

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Beberapa penelitian terdahulu menunjukkan bahwa sistem pengendali perangkat elektronik berbasis *website* dan *Internet of Things* (IoT) mampu digunakan untuk melakukan *monitoring* dan pengendalian perangkat dari jarak jauh serta menampilkan kondisi perangkat secara *real-time*. Teknologi ini banyak diterapkan dengan memanfaatkan mikrokontroler ESP32 sebagai media komunikasi antara perangkat keras dan sistem berbasis *website*.

Penelitian yang dilakukan oleh Achmad Asrori Al Sarfini dan Denny Irawan (2024) menunjukkan bahwa penggunaan mikrokontroler ESP32 berbasis IoT memungkinkan proses *monitoring* dan pengendalian perangkat melalui antarmuka *web* dengan respons yang stabil serta dapat diakses menggunakan komputer maupun *smartphone*[8]. Penelitian tersebut membuktikan bahwa ESP32 dapat digunakan sebagai media komunikasi data dalam sistem kendali berbasis jaringan internet.

Selain itu, penelitian oleh Visayas dkk. (2024) menyatakan bahwa sistem kontrol perangkat elektronik berbasis ESP32 dan protokol HTTP mampu melakukan komunikasi data secara baik antara perangkat dan *website* sehingga proses *monitoring* dan pengendalian dapat dilakukan secara langsung melalui jaringan internet[9]. Sistem berbasis *website* juga dinilai lebih fleksibel karena dapat diakses tanpa memerlukan instalasi aplikasi tambahan dan dapat digunakan pada berbagai perangkat.

Namun demikian, penerapan sistem pengendali *Air conditioner* (AC) berbasis *website* sebagai pengganti *remote konvensional* masih belum dikembangkan secara optimal. Penelitian yang dilakukan oleh Maulana dkk. (2024) menunjukkan bahwa sistem kontrol AC berbasis IoT umumnya masih berfokus pada *monitoring* suhu dan penghematan daya tanpa menyediakan fungsi kendali secara penuh seperti remote AC konvensional[10].

Selain itu, integrasi pengiriman sinyal inframerah (*infrared*) untuk menirukan fungsi remote AC masih menjadi tantangan dalam pengembangan sistem IoT berbasis *website*. Oleh karena itu, diperlukan pengembangan sistem yang mampu mengintegrasikan sistem *monitoring* dan pengendalian berbasis *website* dengan modul inframerah secara akurat agar dapat menggantikan fungsi *remote* AC secara optimal.

2.2 Teori Terkait

2.2.1 Sistem Kendali

Sistem kendali merupakan suatu mekanisme yang berfungsi untuk mengatur, mengendalikan, serta mengawasi proses kerja perangkat agar dapat berjalan sesuai dengan kondisi yang diinginkan[17]. Sistem ini umumnya terdiri dari komponen utama yaitu *input*, proses, dan *output*, di mana *input* berupa perintah atau data yang diberikan, kemudian diproses oleh pengendali untuk menghasilkan *output* berupa aksi pada perangkat yang dikendalikan. Dalam perkembangan teknologi saat ini, sistem kendali banyak diterapkan

pada berbagai perangkat elektronik seperti sistem pendingin ruangan, pencahayaan otomatis, dan sistem *monitoring*. Pada penelitian ini, sistem kendali digunakan melakukan *monitoring* dan pengendalian perangkat *Air conditioner* (AC) melalui sistem berbasis *website* sehingga pengguna dapat melakukan pengaturan perangkat secara lebih mudah, fleksibel, dan dapat diakses dari jarak jauh.

2.2.2 *Air conditioner* (AC)

Air conditioner (AC) merupakan perangkat elektronik yang berfungsi untuk mengatur suhu, kelembapan, serta sirkulasi udara di dalam ruangan agar tetap memberikan kenyamanan bagi pengguna. AC bekerja dengan cara menyerap panas dari udara di dalam ruangan, kemudian melepaskannya ke lingkungan luar ruangan melalui proses *refrigerasi*. Sistem kerja AC melibatkan beberapa komponen utama seperti *kompresor*, *kondensor*, *evaporator*, dan *refrigeran*. Perangkat ini banyak digunakan pada rumah, perkantoran, maupun lingkungan pendidikan untuk menjaga kenyamanan suhu ruangan[18]. Pada penelitian ini, AC menjadi perangkat utama yang dimonitor dan dikendalikan melalui sistem berbasis *website* sehingga pengguna dapat melakukan pengaturan perangkat secara lebih praktis.

2.2.3 *Internet of Things* (IoT)

Internet of Things (IoT) merupakan konsep teknologi yang memungkinkan berbagai perangkat elektronik dapat saling terhubung melalui jaringan internet. Dengan teknologi ini, perangkat seperti

sensor, mikrokontroler, dan sistem komputer dapat saling bertukar data secara otomatis. Data yang diperoleh dari perangkat sensor dapat dikirimkan ke *server* atau *website* untuk diproses dan ditampilkan kepada pengguna. Selain itu, pengguna juga dapat memberikan perintah kepada perangkat melalui jaringan internet[19]. Pada penelitian ini, konsep IoT digunakan untuk mendukung sistem *monitoring* pengendalian AC sehingga perangkat dapat dipantau dan dikendalikan dari jarak jauh melalui *website*.

2.2.4 Website

Website merupakan sistem informasi yang diakses melalui *browser* menggunakan jaringan internet tanpa perlu melakukan instalasi aplikasi tambahan. Keunggulan *website* terletak pada kemudahan akses, fleksibilitas, serta kompatibilitas dengan berbagai perangkat seperti komputer dan *smartphone*[11]. Dalam penelitian ini, *website* digunakan sebagai antarmuka pengguna (*user interface*) untuk melakukan *monitoring* dan pengendalian *Air conditioner* (AC), sehingga pengguna dapat mengirimkan perintah pengendalian secara jarak jauh melalui perangkat yang terkoneksi dengan jaringan internet.

2.2.5 PHP

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang digunakan untuk mengolah data pada sisi *server* dalam pengembangan aplikasi berbasis *website*. PHP Native adalah penggunaan bahasa pemrograman PHP tanpa *framework* tambahan,

sehingga proses pengolahan data dilakukan menggunakan fungsi dasar yang tersedia pada PHP[20]. Dalam penelitian ini, PHP digunakan untuk menerima permintaan dari pengguna melalui *website*, memproses data pengendalian AC, serta berkomunikasi dengan *database* dan mikrokontroler ESP32 melalui *server*.



Gambar 2.1 PHP

Contoh kodingan:

```
<?php
$mode = $_GET['mode'];

if ($mode == "cool") {
    echo "Mode AC: Cool";
} elseif ($mode == "dry") {
    echo "Mode AC: Dry";
} else {
    echo "Mode AC: Fan";
}
?>
```

2.2.6 XAMPP

XAMPP adalah sebuah paket perangkat lunak yang berisi Apache, MySQL, PHP, dan Perl yang digunakan sebagai *server* lokal untuk proses pengembangan aplikasi berbasis *website*. XAMPP memungkinkan pengembang untuk menjalankan sistem *website* di

komputer lokal tanpa memerlukan *server online*[21]. Dalam penelitian ini, XAMPP digunakan sebagai media pengujian sistem *monitoring* pengendali *Air conditioner* (AC) berbasis *website* sebelum diimplementasikan ke *server* sebenarnya.



Gambar 2.2 XAMPP

2.2.7 Database

Database adalah kumpulan data yang tersusun secara sistematis yang digunakan untuk menyimpan, mengelola, serta menyediakan informasi agar dapat diakses dengan mudah sesuai kebutuhan pengguna. Dalam sistem berbasis *website*, *database* berperan penting dalam menyimpan data seperti status perangkat, log aktivitas, dan data pengguna[22]. Pada penelitian ini, *database* digunakan untuk menyimpan informasi terkait pengendalian AC agar dapat diakses dan diperbarui secara langsung.

2.2.8 MySQL

MySQL merupakan sistem manajemen basis data *Database Management System* (DBMS) yang memanfaatkan bahasa SQL (*Structured Query Language*) untuk mengelola data. MySQL bersifat *open source* sehingga dapat digunakan secara bebas serta banyak diterapkan dalam pengembangan *website* karena memiliki kinerja yang

stabil dan mudah digunakan[23]. Dalam penelitian ini, MySQL digunakan untuk menyimpan data admin, data kontrol, serta data *monitoring*.



Gambar 2.3 MySQL

2.2.9 CSS

CSS (*Cascading Style Sheets*) adalah bahasa yang digunakan untuk mengatur tampilan dan desain halaman *website*, seperti warna, tata letak, ukuran teks, dan elemen antarmuka lainnya. CSS berfungsi untuk memisahkan struktur halaman (HTML) dengan tampilan visual sehingga pengembangan *website* menjadi lebih rapi dan mudah dikelola[24]. Dalam penelitian ini, CSS digunakan untuk mengatur tampilan antarmuka sistem *monitoring* pengendali *Air conditioner* (AC) berbasis *website* agar lebih menarik, responsif, dan mudah digunakan oleh pengguna.



Gambar 2.4 CSS

2.2.10 Visual Studio Code

Visual Studio Code adalah editor kode sumber yang digunakan untuk menulis dan mengembangkan program. Aplikasi ini mendukung berbagai bahasa pemrograman seperti PHP, HTML, CSS, dan JavaScript serta memiliki fitur seperti debugging dan ekstensi tambahan[25]. Dalam penelitian ini, Visual Studio Code digunakan sebagai perangkat lunak utama dalam pengembangan sistem *monitoring* pengendali AC berbasis *website*.



Gambar 2.5 Visual Studio Code

2.2.11 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang berfungsi untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak. UML berperan dalam membantu pengembang memahami struktur dan alur kerja sistem sebelum tahap implementasi dilakukan.

Dengan penerapan UML, pengembang dapat memahami cara kerja sistem, interaksi antar komponen, serta alur proses yang terjadi di dalam sistem. Selain itu, UML juga berperan dalam mempermudah komunikasi antara pengembang dengan pengguna agar sistem yang

dibangun dapat beroperasi sesuai dengan kebutuhan dan fungsi yang diharapkan.

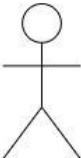
Dalam pengembangan sistem berbasis objek (*Object-Oriented*), UML digunakan sebagai alat bantu dalam menyusun rancangan sistem secara terstruktur dan sistematis. UML biasanya direpresentasikan dalam bentuk berbagai diagram yang memiliki simbol dan notasi tertentu sesuai dengan fungsinya masing-masing[26].

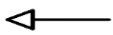
Dalam penelitian ini, UML digunakan untuk menggambarkan perancangan sistem *monitoring* pengendali AC berbasis *website* sebagai pengganti *remote*, seperti *Use Case Diagram*, *Class Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

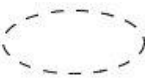
1. *Use Case Diagram*

Use Case Diagram digunakan untuk menggambarkan interaksi antara aktor dengan sistem. Diagram ini menunjukkan fungsi-fungsi yang tersedia dalam sistem serta siapa saja yang dapat mengakses fungsi tersebut.

Tabel 2.1 *Use Case Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Merepresentasikan pengguna maupun pihak lain yang terlibat dan berinteraksi secara langsung dengan sistem.

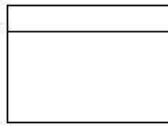
No	Gambar	Nama	Keterangan
2.		<i>Dependency</i>	Menunjukkan adanya hubungan ketergantungan antar elemen, di mana perubahan pada suatu elemen dapat memengaruhi elemen lainnya.
3.		<i>Generalization</i>	Hubungan pewarisan antara aktor atau use case (inheritance).
4.		<i>Include</i>	Menunjukkan bahwa suatu use case selalu menyertakan use case lain.
5.		<i>Extend</i>	Menggambarkan penambahan atau perluasan perilaku pada use case yang terjadi apabila kondisi tertentu terpenuhi.
6.		<i>Association</i>	Hubungan yang menggambarkan interaksi antara aktor dan <i>use case</i> dalam suatu sistem.
7.		<i>System Boundary</i>	Menggambarkan ruang lingkup atau batas sistem yang sedang dirancang.

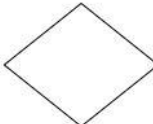
No	Gambar	Nama	Keterangan
8.		<i>Use Case</i>	Menunjukkan berbagai fungsi atau layanan yang tersedia dan dapat digunakan dalam sistem.
9.		<i>Collaboration</i>	Interaksi antar elemen yang bekerja sama dalam sistem.
10.		<i>Note</i>	Catatan tambahan untuk memberikan keterangan pada diagram.

2. *Class Diagram*

Class Diagram adalah salah satu diagram dalam UML yang digunakan untuk menggambarkan struktur sistem secara statis. Diagram ini menunjukkan kelas-kelas yang ada dalam sistem, *atribut* (data) yang dimiliki, *method* (fungsi) yang dapat dilakukan, serta hubungan antar kelas seperti association, inheritance, dan dependency.

Tabel 2.2 *Class Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Class</i>	Sekelompok objek yang memiliki atribut dan metode yang serupa serta digunakan untuk

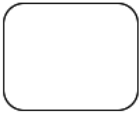
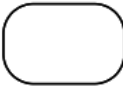
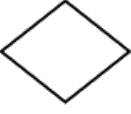
No	Gambar	Nama	Keterangan
			merepresentasikan suatu entitas dalam sistem.
2.		<i>Generalization</i>	Hubungan pewarisan (inheritance), dimana kelas turunan mewarisi atribut dan metode dari kelas induk.
3.		<i>Association</i>	Hubungan antar kelas yang menunjukkan adanya keterkaitan.
8.		<i>Dependency</i>	Hubungan ketergantungan, dimana perubahan pada satu kelas mempengaruhi kelas lain.
9.	1..*	<i>Multiplicity</i>	Menunjukkan jumlah relasi antar kelas (misal: 1..*, 0..1).
10.		<i>N-ary Association</i>	Menggambarkan hubungan yang melibatkan lebih dari dua kelas dalam suatu relasi.

3. *Activity Diagram*

Activity Diagram merupakan diagram yang digunakan untuk menggambarkan alur aktivitas atau proses yang berlangsung dalam suatu sistem, mulai dari tahap awal hingga

akhir, termasuk percabangan serta pengambilan keputusan yang terjadi selama proses tersebut.

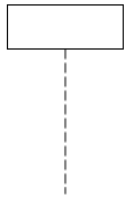
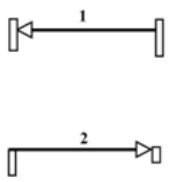
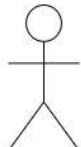
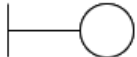
Tabel 2.3 *Activity Diagram*

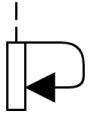
No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Merepresentasikan aktivitas atau proses yang dilakukan dalam sistem selama menjalankan suatu fungsi tertentu.
2.		<i>Action</i>	Aksi yang dilakukan dalam suatu aktivitas.
3.		<i>Initial Node</i>	Menunjukkan titik awal dimulainya suatu aktivitas atau proses dalam sistem.
4.		<i>Final Node</i>	Menunjukkan titik akhir dari suatu aktivitas atau proses dalam sistem.
5.		<i>Fork Node</i>	Memecah satu alur menjadi beberapa alur paralel.
6.		<i>Decision</i>	Percabangan untuk pengambilan keputusan berdasarkan kondisi.
7.		<i>Merge</i>	Menggabungkan kembali beberapa alur menjadi satu alur.
8.		<i>Send Signal</i>	Menunjukkan pengiriman sinyal atau pesan.

4. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan interaksi antar objek berdasarkan urutan waktu.

Tabel 2.4 *Sequence Diagram*

No	Gambar	Nama	Keterangan
1.		<i>LifeLine</i>	Menunjukkan keberadaan objek selama proses interaksi berlangsung.
2.		<i>Message</i>	Mendeskripsikan komunikasi yang terjadi antarobjek beserta informasi mengenai aktivitas atau interaksi yang berlangsung dalam suatu sistem.
3.		<i>Actor</i>	Menggambarkan pengguna atau pihak yang berhubungan secara langsung dengan sistem.
4.		<i>Boundary Class</i>	Antarmuka (<i>interface</i>) antara pengguna dan sistem.
5.		<i>Entity Class</i>	Representasi data atau basis data dalam sistem.
6.		<i>Control Class</i>	Pengatur alur logika yang menghubungkan boundary dan entity.

No	Gambar	Nama	Keterangan
7.		<i>Activation</i>	Merepresentasikan objek yang berperan untuk melakukan suatu aksi atau operasi dalam sistem.
8.		<i>Message</i>	Menunjukkan adanya komunikasi atau interaksi yang terjadi antarobjek dalam sistem.
9.		<i>Self Message</i>	Menunjukkan adanya komunikasi atau pemanggilan operasi yang dilakukan oleh objek terhadap dirinya sendiri.

2.2.12 ESP32

ESP32 merupakan mikrokontroler yang dirancang oleh *Espressif Systems* dan memiliki fitur WiFi serta *Bluetooth* terintegrasi. Berbagai aplikasi *Internet of Things* (IoT) memanfaatkan ESP32 karena memiliki kemampuan untuk melakukan komunikasi jaringan dan pemrosesan data pada berbagai perangkat yang saling terhubung[27]. Mikrokontroler ini dapat digunakan untuk menghubungkan perangkat keras dengan sistem berbasis *website* melalui jaringan internet.

Dalam penelitian ini, ESP32 digunakan sebagai mikrokontroler utama dalam sistem untuk menerima perintah dari *website*, memproses data dari sensor *ultrasonic*, serta mengirimkan sinyal *infrared* ke perangkat *Air conditioner* (AC).

2.2.13 HTTP (*Hypertext Transfer Protocol*)

HTTP (*Hypertext Transfer Protocol*) merupakan protokol yang digunakan untuk pertukaran data antara *client* dan *server* pada jaringan internet[28]. HTTP bekerja menggunakan metode *request* dan *response* untuk mengirim maupun menerima data.

2.2.14 JavaScript

JavaScript adalah bahasa pemrograman yang digunakan untuk menambahkan interaktivitas pada halaman *website*. Melalui *JavaScript*, halaman *website* dapat merespons tindakan pengguna seperti klik tombol, *input* data, maupun pembaruan informasi secara langsung tanpa harus melakukan *reload* halaman.

JavaScript berperan penting dalam pengembangan *website modern* karena mampu menciptakan halaman yang dinamis dan interaktif, seperti animasi, validasi formulir, serta navigasi yang lebih mudah digunakan. Selain itu, *JavaScript* juga digunakan dalam pengembangan sistem berbasis *website* interaktif untuk meningkatkan efektivitas dan pengalaman pengguna.

Dalam sistem ini, *JavaScript* digunakan untuk mengirim perintah dari pengguna ke *server* dan menampilkan data *monitoring* secara dinamis. Selain itu, *JavaScript* juga dapat menghubungkan *website* dengan teknologi *Internet of Things* (IoT) melalui *Application Programming Interface* (API), sehingga memungkinkan interaksi data secara otomatis dan *real-time*[29].