

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

**SKRIPSI**

**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  
UNIVERSITAS BALE BANDUNG  
BANDUNG**

**2025**

## ABSTRAK

Perkembangan teknologi *Internet of Things (IoT)* telah membawa dampak besar dalam 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.

Penelitian ini 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 sensor suhu udara, kelembapan udara, dan kelembapan tanah untuk memantau kondisi lingkungan secara langsung dan terus-menerus. Data dari sensor diolah oleh algoritma untuk menentukan apakah perlu dilakukan penyiraman atau pengkabutan, sesuai hasil klasifikasi kondisi. Sistem ini juga dilengkapi fitur notifikasi melalui antarmuka web yang memungkinkan pengguna memantau dan mengontrol sistem dari jarak 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 pengujian menunjukkan bahwa sistem dapat 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 ini tidak hanya meningkatkan efektivitas proses penyiraman dan pengkabutan tanaman, tetapi juga mendukung untuk praktik pertanian presisi yang hemat sumber daya dan dapat diterapkan oleh petani di *greenhouse*.

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

## **ABSTRACT**

*The development of Internet of Things (IoT) technology has had a major impact on various fields, including the agricultural sector. One prominent implementation is the automation of plant watering and misting systems, especially in greenhouse environments. This condition requires consistent and sustainable environmental management so that plant growth remains optimal and the agricultural products produced are of high quality and efficient.*

*This study aims to design and implement an automated system for watering and misting chili plants based on IoT, with the Naïve Bayes algorithm as a prediction method in decision making. The system uses air temperature, air humidity, and soil moisture sensors to monitor environmental conditions directly and continuously. Data from the sensors is processed by the algorithm to determine whether watering or misting is necessary, according to the results of the condition classification. This system is also equipped with a notification feature via a web interface that allows users to monitor and control the system remotely. 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 work, and maintain microclimate stability in the greenhouse to support optimal and sustainable plant growth.*

*The test results show that the system can run automatically, respond quickly to changes in environmental conditions, and be easily controlled through a web-based application. The application of this system in the Avicenna Greenhouse shows an increase in the efficiency of water use, a reduction in manual intervention, and an improvement in the stability of temperature and humidity in the greenhouse. This system not only increases the effectiveness of the plant watering and fogging process, but also supports for resource-efficient precision farming practices that can be applied by farmers in the greenhouse.*

**Keywords:** *Internet of Things, Misting, Naïve Bayes, Watering.*

## KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT atas rahmat dan hidayah-Nya yang melimpah, sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul “IMPLEMENTASI *INTERNET OF THINGS* UNTUK PENYIRAMAN DAN PENGKABUTAN OTOMATIS PADA TANAMAN MENGGUNAKAN ALGORITMA *NAIVE BAYES* (Studi Kasus di Avicenna *Greenhouse*)”. Shalawat serta salam tidak lupa disampaikan kepada junjungan Nabi Muhammad SAW yang senantiasa memberikan teladan dan petunjuk yang luhur dalam setiap aspek kehidupan.

Ucapan terima kasih yang tulus disampaikan kepada kedua Orang Tua yang senantiasa memberikan kasih sayang, dukungan, dan doa yang tidak terbatas. Serta, ucapan terima kasih juga penulis sampaikan kepada pihak yang telah membantu dalam penyusunan laporan skripsi, di antaranya:

1. Bapak Yudi Herdiana, S.T., M.T., selaku Dekan Fakultas Teknologi Informasi Universitas Bale Bandung.
2. Bapak Yusuf Muharam, S.Kom., M.Kom., selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Bale Bandung Sekaligus dosen pembimbing utama.
3. Bapak Yaya Suharya, S.Kom, M.Kom., selaku pembimbing pendamping.
4. Segenap Bapak/Ibu Dosen di Fakultas Teknologi Informasi.
5. Semua pihak yang telah membantu dan memberikan dukungan kepada penulis untuk menyelesaikan usulan penelitian ini.

Penulis menyadari usulan penelitian ini masih jauh dari sempurna, untuk itu kritik dan saran yang membangun penulis harapkan demi perbaikan dimasa yang akan datang. Akhir kata, penulis berharap semoga laporan skripsi ini diterima dan bermanfaat kepada berbagai pihak.

Bandung, Mei 2025

Penulis

## DAFTAR ISI

<b>ABSTRAK .....</b>	<b>i</b>
<b>ABSTRACT .....</b>	<b>ii</b>
<b>KATA PENGANTAR.....</b>	<b>iii</b>
<b>DAFTAR ISI.....</b>	<b>iv</b>
<b>DAFTAR TABEL .....</b>	<b>vi</b>
<b>DAFTAR GAMBAR.....</b>	<b>vii</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>viii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
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
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>8</b>
2.1    Landasan Teori .....	8
2.2    Dasar Teori .....	13
2.2.1 <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> .....	16
2.2.5    Algoritma <i>Naïve Bayes</i> .....	17
2.2.6    ESP32.....	19
2.2.7 <i>Soil Moisture Sensor</i> .....	20
2.2.8    DHT11 .....	21
2.2.9 <i>Nozzle Sprayer</i> .....	22
2.2.10 <i>Relay</i> .....	23
2.2.11 <i>Unified Modeling language</i> .....	24
2.2.12 <i>Website</i> .....	27
2.2.13 <i>Hosting</i> .....	28

2.2.14	<i>Metode Agile</i> .....	29
2.2.15	<i>Github</i> .....	30
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> .....	38
2.2.23	<i>Draw.io</i> .....	39
2.2.24	<i>Visual Studio Code</i> .....	40
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
<b>BAB III METODOLOGI PENELITIAN .....</b>		<b>46</b>
3.1	Kerangka Pikir.....	46
3.2	Deskripsi.....	47
3.2.1	Identifikasi Masalah .....	47
3.2.2	Pengumpulan Data .....	47
3.2.3	Perencanaan.....	47
3.2.4	Perancangan .....	50
3.2.5	Pengembangan .....	51
3.2.6	Pengujian.....	52
3.2.7	Penerapan .....	52
3.2.8	Peluncuran.....	52
3.2.9	Pembuatan Laporan.....	53
<b>DAFTAR PUSTAKA .....</b>		<b>54</b>
<b>LAMPIRAN.....</b>		<b>58</b>

## DAFTAR TABEL

Tabel 2. 1 Acuan Jurnal Penelitian .....	8
Tabel 2. 2 Pengujian Kelembapan Tanah .....	14
Tabel 2. 3 Pengujian DHT11 .....	15
Tabel 2. 4 <i>Use Case Diagram</i> .....	25
Tabel 2. 5 <i>Activity Diagram</i> .....	26
Tabel 2. 6 <i>Entity Relationship Diagram</i> .....	27
Tabel 3. 1 Perangkat Keras .....	49
Tabel 3. 2 Perangkat <i>Internet of Things</i> .....	49
Tabel 3. 3 Perangkat Lunak .....	49

## DAFTAR GAMBAR

Gambar 2. 1 Konsep <i>Internet of Things</i> .....	13
Gambar 2. 2 Skema <i>Artifial Intelligence dan Machine Learning</i> .....	17
Gambar 2. 3 ESP32 .....	19
Gambar 2. 4 <i>Soil Mastuire Sensor</i> .....	21
Gambar 2. 5 DHT11 .....	22
Gambar 2. 6 <i>Nozzle Sprayer</i> .....	23
Gambar 2. 7 <i>Relay</i> .....	24
Gambar 2. 8 <i>Arsitektur Website</i> .....	28
Gambar 2. 9 <i>Metode Agile</i> .....	29
Gambar 2. 10 <i>GitHub</i> .....	31
Gambar 2. 11 <i>Bahasa Pemrograman Arduino</i> .....	32
Gambar 2. 12 <i>Arduino IDE</i> .....	33
Gambar 2. 13 <i>Arsitektur MQTT</i> .....	34
Gambar 2. 14 <i>Arsitektur MySQL</i> .....	35
Gambar 2. 15 <i>Tampilan Figma</i> .....	36
Gambar 2. 16 <i>Arsitektur Python</i> .....	37
Gambar 2. 17 <i>Arsitektur 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> .....	41
Gambar 2. 21 <i>Arsitektur HTML</i> .....	42
Gambar 2. 22 <i>Arsitektur CSS</i> .....	43
Gambar 2. 23 <i>Arsitektur Javascript</i> .....	44
Gambar 3. 1 <i>Kerangka Pikir</i> .....	46



## DAFTAR LAMPIRAN

Lampiran 1: Hasil Wawancara.....	58
Lampiran 2: Dokumentasi Wawancara.....	60
Lampiran 3: TOR .....	61

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam era digital saat ini, *Internet of Things (IoT)* telah banyak diterapkan dalam berbagai bidang, termasuk pertanian. Salah satu implementasi *IoT* yang semakin berkembang adalah sistem penyiraman dan pengkabutan otomatis yang bertujuan untuk meningkatkan efisiensi dan efektivitas penggunaan air. Dengan memanfaatkan sensor dan algoritma *machine learning*, sistem ini mampu menyesuaikan kondisi penyiraman dan pengkabutan secara *real-time*. Serta dilakukan penerapan algoritma *Naïve Bayes* ke dalam sistem otomatisasi penyiraman tanaman berdasarkan data sensor dapat signifikan meningkatkan efisiensi dan efektifitas dalam manajemen penggunaan air (Alamsyah et al., 2024).

Penelitian ini dilaksanakan di *Avicenna Greenhouse*, yang berlokasi di Kp. Padarek Rt.03 Rw.02, 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, yaitu pada pukul 07.00–08.00 pagi dan 16.00–17.00 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 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 berbasis *Internet of Things* yang dapat mengatur penyiraman dan kelembapan udara secara otomatis, agar pengelolaan tanaman di *Avicenna Greenhouse* menjadi lebih efektif dan efisien.

Beberapa penelitian terdahulu dengan topik yang sama telah dilakukan oleh beberapa peneliti, seperti penelitian yang dilakukan oleh (Alamsyah et al., 2024), Mengungkapkan bahwa di Indonesia, banyak petani masih mengandalkan metode manual sehingga proses penyiraman belum optimal. Penelitian Oleh (M. Iqbal Hasani & Sri Wulandari, 2023), Mengungkapkan bahwa penyiraman tanaman umumnya masih dilakukan secara manual oleh tenaga manusia menggunakan peralatan sederhana seperti gayung, selang dan ember yang memerlukan waktu dan usaha yang signifikan. Penelitian Oleh (Muhamad 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 menunjukkan adanya kebutuhan akan solusi teknologi yang mampu meningkatkan efisiensi dan efektivitas dalam kegiatan budidaya tanaman. Dibandingkan dengan penelitian-penelitian terdahulu, penelitian ini 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* digunakan karena sesuai penelitian ini untuk melakukan klasifikasi dalam mengolah data dari sensor, sehingga dapat membantu dalam pengambilan keputusan seperti menentukan kebutuhan penyiraman dan pengkabutan. Oleh karena itu, penelitian ini difokuskan pada implementasi *Internet of Things* yang dapat mengotomatisasi proses penyiraman dan pengkabutan tanaman di dalam *greenhouse*.

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 *Naive Bayes* sebagai prediksi suhu ruangan, kelembapan tanah pada pot tanaman, kelembapan udara pada ruangan. Sistem ini dikembangkan menggunakan metode *agile* yang terdiri dari perencanaan, perancangan, pengembangan, pengujian, penerapan, peluncuran. Dalam proses perancangannya, aplikasi ini menggunakan sensor suhu, kelembapan udara dan kelembapan tanah. Data dari sensor tersebut dapat dilakukan prediksi untuk penyiraman dan pengkabutan pada tanaman supaya efektif menggunakan algoritma *naive bayes*. Aplikasi dikembangkan menggunakan bahasa pemrograman *Arduino*, *Flask Python*. Hasil dari penelitian ini 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 diharapkan dapat memberikan kemudahan untuk pemilik dalam melakukan penyiraman dan pengkabutan otomatis di *greenhouse*. Kesimpulannya, penelitian ini bertujuan 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*)”.

## 1.2 Rumusan Masalah

Berdasarkan uraian pada latar belakang, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana mengimplementasikan *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 *monitoring* berbasis web yang dapat membantu pengguna memantau kondisi lingkungan *greenhouse* secara *real-time* serta mengendalikan sistem penyiraman dan pengkabutan secara otomatis?

## 1.3 Batasan Masalah

Adapun batasan masalah terhadap penelitian yang diajukan adalah sebagai berikut:

1. Penelitian ini berfokus 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. Penelitian ini menggunakan metode *Agile* dalam pengembangan aplikasi guna meningkatkan fleksibilitas dalam implementasi dan evaluasi.

## 1.4 Tujuan Penelitian

Berdasarkan rumusan masalah dan batasan masalah diatas, maka tujuan penelitian ini adalah sebagai berikut:

1. Mengimplentasikan *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 memungkinkan pengguna memantau kondisi *greenhouse* secara *real-time* dan mengendalikan sistem penyiraman dan pengkabutan secara otomatis.

## 1.5 Metodologi Penelitian

### 1. Metode Pengumpulan Data

Beberapa Metode pengumpulan data digunakan untuk mendapatkan data yang akurat yaitu sebagai berikut:

#### a. Observasi

Metode ini dilakukan dengan mengamati secara langsung kondisi dan kegiatan 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 dilakukan dengan pihak-pihak yang terlibat langsung dalam pengelolaan *greenhouse*, seperti pemilik *greenhouse* atau petani.

#### c. Studi Pustaka

Studi pustaka dilakukan dengan mengumpulkan dan mempelajari literatur dari berbagai sumber seperti jurnal, artikel ilmiah, buku, dan dokumentasi terkait *IoT*, penyiraman otomatis, 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. Metode Pengembangan Sistem

Metode pengembangan perangkat lunak yang digunakan dalam penelitian ini adalah metode *Agile*. *Agile* merupakan metode yang mengandalkan proses berulang dan bertahap dalam siklus pengembangan perangkat lunak atau *Software Development Life Cycle (SDLC)*. Tujuannya adalah 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, pengembangan, pengujian, penerapan, peluncuran.

### 1.6 Sistematika Penulisan

Dalam menyusun laporan skripsi ini diatur dan disusun dalam enam bab, yang masing-masing terdiri dari beberapa sub bab. Adapun urutannya sebagai berikut:

#### **BAB I PENDAHULUAN**

Bagian ini berisi mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penulisan.

#### **BAB II TINJAUAN PUSTAKA**

Bagian ini menjelaskan tentang landasan teori dan dasar teori pendukung dalam penelitian. Tinjauan Pustaka ini bersumber dari buku, jurnal dan website.

#### **BAB III: METODOLOGI PENELITIAN**

Bab ini menjelaskan metodologi penelitian yang dipakai pada tahap-tahap penulis dalam melakukan penelitian di Avicenna *Greenhouse*.

#### **BAB IV: ANALISIS DAN PERANCANGAN**

Bab ini berisi tentang analisis, perancangan *hardware* dan perancangan perangkat lunak, perancangan database, perancangan antarmuka serta penjelasan tentang perancangan perangkat lunak yang akan di bangun.

## **BAB V IMPLEMENTASI DAN PENGUJIAN**

Bab ini berisi penyajian tahap pengembangan aplikasi yang akan dijelaskan tiap langkahnya dan menampilkan tampilan dari setiap fitur dari aplikasi yang dibuat serta penjelasan bagaimana aplikasi berjalan.

## **BAB VI PENUTUP**

Bab ini berisi kesimpulan dan penyajian tahap pembuatan yang dilakukan serta saran untuk implementasi *Internet of Things* selanjutnya.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Landasan Teori

Penulis mempelajari penelitian terdahulu sebagai acuan dalam penelitian yang akan dilakukan, hal ini bertujuan untuk melakukan perbandingan mengenai kesamaan dan perbedaan terhadap penelitian yang terdahulu. Berikut adalah beberapa acuan jurnal penelitian yang digunakan:

Tabel 2. 1 Acuan Jurnal Penelitian

No	Jurnal Penelitian	Masalah	Metode	Kesimpulan
1	<p><b>Judul:</b> Sistem Penyiraman Tanaman Otomatis Menggunakan Logika <i>Fuzzy</i> Dengan Teknologi <i>Internet of Things</i> Berbasis Esp8266 Dan Aplikasi <i>Blynk</i></p> <p><b>Penulis:</b> Ridho Alamsyah, Eddy Ratna Mufidah Ryansyah, Andari Yasinta Permana.</p>	<p>Di Indonesia, banyak petani masih mengandalkan metode manual sehingga proses penyiraman belum optimal.</p>	<p>Penelitian ini menggunakan metode studi literatur, perancangan perangkat, logika <i>fuzzy</i> yaitu Dalam penelitian ini dihasilkan keputusan menggunakan <i>fuzzy logic</i> yang terdiri dari beberapa tahap yaitu fuzzifikasi, mesin inferensi, dan defuzzifikasi. Pada <i>output</i> yang dihasilkan adalah keputusan akhir apakah pompa air otomatis akan menyala atau tidak</p>	<p>Penelitian ini memanfaatkan sensor kelembapan tanah untuk mendeteksi tingkat kelembapan 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 <i>Blynk</i> yang</p>

	<b>Tahun:</b> 2024		dan pengujian perangkat.	berfungsi sebagai antarmuka untuk memantau serta mengendalikan proses penyiraman secara <i>real-time</i> melalui perangkat <i>mobile</i> .
2	<b>Judul:</b> Implementasi <i>Internet of Things (IoT)</i> Pada Sistem Otomatisasi Penyiraman Tanaman Berbasis <i>Mobile</i> <b>Penulis:</b> M. Iqbal Hasani, Sri Wulandari. <b>Tahun:</b> 2023	Penyiraman tanaman umumnya masih dilakukan secara manual oleh tenaga manusia menggunakan peralatan sederhana seperti gayung, selang dan ember yang memerlukan waktu dan usaha yang signifikan	Penelitian ini mengimplementasikan algoritma <i>Naïve Bayes</i> digunakan untuk menganalisis data yang diakuisisi dari sensor dan menentukan tindakan yang harus dilakukan oleh sistem. Lalu melibatkan penerapan metodologi <i>System Development Life Cycle (SDLC)</i> , yang terdiri dari beberapa tahap, yaitu Analisis Kebutuhan, Desain Sistem, Implementasi, Pengujian, Peluncuran dan Pemeliharaan.	Sistem ini memanfaatkan sensor kelembapan tanah untuk mendeteksi tingkat kelembapan, 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 memantau dan mengendalikan proses penyiraman secara <i>real time</i> . Selain itu, penerapan algoritma <i>Naïve</i>

				<i>Bayes</i> terbukti mampu mendukung pengambilan keputusan berdasarkan data sensor secara akurat.
3	<p><b>Judul:</b></p> <p>Implementasi Teknologi Penyiraman Sistem Pengkabutan Otomatis dan <i>Monitoring</i> Pintar Berbasis Tenaga Surya untuk Tempat Budidaya Tanaman Anggrek UD fairus Di Kabupaten Merauke.</p> <p><b>Penulis:</b></p> <p>Muhamad Rusdi, Muriani, Rivaldo Pasca Corputty, Mardiyasa Putra Yoga, Grace Christin Aditya Ronsumbre,</p>	<p>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 kelembapan dan suhu pada area budidaya. Selain</p>	<p>Metode pelaksanaan yang digunakan pada pengabdian ini terdiri dari beberapa tahapan yaitu: diskusi dan observasi, Sosialisasi Program kegiatan dan teknologi, perancangan teknologi, <i>workshop</i> pengoperasian dan pemeliharaan teknologi, evaluasi kegiatan, publikasi dan capaian luaran kegiatan</p>	<p>Penelitian ini memanfaatkan sensor kelembapan tanah untuk mendeteksi tingkat kelembapan lingkungan serta sensor suhu untuk memantau suhu di sekitar tanaman. Sistem yang dikembangkan berfokus pada otomatisasi proses penyiraman dan pengkabutan tanaman secara efisien. Selain itu, aplikasi <i>Blynk</i> digunakan sebagai antarmuka pemantauan dan pengendalian sistem secara <i>real-time</i>, sehingga</p>

	Diah Bayu Titisari. <b>Tahun:</b> 2023	itu, meningkatnya biaya operasional pemeliharaan tanaman anggrek yang disebabkan penggunaan mesin penyiraman berbahan bakar bensin.		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 otomatisasi penyiraman dan pengkabutan tanaman. Penelitian oleh Alamsyah et al. (2024) mengembangkan sistem penyiraman otomatis berbasis ESP8266 dan aplikasi *Blynk*. Sistem tersebut menggunakan sensor kelembapan tanah untuk mengukur tingkat kelembapan media tanam dan mengaktifkan penyiraman secara otomatis, yang dapat dipantau secara *real-time* melalui perangkat *mobile*. Penelitian oleh Hasani dan 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 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, berbeda 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 udara, dan kelembapan tanah 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 menggunakan metode *Agile*, memungkinkan penyesuaian dinamis dalam setiap tahap pengembangan. Dengan pendekatan ini, penelitian ini memberikan kontribusi inovatif terhadap sistem otomasi pertanian

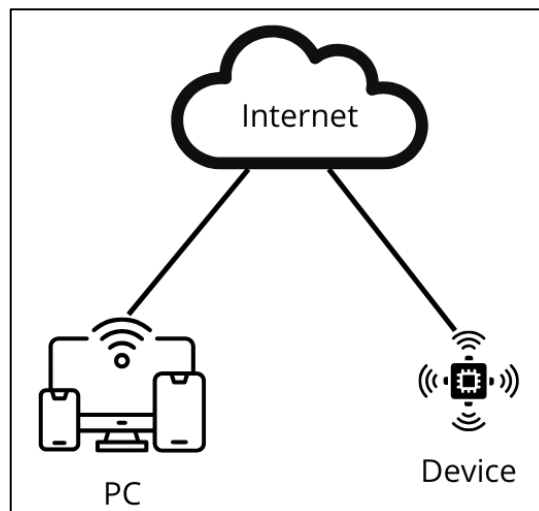
berbasis *IoT* yang lebih efisien, fleksibel, dan responsif terhadap kebutuhan pengguna.

## 2.2 Dasar Teori

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

### 2.2.1 *Internet of Things*

*Internet of Things* didefinisikan kemampuan perangkat yang dapat berkomunikasi, terhubung, dan bertukar data melalui Internet. *Internet of things* adalah teknologi yang memungkinkan adanya kontrol, komunikasi, kerja sama dengan berbagai perangkat keras, data di jaringan internet. (Ifa Susuek Anselmus Talli et al., 2023).



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 internet. Perangkat-perangkat tersebut mengirimkan dan menerima data melalui jaringan internet, sehingga memungkinkan terjadinya integrasi sistem, otomatisasi proses, dan pengambilan keputusan berbasis data secara *real-time*.

### 2.2.2 Penyiraman Tanaman

Menyiram tanaman berarti memberikan air secara langsung ke tanaman. Tujuannya adalah memasok air dan nutrisi yang dibutuhkan akar agar diserap dan didistribusikan ke seluruh bagian tanaman. Menyiram tanaman dilakukan dengan berbagai cara, seperti menuangkan atau meneteskan 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 Berpupuk	66%	Basah
2	Tanah Berpasir	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 2.2 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 yang dicampur 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 adalah salah satu teknik penting dalam merawat tanaman, terutama tanaman hias dan tanaman yang sensitif terhadap perubahan kelembapan. Teknik ini dilakukan dengan cara menyemprotkan butiran air yang sangat halus ke permukaan daun dan batang tanaman secara merata. (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	27,2 °C	71%
9	2022-08-02T07:02:49+00:00	27,8 °C	70%
10	2022-08-02T07:03:25+00:00	19,3 °C	96%
11	2022-08-02T07:03:46+00:00	27,2 °C	86%
12	2022-08-02T07:04:07+00:00	27,3 °C	71%



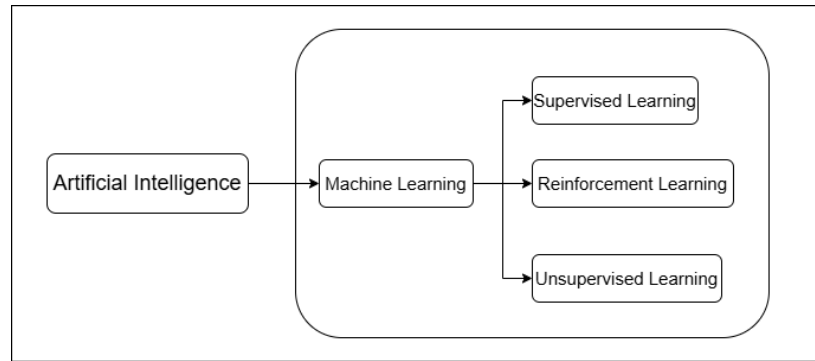
Berdasarkan hasil uji coba terhadap sensor DHT11 yang dilakukan di greenhouse budidaya tanaman cabai, diperoleh data suhu dan kelembapan udara yang bervariasi. Hasil pengujian sebagaimana tercantum pada Tabel 2.3 menunjukkan bahwa sensor mampu mendeteksi suhu dalam rentang 19,3°C hingga 30,4°C dan kelembapan 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 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 27,3°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 (Fahmi et al., 2022).

#### **2.2.4 Machine Learning**

*Machine Learning (ML)* merupakan cabang kecerdasan buatan yang memungkinkan sistem belajar dari data dan membuat keputusan atau prediksi tanpa diprogram secara eksplisit. *ML* merupakan bidang studi yang fokus kepada desain dan analisis algoritma sehingga memungkinkan komputer untuk dapat belajar. Menurut Samuel, *Machine Learning* berisi sebuah algoritma yang bersifat generik atau umum di mana algoritma tersebut dapat menghasilkan sesuatu yang menarik atau bermanfaat dari sejumlah data tanpa harus menulis kode yang spesifik. (Ibnu Daqiqil Id, 2021).



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 *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*. *Supervised Learning* menggunakan dataset berlabel, di mana model dilatih untuk memetakan *input* ke *output* berdasarkan data yang telah diketahui hasilnya. *Unsupervised Learning* bekerja tanpa label dan bertujuan menemukan struktur atau 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 memiliki keunggulan masing-masing tergantung pada karakteristik data dan tujuan analisis yang diinginkan. Dengan memilih pendekatan yang tepat, proses pembelajaran mesin dapat memberikan hasil yang lebih optimal dan relevan terhadap kebutuhan penelitian.

### 2.2.5 Algoritma *Naïve Bayes*

*Naive Bayes* merupakan salah satu algoritma klasifikasi dalam *machine learning* yang didasarkan pada *Teorema Bayes*, yaitu sebuah metode 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 *Naive Bayes* memiliki beberapa varian yang digunakan sesuai dengan tipe data yang dianalisis:

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

Pada penelitian ini, varian yang digunakan adalah *Gaussian Naive Bayes*, yang merupakan salah satu tipe *Naive Bayes*. Prinsip dasarnya sama dengan *Naive Bayes*, namun pada *Naive Bayes* kita tidak dapat menghitung probabilitas data *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  $\mu_y$  adalah rata-rata (*mean*) dan  $\sigma_y^2$  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 adalah nama dari mikrokontroler yang dirancang oleh perusahaan teknologi berbasis 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 dilengkapi dengan port USB Type-C dan Micro USB 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 proyek ini, ESP32 digunakan sebagai pusat kendali sistem *IoT* yang bertugas menerima data dari sensor suhu, kelembapan udara, dan kelembapan tanah. Data yang diterima akan diolah secara lokal, kemudian ESP32 mengambil keputusan untuk mengaktifkan aktuator seperti pompa air (untuk penyiraman) atau kabut (untuk pengkabutan), serta mengirimkan data ke *broker MQTT* melalui jaringan WiFi. Dengan kemampuannya yang multifungsi, ESP32 tidak hanya memproses data dan mengendalikan perangkat, tetapi juga menjadi penghubung antara perangkat keras dan platform monitoring berbasis web secara *real-time*.

### 2.2.7 Soil Moisture Sensor

*Soil moisture* sensor adalah sensor kelembapan yang dapat mendeteksi kelembapan dalam tanah. Sensor ini sangat sederhana, tetapi ideal untuk memantau taman kota, atau tingkat air pada tanaman pekarangan. Sensor ini terdiri dua *probe* untuk melewati arus melalui tanah, kemudian membaca resistansinya untuk mendapatkan nilai tingkat kelembapan. Semakin banyak air membuat tanah lebih mudah menghantarkan listrik (resistansi kecil), sedangkan tanah yang kering sangat sulit menghantarkan listrik (resistansi besar). Sensor ini sangat membantu untuk mengingatkan tingkat kelembapan pada tanaman atau memantau kelembapan tanah (Eka Candra & Maulana Universitas Putera Batam, 2019).



Gambar 2. 4 *Soil Moisture 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 kelembapan dari sensor ini akan dibaca oleh mikrokontroler dan diubah menjadi data persentase kelembapan tanah. Data tersebut selanjutnya dapat digunakan untuk mengambil keputusan, seperti mengaktifkan sistem penyiraman otomatis ketika kelembapan tanah turun di bawah ambang batas yang telah ditentukan.

### 2.2.8 DHT11

Sensor DHT11 adalah modul sensor yang mendeteksi suhu dan kelembapan udara. Modul sensor ini termasuk ke dalam elemen perangkat *resistive* seperti alat pengukur suhu NTC. Cara kerja sensor dht11 dengan mengukur perubahan resistansi yang disebabkan oleh suhu dan kelembapan udara, kemudian mengirimkan data sinyal digital yang terdiri dari pulsa tinggi dan pulsa rendah ke mikrokontroler yang mewakili 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* adalah bagian dari alat semprot yang terdiri dari pipa yang menghubungkan tangki dengan ujung alat semprot. *Nozzle* ini berfungsi untuk mengubah aliran cairan menjadi semprotan dengan pola tertentu dan tekanan yang sesuai dengan kebutuhan. Bentuk dari *nozzle* bisa berbagai macam, seperti *konus*, *flat fan*, dan *full cone*. Setiap bentuk *nozzle* memiliki pola semprotan yang berbeda-beda dan cocok untuk kebutuhan yang berbeda-beda (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. Kombinasi dengan mikrokontroler dan sensor kelembapan, seperti ESP32 dan DHT11, memungkinkan pengoperasian *nozzle* secara otomatis berdasarkan kebutuhan kelembapan tanah yang terdeteksi.

#### 2.2.10 *Relay*

Modul *Relay* adalah sebuah saklar magnet, yang berfungsi untuk memutuskan dan menghubungkan arus listrik. Prinsip kerja *relay* secara umum sama dengan kontaktor magnet yaitu berdasarkan kemagnetan yang dihasilkan oleh kumparan *coil* (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 atau sistem kabut. Modul ini memiliki *optocoupler* dan *transistor* sebagai penguat sinyal dari mikrokontroler, serta jumper pengatur mode aktif. *Optocoupler* (juga disebut *optoisolator*) adalah komponen elektronik yang berfungsi untuk menghubungkan dua rangkaian listrik tanpa kontak langsung, melainkan menggunakan cahaya. *Transistor* adalah komponen elektronik yang bisa bekerja sebagai saklar (*switch*) atau penguat sinyal.

Dalam penelitian ini, *relay* digunakan sebagai aktuator untuk mengontrol penyiraman dan pengkabutan otomatis di *greenhouse* tanaman cabai. Perintah aktivasi *relay* dikendalikan secara otomatis oleh sistem berdasarkan hasil klasifikasi dari algoritma *Naïve Bayes* yang menganalisis data sensor suhu, kelembapan udara, dan kelembapan tanah.

### 2.2.11 *Unified Modeling language*

*UML (Unified Modeling Language)* adalah sebuah bahasa visual yang digunakan untuk menggambarkan dan merancang sistem atau aplikasi secara jelas dan terstruktur. Dengan *UML*, pengembang bisa membuat berbagai diagram untuk menggambarkan bagaimana suatu sistem bekerja, siapa saja yang terlibat, dan bagaimana alur data berjalan di dalamnya (Irhan Hisyam Dwi Nugroho, 2024).

Berikut adalah diagram-diagram dalam *UML*:





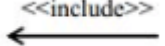

### 1. Use Case Diagram

*Use case diagram* adalah satu dari berbagai jenis diagram *UML (Unified Modelling Language)* yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use Case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya (Dicoding Intern, 2021).

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

Tabel 2. 4 *Use Case Diagram*

Sumber: (Dicoding Intern, 2021)

No	Simbol	Keterangan
1	Aktor 	Mewakili peran orang, system yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
2	<i>Use Case</i> 	Abstraksi dan interaksi antara sistem dan aktor
3	<i>Association</i> 	Abstraksi dari penghubung antara actor dengan <i>use case</i>
4	Generalisasi 	Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
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 fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

### 2. Activity Diagram







*Activity diagram* adalah jenis diagram yang berguna untuk dapat membuat model dari berbagai proses dalam suatu sistem, urutan proses digambarkan secara vertikal. Diagram ini merupakan pengembangan dari *use case* dan menunjukkan

alur aktivitas yang ditampilkan berupa rangkaian menu atau proses bisnis yang ada dalam sistem tersebut (Ucy Sugiarti, 2024).

Daftar simbol pada *activity diagram* dapat dilihat pada tabel berikut:

Tabel 2. 5 *Activity Diagram*

Sumber: (Lawencon International, 2024)

No	Simbol	Keterangan
1	Status Awal 	Sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan / <i>Decision</i> 	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
4	Penggabungan / <i>Join</i> 	Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.
5	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	<i>Swimlane</i> 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi


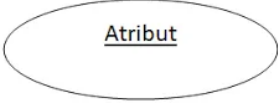
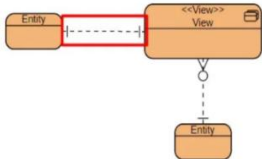
### 3. *Entity Relationship Diagram*

*ERD* (*Entity Relationship Diagram*) adalah sebuah gambar atau diagram yang digunakan untuk menggambarkan hubungan antara entitas dalam sebuah database. Dalam *ERD*, entitas direpresentasikan sebagai kotak dengan atribut yang terkait dengan entitas. Hubungan antara entitas ditunjukkan oleh tanda panah atau garis yang menghubungkannya. (Ayoni Sulthon, 2023).

Daftar simbol pada *Entity Relationship* diagram dapat dilihat pada tabel berikut:

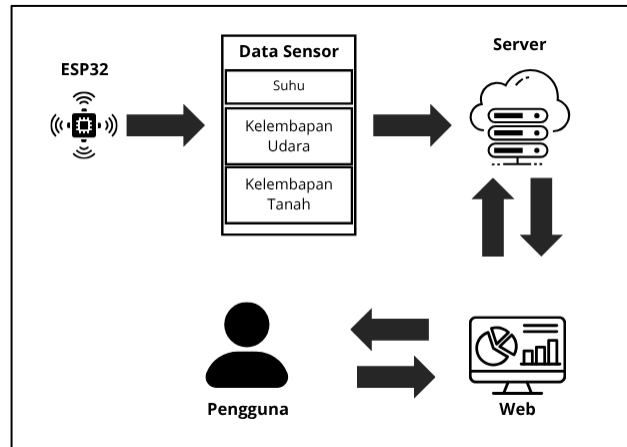
Tabel 2. 6 *Entity Relationship Diagram*

Sumber: (DomaiNesia, 2023)

No	Simbol	Keterangan
1	<p>Entitas / <i>Entity</i></p> 	Entitas adalah sebuah objek berwujud nyata yang dapat dibedakan dengan objek lainnya. Objeknya dapat bersifat konkret maupun abstrak. Data konkrit adalah sesuatu yang benar-benar ada atau dapat dirasakan oleh alat indra, sedangkan abstrak tidak berwujud.
2	<p>Atribut</p> 	Setiap entitas memiliki atribut untuk mendeskripsikan karakteristik dari suatu entitas. Untuk jenisnya dibedakan menjadi beberapa jenis, yaitu Atribut <i>key</i> , atribut yang unik dan berbeda.
3	<p>Relasi</p> 	Hubungan antar entitas untuk menunjukkan adanya koneksi di antara sejumlah entitas yang berasal dari himpunan entitas berbeda.

### 2.2.12 Website

*Website* adalah kumpulan dari halaman-halaman situs yang terdapat dalam sebuah domain atau subdomain yang berada di *dalam World Wide Web (WWW)* di internet. *Website* dapat berisi teks, gambar, video, 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 memungkinkan 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

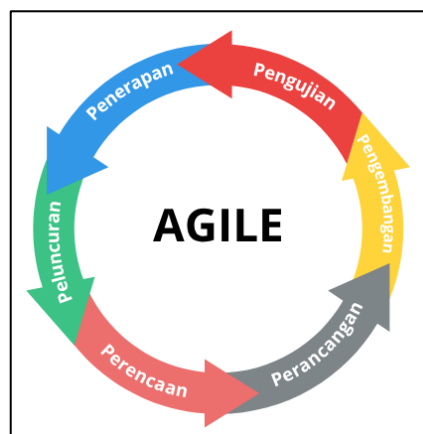
*Hosting* atau *web hosting* adalah layanan penyimpanan yang digunakan untuk menampung seluruh kebutuhan sebuah *website*, seperti file, gambar, video, *script*, *email*, aplikasi, dan *database*, agar *website* tersebut dapat diakses secara *online* melalui internet. Tanpa layanan *web hosting*, sebuah *website* tidak dapat diakses oleh pengguna internet. (Jagoan Hosting Team, 2023).

Terdapat beberapa jenis *web hosting* yang dapat 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*: Menggunakan 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)* merupakan pendekatan modern dalam pengembangan perangkat lunak yang menjadi alternatif unggul dibandingkan metodologi tradisional. Berbeda dengan *SDLC* tradisional yang bersifat linier dan berurutan di mana setiap tahapan pengembangan diselesaikan satu per satu secara sistematis metode *Agile* mengedepankan fleksibilitas serta iterasi berkelanjutan melalui siklus pendek yang disebut *sprint*. (AgileTech Vietnam, 2025).



Gambar 2. 9 Metode Agile

Sumber: (AgileTech Vietnam, 2025)

Berikut adalah tahap-tahap pengembangan perangkat lunak yang digunakan 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. Pengembangan

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 dihasilkan diuji untuk memastikan fungsionalitasnya berjalan dengan baik dan sesuai dengan kebutuhan.

5. Penerapan

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

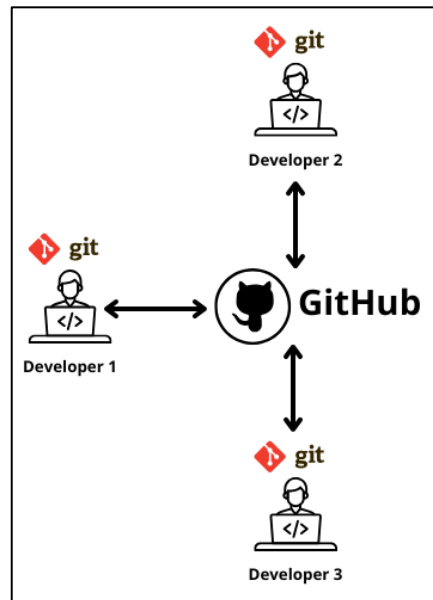
6. Peluncuran

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 kontrol versi *Git* yang memungkinkan pengguna melacak perubahan kode, bekerja dala *GitHub* juga menyediakan berbagai fitur seperti *repository*, *branch*, *pull request*, dan *issue tracker* untuk mempermudah

proses pengembangan tim, serta mengelola kontribusi dari banyak pengembang secara terorganisir (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*

Bahasa pemrograman *Arduino* ditemukan oleh Hernando Barragan pada tahun 2003, pada saat ia mengembangkan sistem *wiring*. *Arduino* menggunakan bahasa pemrogramannya sendiri, yang mirip dengan C++. Terdapat tiga bagian utama yang menyusun bahasa pemrograman *Arduino* yaitu *function*, *value*, dan *structure*. *Function* memungkinkan dalam mengontrol *board*. Dengan menggunakan *function*, analisis data, operasi matematika, dan tugas lainnya dapat dilakukan. *Value* berfungsi mewakili konstanta dan variabel, tipe data yang digunakan seperti *array*, *boolean*, *char*, *float*, dan lainnya yang mirip dengan C++. (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 merupakan penjelasan dari setiap bagian yang terdapat pada struktur program di atas:

#### 1. *Header*

Bagian ini berisi semua pustaka atau *library* yang diperlukan untuk program *Arduino*. Pustaka adalah kumpulan fungsi yang sudah dibuat sebelumnya dan bisa digunakan kembali untuk mempermudah pengembangan program.

#### 2. Deklarasi Variabel *Global*

Bagian ini digunakan untuk mendeklarasikan variabel yang akan digunakan secara global di seluruh program. Variabel global adalah variabel yang dapat diakses dan dimanipulasi dari mana pun dalam program.

### 3. Setup Function

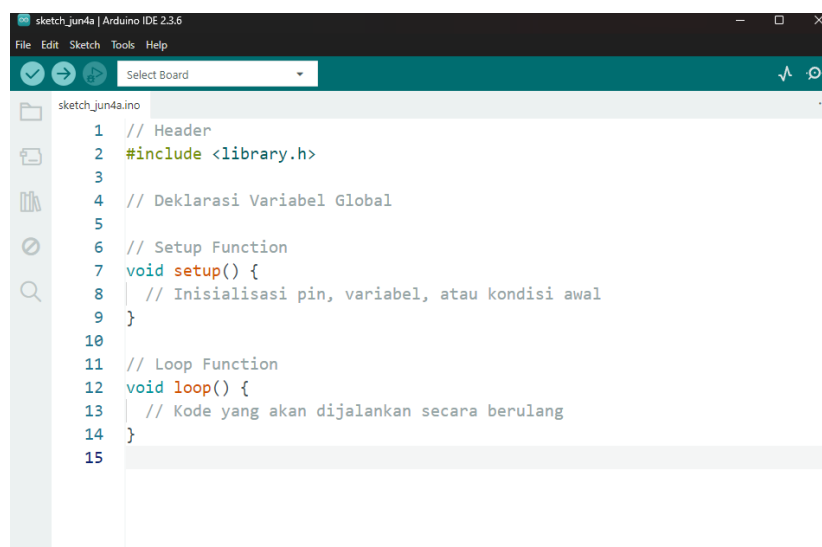
Fungsi *setup()* adalah fungsi khusus yang hanya dieksekusi satu kali saat program Arduino pertama kali dijalankan. Fungsi ini digunakan untuk melakukan inisialisasi, seperti mengatur pin, menginisialisasi variabel, atau menyiapkan kondisi awal.

### 4. Loop Function

Fungsi *loop()* adalah inti dari program *Arduino*. Setelah fungsi *setup()* dieksekusi, Arduino akan terus mengeksekusi fungsi *loop()* secara berulang tanpa henti.

## 2.2.17 *Arduino IDE*

*Arduino IDE (Integrated Development Environment)* adalah perangkat lunak resmi yang digunakan untuk menulis, mengedit, dan mengunggah program ke papan mikrokontroler *Arduino*. *IDE* ini menyediakan antarmuka yang sederhana dan mudah digunakan. 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).

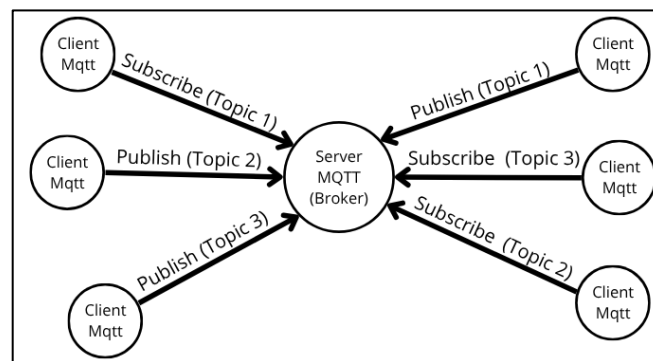


Gambar 2. 12 *Arduino IDE*

Sumber: (Penulis, 2025)

### 2.2.18 MQTT

*MQTT (Message Queuing Telemetry Transport)* adalah protokol *publish-subscribe* yang dirancang khusus untuk mengirimkan pesan antara perangkat IoT dengan *overhead* yang rendah dan konsumsi energi yang efisien. Dalam era *Internet of Things (IoT)* yang semakin berkembang, di mana jutaan perangkat terhubung dan bertukar data, diperlukan protokol komunikasi yang ringan, efisien, dan andal (Indobot Academy, 2023).



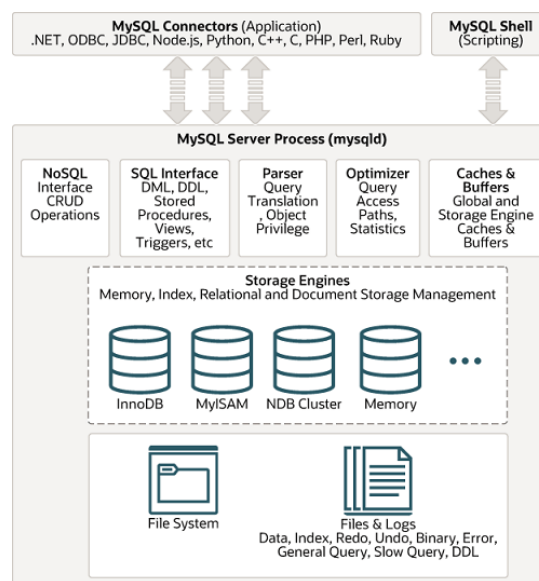
Gambar 2. 13 Arsitektur *MQTT*

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.

### 2.2.19 Mysql

*MySQL* adalah sistem manajemen basis data berbasis *SQL* (*Structured Query Language*) yang memungkinkan pengguna untuk menyimpan, mengelola, dan mengambil data dengan cara yang terstruktur. Sebagai perangkat lunak *open-source*, *MySQL* memberikan kebebasan kepada pengembang untuk memodifikasi dan mendistribusikan perangkat lunak ini sesuai kebutuhan. (Yazid Yusuf, 2024).



Gambar 2. 14 Arsitektur *MySQL*

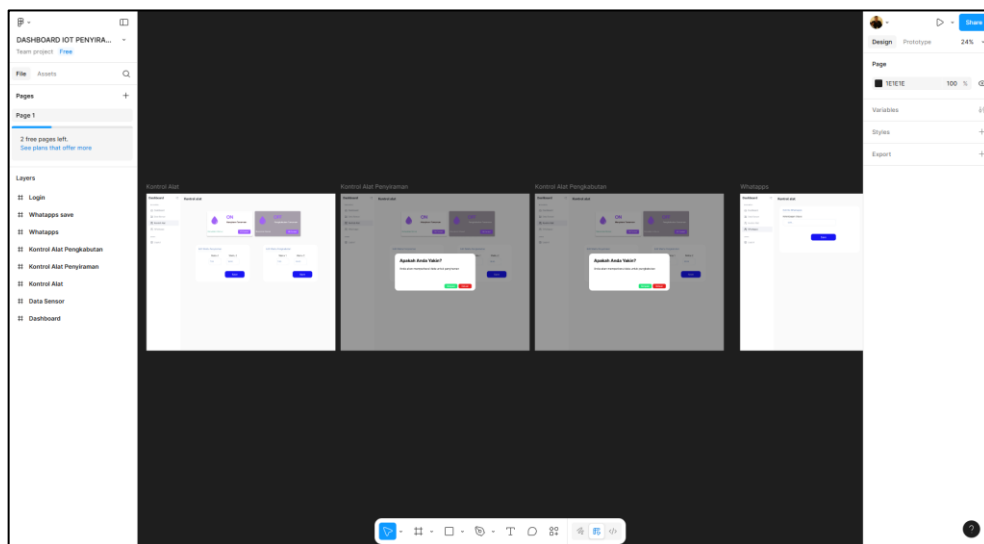
Sumber: (MySQLTutorial.org, 2008)

Gambar di atas menjelaskan arsitektur dari *MySQL* menyediakan *MySQL Connectors* yang mendukung berbagai bahasa pemrograman seperti *Python*, *JavaScript* dan lainnya, 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)

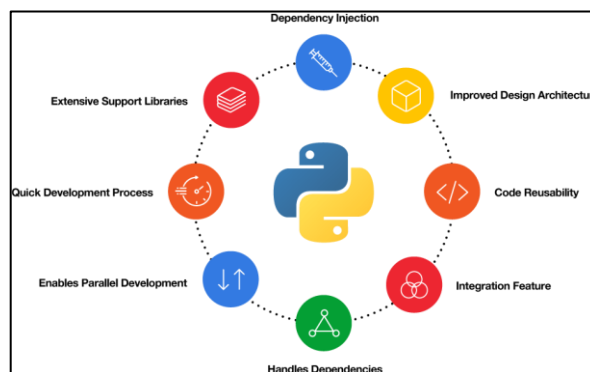
*Figma* merupakan platform desain antarmuka berbasis *cloud* yang mendukung kolaborasi secara *real-time*. Aplikasi ini memungkinkan tim desain dan pengembang bekerja bersama dalam satu ruang kerja digital, tanpa perlu menginstal perangkat lunak tambahan. Sebagaimana ditampilkan pada Gambar, antarmuka *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*: Proyek yang dibuat di *Figma* dapat dengan mudah dibagikan melalui tautan kepada anggota tim (Nadifa Padantya Raihanah, 2023).

### 2.2.21 Python

*Python* adalah pemrograman tingkat tinggi yang diinterpretasikan, berorientasi objek, bahasa dengan semantik dinamis. Data bawaan tingkat tinggi struktur, dikombinasikan dengan penyetoran dinamis dan pengikatan dinamis, membuatnya sangat menarik untuk pengembangan aplikasi 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)

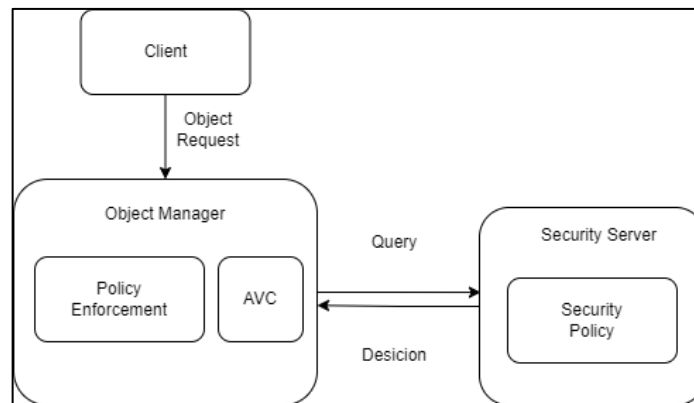
Pada gambar di atas, *Python* ditampilkan sebagai inti dari berbagai fitur utama yang membuatnya unggul dan relevan dalam pengembangan sistem modern, termasuk untuk penelitian ini. Di antaranya adalah:

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

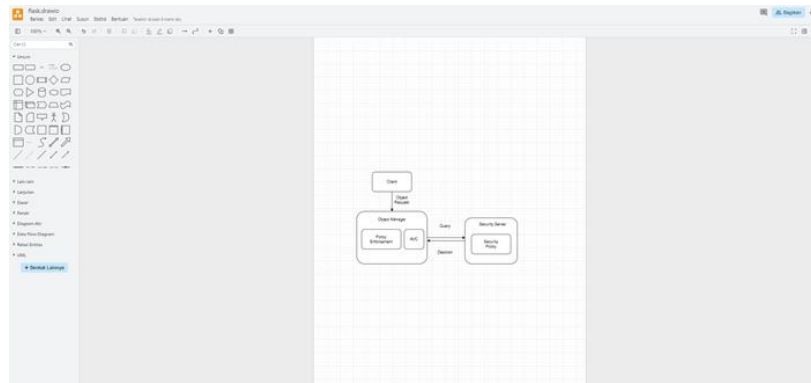
Sumber: (Ding et al., 2012)

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, kelembapan udara, dan kelembapan 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 lain-lain. *Draw.io* juga menyediakan fitur pembuatan diagram berbasis 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*

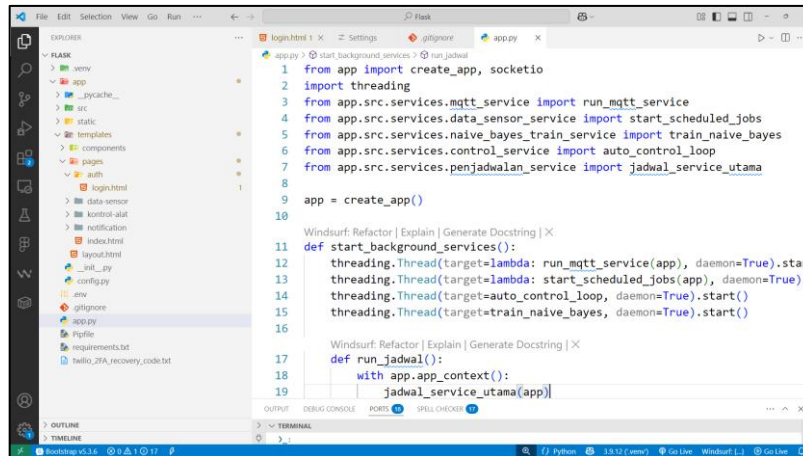
Sumber: (Penulis, 2025)

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 pengkabutan 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.

#### 2.2.24 *Visual Studio Code*

*Visual Studio Code* adalah editor kode yang disederhanakan dengan dukungan untuk operasi pengembangan seperti penelusuran kesalahan, menjalankan tugas, dan kontrol versi. Ini bertujuan untuk menyediakan alat yang dibutuhkan pengembang untuk siklus pembuatan *code-debug* cepat dan meninggalkan alur kerja yang lebih kompleks untuk *IDE* berfitur yang lebih lengkap (Microsoft, 2025).



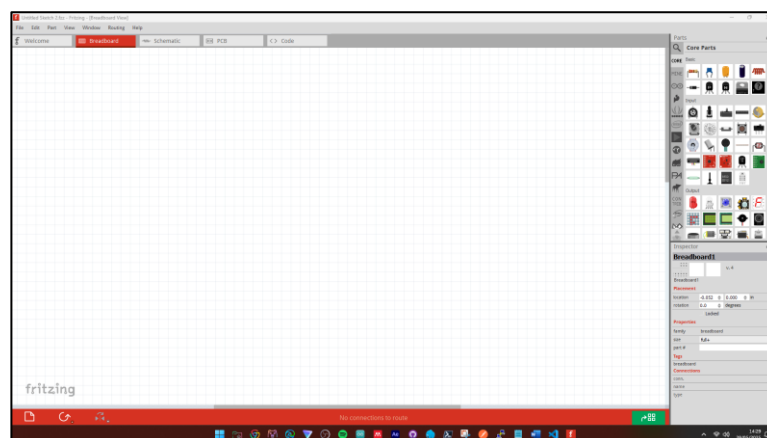
Gambar 2. 19 Visual Studio Code

Sumber: (Penulis,2025)

Gambar di atas menampilkan antarmuka VS Code yang digunakan dalam menulis kode program untuk proyek 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, situs web komunitas, dan layanan dalam pemrosesan dan *arduino*, menumbuhkan ekosistem kreatif yang memungkinkan pengguna untuk mendokumentasikan *prototipe* mereka (Fritzing, 2025).



Gambar 2. 20 Fritzing

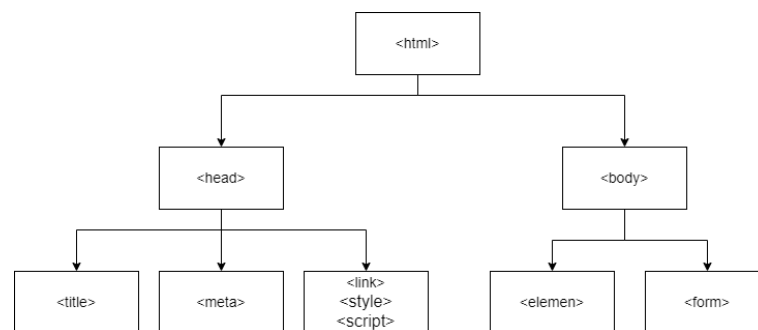
Sumber: (Penulis, 2025)

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, kelembapan udara, dan kelembapan 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 (Publikasi et al., 2025).



Gambar 2. 21 Arsitektur *HTML*

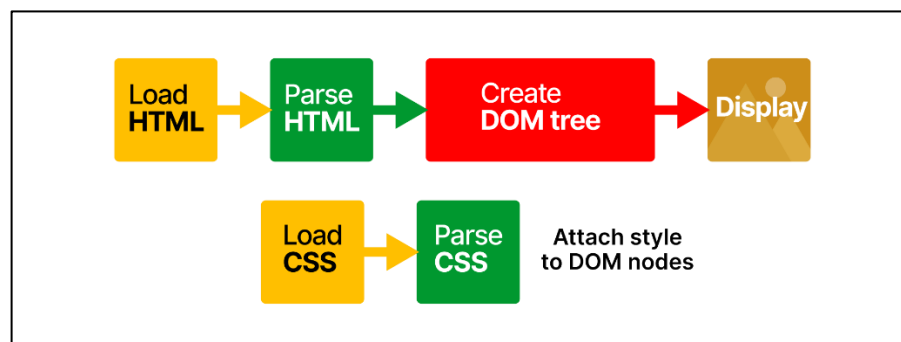
Sumber: (Ade Roni, 2025)

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

*CSS* adalah singkatan dari *Cascading Style Sheets* digunakan untuk menata dan memperindah tampilan halaman web. *CSS* memungkinkan 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)

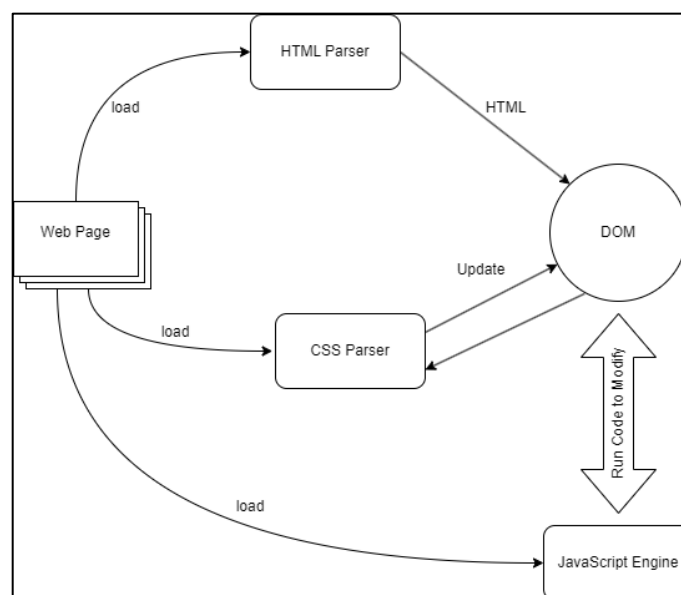
Gambar di atas menjelaskan alur kerja browser saat merender halaman web. Proses diawali dengan pemuatan (*load*) dan penguraian (*parse*) file *HTML* untuk membentuk struktur *DOM* (*Document Object Model*) *tree*. Secara paralel, browser juga memuat dan mengurai file *CSS* untuk membentuk *CSSOM* (*CSS 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, kelembapan udara, dan kelembapan tanah secara *real-time*, serta mengontrol sistem penyiraman dan pengkabutan dari jarak jauh. *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.

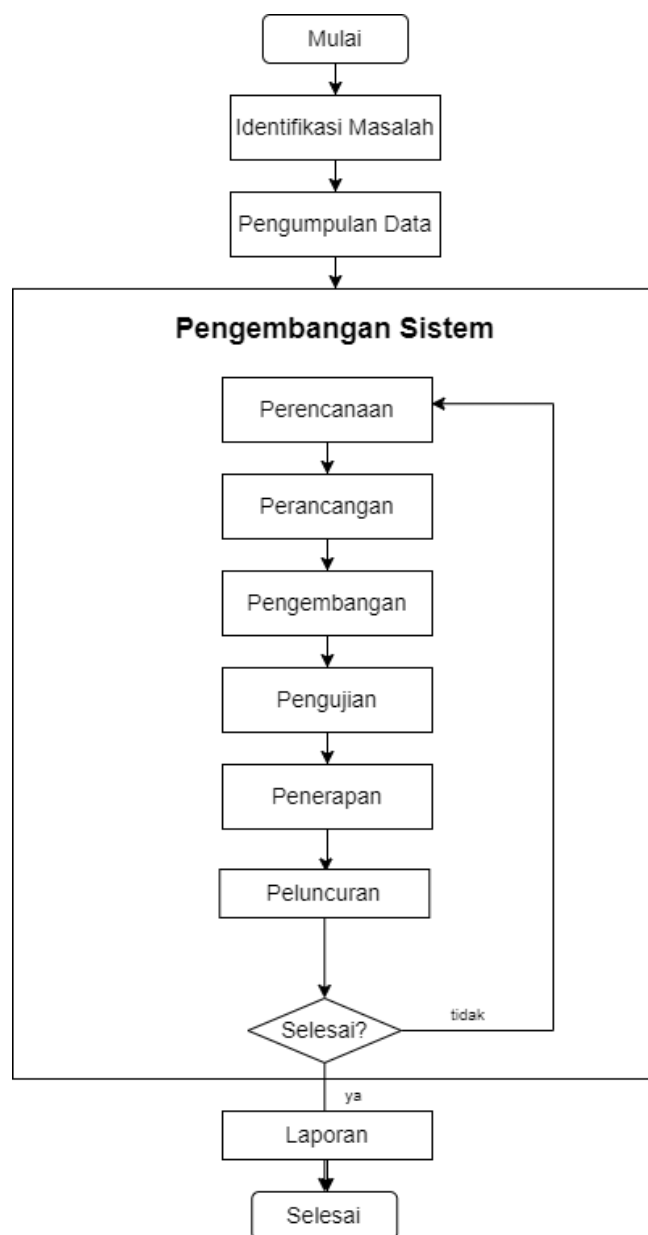
Dalam konteks penelitian ini, *JavaScript* sangat penting dalam pengembangan aplikasi *monitoring* berbasis web untuk sistem penyiraman dan pengkabutan otomatis tanaman cabai di *greenhouse*. Melalui *JavaScript*, antarmuka aplikasi dapat menampilkan data sensor suhu, kelembapan udara, dan kelembapan tanah secara *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.

## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Kerangka Pikir

Kerangka pikir merupakan jalur pemikiran yang dirancang berdasarkan kegiatan penyusun yang dilakukan. Berikut adalah kerangka pikir yang merupakan langkah-langkah yang akan dilakukan dalam penelitian ini:



Gambar 3. 1 Kerangka Pikir

## 3.2 Deskripsi

Penelitian ini menggunakan pendekatan metode *Agile* dalam pengembangan sistem. Berikut adalah tahapan-tahapan penelitian yang dilaksanakan:

### 3.2.1 Identifikasi Masalah

Identifikasi masalah adalah upaya untuk menjelaskan permasalahan. Identifikasi ini dilakukan sebagai langkah awal penelitian. Dimulai dengan meminta izin kepada pemilik *Avicenna Greenhouse* untuk melakukan penelitian kemudian melakukan pengamatan secara langsung terhadap objek penelitian.

### 3.2.2 Pengumpulan Data

Dalam pengumpulan data, penulis menggunakan beberapa metode untuk mendapatkan data yang akurat 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 dan menganalisis data dari berbagai jurnal, buku, website serta sumber relevan lainnya yang membahas konsep *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

Analisis kebutuhan fungsional merupakan proses untuk mengidentifikasi fitur-fitur utama yang harus dimiliki oleh sistem berdasarkan fungsionalitas yang diharapkan. Adapun kebutuhan fungsional sistem yang dirancang 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*.

#### 2. Kebutuhan *Non*-Fungsional

Analisis kebutuhan *non*-fungsional mencakup spesifikasi teknis dan kualitas sistem yang memengaruhi performa dan keandalan sistem secara keseluruhan.

##### 1) Perangkat Keras

Tabel 3. 1 Perangkat Keras

<i>Processor</i>	Intel Dual-Core 3.10GHz
<i>Hard Disk</i>	500 GB
<i>Memory</i>	4 GB
<i>Monitor</i>	Resolusi 1280 x 800px

## 2) Perangkat Internet of Things

Tabel 3. 2 Perangkat *Internet of Things*

Mikrokontroler	ESP32 (Wi-Fi + Bluetooth)
Sensor Suhu dan Kelembapan Udara	DHT11
Sensor Kelembapan Tanah	<i>Soil Moisture Sensor</i>
Pompa Air	<i>Mini water pump DC 3–6V</i>
Pengkabutan	<i>Nozzle Sprayer</i>
Relay Module	2 channel relay 5V
Power Supply & Adaptor	Adapter 5V/2A dan modul <i>step-down</i>
Selang	Selang plastik PE ( <i>Polyethylene</i> )
Modul Sim Wifi	HI-NET H806 <i>Wireless Router</i> 4g
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>

### 3.2.4 Perancangan

#### 1. Perancangan Diagram

##### a. *Use Case Diagram*

Diagram ini menggambarkan 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/menonaktifkan 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 *WhatsApps*
- 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
2. Perancangan Database
- Perancangan ini menggunakan model *Entity Relationship Diagram (ERD)* untuk menggambarkan struktur basis data, relasi antar entitas. Dengan dengan daftar entitas yang akan dibuat adalah sebagai berikut:
- a. Data Sensor
  - b. Riwayat Aksi
  - c. Jadwal Penyiraman
  - d. Hasil Prediksi
3. 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 Pengembangan

Tahap ini melibatkan proses penerjemahan seluruh desain sistem menjadi kode program sesuai dengan spesifikasi yang telah dirancang. Sistem ini dikembangkan menggunakan 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 dilakukan untuk memastikan bahwa seluruh fitur dalam sistem berfungsi dengan baik sesuai dengan spesifikasi dan bebas dari kesalahan atau bug yang signifikan.

Pengujian untuk menguji fungsionalitas sistem tanpa melihat struktur kode secara langsung. Selain itu, dilakukan pengujian oleh pengguna untuk memastikan bahwa sistem sesuai dengan kebutuhan pengguna di lapangan, khususnya dalam konteks *monitoring* dan pengendalian penyiraman dan pengkabutan di *greenhouse*.

### 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 dikonfigurasi 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 memastikan bahwa semua komponen perangkat keras dan perangkat lunak berjalan dengan baik.

### **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.

## DAFTAR PUSTAKA

- Ade Roni. (2025). *Pengertian dan Struktur Dasar HTML*. Aderoni.Com.  
<https://aderoni.com/pemrograman/pengertian-dan-struktur-dasar-html/>
- 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/>
- 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>
- Ayoni Sulthon. (2023, May 27). *Cara membuat ERD: Simbol, Entitas, Atribut Termudah*. DomaiNesia. <https://www.domainesia.com/berita/pengertian-erd-adalah/>
- Dicoding Intern. (2021, May 19). *Contoh Use Case Diagram Lengkap dengan Penjelasannya*. Dicoding.Com. <https://www.dicoding.com/blog/contoh-use-case-diagram/>
- Ding, B., Yao, F., Wu, Y., & He, Y. (2012). Improving Flask Implementation Using Hardware Assisted In-VM Isolation. In *IFIP AICT* (Vol. 376).
- 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 Kelembapan 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>
- 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).
- Ifitah 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>
- Jagoan Hosting Team. (2023, March). *Apa itu Hosting? Jenis, Fungsi dan Cara Kerjanya*. Jagoan Hosting. <https://www.jagoanhosting.com/blog/pengertian-hosting/>
- Kelasplc. (2023). *Pengertian Internet Of Things : Fitur, Arsitektur dan Contohnya*. Kelas PLC. <https://www.kelasplc.com/pengertian-internet-of-things/>
- Kurniawan. (2023, December 9). *Perbedaan Misting dan Menyiram Tanaman*. DaunSuper. <https://daunsuper.com/perbedaan-misting-dan-menyiram-tanaman/>
- 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/starting-guide/the-arduino-software-ide/>
- M. Iqbal Hasani, & 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 Code.  
[https://code.visualstudio.com/docs/supporting/faq#\\_what-is-the-difference-between-visual-studio-code-and-visual-studio-ide](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](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>
- Muhamad Rusdi, Muriani, Rivaldo Pasca Corputty, Mardiyasa Putra Yoga, Grace Christin Aditya Ronsumbre, & Diah Bayu Titisari. (2023).  
IMPLEMENTASI TEKNOLOGI PENYIRAMAN SISTEM  
PENGKABUTAN OTOMATISDAN MONITORING PINTAR BERBASIS  
TENAGA SURYAUNTUK TEMPAT BUDIDAYA TANAMAN  
ANGGREK UDFAIRUS DI KABUPATEN MERAUKE. *JURNAL  
PENGABDIAN MASYARAKAT*, 1(Vol. 1 No. 2 (2023): AKSELERASI:  
*Jurnal Pengabdian Masyarakat*), 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/>
- Nadifa Padantya Raihanah. (2023, January 20). *Apa Itu Figma? Fitur, Kelebihan dan Kekurangan*. Alterra Academy. <https://academy.alterra.id/blog/apa-itu-figma/>
- Pallets. (2025, May 1). *Welcome to Flask*. Pallets.  
<https://flask.palletsprojects.com/en/stable/>
- Publikasi, A. J., Mardiansyah, A., Kasah, B. N., Zamzami, H. R., Arabu, Y., Nasro, M. A., Kristanto, N., Paojiah, R., & Wulandari, Y. (2025).  
PENGENALAN DASAR HTML DAN CSS: LANGKAH PERTAMA  
DALAM PENGEMBANGAN WEB. *Abdi Jurnal Publikasi*, 3(3), 165–170.  
<https://jurnal.portalpublikasi.id/index.php/AJP/index>
- python org. (2025, May 5). *What is Python? Executive Summary*. Python.Org.  
<https://www.python.org/doc/essays/blurb/>
- Rifqi Mulyawan Digital. (2025). *Pengertian JavaScript: Sejarah, Cara Kerja JavaScript dan Manfaatnya*. Rifqi Mulyawan Digital.  
<https://rifqimulyawan.com/blog/pengertian-javascript/>
- 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). 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). *Mengenal Bahasa Pemrograman Arduino Secara Lengkap yang Mudah Dipelajari untuk Pemula*. PT Tekno Gemilang Indonesia. <https://toghr.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. (2012). *Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ 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](https://play.google.com/books/reader?id=F_HHAgAAQBAJ&pg=GBS.PR9.w.10.0.32_29&hl=id)
- 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>
- 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/>
- 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 Desa Drawati Kec. Paseh Kab. Bandung, Prov. Jawa Barat

Berikut adalah hasil wawancara:

No	Pertanyaan	Jawaban
1	Kapan <i>Avicenna Greenhouse</i> mulai dibangun dan digunakan untuk menanam cabai?	<i>Avicenna Greenhouse</i> mulai dibangun dan digunakan sejak tahun 2024
2	Mengapa Bapak memilih menanam cabai di dalam greenhouse?	Supaya tanaman bisa lebih terlindungi dari serangan hama dan cuaca ekstrem.
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 dialirkan melalui selang ke setiap pot.
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 greenhouse ini?	Saat ini, sistem pengkabutan belum diterapkan sama sekali.
7	Untuk penyiraman pada	Biasanya dilakukan satu kali, maksimal dua

	tanaman dilakukan pada waktu jam berapa saja?	kali sehari, yaitu jam 7 pagi dan jam 5 sore.
8	Bagaimana pandangan Bapak terhadap penggunaan sistem <i>IoT</i> untuk otomatisasi penyiraman dan pengkabutan di greenhouse 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



**Adam Setiadi**

Narasumber



**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.  
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

Pemilik Avicenna Greenhouse



Adam Setiadi



Resa Aldiana

