

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Penelitian yang dilakukan oleh Ciptadi dan Hardyanto (2018) mengembangkan sistem menggunakan dua sensor yang terhubung dengan mikrokontroler Arduino Uno. Sensor pertama, DHT11, digunakan untuk mengukur suhu dan kelembapan, sementara sensor YF-S201 digunakan untuk mengukur intensitas nutrisi yang mengalir melalui gully. Teknologi IoT yang diterapkan dalam penelitian ini bertujuan untuk meminimalisasi intervensi manual dan menciptakan sistem hidroponik yang cerdas[2].

Penelitian yang dilakukan oleh Crisnapati et al. (2018) menerapkan teknologi IoT pada tanaman hidroponik menggunakan teknik NFT (Nutrient Film Technique). Sistem ini memonitor dan mengumpulkan informasi dari tanaman, kemudian data tersebut dievaluasi dan dianalisis. Data dikirim menggunakan modul Wi-Fi ESP8266, dan Raspberry Pi 2 Model B digunakan sebagai webserver untuk menyimpan data yang dikirim[3].

Arief et al. (2020) melakukan penelitian dengan membuat smart greenhouse untuk tanaman hidroponik. Sistem ini adalah sistem pintar yang mengatur jumlah nutrisi secara otomatis dan menjaga pH yang tepat untuk tanaman. Pengaturan ini dilakukan secara jarak jauh, terus menerus, dan real-time, dengan pencatatan data historis[4].

Penelitian yang dilakukan oleh Prasetyo et al. (2018) menciptakan sistem hidroponik dengan jenis Deep Flow Technique (DFT), yang menerapkan aliran nutrisi terus menerus dengan genangan setengah dari diameter pipa yang membasahi akar tanaman. IoT diterapkan untuk mengantisipasi perubahan unsur pertumbuhan tanaman. Data unsur pertumbuhan diakuisisi oleh sensor yang terintegrasi dengan Raspberry Pi. Monitoring dilakukan melalui website yang menampilkan data pH, suhu, kelembapan, dan ketinggian air pada tandon hidroponik. Suhu dan kelembapan digunakan sebagai parameter pengendali sirkulasi air, diproses menggunakan Metode Fuzzy Sugeno untuk menyalakan atau mematikan pompa pada sistem hidroponik DFT[5].

Budi Herdiana et al. (2018) dalam penelitiannya yang berjudul “Sistem Smart Urban Gardening Berbasis Internet of Things” bertujuan untuk memonitor suhu, kelembapan, nilai nutrisi air, serta pH air dari jarak jauh, yang dapat dikontrol melalui website atau smartphone. Penelitian ini menggunakan beberapa perangkat keras, seperti NodeMCU (ESP8266), sensor pH, sensor Electric Conductivity (EC), sensor DHT22, mini pump basa, dan mini pump asam[6].

## **2.2 Landasan Teori**

### **2.2.1 IoT (*Internet Of Things*)**

*Internet of Things* (IoT) adalah konsep di mana objek memiliki kemampuan untuk mengirimkan data melalui jaringan tanpa memerlukan interaksi langsung dari manusia atau komputer.



mikrokontroler Arduino telah diprogram dengan program yang disebut bootloader, yang bertindak sebagai perantara antara compiler Arduino dan mikrokontroler itu sendiri. Arduino IDE sendiri dikembangkan menggunakan bahasa pemrograman Java. IDE ini dilengkapi dengan perpustakaan C/C++ yang dikenal sebagai Wiring, yang mempermudah operasi input dan output. Arduino IDE awalnya merupakan hasil adaptasi dari perangkat lunak Processing yang dimodifikasi khusus untuk pemrograman dengan Arduino.

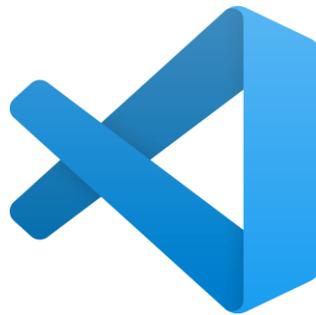


*Gambar 2.2 Software Arduino Ide*

### **2.2.3 Visual Studio Code**

Visual Studio Code adalah editor kode yang dapat digunakan di platform MacOS, Linux, dan Windows. Editor ini mendukung pengeditan kode untuk berbagai bahasa pemrograman seperti JavaScript, TypeScript, dan Node.js. Selain itu, Visual Studio Code juga mendukung bahasa pemrograman seperti PHP, Python, Java, dan .NET. Salah satu keunggulan utama Visual Studio Code adalah ekosistemnya yang luas dan ketersediaan berbagai ekstensi.

Beberapa fitur utamanya mencakup pengeditan kode dasar, intellisense, debugging, marketplace ekstensi, dan integrasi dengan GitHub, yang menjadikannya salah satu editor kode yang paling populer saat ini.



*Gambar 2.3 Software Visual Studio Code*

#### **2.2.4 XAMPP**

XAMPP adalah perangkat lunak server web lokal yang digunakan untuk mengembangkan situs web, aplikasi, dan database secara offline. Banyak orang memilih untuk membuat situs web di komputer secara offline terlebih dahulu untuk memastikan semua berjalan dengan baik sebelum mengunggahnya ke server. XAMPP merupakan perangkat lunak open source yang berbasis server web, mencakup berbagai program dan mendukung berbagai sistem operasi seperti Linux, Windows, MacOS, dan Solaris. Fungsinya utama adalah sebagai server lokal atau localhost, dengan program Apache, MySQL, dan PHP yang sudah terintegrasi di dalamnya. XAMPP hadir sebagai solusi untuk kesulitan dalam instalasi Apache, khususnya dalam menambahkan dukungan untuk PHP dan

MySQL. Dengan demikian, XAMPP memudahkan pengembang yang memerlukan server web di localhost dengan menggunakan satu aplikasi.



*Gambar 2.4 Software XAMPP*

### **2.2.5 Web Browser**

Web browser adalah perangkat lunak yang memungkinkan pengguna untuk mencari, mengakses, dan menampilkan halaman website di internet. Pada dasarnya, website berisi kode seperti JavaScript dan HTML yang tidak dapat langsung dibaca oleh manusia. Web browser menerjemahkan kode tersebut menjadi tulisan, gambar, audio, dan elemen lainnya. Meskipun bukan satu-satunya perangkat lunak yang dapat digunakan untuk mengakses website, web browser sering menjadi pilihan utama karena kemudahannya. Penting untuk diketahui bahwa web browser bukanlah mesin pencari; web browser adalah perangkat lunak yang menampilkan halaman web, seperti Google Chrome, Mozilla Firefox, atau Opera.



*Gambar 2.5 Web Browser*

### 2.2.6 UML (Unified Modelling Language)

UML (*Unified Modelling Language*) adalah “bahasa” pemodelan untuk sistem atau perangkat lunak yang berparadigma “berorientasi objek”.

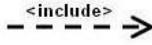
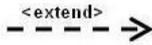
Modeling sebenarnya digunakan untuk menyederhanakan masalah yang kompleks sehingga lebih mudah dipelajari dan dipahami.

Beberapa simbol yang digunakan saat menggambar UML

#### 1. Use Case Diagram

Use case diagram adalah teknik yang digunakan untuk menunjukkan hubungan antara pengguna sistem dengan sistem itu sendiri. Hasilnya adalah skema sederhana yang memungkinkan client membaca dan memahami informasi.

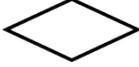
Table 2.1 *Use Case Diagram*

Gambar	Keterangan
	Hubungan dan interaksi antara aktor dan sistem.
	Mewakili peran individu, sistem, atau alat dalam komunikasi menggunakan kasus penggunaan.
	<i>Generalisasi</i> berarti hubungan di mana objek turunan mewarisi struktur data dan perilakunya dari objek induk. Objek induk adalah objek yang berada di tingkat lebih tinggi, atau sering disebut sebagai leluhur.
	<i>Dependency</i> adalah proses di mana hubungan pada suatu elemen independen mempengaruhi elemen lain yang tidak bergantung pada elemen independen tersebut.
	<i>Include</i> berfungsi untuk mengkategorikan use case sumber secara eksplisit.
	<i>Extend</i> berguna untuk mengkategorikan atau menentukan apabila use case target memperluas perilaku dari sumber pada titik tertentu.

## 2. *Activity Diagram* (Aktivitas Diagram)

*Activity diagram* menunjukkan alur kerja (work flow) sebuah proses bisnis dan urutan aktivitasnya. Diagram ini sangat mirip dengan flowchart karena dapat menunjukkan alur kerja dari satu aktivitas ke aktivitas lain atau satu aktivitas ke keadaan sesaat.

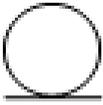
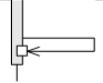
Table 2.2 *Activity Diagram*

<b>Gambar</b>	<b>Keterangan</b>
Status Awal 	Sebuah status awal dapat ditemukan di diagram aktivitas sistem.
Aktivitas 	Aktivitas yang dilakukan sistem dimulai dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan di mana ada lebih dari satu aktivitas yang dapat dipilih.
Penggabungan / <i>join</i> 	Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	sebuah status akhir ditampilkan dalam diagram aktivitas, bersama dengan status akhir yang dilakukan sistem.

## 3. *Sequence Diagram* (Diagram Urutan)

*Sequence Diagram* adalah suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut

Table 2.3 *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> adalah bagian dari sistem yang terdiri dari kumpulan entitas yang membentuk gambaran awal sistem dan berfungsi sebagai dasar untuk penyusunan basis data.
	<i>Boundary Class</i> adalah kumpulan kelas yang bertindak sebagai antarmuka atau interaksi dengan sistem untuk satu atau lebih aktor, seperti form masukan dan pencetakan.
	<i>Control class</i> , suatu entitas yang mengandung logika aplikasi yang tidak bertanggung jawab kepada entitas, contohnya adalah aturan bisnis yang mencakup berbagai objek dan kalkulasi.
	<i>Message</i> , simbol berkomunikasi antar kelas.
	<i>Recursive</i> , menggambarkan pesan yang dikirim secara pribadi.
	<i>Activation</i> , mewakili eksekusi operasi objek, dan panjang kotak ini sama dengan waktu aktivasi operasi.
	<i>Lifeline</i> , garis titik-titik yang menghubungkan objek ke lifeline.

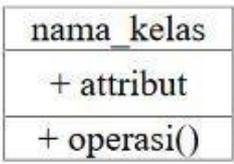
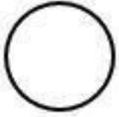
#### 4. *Class Diagram* (Diagram Kelas)

*Class diagram* adalah bagian dari UML. Dalam class diagram ini struktur, deskripsi, dan hubungan antar kelas digambarkan.

Table 2.4 *Multiplicity Class Diagram*

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimal 4

Table 2.5 Simbol *Class Diagram*

<b>Simbol</b>	<b>Keterangan</b>
	Kelas pada struktur sistem.
	<i>Interface</i> , serupa dengan konsep antarmuka dalam pemrograman berbasis objek
	<i>Association</i> adalah hubungan antar kelas yang menggambarkan keterkaitan umum antara mereka, sering kali dengan menyertakan Multiplicity.
	<i>Directed Association</i> adalah hubungan antar kelas di mana satu kelas mengarah ke kelas lain, menunjukkan bahwa satu kelas digunakan atau memanfaatkan kelas lainnya. Biasanya, asosiasi ini juga disertai dengan multiplicity.
	<i>Generalisasi</i> adalah hubungan antar kelas yang menggambarkan hubungan generalisasi-spesialisasi (dari umum ke khusus).
	<i>Dependency</i> adalah hubungan antar kelas yang menunjukkan ketergantungan atau keterkaitan antara kelas.
	<i>Aggregation</i> adalah hubungan antar kelas yang menggambarkan hubungan "seluruh-bagian" (whole-part).