

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terkait**

Pada tahun 2020 Rahmad Doni dan Maulia Rahman mengembangkan sistem pemantauan untuk tanaman hidroponik dengan menggunakan NodeMCU ESP8266 sehingga aplikasi android dapat digunakan untuk memantai tanaman melalui internet. Selanjutnya metode *fuzzy* digunakan untuk menentukan waktu yang tepat untuk menyiram tanaman dan menambah ke tangki penampung hidroponik[8]. Data tanaman yang diperoleh berasal dari sensor DHT11 dan Sensor Air untuk melakukan pengaturan waktu penyiraman serta memungkinkan penambahan air ke dalam tangki hidroponik. Hasil sensor diintegrasikan ke dalam sistem ini untuk memungkinkan sistem beroperasi secara otomatis berdasarkan kondisi cuaca sekaligus memberikan umpan balik tentang level air dalam tangki hidroponik. Jika level air turun di bawah ambang batas yang telah ditentukan, sistem akan menambahkan air secara otomatis untuk menjaga keseimbangan dan ketersediaan air bagi tanaman.

Disisi lain, Andi Firwansyah dan Imam Suharjo dalam penelitiannya menetapkan tujuan untuk mempertahankan rentang kandungan kelembapan optimum dari media tanam sebesar 80%-90% dengan merancang alat yang mampu mengotomatiskan proses penyiraman[9]. Alat ini memeriksa berbagai parameter seperti pH media tanam, kelembapan tanah, dan suhu tanaman menggunakan perangkat lunak android sebagai antarmuka pengguna utama. Data yang dihasilkan oleh sensor seperti sensor modul pH tanah, sensor kelembapan tanah, dan DHT11 dikirim ke database firebase selama penyimpanan yang memungkinkan pengguna mendapatkan informasi. Komponen perangkat keras terdiri dari arduino uno, NodeMCU Esp8266, dan relay 2 channel yang mampu bekerja sama untuk memantau sistem penyiraman dan mengontrol penyiraman tanaman.

Ada juga penelitian yang dilakukan oleh Tio Putra Jaikusuma dalam laporan yang berjudul implementasi aplikasi semut kebun untuk monitoring kontrol tanaman timun, dengan memanfaatkan sensor DHT-22 dan kelembapan tanah yang di rangkai sedemikian rupa menjadi sebuah perangkat monitoring tanaman timun yang dapat digunakan untuk memonitoring tanaman dari jarak jauh[10].

Berikut daftar pengembangan dari penelitian yang di lakukan oleh Tio Putra Jaikusuma:

Tabel 2. 1 Tabel Pengembangan

<b>IMPLEMENTASI APLIKASI SEMUT KEBUN UNTUK MONITORING DAN KONTROL TANAMAN TIMUN (TIO PUTRA JAIKUSUMA)</b>	<b>PENGEMBANGAN HYDROPRO UNTUK BUDIDAYA TANAMAN TOMAT (TUADA RAHADATUL AISY)</b>
Memonitoring <i>temperature</i> , dan <i>humidity</i> .	Memonitoring sensor pH, <i>turbidity</i> , <i>humidity</i> , <i>temperature</i> , dan EC.
Halaman <i>History</i> menampilkan data sensor selama 7 hari.	Halaman <i>History</i> masih tetap tersimpan meskipun sudah melewati 7 hari.
Grafik menampilkan data seluruh sensor selama setiap jam	Grafik menampilkan data setiap sensor selama 1 jam dalam satu hari dengan <i>range</i> tanggal tertentu yang dapat di unduh.
Kontrol sistem berupa tombol <i>on</i> dan <i>off</i> untuk tandon dan tombol auto untuk melakukan penyiraman otomatis.	Kontrol sistem berupa, penyiraman otomatis, vitamin otomatis, pompa vitamin A dan B, isi tandon air serta <i>spray</i> area.

IMPLEMENTASI APLIKASI SEMUT KEBUN UNTUK MONITORING DAN KONTROL TANAMAN TIMUN (TIO PUTRA JAIKUSUMA)	PENGEMBANGAN HYDROPRO UNTUK BUDIDAYA TANAMAN TOMAT (TUADA RAHADATUL AISY)
Belum Ada	<i>Setting</i> waktu penyiraman yang terintegrasi dengan fase penyiraman.
Belum Ada	<i>Setting range</i> pH dan <i>turbidity</i> (PPM) untuk memastikan pH dan <i>turbidity</i> (PPM) tetap berada di <i>range</i> yang ditetapkan.

## 2.2 Landasan Teori

### 2.2.1 Internet Of Things (IoT)

IoT merupakan sebuah istilah yang berasal dari gabungan kata *Internet* dan *Things* (benda). Secara sederhana, IoT dapat diartikan sebagai konsep dimana berbagai benda atau perangkat saling terhubung melalui jaringan internet dan dapat saling berinteraksi serta bertukar data secara otomatis. Teknologi IoT telah banyak dimanfaatkan diberbagai bidang salah satunya dalam sektor pertanian melalui penerapan *smart farming*, untuk meningkatkan efisiensi dan produktivitas pertanian[11].

Menurut (Abdul Basit et al., 2023) (Abdul Basit & Budihartono, 2023) dalam jurnalnya, mengatakan bahwa kebutuhan utama dalam proyek berbasis *Internet of Things* diantaranya *microcontroller* yang berfungsi sebagai pengontrol rangkaian elektronik dan dapat menyimpan sebuah kode program penanaman perintah yang umumnya menggunakan Bahasa pemrograman C

dengan tools arduino IDE untuk mengerjakan tugas tertentu yang dikehendaki oleh pemrogramnya dengan proses penerimaan data yang diberikan oleh sensor[12].

### 2.2.2 Proses Monitoring Tanaman

Dalam Kamus Besar Bahasa Indonesia (KBBI), *monitoring* dapat disebut juga pemantauan. Dimana *monitoring* merupakan proses pengukuran dan evaluasi yang di selesaikan secara berkala, untuk memberikan informasi tentang status dan kecenderungan terhadap suatu objek[13]. Proses *monitoring* tanaman bertujuan untuk mengawasi kondisi dan pertumbuhan tanaman secara berkala, sehingga dapat mengambil tindakan yang tepat untuk meningkatkan produktivitas dan kualitas panen. Dengan adanya perkembangan teknologi proses monitoring dapat dilakukan sedangkan menggunakan aplikasi *mobile*. Dimana data akan secara *real time* dan terstruktur pada fitur fitur yang di sediakan pada sebuah aplikasi.

### 2.2.3 Aplikasi Android

Aplikasi Android merupakan sebuah perangkat lunak yang di rancang untuk berjalan pada sistem operasi android yang di kenalkan oleh google. Aplikasi ini dibuat khusus untuk perangkat seperti smartphone yang menggunakan android sebagai sistem operasinya.

### 2.2.4 ESP32

*Espressif System* memperkenalkan mikrokontoller penerus dari Esp8266 yaitu ESP32[14]. Sama halnya dengan mikrokontroler lainnya, ESP32 digunakan untuk mengirimkan data, data yang ESP32 dikirimkan melalui wifi menuju ke database. ESP32 dapat dilihat pada Gambar 2.1 Gambar ESP32



Gambar 2. 1 Gambar ESP32

#### 2.2.5 Android Studio

Salah satu IDE pengembangan yang dirilis oleh google adalah Android Studio. Melalui Android SDK (*Software Development Kit*), Android Studio menawarkan banyak fitur, seperti *debugger*, *libraries* dan *emulator*[15] Fitur fitur ini dapat membantu *developer* aplikasi Android menjadi lebih produktif. Salah satunya dengan menyediakan berbagai Bahasa pemrograman dart, java, dan Kotlin.

#### 2.2.6 Figma

Figma merupakan sebuah alat berupa perangkat lunak yang digunakan untuk membuat desain, biasanya figma digunakan untuk membuat design sebuah aplikasi baik desain secara *low-Fidelity* maupun *Hight fidelity*. Figma menyediakan berbagai fitur dan *plugin* yang mendukung dapat pembuatan design, melalui fitur *prototype design*, figma mampu melakukan simulasi sebuah rancangan baik aplikasi maupun *website* sesuai fungsional yang semestinya. Ikon figma dapat dilihat pada Gambar 2. 2 Ikon Figma.



Gambar 2. 2 Ikon Figma

#### 2.2.7 Firebase

Firebase merupakan sebuah platform yang dirancang untuk menyimpan sebuah data melalui Firebase database. Firebase

database merupakan penyimpanan database *non relation* atau *nonSQL* yang memfasilitasi penyimpanan berbagai tipe data, namun penting untuk dicatat bahwa metode penyimpanan ini tidak menggunakan *tableSQL* melainkan menggunakan struktur data dalam bentuk JSON[16]. Ikon Firebase dapat dilihat pada Gambar 2. 3 Ikon Firebase.



Gambar 2. 3 Ikon Firebase

#### 2.2.8 Firebase Realtime Database

Aplikasi berbasis *real time*, tentu membutuhkan platform database yang mendukung seperti halnya Firebase Realtime Database. Dengan menggunakan database ini, pengguna dapat mengakses dan melihat data terkini secara langsung, tanpa melakukan *refresh* atau *reload*. Fitur ini berguna untuk aplikasi yang memerlukan pembaruan data secara instan, seperti yang di terapkan pada aplikasi sistem *monitoring*, sehingga lebih *responsive*, interaktif dan efisien.

#### 2.2.9 Kotlin

JetBrains dikenal sebagai perusahaan yang juga mengembangkan IDE Android Studio, salah satu bahasa yang dikembangkan oleh JetBrains adalah kotlin[17]. Kotlin merupakan bahasa pengembangan dari bahasa java, dengan fitur modern yang lebih dari java. Ikon kotlin dapat dilihat pada Gambar 2. 4 Ikon Kotlin.



Gambar 2. 4 Ikon Kotlin

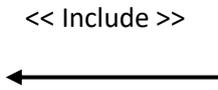
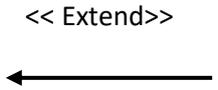
### 2.2.10 UML

Untuk merancang sistem berorientasi objek seperti halnya rancangan perangkat lunak dibutuhkan metode pemodelan visual yang di sebut juga *Unified Modeling Language* (UML)[18]. Merupakan Bahasa baku untuk pendokumentasian, visualisasi, dan perancangan sistem dalam perangkat lunak sehingga membuat pemahaman struktur dan alur kerja perangkat lunak menjadi lebih mudah. UML terdiri dari beberapa konsep yaitu :

#### 1. Use Case

*Use Case Diagram* merupakan representasi visual dari fungsionalitas yang diharapkan dari sebuah sistem serta menggambarkan interaksi antara actor dan sistem. Dalam diagram ini, aktor berperan sebagai entitas, baik manusia maupun sistem lain, yang berinteraksi dan melakukan pekerjaan didalam sistem[19]. *Use case diagram* digunakan untuk memodelkan sistem informasi yang akan dirancang, dengan mendeskripsikan hubungan antara satu atau lebih peran dengan sistem yang di kembangkan. diagram ini membantu dalam memahami tipe interaksi yang terjadi antara pengguna dan sistemnya, sehingga mempermudah proses perancangan dan pengembangan perangkat lunak. *Use case diagram* juga merupakan langkah awal dalam pemodelan sistem, karena mampu menjabarkan aksi yang dilakukan oleh *actor* serta *respons* yang diberikan oleh sistem.

Tabel 2. 2 Simbol Use Case Diagram

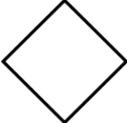
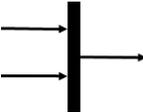
No	Simbol	Nama	Keterangan
1		Aktor	Mewakili peran user, sistem atau alat ketika berkomunikasi dengan <i>use case</i>
2		Use Case	Abstraksi dan interaksi antara sistem dan aktor
3		Association	Abstraksi dari penghubung antara aktor dengan <i>use case</i>
4		Generalisasi	Menunjukkan Spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
5		Include	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsional dari <i>use case</i> lainnya
6		Exclude	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* memodelkan berbagai aliran kegiatan dalam sistem yang dirancang, termasuk bagaimana setiap aliran di mulai, keputusan yang terjadi, serta bagaimana proses tersebut berakhir[19] Dalam UML, *Activity Diagram* atau diagram aktivitas digunakan untuk

memvisualisasikan perilaku dinamis dari suatu sistem atau bagian dari sistem melalui aliran kontrol aksi yang di lakukan.

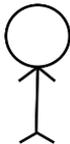
Tabel 2. 3 Simbol Activity Diagram

No	Simbol	Nama	Keterangan
1		Activity	Memperlihatkan bagaimana masing masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Join	Digunukan untuk mewakili aliran (dua input) yang bergabung menjadi satu output
7		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

### 3. Sequence Diagram

*Sequence Diagram* merupakan diagram yang menjelaskan interaksi antar objek dalam suatu sistem berdasarkan urutan waktu[19]. Diagram ini menggambarkan sifat objek dalam *use case* dengan memilih objek yang aktif serta menunjukkan pesan yang dikirim dan diterima antar objek. *Sequence Diagram* juga memvisualisasikan interaksi antara objek didalam dan di sekitar sistem, termasuk pengguna, tampilan, serta komponen lainnya, dengan merepresentasikan pesan yang dikirim. Diagram ini biasanya digunakan untuk menggambarkan *scenario* atau Langkah Langkah yang dilakukan sebagai resnpons terhadap suatu peristiwa untuk menghasilkan *output* tertentu, dengan membantu dalam memahami urutan atau tahapan yang harus dilakukan dalam suatu sistem agar dapat mencapai hasil yang diinginkan.

Tabel 2. 4 Simbol Sequence Diagram

No	Simbol	Nama	Keterangan
1		Actor	Menggambarkan orang yang sedang berinteraksi dengan sistem
2		Entity Class	Menggambarkan hubungan yang akan dilakukan
3		Boundary Class	Menggambarkan sebuah Gambaran dari foem

No	Simbol	Nama	Keterangan
4		Control Class	Menghubungkan penghubung antara boundary dengan tabel
5		A focus of Control & A Life Line	Menggambarkan tempat mulai dan berakhir message
6		A message	Menggambarkan Pengiriman Pesan

#### 4. Class Diagram

*Class Diagram* merupakan diagram yang digunakan dalam pengembangan perangkat lunak. Diagram ini menggambarkan struktur dan hubungan antar kelas dalam suatu sistem. Dengan demikian *Class Diagram* menggambarkan struktur yang berisikan definisi *class* (nama *kelasnya*), atribut, metode serta relasi antar kelasnya.