

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada penelitian yang dilakukan oleh Faisal Arief Deswar dan Rizky Pradana (2021) dalam jurnalnya tentang merancang sistem pengaturan dan pemantauan suhu AC menggunakan *wemos D1 R1*. Hasil penelitian yang dilakukan menunjukkan bahwa sistem mampu mengontrol suhu ruang *server*, menyimpan data suhu AC dan ruangan ke dalam *database*, serta menampilkan informasi tersebut baik melalui aplikasi *Android* maupun *LCD* [8].

Pada penelitian yang dilakukan oleh Safrizal dkk. (2023) dalam jurnal *yang membahas* tentang penerapan teknologi berupa mesin pengembang adonan dan oven otomatis berbasis *mikrokontroler Arduino* dengan kontrol suhu dan waktu. Sistem ini menggunakan sensor suhu hingga 300°C, *servo motor* untuk mengatur *regulator gas*, serta *indikator visual* dan *buzzer* sebagai peringatan. Hasil penelitian menunjukkan bahwa alat mampu meningkatkan kualitas produk hingga 90%, mengefisiensikan waktu produksi sekitar 30% [9].

Pada penelitian yang dilakukan oleh Elok Widayanti, Jasmine Mar'ah Qonita, Retno Ikayanti, dan Nurma Sabila (2023) dalam jurnalnya membahas tentang pengaruh perbedaan metode pengeringan *terhadap kandungan flavonoid* total pada daun jinten. Hasil penelitian menunjukkan kadar *flavonoid* terendah pada pengeringan matahari (3,38 mg QE/g ekstrak), sedang pada oven (3,58 mg QE/g ekstrak), dan tertinggi pada pengeringan angin-angin (5,83 mg QE/g ekstrak). Penelitian ini menyimpulkan bahwa metode pengeringan

berpengaruh signifikan terhadap kandungan *flavonoid*, dengan pengeringan angin-angin sebagai metode terbaik untuk mempertahankan kadar *flavonoid* [10].

Pada *penelitian* Usla & Ikhwan (2023) merancang sistem *monitoring* berbasis *web* untuk penyaluran bantuan sosial agar lebih tepat sasaran. Sistem dikembangkan dengan *PHP, Laravel, MySQL* menggunakan metode Waterfall, serta diuji dengan *black box testing*. Fitur utama *meliputi* pendaftaran akun, pengajuan bantuan, *verifikasi*, dan *monitoring* distribusi oleh admin desa. Hasil penelitian menunjukkan sistem mampu membantu aparat desa memantau distribusi bantuan dan memudahkan masyarakat dalam pengajuan [11].

2.2 Landasan Teori

2.2.1 Internet of Things

Internet of Things (IoT) adalah ide di mana benda-benda fisik bisa saling terhubung dan berbagi data melalui internet tanpa perlu campur tangan manusia secara langsung. [12]. *IoT* terdiri dari beberapa komponen utama yaitu perangkat sensor, *jaringan* komunikasi, dan sistem pengolahan data [13]. Dalam sistem *monitoring* suhu, *IoT* memungkinkan perangkat mengirimkan data secara *real time* ke *server* untuk diproses dan ditampilkan pada aplikasi berbasis *web* [14]. Implementasi *Internet of Things (IoT)* menawarkan berbagai manfaat, seperti kemampuan pemantauan secara *realtime*, pengendalian dari jarak jauh, peningkatan efisiensi operasional, serta kemudahan dalam integrasi dengan platform

berbasis *web*. Dalam penelitian ini, konsep *IoT* digunakan sebagai penghubung antara mesin oven pemanas bahan farmasi dengan sistem *monitoring* berbasis *website* sehingga data suhu dapat dipantau dan dikendalikan secara terintegrasi melalui jaringan internet.

2.2.2 Sistem Informasi

Menurut Gelinas, Oram, dan Wiggins (1990, dalam Riasetiawan, 2007), Sistem informasi adalah sistem yang dibuat oleh manusia, yang mencakup bagian manual dan yang menggunakan komputer. Sistem ini digunakan untuk mengumpulkan, menyimpan, dan mengelola data, serta memberikan informasi kepada pengguna. [15]. Informasi di dalam sistem ini adalah hasil pengolahan data atau fakta sehingga menjadi sesuatu yang bermanfaat bagi pengguna sesuai kebutuhan.

2.2.3 Website

Menurut Indra Griha Tofik Isa dan George Pri Hartawan (2017) *website* merupakan kumpulan halaman yang untuk menampilkan informasi, teks, gambar, video, animasi, suara atau gabungan dari semuanya, baik yang bersifat *dinamis* maupun *statis* [16]. Menurut beberapa penelitian, aplikasi berbasis *web* memiliki keunggulan dalam hal kemudahan akses, fleksibilitas, serta kemudahan pemeliharaan karena pembaruan sistem cukup dilakukan pada sisi *server* tanpa perlu memperbarui perangkat pengguna [17].

2.2.4 Monitoring

Monitoring menurut Rizky Aditya, Viktor Handarianus P, dan Putu Bagus Adidyana A. P (2021) adalah suatu aktivitas pemantauan yang dilakukan secara reguler berdasarkan *indikator* tertentu untuk memastikan kegiatan berjalan sesuai dengan rencana dan langkah-langkah yang sudah ditentukan. Pemantauan berfungsi untuk memberikan informasi mengenai keadaan dan perkembangan suatu aktivitas. serta menjadi masukan bagi evaluasi dan perbaikan proses [18].

2.2.5 Laravel

Menurut Dwiza Riana, Ranga Sanjaya dan Oemie Kalsoem (2018) *Laravel* merupakan *framework* berbasis *PHP* yang menerapkan konsep Model-View-Controller (*MVC*) dalam pengembangan aplikasi *web* [19]. *Laravel* juga memiliki fitur-fitur modern yang membantu developer dalam membuat aplikasi, serta keunggulan dalam penulisan kode program yang lebih singkat, mudah dimengerti, dan ekspresif [20].

2.2.6 PHP

Menurut Anhar (2010), *PHP* (Hypertext Preprocessor) adalah bahasa pemrograman yang digunakan di sisi server web dan bersifat open source. *PHP* adalah sebuah skrip yang terhubung dengan *HTML* dan berjalan di server (skrip tersemat *HTML* di sisi server). *PHP* adalah bahasa pemrograman yang digunakan untuk membuat halaman web yang bisa berubah-ubah. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman tersebut diminta oleh klien. Mekanisme ini membuat

informasi yang diterima klien selalu baru dan terkini. Semua script PHP dijalankan di server tempat script itu berada. [21].

2.2.7 MySQL

Menurut Elisa Usada, Yana Yunianrsyah, dan Noor Rifani (2012) *MySQL* merupakan sistem manajemen basis data yang menggunakan Bahasa SQL (*Structured Query Language*) untuk mengelola dan memanipulasi data [22]. *MySQL* banyak digunakan dalam pengembangan aplikasi berbasis *web* karena kemampuannya dalam menangani data dalam jumlah besar, mendukung transaksi, serta memiliki performa yang stabil [23].

2.2.8 RestAPI

Menurut Sari Noorlima Yanti dan Erni Rihyanti (2021) *Application Programming Interface* atau *API* merupakan integrasi dari dua bagian dari sistem aplikasi. Terdiri dari elemen *function*, *protocols*, dan *tools* lainnya digunakan untuk membuat sebuah aplikasi. *Web API* terdapat beberapa hal mendukung fungsi *CRUD* yang bekerja melalui *HTTP protocol* dengan *method GET, POST, PUT* dan *DELETE* [24].

2.2.9 UML (Unified Modeling Language)

Unified Modeling Language (UML) adalah sebuah bahasa standar yang digunakan untuk menggambarkan perangkat lunak [25]. *UML* dapat digunakan untuk berbagai tujuan seperti *visualisasi*, *spesifikasi*, konstruksi, dan dokumentasi untuk sistem perangkat lunak.

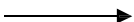
Alat bantu yang *digunakan* dalam perancangan berorientasi objek berbasiskan *UML* sebagai berikut:

1. Use Case Diagram

Use case diagram (use case) merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan system informasi yang akan dibuat [26]. Simbol pada *use case diagram* yaitu:

Tabel 2. 1 Simbol Use Case Diagram

No	Simbol	Pengertian	Keterangan
1.		<i>Use Case</i>	Menggambarkan fungsi yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktir, yang dinyatakan dengan menggunakan kata kerja.
2.		<i>Aktor</i>	<i>Aktor Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktir, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Aktor

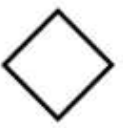
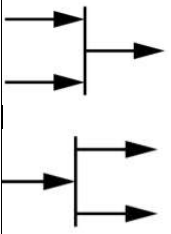
No	Simbol	Pengertian	Keterangan
			berinteraksi.
3.		<i>Association/ Assosiasi</i>	<i>Asosiasi</i> antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
4.		<i>Generalization/ Generalisasi</i>	<i>Asosiasi</i> antara aktor dan <i>Use case</i> yang menggunakan panah Terbuka untuk Mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.

2. Activity Diagram

Activity Diagram merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan [27]. Simbol pada *Activity diagram* yaitu:

Tabel 2. 2 Simbol Activity Diagram

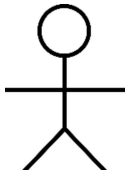
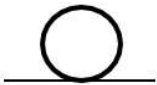
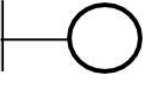
No	Simbol	Pengertian	Keterangan
1.		<i>Activity Fonal</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.

No	Simbol	Pengertian	Keterangan
2.		<i>Node</i>	Pembentukan dan penghancuran objek.
3.		<i>Initial Node</i>	Objek terbentuk atau dihasilkan.
4.		<i>Fork Node</i>	Satu titik tertentu, satu aliran terpecah menjadi beberapa aliran.
5		<i>Decision</i>	Opsi untuk mengambil keputusan.
6		<i>Activity Final</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu

3. Sequence

Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menunjukkan interaksi antara berbagai objek dalam suatu sistem dengan cara yang jelas dan rinci. Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya [28]. Simbol pada *Sequence Diagram* yaitu:

Tabel 2. 3 Simbol Sequence Diagram

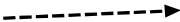
No	Simbol	Pengertian	Keterangan
1.		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem
2.		<i>Entity Class</i>	Menjelaskan hubungan antara batas dan tabel.
3.		<i>Asynchronous Message</i>	Menggambarkan pengiriman pesan
4.		<i>Control Class</i>	Menggambarkan penghubung antara <i>boundary</i> dengan tabel.
5		<i>Boundary Class</i>	Menggambarkan arti sebuah gambaran dari form.
6		<i>A Focus Of Control & A</i>	Waktu yang dibutuhkan suatu objek untuk terus melakukan suatu tindakan atau proses.

4. Class Diagram

Class diagram merupakan diagram yang sering di jumpai pada pemodelan berbasis *UML*. *Class diagram* digunakan untuk menunjukkan interaksi antar *class* di dalam sistem [29].

Tabel 2. 4 Class Diagram

No	Simbol	Deskripsi
1		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai
		

No	Simbol	Deskripsi
		<i>multiplicity</i>
2		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai <i>multiplicity</i>
3		Relasi antar kelas dengan makna generalisasi spesialisasi (umum-khusus)
4		Relasi antar kelas dengan Makna keberuntungan antar kelas
5		Relasi antar kelas dengan makna <i>whole-part</i>