tinggi antara 65% hingga 88%, menunjukkan kondisi udara yang sejuk dan lembap. Pada siang hari (11:00–14:59), suhu meningkat signifikan menjadi 29,3 °C hingga 32,1 °C, sedangkan kelembapan menurun ke rentang 61% hingga 68%, yang menggambarkan kondisi udara panas dan relatif kering.

Memasuki sore hari (15:00–18:59), suhu berangsut turun ke kisaran 28,0 °C hingga 30,4 °C dengan kelembapan meningkat kembali pada rentang 66% hingga 71%, menandakan udara mulai stabil setelah puncak panas siang. Pada malam hari (19:00–04:59), suhu semakin menurun hingga 21,5 °C dengan kelembapan meningkat tajam mencapai 78% hingga 91%, yang mencerminkan kondisi udara lebih dingin dan lembap. Secara keseluruhan, data ini menunjukkan siklus harian yang konsisten dengan iklim tropis, di mana suhu tertinggi terjadi pada siang hari dengan kelembapan rendah, sedangkan suhu terendah terjadi pada malam hingga dini hari dengan kelembapan tinggi. Hal ini membuktikan bahwa sensor mampu merekam perubahan suhu dan kelembapan secara akurat sesuai kondisi lingkungan nyata.

BAB VI

KESIMPULAN

⁷⁵ 6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan oleh penulis melalui beberapa tahapan pada bab-bab sebelumnya, maka penulis dapat menyimpulkan bahwa:

1. Aplikasi penyiraman dan pengkabutan otomatis berbasis *Internet of Things (IoT)* berhasil dibangun dan diimplementasikan menggunakan mikrokontroler *ESP32*, yang mampu membaca data dari sensor suhu, kelembaban udara, serta kelembaban tanah secara *real-time*. ¹⁰
2. Integrasi sistem dengan algoritma *Naive Bayes* berhasil digunakan untuk melakukan prediksi kebutuhan penyiraman dan pengkabutan secara otomatis berdasarkan parameter lingkungan, sehingga dapat membantu menjaga kondisi optimal untuk pertumbuhan tanaman.
3. Sistem *Dashboard monitoring* berbasis web yang dibangun menggunakan *framework Flask* mampu menampilkan data sensor, hasil prediksi, serta memberikan kontrol manual terhadap penyiraman dan pengkabutan, baik melalui jadwal maupun tombol kontrol.
4. Fitur notifikasi *WhatsApp* berhasil diimplementasikan untuk mengirimkan informasi peringatan, prediksi, dan status aktif penyiraman atau pengkabutan kepada pengguna, sehingga memudahkan pemantauan jarak jauh.
5. Berdasarkan hasil pengujian menggunakan metode *black-box testing*, seluruh fitur sistem berjalan dengan baik *sesuai* skenario *yang* telah ditentukan, termasuk proses penyimpanan data sensor dan riwayat aksi ke *database*. ³⁹

6.2 Saran

Berdasarkan hasil penelitian sistem *IoT* untuk penyiraman dan pengkabutan otomatis berbasis *Naive Bayes*, penulis menyadari bahwa masih terdapat beberapa keterbatasan yang dapat diperbaiki dan dikembangkan ke depannya. Adapun beberapa saran yang dapat dipertimbangkan, antara lain:

1. Disarankan menggunakan jaringan WiFi dengan koneksi internet yang stabil dan cepat, terutama jika sistem akan diterapkan di area perkebunan atau *Greenhouse* yang luas dan jauh dari pusat jaringan, agar komunikasi data antara *ESP32* dan *server MQTT* tetap lancar.
2. Penggunaan lebih banyak sensor kelembapan tanah, kelambapan udara dan suhu akan meningkatkan akurasi lebih tinggi dan dapat meningkatkan keandalan sistem dalam mengambil keputusan penyiraman secara otomatis.
3. Penggunaan perangkat *hosting* dengan ukuran 6 *core cpu*, 8-16gb *ram* akan lebih efisien dan memiliki kinerja tinggi sangat disarankan agar proses pengolahan data sensor, prediksi, serta pengiriman notifikasi dapat berjalan lebih cepat, stabil, dan responsif.
4. Sistem keamanan data dapat ditingkatkan dengan menambahkan autentikasi pengguna, enkripsi koneksi *MQTT*, dan log aktivitas untuk mencegah penyalahgunaan sistem.

DAFTAR PUSTAKA

Ade Roni. (2025). *Pengertian dan Struktur Dasar HTML*. Aderoni.Com.
<https://aderoni.com/pemrograman/pengertian-dan-struktur-dasar-html/>

⁵⁴ AgileTech Vietnam. (2025). *Traditional vs Agile SDLC: 7 Key Practices To Skyrocket Your Project With Agile Model in 2025*. AgileTech Vietnam.
<https://agiletech.vn/traditional-sdlc-vs-agile-sdlc/>

⁹ Alamsyah, R., Ryansyah, E., Permana, A. Y., & Mufidah, R. (2024). SISTEM PENYIRAMAN TANAMAN OTOMATIS MENGGUNAKAN LOGIKA FUZZY DENGAN TEKNOLOGI INTERNET OF THINGS BERBASIS ESP8266 DAN APLIKASI BLYNK. *Jurnal Informatika Dan Teknik Elektro Terapan*, 12(2). <https://doi.org/10.23960/jitet.v12i2.4007>

Ariata C. (2024, May 21). *Apa Itu Hosting Web? Pengertian, Fungsi, dan Jenisnya*. Hostinger. ¹⁷ <https://www.hostinger.com/id/tutorial/apa-itu-web-hosting>

Ayoni Sulthon. (2023, May 27). ¹⁷ *Cara membuat ERD: Simbol, Entitas, Atribut Termudah*. DomaiNesia. <https://www.domainesia.com/berita/pengertian-erd-adalah/>

⁴⁸ Ayu Made Supartini, I., Komang Gde Sukarsa, I., & Gusti Ayu Made Srinadi, I. (2017). ANALISIS DISKRIMINAN PADA KLASIFIKASI DESA DI KABUPATEN TABANAN MENGGUNAKAN METODE K-FOLD CROSS VALIDATION. *E-Jurnal Matematika*, 6(2), 106–115.

Ding, B., Yao, F., Wu, Y., & He, Y. (2012). Improving Flask Implementation Using Hardware Assisted In-VM Isolation. In *IFIP AICT* (Vol. 376).

¹¹ Effendi, N., Ramadhani, W., & Farida, F. (2022). Perancangan Sistem Penyiraman Tanaman Otomatis Menggunakan Sensor Kelembapan Tanah Berbasis IoT. *Jurnal CoSciTech (Computer Science and Information Technology)*, 3(2), 91–98. <https://doi.org/10.37859/coscitech.v3i2.3923>

- Eka Candra, J., & Maulana Universitas Putera Batam, A. (2019). Penerapan Soil Moisture Sensor Untuk Desain System Penyiram Tanaman Otomatis. *Seminar Nasional Ilmu Sosial Dan Teknologi*.
- Elga Aris Prastyo. (2025). *Memahami Struktur Program Arduino untuk Pemula*. PT Teknolab Caraka Internasional.
<https://www.arduinoindonesia.id/2024/05/memahami-struktur-program-arduino-untuk-pemula.html>
- Fahmi, A., Fathul Hadi, C., & Yusa, A. M. (2022). Prototype Sistem Monitoring Suhu Dan Kelembaban Udara Pada Tanaman Cabai Berbasis (IOT). *Zetroem, Vol 04. No 02.*
- Figma, Inc. (2025, May 1). *What is Figma?* Figma, Inc.
<https://help.figma.com/hc/en-us/articles/14563969806359-What-is-Figma>
- Fritzing. (2025, May 5). *Fritizing*. Fritzing.
- GitHub, I. (2025, May 1). *About GitHub and Git*. GitHub, Inc.
<https://docs.GitHub.com/en/get-started/start-your-journey/about-GitHub-and-git>
- Hidayat, W., Ardiansyah, M., & Setyanto, A. (2021). Pengaruh Algoritma ADASYN dan SMOTE terhadap Performa Support Vector Machine pada Ketidakseimbangan Dataset Airbnb. *Edumatic: Jurnal Pendidikan Informatika*, 5(1), 11–20. <https://doi.org/10.29408/edumatic.v5i1.3125>
- Ibnu Daqiqil Id. (2021). *MACHINE LEARNING : Teori , Studi Kasus dan Implementasi Menggunakan Python* (Ibnu Daqiqil Id, Ed.). UR PRESS .
<https://play.google.com/books/reader?id=JvBPEAAAQBAJ&pg=GBS.PR1&hl=id>
- Ifa Susuek Anselmus Talli, W., Dedy Irawan, J., & Xaverius Ariwibisono, F. (2023). RANCANG BANGUN SISTEM MONITORING KUALITAS TANAH UNTUK TANAMAN CABAI BERBASIS IOT (INTERNET OF THINGS). *Jurnal Mahasiswa Teknik Informatika*, 7(5).

¹⁷
Iftitah Nurul Laily. (2022, February 7). *Pengertian Website Menurut Para Ahli, Beserta Jenis dan Fungsinya*. Katadata.Co.Id.
<https://katadata.co.id/lifestyle/edukasi/6200a2a9697ec/pengertian-website-menurut-para-ahli-beserta-jenis-dan-fungsinya>

Indobot Academy. (2023, June 27). *Mengenal Protokol MQTT dan Perbedaan dengan HTTP*. Indobot Academy. <https://blog.indobot.co.id/mengenal-protokol-mqtt-dan-perbedaan-dengan-http/>

Irhan Hisyam Dwi Nugroho. (2024, July 12). *Apa itu UML? Pengertian, Jenis, Fungsi, dan Contoh Diagram*. PT Dibimbing Digital Indonesia. <https://dibimbing.id/blog/detail/apa-itu-uml-definisi-fungsi-jenis-contohnya-lengkap>

Kelasplc. (2023). *Pengertian Internet Of Things : Fitur, Arsitektur dan Contohnya*. Kelas PLC. <https://www.kelaspcl.com/pengertian-internet-of-things/>

Kurniawan. (2023, December 9). *Perbedaan Misting dan Menyiram Tanaman*. DaunSuper. <https://daunsuper.com/perbedaan-misting-dan-menyiram-tanaman/>

⁶²
Kusumah, H., & Pradana, R. A. (2019). PENERAPAN TRAINER INTERFACING MIKROKONTROLER DAN INTERNET OF THINGS BERBASIS ESP32 PADA MATA KULIAH INTERFACING. *Journal Cerita*, 5 No 2.

Liam Aljundi. (2024, January 16). *Using the Arduino Software (IDE)*. Arduino. <https://docs.arduino.cc/learn/startng-guide/the-arduino-software-ide/>

⁹
M. Iqbal Hasan, & Sri Wulandari. (2023). Implementasi Internet of Things (IoT) Pada Sistem Otomatisasi Penyiraman Tanaman Berbasis Mobile. *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, 5(3), 149–161. <https://doi.org/10.28926/ilkomnika.v5i3.573>

Microsoft. (2025, May 5). *What is the difference between Visual Studio Code and Visual Studio IDE?* Visual Studio Studio Code.

- https://code.visualstudio.com/docs/supporting/faq#_what-is-the-difference-between-visual-studio-code-and-visual-studio-ide
- Mozilla Foundation. (2025). *What is CSS?* Mozilla Corporation's.
https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Styling_basics/What_is_CSS
- Mozilla Foundation. (2025, May 5). *Introduction Javascript*. Mdn Web Docs.
<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>
- ¹³ Muhamad Rusdi, Muriani, Rivaldo Pasca Corputty, Mardiyasa Putra Yoga, Grace Christin Aditya Ronsumbre, & Diah Bayu Titisari. (2023). **IMPLEMENTASI TEKNOLOGI PENYIRAMAN SISTEM PENGKABUTAN OTOMATIS DAN MONITORING PINTAR BERBASIS TENAGA SURYA UNTUK TEMPAT BUDIDAYA TANAMAN ANGGREK UDFAIRUS DI KABUPATEN MERAUKE. JURNAL PENGABDIAN MASYARAKAT, 1(Vol. 1 No. 2 (2023): AKSELERASI: Jurnal Pengabdian Masyarakat)**, ¹⁴⁶ 53–59.
<https://doi.org/https://doi.org/10.70210/ajpm.v1i2.40>
- MySQLTutorial.org. (2008). *MySQL Architecture*. MySQLTutorial.Org.
<https://www.mysqltutorial.org/mysql-administration/mysql-architecture/>
- ⁷¹ Nadifa Padantya Raihanah. (2023, January 20). *Apa Itu Figma? Fitur, Kelebihan dan Kekurangan*. Alterra Academy. <https://academy.alterra.id/blog/apa-itu-figma/>
- ¹⁵⁰ Nur Ichsanudin, M., Yusuf, M., & Jurusan Rekayasa Sistem Komputer, S. (2022). **PENGUJIAN FUNGSIONAL PERANGKAT LUNAK SISTEM INFORMASI PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA. Jurnal Ilmiah Teknik Dan Ilmu Komputer, 1(2), 1–8.** <https://doi.org/10.55123>
- Pallets. (2025, May 1). *Welcome to Flask*. Pallets.
<https://flask.palletsprojects.com/en/stable/>

- ³⁴ Peryanto, A., Yudhana, A., & Umar, R. (2020). Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation. *Journal of Applied Informatics and Computing (JAIC)*, 4(1), 45. <http://jurnal.polibatam.ac.id/index.php/JAIC>
- Publikasi, A. J., Mardiansyah, A., Kasah, B. N., Zamzami, H. R., Arabu, Y., Nasro, M. A., Kristanto, N., Paojiah, R., & Wulandari, Y. (2025). ³⁵ PENGENALAN DASAR HTML DAN CSS: LANGKAH PERTAMA DALAM PENGEMBANGAN WEB. *Abdi Jurnal Publikasi*, 3(3), 165–170. ³⁶ <https://jurnal.portalpublikasi.id/index.php/AJP/index>
- ³⁷ python.org. (2025, May 5). *What is Python? Executive Summary*. Python.Org. <https://www.python.org/doc/essays/blurb/>
- Rifqi Mulyawan Digital. (2025). ³⁸ *Pengertian JavaScript: Sejarah, Cara Kerja JavaScript dan Manfaatnya*. Rifqi Mulyawan Digital. <https://rifqimulyawan.com/blog/pengertian-javascript/>
- ³⁹ Roihan, A., Abas Sunarya, P., & Rafika, A. S. (2019). IJCIT (Indonesian Journal on Computer and Information Technology) Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper. *IJCIT (Indonesian Journal on Computer and Information Technology)*, 5(1), 75–82.
- S Nursuwars, F. M., & Sujana, D. G. (2018). ⁴⁰ IoT: Kelembaban Tanah dan Suhu Ruang sebagai Parameter Sistem Otomatis Penyiraman Air Bawah dan Atas Tanah. *Jurnal Transistor Elektro Dan Informatika (TRANSISTOR EI)*, 3(3).
- Solahart Handal. (2025, May 5). *Apa itu Nozzle Sprayer?* Solahart Handal. <https://www.solaharthandal.com/jenis-jenis-nozzle-sprayer/>
- Togi. (2021, September 14). ⁴¹ *Mengenal Bahasa Pemrograman Arduino Secara Lengkap yang Mudah Dipelajari untuk Pemula*. PT Tekno Gemilang Indonesia. <https://toghrl.com/bahasa-pemrograman-arduino/>
- Ucy Sugiarti. (2024, November 12). *Activity Diagram: Komponen, Elemen, Beserta Contohnya*. Lawencon International. <https://www.lawencon.com/Activity-diagram/>

Valerie Lampkin, W. T. L. L. O. S. R. N. S. R. X. G. K. N. K. S. F. M. K. D. L.
¹¹⁰
(2012). *Building Smarter Planet Solutions with MQTT and IBM WebSphere*

¹⁴⁸
MO Telemetry: Vol. (1st Edition). IBM Redbooks.
https://play.google.com/books/reader?id=F_HHAgAAQBAJ&pg=GBS.PR9.w.10.0.32_29&hl=id

⁴⁰
Walid, Moh., & Susanto, A. (2024). Penyiraman Otomatis Menggunakan Arduino Uno pada Tanaman Greenhouse MA. Nurul Khoiroh. *Jurnal Ilmu Komputer Dan Informatika*, 4(1), 11–20. <https://doi.org/10.54082/jiki.121>

⁶⁵
Wardani, Hadi, S., & Budiarto, J. (2021). Rancang Bangun Sistem Monitoring Suhu dan Kelembaban Udara Pada Ruang Server Berbasis Wireless Sensor Network. *JURNAL TEKNOLOGI TERPADU*, 9(2).

Web Idea Solution. (2025). Hire Python Developer to Upscale Your Product Development Capabilities. *Web Idea Solution.*
<https://webideasole.com/python-developer/>

⁴⁴
Yazid Yusuf. (2024, December 16). *Apa Itu MySQL? Pengertian MySQL, Cara Kerja, dan Kelebihannya.* Telkom University.
<https://bif.telkomuniversity.ac.id/apa-itu-mysql/>

LAMPIRAN

Lampiran 1 Hasil Wawancara

Wawancara ini bertujuan untuk memperoleh data penelitian tentang proses penyiraman dan pengkabutan di Avicenna *Greenhouse*.

Berikut adalah detail wawancara yang dilakukan:

Narasumber : Resa Aldiana
Jabatan / Posisi : Pemilik *Greenhouse*
Hari, tanggal : 12 Mei 2025
Lokasi : Avicenna *Greenhouse*
Alamat : Kp. Padarek Rt. 03 Rw. 02 ⁵⁶ Desa Drawati Kec. Paseh Kab. Bandung, Prov. Jawa Barat

Berikut adalah hasil wawancara:

No	Pertanyaan	Jawaban
1	Kapan Avicenna <i>Greenhouse</i> mulai dibangun dan digunakan untuk menanam cabai?	Avicenna <i>Greenhouse</i> mulai dibangun dan digunakan sejak tahun 2024
2	Mengapa Bapak memilih menanam cabai di dalam <i>Greenhouse</i> ?	Supaya tanaman bisa lebih terlindungi dari serangan hama dan cuaca ekstrim.
3	Berapa lama waktu yang dibutuhkan tanaman cabai sampai bisa dipanen?	Tanaman cabai biasanya membutuhkan waktu sekitar 4 bulan dari masa tanam hingga panen.
4	Bagaimana metode penyiraman tanaman dilakukan saat ini	Saat ini penyiraman masih manual. Kami menyalakan keran, lalu air dialirkkan melalui selang ke setiap <i>polybag</i> .
5	Apakah ada permasalahan dalam proses penyiraman tanaman?	Kadang penyiraman tergantung pada kehadiran orang. Jika tidak ada yang menyiram, tanaman bisa kekurangan air.
6	Apakah pengkabutan juga sudah dilakukan di <i>Greenhouse</i> ini?	Saat ini, sistem pengkabutan belum diterapkan sama sekali.
7	Untuk penyiraman pada tanaman dilakukan pada waktu	Biasanya dilakukan satu kali, maksimal dua kali sehari, yaitu jam 7 pagi dan jam 5 sore.

	jam berapa saja?	
8	Bagaimana pandangan Bapak terhadap penggunaan sistem <i>IOT</i> untuk otomatisasi penyiraman dan pengkabutan di <i>Greenhouse</i> ini?	Sangat mendukung penggunaan <i>IOT</i> . Dengan adanya sistem otomatis berbasis sensor suhu dan kelembapan, proses penyiraman dan pengkabutan bisa lebih efisien, hemat tenaga, dan menunjang pertumbuhan tanaman lebih optimal.

Bandung, 12 Mei 2025

Pewawancara

Narasumber

Adam Setiadi

Resa Aldiana

Lampiran 2 Dokumentasi Wawancara

Narasumber : Resa Aldiana
Jabatan : Pemilik *Greenhouse*
Tanggal : 12 Mei 2025
Lokasi : Avicenna *Greenhouse*
Alamat : Kp. Padarek Rt. 03 Rw. 02 Desa Drawati Kec. Paseh Kab.
⁵⁶
Bandung, Prov. Jawa Barat



Lampiran 3 TOR

Sebelum melaksanakan penelitian, penulis mengumpulkan data melalui observasi, wawancara, dan studi pustaka di Avicenna Greenhouse. Penelitian ini difokuskan pada proses penyiraman dan pengkabutan tanaman yang masih dilakukan secara manual. Tujuan dari penelitian ini adalah mengimplementasikan sistem *Internet of Things* untuk mengontrol penyiraman dan pengkabutan secara otomatis, serta memanfaatkan data sensor untuk prediksi dan akurasi proses tersebut. Untuk memastikan fokus penelitian adapun batasan masalah yang ditetapkan adalah sebagai berikut:

1. Penelitian ini berfokus pada pengembangan aplikasi penyiraman dan pengkabutan otomatis berbasis *IOT* untuk tanaman cabai di *Greenhouse* Avicenna.
2. Aplikasi yang dikembangkan memanfaatkan sensor kelembapan tanah, suhu, dan kelembapan udara sebagai parameter utama dalam proses penyiraman dan pengkabutan.
3. Aplikasi ini menggunakan algoritma *Naïve bayes* dalam pengambilan keputusan untuk menentukan kapan penyiraman dan pengkabutan dilakukan berdasarkan data sensor.
4. Pengembangan aplikasi mencakup integrasi dengan platform berbasis web untuk monitoring dan pengendalian sistem secara *real-time*, tetapi tidak mencakup fitur lanjutan seperti rekomendasi pemupukan atau analisis pertumbuhan tanaman.
5. Penelitian ini menggunakan metode *Agile* dalam pengembangan aplikasi guna meningkatkan fleksibilitas dalam implementasi dan evaluasi.

Bandung, 12 Mei 2025

Mahasiswa



Adam Setiadi

Pemilik Avicenna *Greenhouse*



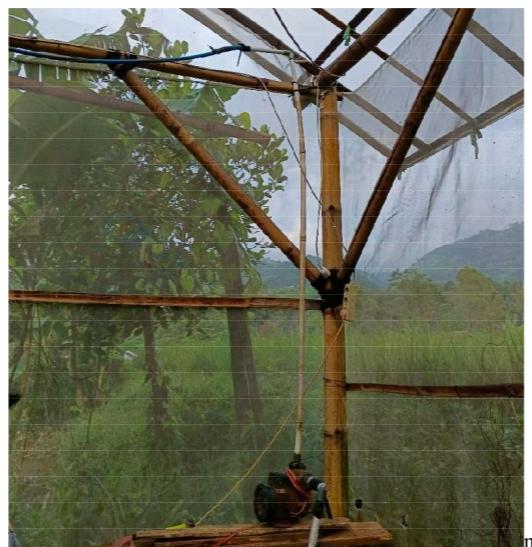
Resa Aldiana

Lampiran 4 Instalasi Alat Di *Greenhouse*

1. Tata Letak Alat Mikrokontroller box



2. Tata Letak Pompa air pengkabutan



3. Tata Letak Pompa air penyiraman



4. Tata Letak Terminal Listrik



5. Tata Letak Selang Penyiraman



6. Tata Letak Selang Pengkabutan



7. Tata Letak Sensor *DHT11*



8. Tata Letak Sensor *Soil moisture*



Lampiran 5 Listing Program

Seluruh kode program yang digunakan dalam penelitian dan pengujian pada skripsi ini disimpan secara lengkap pada *repository GitHub*. *Repository* tersebut memuat seluruh komponen sistem, termasuk pemrosesan data, pelatihan model *Naive Bayes*, pengujian *cross validation*, antarmuka aplikasi serta kode *arduino*. Kode dapat diakses melalui tautan berikut:

Source Code: <https://GitHub.com/madamroger007/flask-penyiraman-dan-pengkabutan.git>

SKRIPSI_IOT_UP2

ORIGINALITY REPORT

19%	17%	8%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.itn.ac.id Internet Source	1%
2	www.scribd.com Internet Source	1%
3	repository.ub.ac.id Internet Source	1%
4	www.coursehero.com Internet Source	<1%
5	Submitted to Universitas Brawijaya Student Paper	<1%
6	Submitted to Universitas Putera Batam Student Paper	<1%
7	jurnal.polibatam.ac.id Internet Source	<1%
8	Submitted to Universitas Pamulang Student Paper	<1%
9	journal.lembagakita.org Internet Source	<1%
10	etheses.uin-malang.ac.id Internet Source	<1%
11	prin.or.id Internet Source	<1%
12	repota.jti.polinema.ac.id Internet Source	<1%

		<1 %
13	e-publisher.my.id Internet Source	<1 %
14	Submitted to Universitas Pancasila Student Paper	<1 %
15	Submitted to Universitas Budi Luhur Student Paper	<1 %
16	journal.eng.unila.ac.id Internet Source	<1 %
17	kc.umn.ac.id Internet Source	<1 %
18	AZHAR AZHAR, Sudi M. Al Sasongko, Djul Fikry Budiman. "IMPLEMENTASI PURWARUPA WIRELESS SENSOR NETWORK UNTUK MONITORING DAN PENYIRAMAN OTOMATIS PADA TANAMAN MINT MENGGUNAKAN ESP32 BERBASIS IoT-LoRa", Jurnal Informatika dan Teknik Elektro Terapan, 2024 Publication	<1 %
19	text-id.123dok.com Internet Source	<1 %
20	Wahyu Firmansyah, Lukman Rosyidi, Sirojul Munir. "Utilization of the Internet of Things in Smart Garden Systems for Plant Control and Monitoring", DBESTI: Journal of Digital Business and Technology Innovation, 2025 Publication	<1 %
21	docplayer.info Internet Source	<1 %

Submitted to Universitas Riau

22

<1 %

23

mc.dfrobot.com.cn

Internet Source

<1 %

24

Rizky Pamungkas, Aulia Ullah, Ahmad Faizal, Hilman Zarory. "Sistem Penyiraman Pintar dan Monitoring Tanaman Sawi Otomatis Berbasis Esp 32 dan Sensor Kelembapan Tanah Menggunakan Metode Fuzzy Mamdani", JURNAL AI-AZHAR INDONESIA SERI SAINS DAN TEKNOLOGI, 2025

Publication

<1 %

25

Tundo, Sodik, Kiki Setiawan, Raisah Fajri Aula. "Penerapan IoT dengan Algoritma Fuzzy dan Mikrokontroler ESP32 dalam Monitoring Penyiraman", Jurnal Indonesia : Manajemen Informatika dan Komunikasi, 2024

Publication

<1 %

26

www.researchgate.net

Internet Source

<1 %

27

[Submitted to fpptijateng](#)

Student Paper

<1 %

28

jim.teknokrat.ac.id

Internet Source

<1 %

29

polynoe.lib.uniwa.gr

Internet Source

<1 %

30

repository.its.ac.id

Internet Source

<1 %

31

ejournal.fhuki.id

Internet Source

<1 %

repository.uniks.ac.id

32	Internet Source	<1 %
33	smart.stmikplk.ac.id Internet Source	<1 %
34	repo.undiksha.ac.id Internet Source	<1 %
35	digilib.uin-suka.ac.id Internet Source	<1 %
36	Andi Budiman, Uray Ristian, Hirzen Hasfani. "Sistem Pemantauan Dan Kendali Pembibitan Anggur Berbasis Internet Of Things (IoT) (Studi Kasus: Smart Garden FMIPA Untan)", Coding Jurnal Komputer dan Aplikasi, 2023 Publication	<1 %
37	id.scribd.com Internet Source	<1 %
38	jurnal.portalpublikasi.id Internet Source	<1 %
39	ipsikom.unipem.ac.id Internet Source	<1 %
40	jiki.jurnal-id.com Internet Source	<1 %
41	journal.thamrin.ac.id Internet Source	<1 %
42	pdfcoffee.com Internet Source	<1 %
43	repo.iainbatusangkar.ac.id Internet Source	<1 %
44	repository.ittelkom-pwt.ac.id Internet Source	<1 %

45	repository.unhas.ac.id Internet Source	<1 %
46	repository.uin-suska.ac.id Internet Source	<1 %
47	repository.unsri.ac.id Internet Source	<1 %
48	www.grafati.com Internet Source	<1 %
49	ejournal.instiki.ac.id Internet Source	<1 %
50	repo.usni.ac.id Internet Source	<1 %
51	Submitted to Escuela Politecnica Nacional Student Paper	<1 %
52	Submitted to Forum Perpustakaan Perguruan Tinggi Indonesia Jawa Timur Student Paper	<1 %
53	Submitted to Universitas Muhammadiyah Palembang Student Paper	<1 %
54	Submitted to York St John University Student Paper	<1 %
55	repository.upnvj.ac.id Internet Source	<1 %
56	www.drawati.desa.id Internet Source	<1 %
57	digilib.unila.ac.id Internet Source	<1 %
58	ejournal.istn.ac.id Internet Source	<1 %

		<1 %
59	idoc.pub Internet Source	<1 %
60	Submitted to Sultan Agung Islamic University Student Paper	<1 %
61	Submitted to Telkom University Student Paper	<1 %
62	sipora.polije.ac.id Internet Source	<1 %
63	www.aditekjayaputra.com Internet Source	<1 %
64	Submitted to Universidad de Guayaquil Student Paper	<1 %
65	ejournal.universitasmandiri.ac.id Internet Source	<1 %
66	library.poltekkesdepkes-sby.ac.id Internet Source	<1 %
67	repository.mercubuana.ac.id Internet Source	<1 %
68	www.kompasiana.com Internet Source	<1 %
69	Submitted to Sekolah Teknik Elektro & Informatika Student Paper	<1 %
70	comserva.publikasiindonesia.id Internet Source	<1 %
71	elibrary.bsi.ac.id Internet Source	<1 %

72	Ridwan Bagus, Ali Romli. "Mobile Health Monitoring Application as an Effort to Detect Stunting in Early Childhood Based on Android", INOVTEK Polbeng - Seri Informatika, 2024	<1 %
Publication		
73	Submitted to Universidad de Burgos UBUCEV	<1 %
	Student Paper	
74	accounting.binus.ac.id	<1 %
	Internet Source	
75	ejournal.unibba.ac.id	<1 %
	Internet Source	
76	intezer.com	<1 %
	Internet Source	
77	repository.atmaluhur.ac.id	<1 %
	Internet Source	
78	id.123dok.com	<1 %
	Internet Source	
79	repository.stmikroyal.ac.id	<1 %
	Internet Source	
80	repository.usni.ac.id	<1 %
	Internet Source	
81	Noverta Effendi, Witri Ramadhani, Fitri Farida. "Perancangan Sistem Penyiraman Tanaman Otomatis Menggunakan Sensor Kelembapan Tanah Berbasis IoT", Jurnal CoSciTech (Computer Science and Information Technology), 2022	<1 %
	Publication	
82	eprints.poltekegal.ac.id	<1 %
	Internet Source	

83	Submitted to poltekssn Student Paper	<1 %
84	123dok.com Internet Source	<1 %
85	Magnus Lie Hetland, Fabio Nelli. "Chapter 29 Activity 5: Web Scraping with Requests and BeautifulSoup", Springer Science and Business Media LLC, 2024 Publication	<1 %
86	Submitted to Middle East College of Information Technology Student Paper	<1 %
87	Muhammad Wildan, Seftian M. Gilang, Novan Wijaya. "PERANCANGAN SMART HOME DENGAN INTERNET OF THINGS (IoT) MENGGUNAKAN CISCO PACKET TRACER", DEVICE : JOURNAL OF INFORMATION SYSTEM, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY, 2025 Publication	<1 %
88	Nur Cahyono, Suprianto Suprianto. "Perancangan Sistem Informasi Penyewaan Alat Outdoor Berbasis Web menggunakan Metode Waterfall pada InOutdoors Rental Sidoarjo", Journal of Internet and Software Engineering, 2024 Publication	<1 %
89	journal.unublitar.ac.id Internet Source	<1 %
90	repository.teknokrat.ac.id Internet Source	<1 %

91	www.theseus.fi Internet Source	<1 %
92	Submitted to SDM Universitas Gadjah Mada Student Paper	<1 %
93	Submitted to University of Derby Student Paper	<1 %
94	eprints.uny.ac.id Internet Source	<1 %
95	journal.irpi.or.id Internet Source	<1 %
96	library.stmikgici.ac.id Internet Source	<1 %
97	repositori.umrah.ac.id Internet Source	<1 %
98	Submitted to unimal Student Paper	<1 %
99	Babeş-Bolyai University Publication	<1 %
100	Submitted to Fakultas Teknik Student Paper	<1 %
101	Submitted to Swinburne University of Technology Student Paper	<1 %
102	Submitted to UIN Sultan Maulana Hasanudin Student Paper	<1 %
103	Submitted to University of Sydney Student Paper	<1 %
104	elib.unikom.ac.id Internet Source	<1 %

105	eprints.umm.ac.id Internet Source	<1 %
106	Submitted to Badan PPSDM Kesehatan Kementerian Kesehatan Student Paper	<1 %
107	Submitted to Defense University Student Paper	<1 %
108	Submitted to Universitas Negeri Semarang Student Paper	<1 %
109	Submitted to Universitas Pelita Harapan Student Paper	<1 %
110	dspace.uii.ac.id Internet Source	<1 %
111	ejournal.uigm.ac.id Internet Source	<1 %
112	forum.chirpstack.io Internet Source	<1 %
113	journal.aritekin.or.id Internet Source	<1 %
114	jurnal.umt.ac.id Internet Source	<1 %
115	Submitted to KYUNG HEE UNIVERSITY Student Paper	<1 %
116	Submitted to STT Wastukancana Purwakarta Student Paper	<1 %
117	Submitted to Universitas Muria Kudus Student Paper	<1 %
118	Submitted to Universitas Musamus Merauke Student Paper	<1 %

- 119 Submitted to University Tun Hussein Onn Malaysia <1 %
Student Paper
-
- 120 digilib.unimed.ac.id <1 %
Internet Source
-
- 121 doku.pub <1 %
Internet Source
-
- 122 git.io-warnemuende.de <1 %
Internet Source
-
- 123 moam.info <1 %
Internet Source
-
- 124 savitrigita.wordpress.com <1 %
Internet Source
-
- 125 Doni Prastyo, Dede Irawan, Imam Halim Mursyidin. "Sistem Informasi Terpusat untuk Manajemen Dokumen, Penelitian, dan Pengabdian kepada Masyarakat", bit-Tech, 2025 <1 %
Publication
-
- 126 Pitjont Tomatala, Petrus Paulus Letsoin, Evangelin Martha Yulia Kadmaer. "The Nursery Technique of Juvenile Sandfish, *Holothuri scabra*", Jurnal Ilmiah PLATAK, 2020 <1 %
Publication
-
- 127 Submitted to Politeknik STMI Jakarta <1 %
Student Paper
-
- 128 Submitted to UNIVERSITAS BUDI LUHUR <1 %
Student Paper
-
- 129 Submitted to Universitas Airlangga <1 %
Student Paper

- 130 eprints.utdi.ac.id <1 %
Internet Source
- 131 naos-be.zcu.cz <1 %
Internet Source
- 132 Lady D. Khartiono, Julius Sampekalo, Winda M. Mingkid. "Physical evaluation on freshwater crayfish, Cherax quadricarinatus, feed using several gluten materials", AQUATIC SCIENCE & MANAGEMENT, 2014 <1 %
Publication
- 133 Submitted to Universitas Pendidikan Ganesha <1 %
Student Paper
- 134 electrician.unila.ac.id <1 %
Internet Source
- 135 jurnal.unprimdn.ac.id <1 %
Internet Source
- 136 repository.nurulfikri.ac.id <1 %
Internet Source
- 137 repository.unanda.ac.id <1 %
Internet Source
- 138 repository.unim.ac.id <1 %
Internet Source
- 139 repository.unwidha.ac.id <1 %
Internet Source
- 140 toghr.com <1 %
Internet Source
- 141 Sofyh Zacksavira, Cindy Anggraeni Br Sebayang, Jonathan Imanuel, Grestia Br Sibarani. "Implementasi Algoritma Decision Tree untuk Deteksi Depresi Pada Pelajar", <1 %

Innovative: Journal Of Social Science Research, 2025

Publication

142	accesson.kr Internet Source	<1 %
143	ejournal.arimbi.or.id Internet Source	<1 %
144	github.com Internet Source	<1 %
145	gitlab.ow2.org Internet Source	<1 %
146	jurnal.staialhidayahbogor.ac.id Internet Source	<1 %
147	library.binus.ac.id Internet Source	<1 %
148	ninercommons.charlotte.edu Internet Source	<1 %
149	repo.iain-tulungagung.ac.id Internet Source	<1 %
150	repository.upnjatim.ac.id Internet Source	<1 %
151	rifqimulyawan.com Internet Source	<1 %
152	www.csauthors.net Internet Source	<1 %
153	Angga Ariawan. "Smart Sprout: Irigasi Cerdas Berbasis ALoT untuk Pertanian Modern dan Ramah Lingkungan", bit-Tech, 2024 Publication	<1 %

- 154 G. R. Kanagachidambaresan. "Internet of Things Using Single Board Computers", Springer Science and Business Media LLC, 2022 <1 %
Publication
-
- 155 Jumawal, Aris Sudianto, Zulkipli, Yupi Kuspandi Putra. "Pengembangan Aplikasi Mobile Sistem Pengendalian Oven Tembakau Berbasis Internet of Things", Infotek: Jurnal Informatika dan Teknologi, 2025 <1 %
Publication
-
- 156 Kusmayuda Kusmayuda, Yus Sholva, Haried Novriando. "SISTEM INFORMASI GEOGRAFIS PENYEBARAN OPTICAL DISTRIBUTION POINT JARINGAN FIBER OPTIC DI PT PLN COMNETS PLUS KPW KALIMANTAN BARAT", Jurnal Informatika dan Teknik Elektro Terapan, 2024 <1 %
Publication
-
- 157 Submitted to Liverpool John Moores University <1 %
Student Paper
-
- 158 Yustina Heny Wardhani, Kirana Anisyah, Yang Agita Rindri, Sari Mubaroh. "APLIKASI PERCAKAPAN BAHASA INGGRIS-BAHASA INDONESIA UNTUK TURIS MANCANEGARA BERBASIS ANDROID", TEKNIMEDIA: Teknologi Informasi dan Multimedia, 2025 <1 %
Publication
-
- 159 begawe.unram.ac.id <1 %
Internet Source
-
- 160 de.scribd.com <1 %
Internet Source

161	eprints.uns.ac.id Internet Source	<1 %
162	finkom.repository.unbin.ac.id Internet Source	<1 %
163	innovpedagogy.od.ua Internet Source	<1 %
164	penerbit.aksaraglobal.co.id Internet Source	<1 %
165	perpusteknik.com Internet Source	<1 %
166	repositori.uin-alauddin.ac.id Internet Source	<1 %
167	repository.itk.ac.id Internet Source	<1 %
168	repository.unpas.ac.id Internet Source	<1 %
169	rohmadi.com Internet Source	<1 %
170	sipil.polines.ac.id Internet Source	<1 %
171	suara.independentnews.id Internet Source	<1 %
172	uiye.gilead.org.il Internet Source	<1 %
173	www.e-journal.stmiklombok.ac.id Internet Source	<1 %
174	www.springerprofessional.de Internet Source	<1 %

- 175 Akhmad Zulkifli, Yulisman Yulisman, Uci Rahmalisa. "Penerapan Teknologi Informasi untuk Keterampilan Pertanian Tanaman Hidroponik Berbasis Internet of Things pada Siswa SLB Negeri Pembina Pekanbaru", Bertuah: Jurnal Pengabdian kepada Masyarakat, 2024
Publication <1 %
- 176 Dwi Cita Septia Candra, Lutfi Syarifullah, Muhammad Nur Faiz. "SISTEM INFORMASI PEMBAYARAN UANG SEKOLAH DENGAN MODEL MVC DAN MENGGUNAKAN NOTIFIKASI SMS GATEWAY", Jurnal Sistem Informasi dan Informatika (Simika), 2021
Publication <1 %
- 177 Fitra Rahim, Yusuf Ramadhan Nasution, Supiyandi. "Implementasi Algoritma ChaCha20 Pada Pengamanan File Citra Bitmap", JURNAL FASILKOM, 2024
Publication <1 %
- 178 Indriani M, Hisma Abduh, Rinto Suppa. "SISTEM PENGONTROLAN PENYIRAM TANAMAN BERBASIS INTERNET OF THINGS", Jurnal Informatika dan Teknik Elektro Terapan, 2025
Publication <1 %
- 179 Johar Saputra Irsandi, Iskandar Fitri, Novi Dian Nathasia. "Sistem Informasi Pemasaran dengan Penerapan CRM (Customer Relationship Management) Berbasis Website menggunakan Metode Waterfall dan Agile", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2021
Publication <1 %

-
- 180 Satrio Bagus Prakoso. "Rancang Bangun Smart Doorbell Berbasis Internet of Things (IoT) Menggunakan ESP32-CAM via Telegram", YASIN, 2025
Publication <1 %
-
- 181 Uray Ristian, Hirzen Hasfani, Uray Syaziman Kesuma Wijaya. "IMPLEMENTASI CHANGE DETECTION UNTUK EFISIENSI PENGIRIMAN PAKET DATA PADA IOT MASTER-SLAVE DI GREENHOUSE", Djtechno: Jurnal Teknologi Informasi, 2024
Publication <1 %
-
- 182 adoc.pub
Internet Source <1 %
-
- 183 anoatik.uho.ac.id
Internet Source <1 %
-
- 184 databaseartikel.com
Internet Source <1 %
-
- 185 dirdosen.budiluhur.ac.id
Internet Source <1 %
-
- 186 docobook.com
Internet Source <1 %
-
- 187 dokumen.pub
Internet Source <1 %
-
- 188 e-journal.hamzanwadi.ac.id
Internet Source <1 %
-
- 189 iqbalramon.blogspot.com
Internet Source <1 %
-
- 190 jurnal.fmipa.unila.ac.id
Internet Source <1 %

191	kurzy.kpi.fei.tuke.sk	<1 %
Internet Source		
192	mafiadoc.com	<1 %
Internet Source		
193	ph01.tci-thaijo.org	<1 %
Internet Source		
194	rama.unimal.ac.id	<1 %
Internet Source		
195	repo.palcomtech.ac.id	<1 %
Internet Source		
196	repository.bsi.ac.id	<1 %
Internet Source		
197	repository.pnj.ac.id	<1 %
Internet Source		
198	repository.poltekbangplg.ac.id	<1 %
Internet Source		
199	repository.uin-malang.ac.id	<1 %
Internet Source		
200	wongbantur.blogspot.com	<1 %
Internet Source		
201	www.omall.co.ke	<1 %
Internet Source		
202	www.uni-oldenburg.de	<1 %
Internet Source		
203	Arnan Dwi Pangestu, Herliyani Hasanah, Moh. Muhtarom. "Rancang Bangun Sistem Monitoring dan Kendali Tanaman Pintar Berbasis Android", Generation Journal, 2025	<1 %
	Publication	

- 204 Supriadi Sahibu, Surahman, Andani Ahmad. "Teknologi Irrigasi Berbasis IoT: Integrasi Sensor Nirkabel (SN), Energi Surya dan Logika Fuzzy", JURNAL FASILKOM, 2025 **<1 %**
Publication
-
- 205 Gilang Setiawan, Joni Maulindar, Afu Ichsan Pradana. "Perancangan Sistem Monitoring Penyiraman Dan Pemupukan Pada Tanaman Hias Sri Rezeki", INTECOMS: Journal of Information Technology and Computer Science, 2024 **<1 %**
Publication
-
- 206 Handa Gustiawan, Muhammad Ridwan Effendi. "Pengembangan Aplikasi Sistem Informasi Pelatihan Pegawai Berbasis Website Pada PT Inacon Luhur Pertiwi Jakarta", Jurnal Teknik dan Science, 2025 **<1 %**
Publication
-
- 207 Jumadi Mabe Parenreng, Andi Ferry Adlian Tri Andani, Muhammad Yahya, Fhatiah Adiba. "Internet-based Design of Hydroponic Plants Monitoring and Automation Control Systems", Internet of Things and Artificial Intelligence Journal, 2024 **<1 %**
Publication
-
- 208 Yuriya Kato. "The Influence of the Hubbard U Term in the Titanium Dioxide Model on DFT+U calculation", American Chemical Society (ACS), 2025 **<1 %**
Publication
-

Exclude quotes Off

Exclude bibliography Off

Exclude matches Off