

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. TEORI TERKAIT

Kajian terhadap penelitian terdahulu perlu dilakukan untuk mengetahui metode dan teknologi *Internet of Things* (IoT) serta pengembangan *website* yang telah diterapkan pada sistem *monitoring* suhu dan kelembapan. Selain itu, kajian ini bertujuan untuk mengidentifikasi kelebihan, keterbatasan, serta celah penelitian yang masih dapat dikembangkan sebagai dasar dalam menentukan pendekatan yang digunakan pada penelitian ini. Rincian perbandingan jurnal terdahulu dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Penelitian Terdahulu

| No | Judul                                                                                                                                                          | Metode                                                             | Hasil Evaluasi                                                                                                                      | Keterbatasan                                                                                                                                                  |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Sistem <i>Monitoring</i> Suhu dan Kelembaban pada Budidaya Jamur Tiram Berbasis <i>Internet of Things</i> Menggunakan <i>MQTT</i> dan Telegram <i>BOT</i> [11] | <i>IoT</i> , Protokol <i>MQTT</i> , Notifikasi Telegram <i>BOT</i> | Sistem mampu mengirimkan data suhu dan kelembapan secara <i>real-time</i> untuk mengurangi kesalahan pencatatan manual di lapangan. | Antarmuka pemantauan lebih berfokus pada penyampaian notifikasi teks di Telegram, belum menyediakan visualisasi <i>website</i> terdedikasi yang mudah dibaca. |

|    |                                                                                                                    |                                                         |                                                                                                                       |                                                                                                                                                     |
|----|--------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 2. | <i>Weather-Baglog Parameters Monitoring System Based IoT-MQTT-NodeRED for Mushroom Cultivation Room</i> [12]       | <i>IoT, Protokol MQTT, Platform Node-RED</i>            | Berhasil memantau kondisi cuaca secara spesifik pada baglog jamur dan menampilkan data secara <i>real-time</i> .      | Antarmuka bawaan dari <i>Node-RED</i> cenderung bersifat teknis dan kaku, sehingga kurang <i>user-friendly</i> untuk dioperasikan oleh petani awam. |
| 3. | Pengawasan dan Pemantauan Pertumbuhan Jamur Tiram dengan Memanfaatkan <i>Teknologi Internet of Things</i> (IoT)[5] | Perangkat Keras ( <i>Hardware</i> ) berbasis <i>IoT</i> | Membuktikan bahwa pengawasan jarak jauh berbasis <i>IoT</i> mampu meningkatkan efisiensi proses budidaya jamur tiram. | Berfokus kuat pada keandalan instrumen <i>hardware</i> , namun kurang menguraikan pengembangan antarmuka ( <i>UI/UX</i> ) bagi pengguna akhir.      |
| 4. | Simulasi <i>Internet of Things</i> (IoT) Pada Budi Daya Jamur Tiram [8]                                            | Simulasi Jaringan <i>IoT</i>                            | Berhasil melakukan simulasi sistem otomatisasi lingkungan budidaya jamur untuk menjaga parameter suhu dan kelembapan. | Riset masih terbatas pada tahap simulasi, belum diimplementasikan menjadi sistem <i>website</i> interaktif yang siap                                |

|    |                                                                                                                                                              |                                             |                                                                                                                                      |                                                                                                                                |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
|    |                                                                                                                                                              |                                             |                                                                                                                                      | pakai di kebun.                                                                                                                |
| 5. | Implementasi <i>Internet Of Things</i> (IoT) Dalam <i>Monitoring</i> Suhu Dan Kelembaban Ruang Produksi Obat Non Steril Menggunakan Arduino Berbasis Web[13] | Mikrokontroler Arduino, Sistem Berbasis Web | Data dikirim ke <i>server web</i> secara stabil, memungkinkan akses pemantauan yang fleksibel dan penyimpanan data yang terstruktur. | Objek pemantauan ditujukan untuk ruang produksi obat, tidak didesain spesifik untuk kebutuhan informasi pertanian jamur tiram. |

## 2.2. LANDASAN TEORI

*Website* merupakan media berbasis internet yang berfungsi sebagai antarmuka (*interface*) untuk menyampaikan informasi kepada pengguna melalui halaman web yang dapat diakses menggunakan *web browser*[14]. Dalam sistem berbasis *Internet of Things* (IoT), *website* berperan sebagai penghubung antara perangkat sensor dengan pengguna sehingga data hasil pengukuran dapat ditampilkan secara *real-time*, disimpan dalam basis data, dan diakses dari berbagai lokasi melalui jaringan internet[15].

Arsitektur *website* pada sistem *monitoring* umumnya terdiri atas dua komponen utama, yaitu *frontend* dan *backend*. *Frontend* berfungsi menyajikan antarmuka pengguna (*user interface*) dalam bentuk halaman web, sedangkan *backend* bertugas mengelola logika aplikasi, komunikasi dengan basis data, serta pertukaran data antara *website* dan perangkat

*Internet of Things* (IoT)[16]. Dengan adanya integrasi tersebut, pengguna dapat memantau kondisi lingkungan budidaya secara lebih mudah tanpa harus berada langsung di lokasi kumbung.

Pada penelitian ini, *website* digunakan sebagai media *monitoring* suhu dan kelembapan rumah jamur tiram. Data yang dikirimkan oleh sensor ditampilkan dalam bentuk nilai *real-time*, grafik historis, indikator kondisi, serta riwayat pengukuran sehingga pengguna dapat mengetahui perubahan kondisi lingkungan secara cepat dan akurat[17]. Penyajian data secara visual membantu pengguna dalam melakukan evaluasi terhadap kondisi lingkungan budidaya serta mendukung pengambilan keputusan secara lebih efektif[18].

Suhu dan kelembapan merupakan parameter lingkungan yang sangat memengaruhi pertumbuhan jamur tiram (*Pleurotus ostreatus*). Berbagai penelitian menunjukkan bahwa suhu optimum untuk pertumbuhan jamur tiram berada pada kisaran 22–25°C, dengan kelembapan relatif 80–90% RH. Kondisi lingkungan yang berada di luar kisaran tersebut dapat menghambat pertumbuhan miselium, pembentukan tubuh buah, serta menurunkan kualitas hasil budidaya. Oleh karena itu, pengendalian suhu dan kelembapan menjadi aspek penting dalam proses budidaya jamur tiram[19].

Berdasarkan parameter tersebut, sistem *monitoring* pada penelitian ini menerapkan klasifikasi kondisi lingkungan sebagaimana ditunjukkan pada Tabel 2.2. & 2.3. Informasi status lingkungan ditampilkan secara

otomatis pada *website* sehingga pengguna dapat mengetahui apakah kondisi kumbung berada pada kategori Optimal, Terlalu Dingin, Terlalu Panas, Terlalu Kering, atau Terlalu Lembap.

Tabel 2. 2 Nilai Threshold Kondisi Lingkungan Budidaya Jamur Tiram

| <b>Parameter</b>    | <b>Nilai Threshold</b> | <b>Keterangan</b>              |
|---------------------|------------------------|--------------------------------|
| Suhu Minimum        | 22°C                   | Batas bawah Suhu Optimum       |
| Suhu Maksimum       | 25°C                   | Batas atas Suhu Optimum        |
| Kelembapan Minimum  | 80% RH                 | Batas bawah Kelembapan Optimum |
| Kelembapan Maksimum | 90% RH                 | Batas atas kelembapan optimum  |

Tabel 2. 3 Parameneter Suhu & Kelembapan

| <b>Parameter</b> | <b>Nilai Threshold</b> | <b>Status</b>  |
|------------------|------------------------|----------------|
| Suhu             | < 22°C                 | Terlalu Dingin |
| Suhu             | 22–25°C                | Optimal        |
| Suhu             | > 25°C                 | Terlalu Panas  |
| Kelembapan       | < 80% RH               | Terlalu Kering |
| Kelembapan       | 80–90% RH              | Optimal        |
| Kelembapan       | > 90% RH               | Terlalu Lembap |

### 2.2.1. Visual Studio Code

Visual Studio Code adalah sebuah aplikasi editor code *open source* yang dikembangkan oleh Microsoft untuk sistem operasi *Windows*, *Linux*, dan *MacOS*. Visual Studio Code memudahkan penulisan code yang mendukung berbagai jenis pemrograman, seperti *C++*, *C#*, *Java*, *Python*, *PHP*, dan lain-lain[20]. Logo Visual Studio Code yang digunakan sebagai lingkungan pengembangan aplikasi pada penelitian ini dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Logo Visual Studio Code

### 2.2.2. Arduino IDE

Aplikasi untuk membuat *sketch* pemrograman, yang berfungsi sebagai alat untuk memprogram board yang ingin diprogram. *Software* ini membantu dalam mengedit, membuat mengunggah ke board yang dipilih, dan menulis program tertentu. Arduino IDE berbasis *Java* dan dilengkapi dengan *library C/C++* untuk mempermudah operasi *input* dan *output*[21]. Logo Arduino IDE yang digunakan sebagai perangkat lunak pemrograman mikrokontroler pada penelitian ini dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Logo Arduino IDE

### 2.2.3. Laragon

Perangkat lunak Laragon merupakan salah satu *tools* yang digunakan sebagai lingkungan pengembangan lokal (*local development environment*) dalam pembuatan aplikasi berbasis web. Laragon menyediakan berbagai database, serta *PHP* sebagai Bahasa pemrograman yang saling terintegrasi dalam satu *system*. Dengan adanya lingkungan pengembangan lokal. Proses pembuatan dan pengujian aplikasi dapat dilakukan secara lebih efisien tanpa harus menggunakan server online. Hal ini sejalan dengan penelitian yang menyatakan bahwa penggunaan *framework* Laravel dengan dukungan web server dan database *MySQL* dapat mempermudah pengembangan aplikasi berbasis web serta meningkatkan efisiensi pengelolaan *system*[22]. Logo Laragon yang digunakan sebagai lingkungan pengembangan lokal pada penelitian ini ditunjukkan pada Gambar 2.3.



Gambar 2. 3 Logo Laragon

#### 2.2.4. MySQL

*MySQL* merupakan salah satu system manajemen basis data (Database Management System/DBMS) yang paling banyak digunakan dalam pengembangan aplikasi berbasis web dinamis. *MySQL* bersifat *open source* dan mampu menangani pengelolaan data dalam jumlah besar serta dukungan penggunaan multiuser secara bersamaan. Selain itu, *MySQL* sering digunakan Bersama Bahasa pemrograman seperti *PHP* dalam membangun aplikasi web karena kemudahan integrasi dan kinerjanya yang cukup baik. Hal ini didukung oleh penelitian yang menyatakan bahwa *MySQL* merupakan *DBMS* yang banyak digunakan dalam pengembangan sistem informasi berbasis web karena bersifat *open source*, mudah digunakan, serta mampu menangani pengolahan data secara efisien[23].

#### 2.2.5. PhpMyAdmin

Dalam pengembangan sistem informasi berbasis web, *PhpMyAdmin* digunakan sebagai alat bantu untuk pengelolaan basis data *MySQL*

karena menyediakan antarmuka yang memudahkan proses administrasi database[24]. Logo *PhpMyAdmin* yang digunakan sebagai alat bantu pengelolaan basis data pada penelitian ini ditunjukkan pada Gambar 2.4.



Gambar 2. 4 Logo PhpMyAdmin

#### 2.2.6. *HyperText Markup Language (HTML)*

*HyperText Markup Language (HTML)* merupakan bahasa markup standar yang digunakan untuk membangun dan menyusun struktur dasar halaman web. *HTML* berfungsi sebagai struktur atau kerangka dasar sebuah halaman web yang kemudian dapat dikombinasikan dengan *CSS* dan *JavaScript* untuk menghasilkan halaman web yang responsif[25].

#### 2.2.7. *Cascading Style Sheet (CSS)*

*Cascading Style Sheet (CSS)* merupakan bahasa *stylesheet* yang digunakan untuk mengatur tampilan dan desain halaman web yang dibuat menggunakan *HTML*. *CSS* memungkinkan pengembang untuk mengatur berbagai aspek visual, seperti warna, tata letak, ukuran teks, jenis huruf, dan jarak antar elemen sehingga halaman web menjadi

lebih menarik dan mudah digunakan. Selain itu, CSS juga mendukung pembuatan tampilan yang responsif sehingga *website* dapat menyesuaikan ukuran layar pada berbagai perangkat, baik komputer maupun perangkat mobile[26].

#### 2.2.8. Tailwind CSS

*Tailwind CSS* merupakan *framework CSS* yang menerapkan konsep *utility-first*, yaitu menyediakan kumpulan kelas utilitas yang dapat digunakan secara langsung pada elemen *HTML* untuk membangun antarmuka pengguna. Dengan pendekatan tersebut, pengembang dapat membuat desain *website* tanpa harus menulis banyak kode CSS secara terpisah. *Tailwind CSS* menyediakan berbagai utilitas untuk mengatur tata letak, warna, tipografi, ukuran, serta desain *responsif* sehingga tampilan *website* dapat menyesuaikan berbagai ukuran perangkat. Penggunaan *Tailwind CSS* dapat mempercepat proses pengembangan antarmuka dan memberikan fleksibilitas yang lebih tinggi dalam menentukan gaya dan tata letak sesuai kebutuhan *website*[27]. Logo *Tailwind CSS* yang digunakan sebagai *framework* antarmuka pada penelitian ini ditunjukkan pada Gambar 2.5.



Gambar 2. 5 Tailwind CSS

### 2.2.9. PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman yang berjalan pada sisi server (*server-side scripting*) dan banyak digunakan dalam pengembangan aplikasi berbasis web. *PHP* berfungsi untuk mengolah data, menjalankan logika aplikasi, mengelola interaksi dengan basis data, serta menghasilkan halaman web yang dinamis sesuai kebutuhan pengguna. Bahasa pemrograman ini bersifat *open source*, mudah dipelajari, dan mendukung berbagai sistem operasi maupun web server sehingga menjadi salah satu bahasa pemrograman yang populer dalam pengembangan sistem informasi berbasis web. Pada implementasinya, *PHP* sering digunakan untuk mengelola data yang tersimpan dalam basis data dan mendukung proses pengolahan informasi pada aplikasi web[28].

### 2.2.10. JavaScript

*JavaScript* merupakan salah satu bahasa pemrograman web yang digunakan untuk memperkaya tampilan dan fungsionalitas aplikasi web. *JavaScript* banyak digunakan dalam pengembangan *website* bersama *HTML* dan *CSS* untuk mendukung pembuatan aplikasi web yang lebih interaktif serta sesuai dengan kebutuhan pengguna[29].

### 2.2.11. Laravel

Laravel merupakan *framework PHP* yang digunakan untuk mempermudah proses pengembangan aplikasi berbasis web. Laravel menyediakan berbagai fitur yang mendukung pengembangan aplikasi,

seperti sistem *routing*, autentikasi, manajemen basis data, serta struktur kode yang terorganisir sehingga memudahkan pengembang dalam membangun dan memelihara aplikasi. Dengan berbagai fitur yang tersedia, Laravel dapat membantu meningkatkan efisiensi dan produktivitas dalam proses pengembangan *website*. Selain itu, Laravel banyak digunakan dalam pembangunan sistem informasi berbasis web karena memiliki sintaks yang mudah dipahami dan mendukung pengembangan aplikasi yang terstruktur[30].

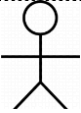
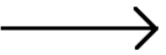
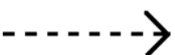
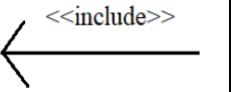
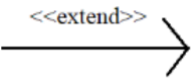
#### 2.2.12. UML (*Unified Modeling Language*)

UML (*Unified Modeling Language*) merupakan salah satu alat bantu yang digunakan dalam perancangan dan pengembangan perangkat lunak berbasis *object-oriented*. UML digunakan untuk memvisualisasikan, merancang, serta mendokumentasikan sistem melalui berbagai diagram seperti *use case* diagram, *activity* diagram, dan *class* diagram. Dengan adanya UML, pengembang dapat memahami alur sistem secara lebih terstruktur dan sistematis sebelum tahap implementasi dilakukan. Hal ini didukung oleh penelitian yang menyatakan bahwa UML berperan penting dalam membantu proses analisis dan perancangan sistem karena mampu memodelkan kebutuhan sistem secara visual dan terstruktur[31].

Ada beberapa diagram UML yang biasa digunakan dalam pengembangan sistem, yaitu:

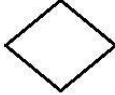
1. *Use case*: Menggambarkan dari fungsionalitas yang dibutuhkan menurut sebuah sistem, dan merepresentasikan sebuah hubungan antara actor dan sistem.

Tabel 2. 4 Simbol *Use Case*

| Simbol                                                                              | Nama               | Keterangan                                                                                                                       |
|-------------------------------------------------------------------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Actor</i>       | Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .                                  |
|    | <i>Use case</i>    | Abstraksi dan interaksi antara sistem dan aktor                                                                                  |
|   | <i>Association</i> | Abstraksi dari penghubung antara aktor dan <i>use case</i> .                                                                     |
|  | Generalisasi       | Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .                                               |
|  | <i>include</i>     | Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.                        |
|  | <i>extend</i>      | Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi. |

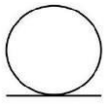
2. *Activity Diagram*: Menggambarkan perilaku dinamis dari suatu sistem atau bagian dari sistem melalui aliran kontrol antara aksi yang dilakukan sistem.

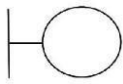
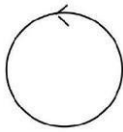
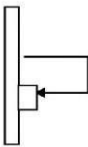
Tabel 2. 5 Simbol *Activity Diagram*

| Simbol                                                                              | Nama                             | Keterangan                                                                                           |
|-------------------------------------------------------------------------------------|----------------------------------|------------------------------------------------------------------------------------------------------|
|    | Status awal                      | Sebuah diagram aktivitas memiliki sebuah status awal.                                                |
|    | Aktivitas                        | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.                       |
|    | Percabangan atau <i>Decision</i> | Percabangan dimana ada pilihan aktivitas yang lebih dari satu.                                       |
|  | Penggabungan atau <i>Join</i>    | Penggabungan dimana yang mana lebih dari satu aktivitas lalu digabungkan jadi satu.                  |
|  | Status Akhir                     | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.           |
|  | <i>Swimlane</i>                  | <i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

3. *Sequence Diagram*: Biasa digunakan untuk menggambarkan skenario atau langkah-langkah yang dilakukan sebagai respon dari sebuah event tertentu untuk menghasilkan output tertentu.

Tabel 2. 6 Simbol *Sequence Diagram*

| Simbol                                                                              | Nama                | Keterangan                                                  |
|-------------------------------------------------------------------------------------|---------------------|-------------------------------------------------------------|
|  | <i>Entity Class</i> | Gambaran sistem sebagai landasan dalam menyusun basis data. |

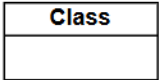
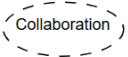
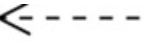
|                                                                                    |                       |                                                                           |
|------------------------------------------------------------------------------------|-----------------------|---------------------------------------------------------------------------|
|   | <i>Boundary Class</i> | Menangani komunikasi antar lingkungan sistem.                             |
|   | <i>Control Class</i>  | Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika. |
|   | <i>Recursive</i>      | Pesan untuk dirinya.                                                      |
|   | <i>Activation</i>     | Mewakili proses durasi aktivasi sebuah operasi.                           |
|  | <i>Life Line</i>      | Komponen yang digambarkan garis putus terhubung dengan objek              |

#### 4. Class Diagram

*Class Diagram* merupakan gambaran struktur dan deskripsi dari *class*, *package*, dan objek yang saling keterkaitan seperti diantaranya pewarisan, asosiasi, dan lainnya.

Tabel 2. 7 Tabel *Class Diagram*

| Gambar                                                                              | Nama                    | Keterangan                                                                                                             |
|-------------------------------------------------------------------------------------|-------------------------|------------------------------------------------------------------------------------------------------------------------|
|  | <i>Generalization</i>   | Hubungan di mana objek anak ( <i>descendant</i> ) mewarisi perilaku dan struktur dari objek induk ( <i>ancestor</i> ). |
|  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.                                                            |

| Gambar                                                                              | Nama                      | Keterangan                                                                                                                                                   |
|-------------------------------------------------------------------------------------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <i>Class</i>              | Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.                                                                                     |
|    | <i>Collaborati<br/>on</i> | Deskripsi dari urutan aksi yang ditampilkan dalam sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.                                        |
|    | <i>Dependency</i>         | Operasi yang benar-benar dilakukan oleh suatu objek.                                                                                                         |
|  | <i>Dependency</i>         | Hubungan di mana perubahan yang terjadi pada suatu elemen ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri. |
|  | <i>Association</i>        | Apa yang menghubungkan antara objek satu dengan objek lainnya.                                                                                               |