

BAB II

TINJAUAN PUSTAKA

2.1 Teori Terkait

Dalam penyusunan tugas akhir ini, penulis mengambil referensi dari beberapa penelitian sebelumnya yang berkaitan dengan sistem informasi kost berbasis *website*. Penelitian-penelitian tersebut digunakan sebagai acuan dalam merancang dan mengembangkan sistem yang sesuai dengan kebutuhan pengelolaan kost. Berikut beberapa penelitian terdahulu yang relevan dengan tugas akhir ini:

Penelitian yang dilakukan oleh Yuti Dewita Arimbi, Dwi Kartinah, dan Anggit Nila Waska Della (2022) yang berjudul “Sistem Informasi Kost Putri Malika Berbasis *Website*” mengembangkan sistem informasi kost menggunakan *framework* Laravel dan *database* MySQL. Penelitian ini menekankan pentingnya pengelolaan data penghuni serta pencatatan pembayaran yang terstruktur. Fitur yang disediakan meliputi pencatatan data penghuni, informasi kamar, serta proses administrasi kost. Hasil penelitian menunjukkan bahwa sistem yang dibangun mampu mengurangi kesalahan pencatatan karena seluruh proses telah terdigitalisasi dan terkelola dengan baik[5].

Penelitian yang dilakukan oleh Novrini Hasti dan Muhammad Inal Zaelani (2023) pada Rumah Kost Budi Sari mengembangkan sistem berbasis *website* untuk mendukung kegiatan administrasi kost. Fokus penelitian ini

terletak pada pengorganisasian data kamar, pencatatan pembayaran bulanan, serta penyajian laporan bagi pemilik kost. Sistem yang dibangun membantu pemilik dalam menyimpan histori transaksi sehingga memudahkan proses pengecekan pembayaran tanpa harus mencari data secara manual. Hasil penelitian ini menunjukkan bahwa digitalisasi sistem kost menjadi kebutuhan penting, terutama pada kost dengan jumlah penghuni yang cukup banyak[6].

Penelitian yang dilakukan oleh Steven dan Kevin Christianto (2021) melalui aplikasi *AturKost* menghadirkan sistem dengan fitur yang lebih lengkap. Selain pendataan penghuni dan pembayaran, sistem ini juga menyediakan pengelolaan fasilitas kost serta akses bagi penyewa untuk melihat informasi kamar melalui akun masing-masing. Penelitian ini menekankan pentingnya komunikasi antara pemilik dan penghuni yang dapat berjalan lebih efektif melalui sistem berbasis *website*. Selain itu, adanya pembagian hak akses antara pengelola dan penghuni membuat sistem menjadi lebih terstruktur dan aman[7].

Berdasarkan beberapa penelitian tersebut, dapat disimpulkan bahwa sistem informasi kost berbasis *website* memiliki peran penting dalam membantu proses pengelolaan data penghuni, pengaturan data kamar, serta pencatatan pembayaran secara lebih terstruktur dan efisien. Namun, penelitian-penelitian sebelumnya masih berfokus pada aspek administrasi dan belum banyak yang mengintegrasikan teknologi *Internet of Things* (IoT) sebagai pendukung sistem monitoring. Oleh karena itu, penelitian ini dikembangkan untuk melengkapi penelitian sebelumnya dengan

menambahkan integrasi IoT guna mendukung proses monitoring kondisi kamar secara lebih efektif.

2.2 Landasan Teori

2.2.1 Sistem Informasi

Sistem informasi merupakan suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung kegiatan operasional, bersifat manajerial, serta menyediakan informasi yang dibutuhkan dalam proses pengambilan keputusan[8]. Sistem informasi terdiri dari komponen yang saling berhubungan seperti manusia, perangkat keras, perangkat lunak, data, dan prosedur yang bekerja secara terintegrasi untuk menghasilkan informasi yang akurat dan relevan.

Dalam penelitian ini, sistem informasi digunakan untuk membantu pengelolaan data penghuni kost, pencatatan pembayaran, serta monitoring kondisi kamar secara terpusat melalui platform berbasis *website* sehingga informasi dapat diakses secara *real-time* dan lebih efisien.

2.2.2 Website

Website adalah kumpulan halaman yang dapat diakses melalui jaringan internet dan berisi informasi dalam bentuk teks, gambar, maupun elemen interaktif lainnya. *Website* dibangun menggunakan teknologi seperti HTML, CSS, dan JavaScript, serta dapat

dikembangkan menggunakan *framework* agar memiliki struktur yang lebih baik dan aman[9]. *Website* banyak digunakan sebagai media sistem informasi karena memiliki keunggulan dalam hal aksesibilitas, fleksibilitas, serta kemudahan dalam menampilkan data secara *real-time*.

2.2.3 Internet of Things (IoT)

Internet of things (IoT) merupakan konsep yang memungkinkan berbagai perangkat fisik untuk saling terhubung dan berkomunikasi melalui jaringan internet. Teknologi ini memungkinkan pengumpulan dan pertukaran data secara otomatis tanpa interaksi langsung dari pengguna. Penerapan IoT dalam sistem monitoring memberikan kemudahan dalam memperoleh informasi kondisi secara *real-time* sehingga dapat meningkatkan efisiensi dalam proses pengawasan.

2.2.4 Hypertext Preprocessor (PHP)

Hypertext preprocessor (PHP) merupakan salah satu jenis Bahasa pemrograman yang diperlukan untuk membuat sebuah aplikasi *website* dan mengaitkannya dengan server. Untuk membangun sebuah aplikasi *website* PHP memerlukan sebuah *add-on Hypertext Markup Language* (HTML).



Gambar 2. 1 Logo Php

2.2.5 Laravel

Laravel merupakan framework PHP yang digunakan untuk membangun aplikasi berbasis *web* dengan struktur yang terorganisir dan mudah dikembangkan. *Laravel* menyediakan berbagai fitur seperti routing, middleware, ORM (*Eloquent*), serta template engine *Blade* yang mendukung proses pengembangan sistem secara efisien[10].



Gambar 2. 2 Logo Laravel

2.2.6 Database

Basis data merupakan data yang saling terhubung dan berkaitan dengan subjek tertentu pada tujuan tertentu pula. *Database* merupakan suatu koleksi terstruktur dari data yang saling terkait, disimpan dalam media penyimpanan komputer, dan dapat diakses serta dikelola menggunakan perangkat lunak khusus. *Database* digunakan untuk menyimpan, mengelola, dan mengorganisir data dengan tujuan memberikan akses yang efisien, aman, dan terstruktur terhadap informasi[11].

2.2.7 *HyperText Markup Language* (HTML)

HTML merupakan bahasa dasar yang digunakan untuk membangun struktur halaman *website*. HTML menyusun elemen-elemen seperti teks, tabel, form, dan gambar yang ditampilkan pada browser. Penggunaan HTML menjadi dasar dalam pembuatan tampilan halaman sistem berbasis *web*.



Gambar 2. 3 Logo HTML

2.2.8 Visual Studio Code

Visual Studio Code (VS Code) adalah editor kode sumber (*source code editor*) yang dikembangkan oleh Microsoft. VS Code banyak digunakan oleh programmer untuk menulis, mengedit, dan menjalankan berbagai bahasa pemrograman seperti Python, JavaScript, PHP, Java, C++, dan lainnya [12].



Gambar 2. 4 Logo Visual Studio Code

2.2.9 Tailwind CSS

Tailwind CSS merupakan *framework* CSS berbasis *utility-first* yang menyediakan berbagai *class* siap pakai untuk mempercepat proses perancangan tampilan *website*. *Framework* ini memudahkan pengembang dalam membuat desain yang konsisten dan responsif tanpa harus menulis kode CSS secara manual dalam jumlah besar[13].

2.2.10 Laragon

Laragon merupakan perangkat lunak yang digunakan sebagai *server* lokal untuk menjalankan aplikasi berbasis *web*. Laragon menyediakan lingkungan pengembangan yang lengkap, meliputi *web server*, *database*, dan *interpreter* bahasa pemrograman. Penggunaan Laragon memudahkan proses pengembangan, pengujian, dan debugging aplikasi sebelum diimplementasikan ke server[14].



Gambar 2. 5 Logo Laragon

2.2.11 Payment Gateway Midtrans

Midtrans merupakan *payment gateway* yang mempermudah proses pembayaran online serta *menyediakan* mode *sandbox* untuk pengujian transaksi. Fitur ini memungkinkan pengembang mensimulasikan berbagai metode pembayaran guna memastikan sistem berjalan aman sebelum digunakan secara resmi.

2.2.12 *Unified Modeling Language (UML)*

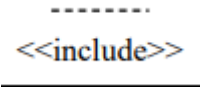
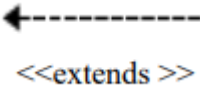
Unified modeling language (UML) adalah sekumpulan alat yang biasanya digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek[15]. UML memvisualisasikan konstruksi sistem dalam beberapa diagram yang sering digunakan yaitu diagram *use case*, *activity diagram*, dan *class diagram*.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use Case Diagram*

Use case diagram adalah deskripsi tentang bagaimana pengguna (aktor) berinteraksi dengan sistem untuk mencapai suatu tujuan tertentu. Dalam pengembangan perangkat lunak, *Use Case* digunakan untuk memahami kebutuhan sistem dan menggambarkan bagaimana suatu fitur yang digunakan.

Tabel 2. 1 Use Case Diagram

No	Simbol	Keterangan
1.		<i>Use case</i> merupakan deskripsi fungsional yang telah disediakan oleh sistem sebagai entitas yang menghasilkan hasil yang terukur untuk suatu actor.
2.		Actor merupakan himpunan peran untuk berinteraksi dengan <i>use case</i>
3.		Association merupakan garis yang menghubungkan objek satu dengan objek yang lain
4.		Include merupakan gambaran jika <i>usecase</i> dipanggil oleh <i>usecase</i> lain.
5.		Dependency merupakan garis panah yang menunjukkan jika actor berinteraksi secara pasif
6.		Extend merupakan gambar jika memperluas <i>usecase</i> target.

2. Activity Diagram

Activity diagram atau diagram aktivitas merupakan diagram yang bersifat statis, yang menggambarkan aktivitas dari suatu sistem bisnis.

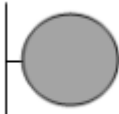
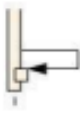
Tabel 2. 2 Activity Diagram

No	Simbol	Keterangan
1.		End point atau final Node merupakan gambaran akhir dari suatu aktivitas
2.		Start Point merupakan awal dari suatu aktivitas yang peletakkannya pada pojok kiri atas
3.		Fork atau join digunakan untuk memarallelkan suatu kegiatan atau penggabungan 2 kegiatan parallel menjadi satu
4.		Activity merupakan gambaran dari suatu proses
5.		Decision merupakan pilihan pengambilan suatu keputusan false or true

3. Sequence Diagram

Sequence diagram atau diagram urutan mendeskripsikan diagram interaksi yang mengirimkan pesan dan diterima antar objek.

Tabel 2. 3 *Sequence Diagram*

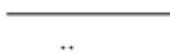
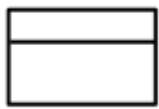
No	Simbol	Keterangan
1.		Entity Class merupakan bagian sistem yang membentuk deskripsi awal sistem dan berisi kumpulan kelas dalam bentuk entitas yang mendasari untuk membuat <i>database</i>
2.		Control Class merupakan gambaran penghubung antara Boundaryclass dengan suatu table
3.		Boundary Class merupakan gambaran dari penggambaran table
4.		Pesan atau message Menunjukkan pengiriman pesen antar class.
5.		Self message menunjukkan pengiriman suatu pesan yang akan dikirim ke objek itu sendiri.

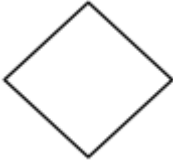
No	Simbol	Keterangan
6.		Lifeline garis titik yang terhubung ke objek disepanjang garis lifeline memiliki aktivitas.
7.		Lifeline garis titik yang terhubung ke objek disepanjang garis lifeline memiliki aktivitas.

4. *Class Diagram*

Class diagram merupakan diagram yang bersifat statis, dalam diagram ini memperlihatkan himpunan kelas, antarmuka, serta relasi. Simbol dari *Class Diagram*.

Tabel 2. 4 Class Diagram

No	Simbol	Keterangan
1.		Generallization, merupakan dimana objek descendent membagikan perilaku dan struktur data objek induknya
2.		<i>Class</i> , adalah kumpulan objek yang saling berbagi.

No	Simbol	Keterangan
3.		Nary Association, digunakan untuk asosiasi terhindar dengan objek lainya.
4.		Collaboration merupakan deskripsi urutan aksi yang ditampilkan suatu sistem yang memiliki konsekuensi terukur bagi actor
5.		Merupakan operasi yang valid dilakukan oleh suatu objek
6.		Merupakan garis penah yang menunjukkan jika actor berinteraksi secara pasif
7.		Association merupakan garis yang menghubungkan objek satu dengan objek