

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada penelitian yang dilakukan oleh Ezha Tiara,dkk (2023) yang berjudul “Sistem Pemantauan dan Kendali Kelembapan Tanah dan pH pada Tanaman Anggur Berbasis Android”. Budidaya anggur di Indonesia masih belum merata meskipun prospek penjualannya tinggi dan sangat populer di Indonesia. Hal ini disebabkan oleh anggur memerlukan perhatian dan perawatan yang intensif dalam waktu yang lama agar anggur dapat menghasilkan hasil panen. *Greenhouse* menyediakan kondisi lingkungan yang ideal bagi tanaman, namun terbatas dalam kemampuannya untuk memantau dan mengendalikan faktor-faktor pertumbuhan secara *real-time*. Oleh karena itu, dibutuhkan suatu sistem yang dapat memantau dan mengendalikan kelembapan tanah dan pH air pada tanaman anggur di dalam greenhouse melalui aplikasi android. Penelitian ini menggunakan NodeMCU ESP32 untuk mengendalikan keseluruhan komponen perangkat keras dan perangkat lunak. Sensor capacitive soil moisture untuk pembacaan nilai kelembapan tanah, dan sensor pH Meter untuk pembacaan nilai tingkat keasaman air. Pengguna dapat melakukan pengendalian perangkat keras secara manual ataupun otomatis melalui antarmuka aplikasi android. Sistem kendali otomatis dapat berfungsi dengan baik. Ketika nilai kelembapan tanah

berada di bawah 60% maka pompa air akan hidup (on), sedangkan apabila nilai pH dibawah 5,5 maka pompa alkali juga akan hidup (on)[3].

Selanjutnya pada penelitian yang dilakukan oleh Firdaus (2023) yang berjudul “Sistem Manajemen Pengairan Pada Budidaya Tanaman Anggur Berbasis *Internet of Things (Iot)*”. Dalam budidaya anggur di daerah tropis, penting untuk mengatur kelembapan tanah. Pengendalian kelembapan tanah dapat dilakukan melalui penyiraman air. Namun, penyiraman manual memerlukan pemantauan yang terus-menerus terhadap kelembapan tanah. Oleh karena itu, diperlukan sistem berbasis *Internet of Things (IoT)* yang dapat memantau dan mengendalikan penyiraman tanaman secara otomatis. ESP32 digunakan sebagai mikrokontroler, sensor kelembapan tanah untuk pengukuran, dan *relay* untuk mengendalikan *solenoid valve* dan pompa air. Informasi hasil pengukuran dan status kontrol ditampilkan melalui *website*, dimana pengguna juga dapat menetapkan jadwal penyiraman. Sistem akan melakukan penyiraman ketika kelembapan tanah mencapai $\leq 60\%$, atau sesuai dengan jadwal yang telah ditetapkan pengguna jika kelembapan tanah berada di rentang 61-74%. Hasil uji menunjukkan tingkat kesalahan pengukuran kelembapan tanah oleh tiga sensor dengan rata-rata masing-masing adalah 2,24%, 1,68%, dan 1,11%. Waktu respon rata-rata *relay* dan pompa air adalah 2,32 detik hingga 3,76 detik. Meskipun demikian, perlu diingat bahwa respons ini dapat mempengaruhi kelembapan tanah, yang bisa melebihi 75% [4].

Pada penelitian yang dilakukan Kusuma Raharja (2022) yang berjudul “Pengaplikasian *Internet of Things* Untuk *Monitoring* Lingkungan Lahan Tanaman Anggur”. Perkembangan teknologi informasi dan komunikasi sangat pesat, dengan salah satu fokus utamanya adalah teknologi *Internet of Things (IoT)*. Teknologi ini dapat dimanfaatkan dalam berbagai aplikasi, termasuk alat *telemetry* di bidang pertanian untuk memantau lahan tanaman anggur secara *real-time* dan mengontrolnya dari jarak jauh. Parameter yang dapat dimonitor meliputi kelembapan tanah, intensitas cahaya, kelembapan udara, dan suhu udara. Penelitian ini bertujuan untuk merancang dan membangun alat *telemetry* berbasis *IoT* untuk lingkungan lahan tanaman anggur. Aplikasi ini cocok untuk *smart farming*, terutama bagi pengelola tanaman anggur. Alat yang dikembangkan terdiri dari tiga sensor: DHT11 untuk kelembapan dan suhu, LDR untuk intensitas cahaya, dan YL-69 untuk kelembapan tanah. Arduino ESP32 digunakan sebagai unit pemrosesan semua variabel tersebut, dan menggunakan aplikasi *Blynk* untuk menampilkan hasil pada *smartphone*. Selain itu, alat ini memiliki kemampuan untuk melakukan penyiraman otomatis dengan pompa air dan melacak tanaman anggur melalui kamera. Hasil menunjukkan akurasi rata-rata 72,85% untuk intensitas cahaya, 93,75% untuk suhu, dan 87,25% untuk kelembapan tanah, berdasarkan pengujian terhadap tiga parameter lingkungan dan dibandingkan dengan alat sejenis [5].

Pada penelitian yang dilakukan Restu (2023) yang berjudul “Rancang Bangun Sistem *Monitoring* Serta Sistem Otomatisasi Penyiraman Dan

Pemberian Vitamin Pada Tanaman Anggrek”. Anggrek adalah tanaman yang digemari banyak orang. Dalam budidaya anggrek, pemantauan kondisi tanaman sangat penting, karena serangan hama menjadi salah satu faktor yang menurunkan ekspor bunga anggrek. Merawat tanaman anggrek membutuhkan perhatian terhadap suhu, kelembapan, dan intensitas cahaya yang sesuai untuk pertumbuhan optimal. Oleh karena itu, dibuat alat untuk mengatasi masalah perawatan anggrek, seperti kekurangan air dan nutrisi vitamin. Alat ini menggunakan sensor DHT22 yang bekerja ketika suhu dan kelembapan kurang dari ideal, serta sensor intensitas cahaya untuk memantau kebutuhan cahaya matahari. Hasil pengujian menunjukkan bahwa alat ini dapat secara otomatis memantau suhu, kelembapan udara, dan intensitas cahaya [6].

Pada penelitian yang dilakukan oleh Septyanto (2023) yang berjudul “Sistem *Monitoring* Penyiraman Tanaman Otomatis Menggunakan Mikrokontroler ESP8266, DHT11, Dan Soil Berbasis *Web*”. Tanaman di taman sering disiram secara manual, yang dianggap kurang efisien karena memakan waktu dan seringkali terlupakan, terutama jika tanaman tersebut jauh dari rumah pemiliknya. Jika tanaman hias yang perlu disiram, masalahnya menjadi lebih rumit. Teknologi penyiraman otomatis untuk taman kota adalah solusi yang dapat digunakan. Teknologi ini akan menyiram tanaman secara otomatis, menghindari kekeringan. Solusi terbaik untuk menerapkan penyiraman otomatis ini adalah teknologi *Internet of Things* (IoT). Tujuan dari penelitian ini adalah untuk membuat alat yang memantau

penyiraman otomatis pada tanaman yang menggunakan sensor suhu DHT11 dan sensor kelembapan tanah sebagai *input* dan *output*. Mikrokontroler NodeMcu ESP-8266 berfungsi sebagai perantara yang menghubungkan alat ke internet melalui *WiFi*. Ini memungkinkan sensor DHT11 dan sensor kelembapan tanah untuk menyimpan dan menampilkan data suhu dan kelembapan tanah pada layar LCD dan *server web* [7].

2.2 Landasan Teori

2.2.1 Sistem *Monitoring*

Sistem *monitoring* adalah proses pengumpulan data secara *real-time* dari berbagai sumber dikenal sebagai sistem pemantauan. Tahapannya meliputi pengumpulan data, analisis data, dan penyajian data sebagai informasi. Semua tahapan ini dilakukan sesuai dengan konsep dasar *input-proses-output*. Tujuannya adalah untuk memudahkan pengguna untuk melihat setiap perubahan dan perkembangan kondisi [8].

2.2.2 Arduino IDE

Arduino IDE atau *Integrated Development Environment*, adalah *platform* yang dipakai untuk mengirim instruksi ke *board* Arduino, sehingga *board* tersebut dapat menghasilkan *output* sesuai dengan program yang dibuat. Perangkat lunak Arduino terdiri dari aplikasi yang berbasis pada bahasa pemrograman *processing*, yang dipakai untuk menuliskan kode program untuk *board* Arduino Uno. Perangkat

lunak Arduino mengkombinasikan elemen dari bahasa *C++* dan *Java*. Arduino IDE dapat diinstal di berbagai sistem operasi, termasuk Linux, MacOS, dan Windows. IDE Arduino adalah lingkungan pengembangan terpadu yang menggunakan bahasa pemrograman *C* [9].

2.2.3 Visual Studio Code

Visual Studio Code merupakan program buatan Microsoft yang fokus pada pengembangan aplikasi *web*, terutama menggunakan *ASP.NET* dan *Node.js*. Aplikasi ini juga dapat digunakan untuk mengembangkan teknologi sejenis seperti *HTML*, *CSS*, *Less*, *Sass*, dan *JSON*. Visual Studio Code mendukung berbagai bahasa pemrograman termasuk *Batch*, *C++*, *PHP*, *SQL*, *Ruby*, *Razor*, *Visual Basic*, *Java*, *XML*, dan banyak lagi [10].

2.2.4 XAMPP

XAMPP merupakan singkatan dari X (yang menunjukkan empat sistem operasi): Apache, MySQL, PHP, dan Perl. Perangkat lunak ini berfungsi sebagai *server* lokal dan terdiri dari beberapa komponen, seperti program Apache HTTP Server, MySQL, dan penerjemah PHP[11].

2.2.5 Chrome

Google Chrome adalah *browser* yang bisa digunakan di berbagai *platform*, dibuat oleh Google. Awalnya diluncurkan pada tahun 2008 untuk Windows, kemudian dikembangkan juga untuk Linux, macOS, iOS, dan Android. Versi yang digunakan dalam penelitian ini adalah Google Chrome 79.0. Selain itu, *browser* ini memiliki berbagai produk yang dapat diunduh, seperti *Extension*, Chrome Apps, dan *Themes* [12].

2.2.6 Database MySQL

MySQL adalah perangkat lunak *database* yang bersifat *open source* yang digunakan untuk mengelola basis data. Ini adalah aplikasi *database management system*(DBMS) yang tersedia secara gratis, dapat diandalkan, terus diperbarui, dan sering digunakan bersama dengan *web server* untuk mempermudah proses instalasi [10].

2.2.7 PhpMyAdmin

PhpMyAdmin adalah aplikasi *web* yang dikembangkan oleh *PhpMyAdmin.net*. Fungsinya adalah untuk mengelola *database* MySQL. Untuk mengakses PhpMyAdmin pada layar utama WAMP, setelah itu dapat memilih opsi PhpMyAdmin [13].

2.2.8 Bootstrap

Bootstrap merupakan suatu kerangka kerja *front-end* yang tersedia secara gratis untuk mempercepat dan menyederhanakan pengembangan *web*. Di dalamnya terdapat *template* desain berbasis

HTML dan CSS untuk berbagai elemen antarmuka seperti tipografi, bentuk, tombol, navigasi, dan elemen lainnya, bersama dengan opsi tambahan untuk ekstensi JavaScript [14].

2.2.9 Website

Website adalah sekumpulan halaman yang terhubung satu sama lain dan menampilkan informasi dalam bentuk teks, gambar, animasi, suara, atau kombinasi dari semua ini. Halaman dapat bersifat statis atau dinamis, dan dapat terhubung ke jaringan halaman lainnya [15].

2.2.10 PHP (*Hypertext Preprocessor*)

PHP juga dikenal sebagai *preprocessor hypertext* adalah bahasa pemrograman yang umum digunakan untuk membuat dan merawat situs *web*. Ini dapat digabungkan dengan HTML. Pada tahun 1994, Rasmus Lerdorf memulai PHP. Dia awalnya menggunakan *personal home page tools*, tetapi kemudian berganti nama menjadi FI (*forms interpret*). Tetapi setelah versi 3.0, nama bahasa diubah menjadi PHP (*hypertext preprocessor*) [15].

2.2.11 NodeMCU ESP8266

NodeMCU adalah sebuah *platform Internet of Things (IoT)* yang bersifat *open-source*. Ini termasuk perangkat keras berupa seri Sistem ESP8266 dari *espressif systems*, dan menggunakan bahasa skrip Lua sebagai *firmware*-nya. Secara default, istilah NodeMCU mengacu pada *firmware* yang digunakan dan bukan pada perangkat keras dari kit pengembangan, dan NodeMCU juga dapat dianggap

sebagai papan Arduino yang digunakan dengan ESP8266. Selain bahasa Lua, NodeMCU juga mendukung penggunaan perangkat lunak Arduino IDE dengan beberapa modifikasi. Arduino IDE menambahkan URL khusus untuk mengunduh *board* NodeMCU di dalam *board manager* [16].


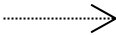

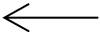
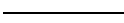


2.2.12 UML (*unified modeling language*)

Hasil konversi berbagai objek ke dalam bentuk representasi pemodelan untuk menggambarkan perangkat lunak sering kali disebut sebagai UML. Dengan menggunakan pemodelan UML, kita bisa memahami alur, proses kerja, dan struktur yang membentuk suatu sistem dengan jelas dan spesifik. Penyajian informasi melalui susunan objek ini membuatnya sederhana dan mudah dimengerti [17]. UML meliputi diagram *use case*, *sequence diagram*, dan kelas diagram.

1. Diagram *Use Case*

Diagram *use case* digunakan untuk mendeskripsikan fungsionalitas sistem, termasuk fungsi atau rangkaian kegiatan yang dapat dilakukan oleh sistem dari perspektif aktor sebagai penggunanya. Diagram *use case* menunjukkan bagaimana sistem dapat melakukan sesuatu melalui aktor sebagai pelaksananya [17].

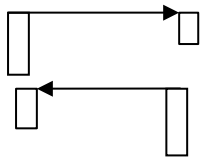





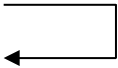
Tabel 2. 1 Simbol-Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		<i>Actor</i>	mengidentifikasi jenis peran yang dimainkan oleh pengguna saat berinteraksi dengan kasus.
2		<i>Dependency</i>	Dalam hubungan ini, elemen yang tidak mandiri bergantung pada elemen yang mandiri.
3		<i>Generalization</i>	Dalam hubungan ini, objek anak (descendant) mewarisi perilaku dan struktur data dari objek induk (ancestor).
4		<i>Extend</i>	menentukan bahwa use case target memperluas perilaku dari use case sumber pada waktu tertentu.
5		<i>Association</i>	Apa yang menyambungkan antara satu objek dengan objek lainnya
6		<i>System</i>	Mendefinisikan paket yang menampilkan sistem dengan batasan tertentu.
7		<i>Usecase</i>	urutan tindakan yang diambil oleh sistem untuk menghasilkan hasil yang dapat diukur bagi seorang aktor

2. *Sequence* Diagram

Sequence diagram mengilustrasikan interaksi antara aktor yang bertindak sebagai pelaku dan objek serta hubungannya saat beroperasi di dalam sistem, terutama saat menyelesaikan tujuan atau fungsi tertentu. Diagram urutan digunakan untuk mewakili proses ini dengan jelas dan terperinci.

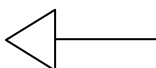
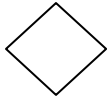
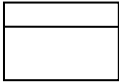

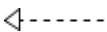
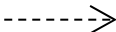
Tabel 2. 2 Simbol-Simbol *Sequence* Diagram

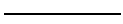
No	Gambar	Nama	Keterangan
1.		<i>Message</i>	Spesifikasi komunikasi antar objek yang berisi data tentang aktivitas yang terjadi.
2.		<i>Actor</i>	menggambarkan individu-individu yang berinteraksi dengan sistem.
3.		<i>Boundary Class</i>	Menggambarkan penggambaran dari form.
4.		<i>Entity Class</i>	menunjukkan hubungan antara kegiatan yang akan dilakukan.
5.		<i>Control Class</i>	menggambarkan hubungan antara batas dan tabel.
6.		<i>Activation</i>	sebagai objek yang dapat melakukan sesuatu.
7.	<i>Message</i>	<i>Message</i>	Mengindikasikan komunikasi antara objek dengan objek
8.		<i>Self Message</i>	Mengindikasikan kembalinya informasi ke dalam sebuah objek

3. Class Diagram

Class diagram ini digunakan untuk menggambarkan keberadaan kelas dan hubungannya dengan kelas lain dalam suatu sistem dengan cara yang logis, melalui koneksi antarkelas yang saling terhubung. Dalam diagram kelas, setiap diagram mewakili pandangan struktur kelas dari sistem tersebut.

Tabel 2. 3 Simbol-Simbol *Class* Diagram

No	Gambar	Nama	Keterangan
1.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>)
2.		<i>Nary Association</i>	Upaya untuk menghindari hubungan dengan lebih dari dua hal.
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	deskripsi dari urutan tindakan yang ditampilkan oleh sistem yang menghasilkan hasil yang dapat diukur untuk suatu aktor
5.		<i>Dependency</i>	Operasi yang benar - benar dilakukan oleh suatu objek.
6.		<i>Dependency</i>	hubungan di mana elemen yang tidak mandiri bergantung pada elemen mandiri jika diubah,

No	Gambar	Nama	Keterangan
7.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.