

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem Informasi

2.1.1. Pengertian Sistem

Sistem merupakan sekumpulan elemen yang saling berinteraksi dan bekerja secara terkoordinasi untuk mencapai tujuan tertentu. Elemen-elemen tersebut dapat berupa manusia, prosedur, data, perangkat keras, maupun perangkat lunak, yang secara bersama-sama membentuk suatu kesatuan yang terstruktur. Sebuah sistem memiliki karakteristik seperti input, proses, output, kontrol, dan mekanisme umpan balik yang memastikan seluruh bagian bekerja secara harmonis. Keberadaan hubungan antar komponen menjadi faktor penting agar suatu sistem dapat berjalan secara efektif, teratur, dan dapat menghasilkan keluaran yang sesuai dengan kebutuhan pengguna maupun organisasi.

Sistem dapat menjadi landasan untuk menciptakan proses kerja yang lebih teratur, terutama ketika diperlukan mekanisme yang mampu menghubungkan pengguna, alur pelaporan, serta kegiatan operasional lainnya. Pemahaman tentang sistem memberikan fondasi dalam merancang alur kerja yang terstruktur sehingga proses pelaporan, pengolahan data, serta tindak lanjut terhadap suatu kejadian dapat dijalankan secara otomatis, konsisten, dan dapat dipertanggungjawabkan. Ketika elemen-elemen seperti pengguna, fasilitas, data kerusakan, serta proses penanganan disatukan dalam suatu sistem yang terorganisir, kegiatan pencatatan, pemantauan, dan penyediaan informasi menjadi lebih cepat, akurat, dan mudah diakses oleh pihak yang membutuhkan. Dengan demikian, keberadaan sistem mampu

meningkatkan efisiensi, koordinasi, dan kualitas layanan dalam pengelolaan aktivitas operasional di lingkungan universitas.

2.1.2. Pengertian Informasi

Informasi merupakan hasil dari proses pengolahan data yang telah melalui tahapan seleksi, pengorganisasian, dan interpretasi sehingga memiliki makna bagi penerimanya. Informasi menjadi elemen penting dalam pengambilan keputusan karena memberikan gambaran faktual mengenai suatu keadaan berdasarkan data yang akurat, relevan, dan dapat dipertanggungjawabkan. Informasi yang baik harus memenuhi beberapa kriteria, seperti ketepatan waktu, kejelasan, keakuratan, kelengkapan, serta relevansi dengan kebutuhan penggunanya. Dalam ruang lingkup sistem informasi, informasi berperan sebagai output utama yang membantu penggunanya memahami situasi, mengidentifikasi masalah, maupun mengevaluasi proses kerja yang sedang berjalan. Sistem informasi perlu diperoleh karena bagi manajemen informasi sangat penting guna untuk mengambil keputusan (Rohman & Bhakti, 2023).

Informasi berperan penting sebagai fondasi yang mendukung proses pelaporan, pemantauan, dan pengambilan keputusan dalam sistem yang dirancang untuk menangani kerusakan fasilitas ruangan di sebuah institusi Pendidikan. Kualitas informasi yang dihasilkan dari proses pelaporan menentukan seberapa cepat dan tepat pihak terkait dapat menindaklanjuti kerusakan yang terjadi. Informasi yang akurat mengenai jenis kerusakan, lokasi, waktu kejadian, serta status penanganan mempermudah pihak pengelola fasilitas dalam melakukan verifikasi dan menentukan prioritas perbaikan. Keberadaan informasi yang tersusun

secara sistematis dan mudah diakses juga membantu meningkatkan efisiensi operasional serta memastikan transparansi dalam proses penanganan kerusakan, sehingga seluruh pihak yang terlibat dapat memahami kondisi fasilitas secara real time dan melakukan tindakan yang diperlukan tanpa keterlambatan.

2.1.3. Pengertian Sistem Informasi

Sistem informasi merupakan sebuah proses mengumpulkan, memproses, menganalisis serta menyebarkan informasi untuk tujuan tertentu yang melibatkan orang, perangkat keras, perangkat lunak maupun basis data (Rahmi et al., 2023). Sistem informasi tidak hanya berfungsi sebagai alat bantu pengolahan data, tetapi juga sebagai mekanisme yang memastikan bahwa informasi yang dihasilkan relevan, akurat, tepat waktu, serta dapat digunakan untuk meningkatkan efektivitas dan efisiensi operasional. Melalui integrasi berbagai komponen tersebut, sistem informasi mampu menciptakan alur kerja yang lebih teratur, meminimalkan kesalahan manusia, mempercepat penyampaian informasi, dan menambah nilai bagi organisasi melalui otomatisasi proses serta penyediaan informasi strategis.

Keberadaan sistem informasi menjadi sangat penting ketika organisasi membutuhkan proses yang cepat, terarah, dan terdokumentasi secara baik seperti pada pengelolaan fasilitas ruangan. Sistem informasi berperan untuk menghubungkan pelapor, petugas, dan pihak pengelola melalui aliran data yang sistematis sehingga setiap laporan kerusakan dapat diterima, dicatat, diverifikasi, dan ditindaklanjuti secara lebih terstruktur. Pemanfaatan sistem informasi memungkinkan seluruh proses berjalan lebih transparan karena setiap aktivitas terekam dalam basis data, mulai dari waktu pelaporan, jenis kerusakan, respon yang

diberikan, hingga penyelesaian perbaikan. Dengan dukungan teknologi berbasis *web*, informasi mengenai kerusakan dapat diakses kapan saja, mempermudah pemantauan, meminimalkan keterlambatan penanganan, serta meningkatkan kualitas layanan fasilitas yang berdampak langsung pada kenyamanan dan kelancaran aktivitas akademik.

2.2. Alat Bantu dalam Pengembangan Sistem

Alat bantu dalam pengembangan sistem merupakan komponen yang berperan penting untuk mempermudah, mempercepat, dan meningkatkan ketepatan proses pembangunan perangkat lunak. Alat bantu ini mencakup perangkat untuk pemodelan, perancangan, pengkodean, hingga pengujian, sehingga pengembang dapat bekerja lebih terstruktur dan terarah. Contohnya adalah *UML (Unified Modeling Language)* yang digunakan untuk memvisualisasikan kebutuhan dan alur proses, ERD untuk memodelkan struktur basis data, text editor atau IDE untuk mempermudah penulisan kode, serta tools debugging dan testing untuk memastikan fungsi berjalan sebagaimana mestinya. Kehadiran alat bantu tersebut tidak hanya mendukung dokumentasi teknis, tetapi juga meminimalkan kesalahan selama proses pengembangan, meningkatkan efisiensi kerja, dan menghasilkan sistem yang lebih mudah dipelihara.

Penggunaan alat bantu sangat mendukung proses pembuatan sistem pelaporan dan penanganan kerusakan fasilitas ruangan karena kebutuhan terhadap pemodelan proses, desain basis data, dan implementasi kode harus disusun secara jelas dan sistematis. UML membantu menggambarkan alur pelaporan kerusakan mulai dari pengguna hingga pihak teknisi, sementara ERD memudahkan dalam

merancang tabel-tabel seperti laporan kerusakan, status perbaikan, dan data ruangan agar hubungan antar data tersusun dengan benar. IDE seperti *Visual Studio Code* mempercepat penulisan kode PHP, menyediakan fitur auto-complete, integrasi debugging, serta kemampuan mengelola struktur folder proyek. Tools pengujian juga mempermudah pengecekan fungsi pelaporan, validasi data, dan perubahan status perbaikan sehingga sistem yang dibangun lebih stabil dan akurat. Kombinasi alat bantu tersebut membuat proses pengembangan berjalan lebih cepat, rapi, dan sesuai kebutuhan pengguna.

2.2.1. *Unified Modeling Language (UML)*

Unified Modeling Language atau yang biasa dikenal dengan singkatan UML, adalah bahasa standar yang digunakan untuk menggambarkan, menjelaskan, dan membangun perangkat lunak (Elis & Voutama, 2023). UML menyediakan berbagai jenis *Diagram* seperti use case, activity, sequence, class, dan lainnya yang membantu pengembang memahami kebutuhan pengguna, alur proses, interaksi antar komponen, hingga struktur data dalam sistem. Dengan UML, sistem yang kompleks dapat dipecah menjadi bagian-bagian yang lebih mudah dianalisis, sehingga mempermudah komunikasi antara analis, developer, dan stakeholder. Keberadaan notasi yang baku membuat UML menjadi bahasa pemodelan yang fleksibel dan dapat diaplikasikan di berbagai metodologi pengembangan, baik waterfall, agile, maupun iterative.

Penggunaan UML sangat membantu menggambarkan kebutuhan dan alur kerja pada sistem informasi pelaporan dan penanganan kerusakan fasilitas ruangan. Penggunaan UML tidak terbatas untuk metodologi tertentu, walaupun pada





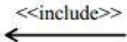
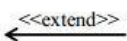
kenyataannya UML ini paling banyak digunakan pada metodologi berorientasi objek (Suharni et al., 2023). *Diagram use case* dapat menampilkan hubungan antara mahasiswa, dosen, teknisi, dan admin dengan fungsionalitas sistem, sementara *Activity Diagram* mampu menunjukkan urutan aktivitas mulai dari proses pelaporan kerusakan, verifikasi, hingga penyelesaian. *Sequence Diagram* memperlihatkan interaksi pesan antara pengguna dan modul sistem ketika laporan diproses, sedangkan *Class Diagram* memberikan gambaran struktur data seperti entitas ruangan, laporan kerusakan, pengguna, dan status penanganan. Pemanfaatan *Diagram-Diagram* tersebut mempermudah perancangan sistem agar lebih terstruktur, mudah diimplementasikan, serta memastikan seluruh kebutuhan pengguna terwakili secara jelas sebelum masuk ke tahap pengembangan menggunakan PHP dan basis data *MySQL*.

1. Use Case Diagram

Use Case Diagram merupakan salah satu alat pemodelan yang digunakan dalam *Unified Modeling Language (UML)* untuk menggambarkan hubungan antara aktor dan fungsi-fungsi yang tersedia dalam sebuah system. Dengan UML, proses perancangan sistem menjadi lebih mudah dipahami karena disajikan secara visual dan merupakan bahasa standar dalam pembuatan blueprint perangkat lunak (Pangestu et al., 2024). *Diagram* ini berfungsi untuk memperlihatkan apa saja yang dapat dilakukan oleh pengguna atau entitas lain terhadap sistem tanpa menjelaskan detail teknis bagaimana fungsi tersebut dijalankan. *Use Case Diagram* membantu pengembang memahami kebutuhan sistem dari sudut pandang pengguna sehingga seluruh proses bisnis dapat digambarkan secara jelas, terstruktur, dan mudah

dipahami. Selain itu, *Diagram* ini menjadi dasar penting untuk merancang alur kerja sistem, mendefinisikan kebutuhan fungsional, serta memastikan bahwa fitur yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna.

Tabel 2. 1. Simbol-Simbol Use Case Diagram

Simbol	Keterangan
	Aktor : 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 dengan use case
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case
	Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi







Sumber: <https://badoystudio.com/uml/simbol-use-case-Diagram/>

Peran *Use Case Diagram* sangat signifikan karena sistem informasi pelaporan dan penanganan kerusakan fasilitas ruangan melibatkan banyak aktor dengan fungsi yang berbeda-beda, seperti mahasiswa atau dosen sebagai pelapor, teknisi sebagai penanganan, serta admin atau petugas fasilitas sebagai pengelola laporan. Usecase diagram mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat (Sintaro, 2022).

2. Activity Diagram

Activity Diagram merupakan salah satu *Diagram* dalam UML yang menggambarkan alur aktivitas atau proses yang terjadi dalam sebuah sistem secara runtut dari awal hingga akhir. Dengan UML maka pada tahap perancangan sistem lebih mudah dipahami karena dalam bentuk visualisasi dan menjadi bahasa standart dalam penulisan blue print software. (Ramdany, 2024). *Diagram* ini menampilkan langkah-langkah kerja, keputusan, aliran data, hingga aktor yang terlibat dalam menjalankan suatu proses. Dengan simbol seperti *initial node*, *activity*, *decision*, *merge*, *fork*, *join*, dan *final node*, *Activity Diagram* memudahkan perancang sistem dalam memahami bagaimana sebuah proses berjalan, bagaimana keputusan dibuat, serta bagaimana aliran aktivitas berpindah dari satu keadaan ke keadaan lainnya.

Tabel 2. 2. Simbol 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 / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	Penggabungan / Join	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
	Swimlane	Swimlane memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi





Sumber: <https://www.dicoding.com/blog/apa-itu-activity-Diagram/>

Activity Diagram menggambarkan rangkaian aktivitas yang terjadi mulai dari pelapor memasukkan data kerusakan ruangan, alur verifikasi oleh admin, hingga proses penanganan oleh petugas teknisi. Setiap langkah seperti mengisi formulir laporan, menyimpan data ke sistem, memvalidasi informasi, mengubah status laporan, serta memberikan notifikasi tercermin dalam alur *Diagram* yang saling terhubung. Penggambaran aktivitas ini memberikan visualisasi yang jelas mengenai bagaimana sistem mendukung proses pelaporan, siapa saja yang berperan dalam alur tersebut, dan bagaimana setiap keputusan memengaruhi aktivitas selanjutnya. Dengan demikian, *Diagram* tersebut membantu memastikan bahwa seluruh proses kerja tercakup secara utuh dan konsisten dengan kebutuhan pengguna serta alur penanganan kerusakan fasilitas ruangan di lingkungan universitas.

3. *Class Diagram*

Class diagram adalah merupakan hubungan antar kelas dan penjelasan detail tiap tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan- aturan dan tanggung jawab entitas yang menentukan perilaku system (Ramdany, 2024). *Diagram* ini membantu memperjelas bagaimana suatu sistem dibangun dari komponen-komponen logis yang saling terhubung sehingga pengembang dapat memahami alur data, keterkaitan fungsi, serta batasan yang berlaku di dalam sistem. Dengan menampilkan struktur statis, *Class Diagram* memberikan gambaran menyeluruh mengenai entitas inti yang berperan dalam sistem sehingga mempermudah proses desain, pengembangan, dan pemeliharaan perangkat lunak.

Tabel 2. 3. Simbol-Simbol Class Diagram

asosiasi / <i>association</i> 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
asosiasi berarah / <i>directed association</i> 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
generalisasi 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
kebergantungan / <i>dependency</i> 	relasi antar kelas dengan makna kebergantungan antar kelas
agregasi / <i>aggregation</i> 	relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

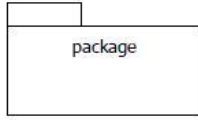
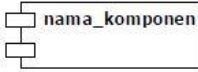

Sumber: <https://pccontrol.wordpress.com/2013/01/21/pengetahuan-dasar-dan-contoh-Diagram-kelas-class-Diagram/>

Struktur *Class Diagram* sangat relevan untuk menggambarkan hubungan antar entitas seperti pengguna, laporan kerusakan, fasilitas, dan proses penanganan yang saling berinteraksi dalam alur kerja system. Class diagram juga dapat memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain (logical view) dari suatu sistem (Syahputra, 2025). Setiap kelas memiliki tanggung jawab masing-masing, misalnya kelas pengguna yang menyimpan data identitas dan hak akses, kelas laporan yang menampung informasi kerusakan, serta kelas penanganan yang mengelola proses verifikasi dan status perbaikan. Hubungan antar kelas tersebut membentuk alur yang logis sehingga memudahkan pengembangan sistem pelaporan kerusakan ruangan berbasis PHP dan *MySQL*, sekaligus memastikan setiap fungsi saling terintegrasi dengan baik mulai dari proses input data, pengolahan, hingga penyajian informasi kepada pihak pengelola fasilitas.

4. *Component Diagram*

Component Diagram merupakan salah satu jenis *Diagram* dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan struktur fisik dari sebuah sistem perangkat lunak. UML adalah set standar teknik yang menyediakan representasi grafis cukup kaya untuk model proyek pengembangan sistem dari analisis melalui implementasi diagram (Asra et al., 2023). *Diagram* ini menampilkan bagaimana komponen-komponen perangkat lunak, seperti modul, *library*, *package*, maupun file eksekusi, saling berinteraksi dan berhubungan melalui *interface* yang tersedia. Fokus utama *Component Diagram* adalah memperlihatkan arsitektur perangkat lunak dari sisi *high-level*, menunjukkan bagaimana setiap bagian sistem dibangun, apa saja dependensinya, serta bagaimana komponen tersebut bekerja bersama untuk membentuk satu kesatuan sistem yang utuh.

Tabel 2. 4. Simbol-Simbol *Component Diagram*

Simbol	Deskripsi
Package 	package merupakan sebuah bungkus dari satu atau lebih komponen
Komponen 	Komponen sistem
Kebergantungan / <i>dependency</i> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
Antarmuka / <i>interface</i>	sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen

Sumber: <https://www.nesabamedia.com/component-Diagram/>



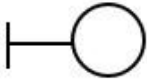



UML melalui penggunaan *Component Diagram* dapat menggambarkan pembagian sistem ke dalam komponen seperti modul pelaporan, modul verifikasi, modul pemrosesan data, modul admin, serta komponen basis data yang menyimpan data kerusakan dan status penanganannya sehingga struktur dan hubungan antar bagian sistem dapat terlihat lebih jelas. UML hanya berfungsi untuk melakukan pemodelan, sehingga penggunaannya tidak terbatas pada metodologi tertentu, meskipun pada praktiknya UML paling banyak diterapkan pada metodologi berorientasi objek. (Bayu et al., 2021). Setiap modul digambarkan dengan hubungan dependensi yang menunjukkan alur interaksi, misalnya komponen pelaporan yang berhubungan dengan komponen verifikasi sebelum data diteruskan ke komponen penanganan kerusakan. Representasi ini membantu menunjukkan bagaimana aplikasi berbasis PHP yang dihasilkan tersusun dari elemen-elemen perangkat lunak yang saling terhubung dan bagaimana *MySQL* berperan sebagai penyedia layanan penyimpanan data. Selain itu, *Component Diagram* mempertegas bagaimana antarmuka antar komponen dibangun untuk memastikan proses pelaporan, pengelolaan, dan pembaruan status dapat berjalan secara terintegrasi dan konsisten sehingga seluruh kebutuhan sistem tercapai secara efisien.

5. *Sequence Diagram*

Sequence Diagram menggambarkan alur interaksi antar objek atau aktor dalam sebuah sistem secara berurutan berdasarkan waktu. *Diagram* ini menunjukkan pesan yang dikirimkan dari satu objek ke objek lainnya, termasuk kapan proses dimulai, bagaimana langkah-langkah berlangsung, dan kapan suatu aksi berakhir, hal ini dilakukan karna adanya analisis kebutuhan system yang

memang harus ada suatu system. Analisis Kebutuhan Sistem merupakan proses analisa yang diperlukan untuk sistem, proses ini meliputi tiga kebutuhan sistem yaitu kebutuhan data, kebutuhan antarmuka dan kebutuhan fungsional (Mahdalena et al., 2023). Elemen utama di dalamnya terdiri dari *lifeline*, *activation bar*, dan *messages* yang dapat berupa *synchronous*, *asynchronous*, maupun *return message*. *Sequence Diagram* membantu memahami dinamika proses secara runtut sehingga mempermudah pengembang dalam memvisualisasikan bagaimana sebuah fitur bekerja dari awal hingga akhir, serta memastikan tidak ada langkah fungsional yang terlewat saat merancang sistem.

Tabel 2. 5. Simbol Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sisitem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

Sumber: <https://ilmuteknologi007.blogspot.com/2017/02/unified-modelling-language-uml.html>

Penerapan *Sequence Diagram* memberikan gambaran jelas mengenai jalannya proses pelaporan, verifikasi, dan penanganan kerusakan fasilitas ruangan yang melibatkan aktor seperti mahasiswa, petugas fasilitas, dan *administrator*. Setiap interaksi, mulai dari pengiriman laporan kerusakan, proses pengecekan data ke dalam sistem, hingga perubahan status perbaikan, dapat divisualisasikan secara runtut sehingga memudahkan analisis kebutuhan sistem maupun alur kerja yang harus dijalankan oleh masing-masing aktor. *Diagram* ini juga membantu memastikan bahwa setiap fungsi yang dibangun saling terhubung dengan logis, mencerminkan aliran kerja nyata, serta mendukung terciptanya sistem informasi yang lebih terstruktur, efisien, dan mudah dipahami oleh pengembang maupun pihak pengelola fasilitas.

2.3. Database

Database atau basis data adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut (Ahmad et al., 2021). Database memungkinkan penyimpanan informasi dalam jumlah besar dengan struktur yang teratur sehingga proses pencarian, pemrosesan, dan pengolahan data dapat dilakukan dengan cepat dan akurat. MySQL, salah satu DBMS yang banyak digunakan, menyediakan kemampuan untuk menangani data secara terstruktur melalui tabel, relasi, serta operasi CRUD (Create, Read, Update, Delete) yang mendukung kebutuhan sistem berskala kecil hingga besar. Keandalan Database sangat bergantung pada integritas data, keamanan, serta kemampuan menangani banyak permintaan secara bersamaan.

Penggunaan *Database MySQL* memberikan fondasi bagi sistem pelaporan dan penanganan kerusakan fasilitas ruangan agar mampu mengelola setiap informasi secara terstruktur dan dapat diproses tanpa menimbulkan duplikasi atau kehilangan data. Banyak cara yang dapat dimanfaatkan untuk meningkatkan kinerja sistem, salah satunya dengan menyimpan data menggunakan XAMPP yang menyediakan lingkungan server lokal lengkap untuk menjalankan aplikasi berbasis web. Pada aplikasi ini database akan disimpan dan diolah sehingga proses pengelolaan data dapat berjalan lebih cepat dan terstruktur. (Sari et al., 2022). Setiap laporan kerusakan—seperti data ruangan, jenis kerusakan, status penanganan, dan informasi pengguna—dapat disimpan dalam tabel yang saling terhubung sehingga memudahkan proses pelacakan mulai dari pelaporan hingga tahap perbaikan. Struktur relasional *MySQL* membantu sistem mengidentifikasi prioritas kerusakan, menampilkan informasi terbaru kepada admin maupun petugas, serta menghasilkan laporan yang akurat. Dengan dukungan query yang efisien, *Database* memungkinkan sistem bekerja lebih cepat, responsif, dan andal ketika mengelola banyak laporan sekaligus. Ini menjadikan *MySQL* sebagai komponen penting dalam memastikan sistem mampu memberikan informasi yang tepat, real-time, dan mudah diakses oleh pihak yang bertanggung jawab terhadap pemeliharaan fasilitas.

2.3.1. *MySQL*

MySQL merupakan sistem manajemen basis data relasional (RDBMS) yang banyak digunakan karena bersifat open-source, memiliki performa tinggi, serta mampu menangani volume data besar dengan struktur yang terorganisir. *MySQL* dibagi menjadi dua lisensi, pertama adalah Free Software dimana perangkat lunak

dapat diakses oleh siapa saja (Arimbi et al., 2022). *MySQL* mendukung bahasa SQL sebagai standar pengelolaan data sehingga proses pembuatan tabel, pengolahan data, hingga eksekusi query dapat dilakukan dengan cepat dan efisien. Keunggulan lain *MySQL* terletak pada stabilitas, fleksibilitas, serta kompatibilitasnya dengan berbagai platform dan bahasa pemrograman. Arsitektur *server-client* yang dimilikinya memungkinkan proses pengolahan data terpusat sehingga mudah diakses oleh berbagai aplikasi berbasis *web*. Dengan fitur seperti indexing, relasi antar tabel, serta pengamanan akses, *MySQL* menjadi salah satu DBMS yang paling banyak digunakan di dunia untuk pengembangan aplikasi skala kecil hingga besar.



Gambar 2. 1. MySQL

Sumber: <https://id.wikipedia.org/wiki/MySQL>

MySQL sangat relevan digunakan untuk membangun sistem pelaporan kerusakan fasilitas karena mampu menampung data laporan, data ruangan, data pengguna, dan data proses penanganan kerusakan secara terstruktur dan terhubung satu sama lain. Sistem ini dapat digunakan dengan menggunakan Bahasa pemrograman yang sederhana. Dengan bahasa pemrograman yang sederhana, rancangan sistem ini dapat mempermudah dan mengurangi kesalahan pada saat

pemrosesan semua data penggajian, sehingga dapat menghasilkan laporan penggajian yang lebih tepat dan efektif (Sugiarti et al., 2024). Setiap proses yang terjadi di dalam sistem, seperti pembuatan laporan baru, perubahan status penanganan, atau pemrosesan verifikasi, dapat direkam dan dikelola secara konsisten melalui *query SQL*. Kemampuannya dalam menangani transaksi data yang terus diperbarui membuat alur pelaporan menjadi lebih rapi, terpantau, dan akurat. Selain itu, *MySQL* bekerja sangat baik ketika diintegrasikan dengan aplikasi berbasis PHP, sehingga alur komunikasi antara antarmuka pengguna dan *Database* menjadi lebih stabil dan responsif.

2.3.2. *Structure Query Language (SQL)*

SQL merupakan bahasa pemrograman yang digunakan untuk mengakses basis data relasional dan memudahkan praktisi data dalam mengolah data (Kalsum Siregar et al., 2024). *SQL* menyediakan seperangkat perintah untuk membuat, mengubah, menghapus, serta mengambil data dari tabel secara terstruktur dan efisien. Bahasa ini memiliki beberapa kategori perintah, seperti *Data Definition Language (DDL)* untuk pendefinisian struktur *Database*, *Data Manipulation Language (DML)* untuk operasi tambah, ubah, dan hapus data, *Data Control Language (DCL)* untuk pengaturan hak akses, serta *Data Query Language (DQL)* yang berfungsi melakukan pencarian dan pengambilan informasi. Kemampuan *SQL* dalam menangani data dalam jumlah besar, menjaga konsistensi, serta mendukung relasi antar tabel membuatnya menjadi fondasi utama dalam pembangunan berbagai sistem informasi modern. Dengan sintaks yang terstandarisasi dan mudah dipahami, *SQL* memungkinkan pengembang

membangun *Database* yang terstruktur, fleksibel, dan dapat diakses sesuai kebutuhan aplikasi.



Gambar 2. 2. SQL

Sumber: <https://www.fanruan.com/id/blog/sql>

Penggunaan SQL memberikan peran krusial dalam pengembangan sistem pelaporan dan penanganan kerusakan fasilitas ruangan, karena seluruh informasi yang berkaitan dengan laporan kerusakan, data pengguna, status penanganan, hingga histori perbaikan perlu disimpan dalam basis data yang terorganisir. Dalam membuat aplikasi yang sudah menggunakan database maka akan terdapat perintah SQL (*Structur query Language*) untuk melakukan pengolahan data pada aplikasi tersebut, misalnya dalam proses penyimpanan data, mengedit data dan menghapus data (Lase, 2023). SQL memungkinkan pembentukan tabel yang saling terhubung sehingga setiap laporan dapat direlasikan dengan data ruangan, pelapor, dan petugas yang menangani kerusakan tersebut. Operasi *SELECT* digunakan untuk menampilkan daftar laporan secara *real time*, *INSERT* untuk menyimpan laporan baru, *UPDATE* untuk mengubah status penanganan, dan *DELETE* apabila terdapat data yang harus dihapus. Struktur *Database* yang dirancang menggunakan SQL

membantu menghasilkan alur data yang lebih rapi, mempermudah proses pencarian informasi, dan meningkatkan akurasi pelaporan. Selain itu, kemampuan SQL dalam memastikan integritas data membuat sistem dapat berfungsi lebih stabil, konsisten, dan mampu mendukung kebutuhan pengelolaan fasilitas secara lebih efektif.

2.3.3. Konsep Dasar *Database MySQL*

MySQL merupakan salah satu sistem manajemen basis data relasional (RDBMS) yang paling banyak digunakan karena bersifat *open-source*, stabil, dan mampu menangani data dalam jumlah besar dengan performa yang tinggi. Penggunaan *MySQL* ini juga dapat dikombinasikan dengan HTML dan PHP yang merupakan sebuah Bahasa pemrograman yang sederhana dan mudah untuk dipahami. Penggunaan kombinasi HTML, PHP, dan *MySQL* dalam pengembangan sistem informasi telah banyak terbukti efektif dan menguntungkan, baik dalam bidang bisnis maupun pendidikan (Arif, 2023). *MySQL* menggunakan bahasa *SQL (Structured Query Language)* sebagai dasar untuk melakukan pengelolaan data seperti penyimpanan, pengubahan, penghapusan, dan pengambilan informasi. Struktur data dalam *MySQL* tersusun dalam bentuk tabel yang saling terhubung melalui relasi, sehingga proses pengelolaan data menjadi lebih terorganisir, konsisten, dan efisien. Keunggulan lain dari *MySQL* meliputi skalabilitas, kompatibilitas dengan berbagai bahasa pemrograman seperti PHP, keamanan data yang baik, serta kemampuan menangani transaksi yang kompleks melalui mekanisme engine seperti *InnoDB*. Kombinasi fitur-fitur ini menjadikan *MySQL* sangat ideal digunakan dalam pengembangan aplikasi *web* berskala kecil hingga besar.

MySQL berperan penting sebagai media penyimpanan utama yang mengelola seluruh data pelaporan dan penanganan kerusakan fasilitas ruangan. Pada penerapannya juga *MySQL* dapat digunakan secara gratis dan penggunaannya juga mudah. Dalam bahasan penelitian yang lainnya Salah satu perangkat lunak web server yang dapat diunduh secara gratis adalah interpreter PHP dan database *MySQL* yang dapat digunakan untuk membuat soal-soal tes atau penilaian interaktif, yang dapat langsung diberikan, memberikan umpan balik kepada para peserta (Raharjo et al., 2022). Data seperti jenis kerusakan, lokasi ruangan, identitas pelapor, status penanganan, serta riwayat perbaikan dapat disimpan secara sistematis dalam tabel-tabel yang saling terhubung sehingga memudahkan proses pencarian dan analisis. Relasi antar tabel memungkinkan proses verifikasi laporan menjadi lebih cepat karena setiap informasi dapat ditelusuri secara real time dan akurat.

2.3.4. Tipe-Tipe Data pada *MySQL*

MySQL menyediakan beragam tipe data yang dirancang untuk menampung berbagai bentuk informasi sehingga penyimpanan data menjadi lebih terstruktur, efisien, dan sesuai kebutuhan aplikasi. Setiap tipe data memiliki karakteristik khusus yang memungkinkan sistem membedakan antara angka, teks, tanggal, waktu, maupun data biner, sehingga proses pengolahan dan validasi data dapat dilakukan dengan lebih akurat. Pengelompokan tipe data ini juga membantu pengembang dalam menentukan format penyimpanan yang tepat untuk setiap atribut dalam tabel, mengoptimalkan performa query, serta menjaga konsistensi dan

integritas data di dalam basis data. Terdapat beberapa tipe-tipe data pada *MySQL* yaitu sebagai berikut.

1. Tipe Data Numerik

Tipe data numerik pada *MySQL* digunakan untuk menyimpan nilai berupa angka, baik dalam bentuk bilangan bulat maupun bilangan pecahan. Tipe data ini memungkinkan penyimpanan nilai dengan rentang dan tingkat presisi yang berbeda sesuai kebutuhan, mulai dari angka kecil, angka berukuran sedang, hingga angka yang sangat besar. Selain itu, tipe numerik juga mendukung penyimpanan angka desimal yang membutuhkan ketelitian tinggi, seperti nilai perhitungan atau keuangan. Dengan adanya variasi tipe numerik, pengembang dapat memilih format penyimpanan yang paling efisien dan akurat untuk setiap data yang berkaitan dengan perhitungan numerik. Untuk tipe tipe pada data ini yaitu sebagai berikut.

- 1) *TINYINT* digunakan untuk menyimpan bilangan bulat berukuran sangat kecil.
- 2) *SMALLINT* digunakan untuk menyimpan bilangan bulat yang lebih besar dari *TINYINT* namun masih dalam kisaran kecil.
- 3) *MEDIUMINT* digunakan untuk menyimpan bilangan bulat berukuran sedang yang membutuhkan ruang penyimpanan lebih besar.
- 4) *INT* / *INTEGER* digunakan untuk menyimpan bilangan bulat standar yang paling umum dipakai dalam tabel.
- 5) *BIGINT* digunakan untuk menyimpan bilangan bulat sangat besar yang melebihi kapasitas tipe *INT*.
- 6) *FLOAT* digunakan untuk menyimpan angka pecahan dengan presisi yang tidak terlalu tinggi.

- 7) *DOUBLE* digunakan untuk menyimpan angka pecahan dengan presisi lebih tinggi dibandingkan *FLOAT*.
- 8) *DECIMAL (M, D)* digunakan untuk menyimpan angka pecahan dengan presisi tetap, biasanya untuk nilai yang membutuhkan akurasi seperti data keuangan.

2. Tipe Data String / Karakter

Tipe data string atau karakter pada *MySQL* digunakan untuk menyimpan data dalam bentuk teks, mulai dari teks pendek hingga teks dengan panjang yang sangat besar. Tipe data ini mencakup format penyimpanan dengan panjang tetap maupun panjang dinamis, sehingga memungkinkan pengembang memilih jenis yang paling efisien sesuai kebutuhan data. Penggunaan tipe string sangat penting untuk menyimpan informasi seperti nama, alamat, deskripsi, pesan, atau data lain yang berbasis huruf, angka, dan simbol. Fleksibilitas tipe data ini membantu memastikan bahwa penyimpanan teks dapat dilakukan secara optimal tanpa menghabiskan ruang memori yang berlebihan, sekaligus tetap menjaga kecepatan proses pengolahan data. Untuk tipe tipe pada data ini yaitu sebagai berikut.

- 1) *CHAR(n)* digunakan untuk menyimpan teks dengan panjang tetap sehingga ruang penyimpanan akan selalu terpakai penuh sesuai jumlah karakter yang ditentukan.
- 2) *VARCHAR(n)* digunakan untuk menyimpan teks dengan panjang dinamis sehingga hanya menggunakan ruang sesuai jumlah karakter yang diisi.
- 3) *TINYTEXT* digunakan untuk menyimpan teks berukuran sangat pendek dengan kapasitas penyimpanan yang terbatas.

- 4) *TEXT* digunakan untuk menyimpan teks berukuran sedang yang lebih panjang dari *TINYTEXT*.
- 5) *MEDIUMTEXT* digunakan untuk menyimpan teks yang lebih panjang lagi, cocok untuk data dengan jumlah karakter besar.
- 6) *LONGTEXT* digunakan untuk menyimpan teks yang sangat panjang dengan kapasitas terbesar di antara tipe data teks lainnya.

3. Tipe Data Binary

Tipe data binary pada *MySQL* digunakan untuk menyimpan data dalam bentuk biner, yaitu data mentah yang tidak ditafsirkan sebagai teks melainkan sebagai rangkaian byte. Tipe data ini biasanya digunakan untuk menyimpan informasi seperti file, gambar, dokumen, atau data digital lain yang membutuhkan format penyimpanan non-teks. Penggunaan tipe binary memungkinkan sistem menyimpan data dalam ukuran kecil hingga sangat besar sesuai kebutuhan, sekaligus menjaga struktur asli dari file yang disimpan. Dengan dukungan tipe binary, *MySQL* dapat mengelola berbagai jenis data kompleks yang tidak bisa direpresentasikan dalam bentuk numerik maupun karakter. Untuk tipe tipe pada data ini yaitu sebagai berikut.

- 1) *TINYBLOB* digunakan untuk menyimpan data biner berukuran sangat kecil seperti file dengan ukuran beberapa kilobyte.
- 2) *BLOB* digunakan untuk menyimpan data biner berukuran kecil hingga menengah yang lebih besar dari *TINYBLOB*.
- 3) *MEDIUMBLOB* digunakan untuk menyimpan data biner berukuran sedang dengan kapasitas yang lebih besar dari *BLOB*.

- 4) *LONGBLOB* digunakan untuk menyimpan data biner berukuran sangat besar yang membutuhkan kapasitas penyimpanan maksimal.

4. Tipe Data Tanggal & Waktu

Tipe data tanggal dan waktu pada *MySQL* digunakan untuk menyimpan informasi yang berkaitan dengan tanggal, jam, atau kombinasi keduanya, sehingga data yang berhubungan dengan waktu dapat dicatat secara akurat dan terstruktur. Penggunaan tipe data ini sangat penting untuk mencatat aktivitas seperti tanggal pembuatan data, waktu pembaruan, jadwal, log sistem, maupun riwayat proses. *MySQL* menyediakan beberapa format penyimpanan yang berbeda agar pengembang dapat memilih jenis yang paling sesuai, baik untuk menyimpan tanggal saja, waktu saja, maupun keduanya secara bersamaan. Dengan adanya tipe data ini, pengelolaan informasi berbasis waktu dapat dilakukan dengan lebih konsisten, efisien, dan mudah diolah kembali untuk keperluan analisis maupun pelacakan. Untuk tipe tipe pada data ini yaitu sebagai berikut.

- 1) *DATE* digunakan untuk menyimpan informasi tanggal dalam format tahun-bulan-hari.
- 2) *TIME* digunakan untuk menyimpan informasi waktu dalam format jam-menit-detik.
- 3) *DATETIME* digunakan untuk menyimpan kombinasi lengkap antara tanggal dan waktu dalam satu kolom.
- 4) *TIMESTAMP* digunakan untuk mencatat tanggal dan waktu secara otomatis berdasarkan waktu *server*; biasanya untuk mencatat waktu pembuatan atau pembaruan data.

- 5) *YEAR* digunakan untuk menyimpan informasi tahun dalam format dua digit atau empat digit.

5. Tipe Data *Boolean*

Tipe data Boolean pada *MySQL* digunakan untuk menyimpan nilai logika yang hanya memiliki dua kemungkinan, yaitu benar atau salah. Meskipun *MySQL* tidak memiliki tipe Boolean murni, nilai ini direpresentasikan melalui tipe *TINYINT* dengan nilai 1 sebagai benar dan 0 sebagai salah. Penggunaan tipe data ini sangat penting dalam pengambilan keputusan pada sistem, seperti menentukan status aktif atau tidak, valid atau tidak, serta berbagai kondisi logis lainnya. Dengan adanya tipe Boolean, proses pengolahan data yang membutuhkan penilaian kondisi menjadi lebih sederhana, efisien, dan mudah digunakan dalam berbagai operasi logika. Untuk tipe tipe pada data ini yaitu sebagai berikut.

- 1) *BOOLEAN / BOOL* digunakan untuk merepresentasikan nilai logika yang hanya bernilai benar atau salah dan disimpan sebagai *TINYINT* (1).

6. Tipe Data *Spatial* (Geometri)

Tipe data spatial atau geometri pada *MySQL* digunakan untuk menyimpan dan mengolah data yang berhubungan dengan posisi, bentuk, dan lokasi dalam ruang dua dimensi. Tipe data ini memungkinkan penyimpanan berbagai objek geometri seperti titik, garis, dan area sehingga sangat berguna untuk aplikasi yang memerlukan pemetaan atau analisis lokasi. *MySQL* menyediakan fungsi-fungsi khusus untuk memanipulasi dan mengevaluasi data geospasial, seperti menghitung jarak, menentukan hubungan antar objek, atau memeriksa apakah suatu titik berada dalam suatu area. Dengan dukungan tipe data spatial, penyimpanan informasi

berbasis koordinat dapat dilakukan secara efisien dan akurat, terutama untuk sistem yang memerlukan representasi lokasi atau struktur wilayah. Untuk tipe tipe pada data ini yaitu sebagai berikut.

- 1) *POINT* digunakan untuk menyimpan satu koordinat lokasi dalam bentuk titik tunggal.
- 2) *LINESTRING* digunakan untuk menyimpan rangkaian beberapa titik yang terhubung sehingga membentuk sebuah garis.
- 3) *POLYGON* digunakan untuk menyimpan sekumpulan titik yang membentuk area tertutup seperti bentuk segi banyak.
- 4) *MULTIPOINT* digunakan untuk menyimpan beberapa titik terpisah dalam satu kolom sebagai satu kesatuan data.
- 5) *MULTILINESTRING* digunakan untuk menyimpan kumpulan beberapa garis yang berdiri sendiri namun disimpan sebagai satu objek.
- 6) *MULTIPOLYGON* digunakan untuk menyimpan beberapa area tertutup yang terpisah dalam satu kolom sebagai satu objek.
- 7) *GEOMETRY* digunakan sebagai tipe umum untuk menyimpan berbagai bentuk data geometri tanpa menentukan jenis geometri tertentu.

2.4. Sekilas Tentang PHP

PHP merupakan bahasa pemrograman sisi *server* yang dirancang khusus untuk membangun aplikasi *web* dinamis dan interaktif. PHP dapat dipadukan dengan berbagai teknologi seperti *HTML*, *CSS*, dan *JavaScript* sehingga mampu menghasilkan halaman *web* yang dapat memproses permintaan pengguna secara real time. Secara khusus, PHP dirancang untuk membentuk web dinamis. Artinya

ia dapat membentuk satu tampilan berdasarkan permintaan terkini. Misalnya dapat ditampilkan isi basis data kehalaman web (Sitorus & Sakban, 2021). Keunggulan PHP terletak pada kemudahan sintaks, fleksibilitas, serta dukungannya terhadap berbagai jenis *server* dan sistem operasi. Selain itu, PHP memiliki kemampuan terintegrasi dengan berbagai jenis basis data, termasuk *MySQL*, yang menjadikannya salah satu bahasa pemrograman paling populer untuk pengembangan aplikasi *web*. PHP juga memiliki komunitas besar serta banyak pustaka dan *framework* pendukung sehingga memudahkan proses pengembangan, pemeliharaan, dan pengembangan fitur baru.

Penggunaan PHP memberikan kemampuan untuk membangun sistem berbasis *web* yang responsif dan mudah diakses melalui browser tanpa memerlukan instalasi khusus pada perangkat pengguna. PHP memungkinkan pengelolaan data secara dinamis, seperti memproses laporan, menampilkan status, melakukan validasi input, dan berinteraksi langsung dengan *Database MySQL* untuk menyimpan maupun mengambil data secara efisien. Dukungan fitur-fitur PHP seperti pengelolaan *session*, pengunggahan file, autentikasi pengguna, serta pemanggilan fungsi secara modular membantu membangun alur kerja sistem pelaporan kerusakan yang lebih terstruktur dan aman

2.5. Sistem Informasi Pelaporan dan Penanganan Kerusakan Fasilitas

Sistem Informasi Pelaporan dan Penanganan Kerusakan Fasilitas merupakan suatu aplikasi yang dirancang untuk mempermudah proses pelaporan, pencatatan, dan pemantauan perbaikan fasilitas yang mengalami kerusakan (Britney Hana Maria et al., 2022). Sistem ini mengintegrasikan seluruh aktivitas mulai dari laporan

kerusakan yang diajukan oleh pengguna, verifikasi oleh petugas, hingga tindak lanjut perbaikan oleh tim teknisi. Dengan adanya sistem ini, setiap laporan dapat tercatat secara rapi, waktu respon terhadap kerusakan dapat dipantau, dan setiap tindakan perbaikan terdokumentasi secara jelas. Penggunaan teknologi berbasis *web* memungkinkan akses yang mudah dari berbagai perangkat, sehingga meminimalkan kendala komunikasi dan meningkatkan transparansi serta akuntabilitas. Selain itu, sistem ini menyediakan laporan statistik kerusakan, sehingga pihak manajemen dapat mengambil keputusan yang tepat mengenai perbaikan, pengadaan, atau pemeliharaan fasilitas secara berkala.

Implementasi sistem tersebut memberikan kemudahan dalam mengelola informasi terkait kondisi fasilitas, mengurangi risiko kerusakan yang tidak tertangani, dan meningkatkan efisiensi operasional secara keseluruhan (Fauzi et al., 2023). Dengan fitur yang lengkap mulai dari input laporan, notifikasi otomatis, hingga pelacakan status perbaikan, sistem ini mampu memastikan setiap kerusakan tertangani tepat waktu serta memberikan rekam jejak yang dapat dianalisis untuk pengambilan keputusan strategis di masa depan. Selain itu, penggunaan sistem ini memungkinkan koordinasi antarpetugas menjadi lebih efektif karena semua data tersentralisasi dan dapat diakses secara real-time.

2.5.1. Pengertian Sistem Informasi Pelaporan dan Penanganan Kerusakan Fasilitas

Sistem Informasi Pelaporan dan Penanganan Kerusakan Fasilitas merupakan sebuah mekanisme terstruktur yang dirancang untuk mempermudah proses pelaporan, pendataan, dan pemantauan kondisi fasilitas yang mengalami kerusakan.

Sistem ini memanfaatkan teknologi informasi untuk mencatat setiap laporan kerusakan secara cepat dan akurat, memprioritaskan penanganan berdasarkan tingkat urgensi, serta menyimpan riwayat perbaikan secara rapi. Dengan sistem ini, komunikasi antara pihak pengguna fasilitas dan tim pemeliharaan menjadi lebih efisien, sehingga setiap permasalahan dapat ditangani dengan tepat waktu. Tidak hanya membantu mempercepat proses perbaikan, sistem juga memungkinkan evaluasi kinerja pemeliharaan, analisis frekuensi kerusakan, serta pengambilan keputusan yang lebih berbasis data. Selain itu, sistem ini dapat diakses secara digital sehingga meminimalisir penggunaan dokumen fisik, mempermudah tracking status laporan, dan memberikan transparansi bagi semua pihak terkait.

Penerapan sistem informasi pelaporan dan penanganan kerusakan fasilitas memberikan manfaat nyata dalam pengelolaan sarana yang ada, sehingga setiap kerusakan dapat ditangani lebih cepat dan tepat. Dengan adanya pencatatan yang sistematis dan terpusat, pihak pengelola fasilitas dapat mengidentifikasi pola kerusakan, memperkirakan kebutuhan perbaikan, dan mengalokasikan sumber daya secara lebih efisien. Sistem ini juga memungkinkan koordinasi yang lebih baik antarbagian, meminimalkan risiko kelalaian, serta memastikan bahwa fasilitas yang digunakan selalu dalam kondisi optimal. Implementasi sistem berbasis *web* dengan *Database* yang handal mendukung kelancaran proses pelaporan dan penanganan, sehingga seluruh alur kerja dari pelaporan hingga perbaikan dapat dilakukan dengan transparan, akurat, dan dapat dipertanggungjawabkan, sekaligus meningkatkan kualitas layanan dan kenyamanan pengguna fasilitas.

2.5.2. Kosep Dasar Sistem Informasi Pelaporan dan Penanganan Kerusakan Fasilitas

Sistem informasi pelaporan dan penanganan kerusakan fasilitas adalah suatu sistem yang dirancang untuk mempermudah proses identifikasi, pelaporan, dan penanganan kerusakan fasilitas secara lebih cepat, terstruktur, dan efisien. Sistem ini memanfaatkan teknologi informasi untuk mencatat laporan kerusakan secara digital, mulai dari input data kerusakan oleh pengguna, verifikasi laporan oleh petugas terkait, hingga pemantauan status perbaikan hingga selesai. Dengan adanya sistem ini, alur komunikasi antara pengguna fasilitas dan pihak pengelola menjadi lebih transparan, mengurangi kemungkinan laporan hilang atau terlambat ditindaklanjuti, serta mempermudah manajemen fasilitas dalam membuat laporan rutin dan analisis kerusakan. Selain itu, integrasi basis data memungkinkan penyimpanan riwayat kerusakan, prioritas perbaikan, dan pengaturan jadwal pemeliharaan sehingga dapat meningkatkan efektivitas dan efisiensi operasional secara keseluruhan.

Implementasi sistem informasi pelaporan dan penanganan kerusakan fasilitas sangat membantu dalam mempercepat proses identifikasi masalah dan tindakan perbaikan, sekaligus meminimalkan gangguan yang timbul akibat fasilitas yang rusak. Dengan mekanisme yang terstruktur, setiap laporan kerusakan dapat diproses dengan lebih akurat, memudahkan petugas dalam menentukan langkah perbaikan yang tepat dan memonitor progres secara real-time. Selain itu, sistem ini juga mendukung pengambilan keputusan berbasis data, karena setiap kerusakan yang terjadi dapat dianalisis untuk menemukan pola, frekuensi, dan penyebab utama kerusakan. Hal ini memungkinkan pihak pengelola untuk membuat strategi

pemeliharaan preventif yang lebih efektif, mengurangi biaya perbaikan, dan meningkatkan kenyamanan serta produktivitas pengguna fasilitas secara keseluruhan.

2.6. Sekilas Tentang *Web Server*

Web server merupakan perangkat lunak yang menyajikan konten ke perangkat klien, seperti browser web, melalui internet atau jaringan local (Saputra et al., 2023). *Web server* memainkan peran penting dalam penyajian konten di internet, mulai dari halaman statis hingga aplikasi *web* yang kompleks. Beberapa *web server* populer meliputi *Apache*, *Nginx*, dan *Microsoft IIS*, yang masing-masing memiliki kelebihan dalam hal performa, konfigurasi, dan keamanan. Selain itu, *web server* juga menangani berbagai fungsi penting seperti manajemen koneksi, autentikasi pengguna, enkripsi data, dan pemrosesan skrip sisi server, misalnya PHP. Kecepatan, keandalan, dan kemampuan *web server* untuk menangani banyak permintaan sekaligus menjadi faktor kunci dalam pengalaman pengguna yang lancar, terutama ketika sistem yang dijalankan membutuhkan interaksi real-time dan penyimpanan data yang konsisten. Dengan kemampuannya mengelola komunikasi antara klien dan *server*, *web server* menjadi fondasi utama dalam penyebaran informasi digital secara efisien.



Web Server

Gambar 2. 3. Web Server

Sumber: <https://www.jetorbit.com/blog/pengertian-web-server-dan-fungsinya/>

Dalam implementasi sistem pelaporan dan penanganan kerusakan ruangan, *web server* berperan sebagai jembatan yang memungkinkan data dari pengguna dapat diterima, diproses, dan disimpan secara terstruktur dalam basis data. Setiap laporan kerusakan yang dikirimkan melalui formulir online diterima oleh *web server*, yang kemudian mengeksekusi skrip PHP untuk menyimpan informasi tersebut ke dalam *MySQL*, mengupdate status perbaikan, dan menampilkan notifikasi bagi petugas terkait. Dengan kemampuan *web server* menangani permintaan dari banyak pengguna sekaligus, proses pelaporan menjadi lebih cepat dan transparan, serta memastikan bahwa data kerusakan ruangan tersimpan dengan aman dan dapat diakses kapan saja.

2.7. Aplikasi Edraw

Edraw adalah aplikasi perangkat lunak yang dirancang untuk membuat berbagai jenis *Diagram* dan visualisasi secara mudah dan efisien. Aplikasi ini menawarkan fitur drag-and-drop yang intuitif, berbagai template siap pakai, dan

simbol-simbol standar untuk *Diagram* alur, *Diagram* jaringan, *Diagram* organisasi, peta konsep, serta mind map. Edraw juga mendukung ekspor ke berbagai format seperti PDF, Word, PowerPoint, dan gambar, sehingga memudahkan pengguna untuk membagikan hasil kerjanya. Keunggulan lain dari Edraw adalah antarmuka yang user-friendly, memungkinkan pengguna pemula maupun profesional untuk membuat *Diagram* yang kompleks tanpa perlu keahlian desain khusus. Dengan kemampuan integrasi dan fleksibilitas yang tinggi, Edraw mempermudah perancangan visualisasi data dan proses, serta membantu menyajikan informasi secara lebih sistematis dan terstruktur.



Gambar 2. 4. Aplikasi Edraw

Sumber: <https://www.nesabamedia.com/download-edrawmax/>

Dalam pengembangan sistem informasi, Edraw dapat digunakan untuk merancang alur kerja, struktur *Database*, dan *Diagram* proses dengan lebih jelas dan terorganisir. *Diagram* alur yang dibuat dengan Edraw membantu menggambarkan setiap langkah dalam proses pelaporan dan penanganan kerusakan ruangan, sehingga memudahkan pemahaman bagi pengembang maupun pengguna sistem. Penggunaan Edraw memungkinkan visualisasi proses yang kompleks menjadi lebih sederhana, mempercepat identifikasi titik masalah, dan memastikan setiap fungsi sistem tertata dengan baik. Selain itu, dokumentasi *Diagram* yang rapi

dan mudah dibaca mendukung evaluasi dan perbaikan sistem secara berkelanjutan, sehingga implementasi sistem informasi dapat lebih efisien, transparan, dan efektif.

2.8. Aplikasi *Visual Studio Code*

Visual Studio Code adalah sebuah editor kode sumber yang dikembangkan oleh Microsoft dan terkenal karena kecepatan, fleksibilitas, dan kemampuannya yang mendukung berbagai bahasa pemrograman. Aplikasi ini memiliki antarmuka yang sederhana namun powerful, dilengkapi dengan fitur seperti *syntax highlighting*, *code completion*, *debugging*, dan integrasi *version control* melalui Git. Selain itu, VS Code mendukung berbagai *extensions* yang memungkinkan pengguna menambahkan fungsi tambahan sesuai kebutuhan, mulai dari pengembangan *web*, aplikasi desktop, hingga pengolahan *Database*. Kemampuannya yang ringan membuatnya dapat berjalan lancar pada berbagai sistem operasi seperti Windows, macOS, dan Linux, sehingga banyak pengembang memilihnya sebagai alat utama untuk menulis, mengedit, dan mengelola kode secara efisien.



Gambar 2. 5. Aplikasi Visual Studio Code

Sumber: <https://alfanbro99.wordpress.com/2019/02/28/mengenal-lebih-dekat-dengan-visual-studio-code/>

Penggunaan *Visual Studio Code* sangat membantu dalam pembuatan sistem informasi karena kemampuannya dalam mengelola kode dengan lebih terstruktur

dan rapi, serta memudahkan integrasi dengan bahasa pemrograman seperti PHP dan basis data *MySQL*. Dengan fitur *debugging* dan *code completion*, proses pengembangan aplikasi menjadi lebih cepat dan minim kesalahan, sementara integrasi dengan *extensions* memungkinkan pengelolaan proyek yang lebih kompleks tanpa harus berpindah aplikasi lain. Selain itu, kemampuan VS Code untuk mengorganisir file proyek, menandai error secara otomatis, dan menyediakan antarmuka yang interaktif membuat seluruh proses pembuatan sistem menjadi lebih sistematis, efisien, dan mudah dipantau dari awal hingga akhir pengembangan.

2.9. Aplikasi XAMPP

XAMPP adalah paket perangkat lunak open-source yang menyediakan lingkungan server lokal untuk menjalankan aplikasi berbasis *web*. *XAMPP* terdiri dari beberapa komponen utama, seperti Apache sebagai *web server*, *MySQL* atau MariaDB sebagai sistem manajemen basis data, PHP sebagai bahasa pemrograman server-side, dan Perl sebagai bahasa tambahan. Dengan *XAMPP*, pengguna dapat dengan mudah membuat, menguji, dan mengembangkan aplikasi *web* secara lokal tanpa harus terkoneksi ke server online. Instalasinya yang sederhana dan antarmuka yang ramah membuat *XAMPP* populer di kalangan pengembang pemula maupun profesional. Selain itu, *XAMPP* mendukung berbagai platform, termasuk Windows, Linux, dan macOS, sehingga fleksibel untuk digunakan di berbagai perangkat.



Gambar 2. 6. Aplikasi XAMPP

Sumber: <https://berita.smkbundakandung.sch.id/2023/09/20/cara-meng-install-XAMPP/>

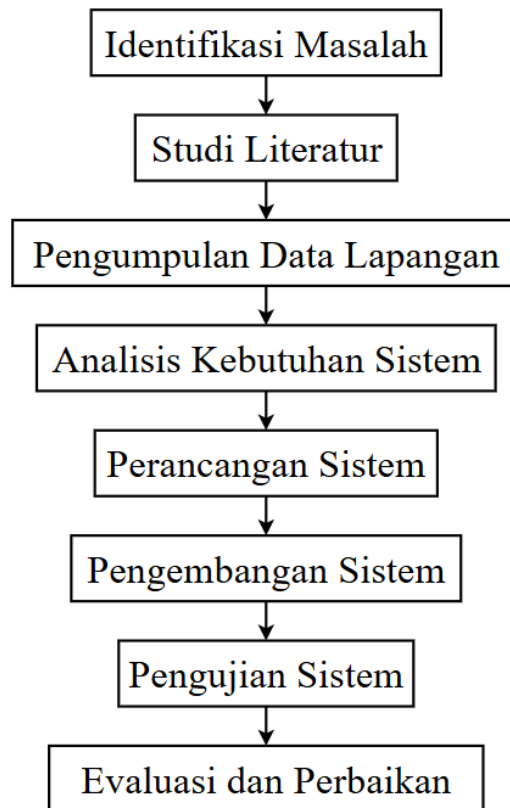
Penggunaan *XAMPP* memberikan kemudahan dalam membangun dan mengelola sistem informasi berbasis *web*, terutama yang membutuhkan interaksi dengan basis data secara langsung. Dengan lingkungan server lokal ini, pengembangan sistem pelaporan dan penanganan kerusakan ruangan dapat dilakukan secara lebih efisien, karena setiap fitur dari aplikasi dapat diuji secara menyeluruh sebelum diimplementasikan secara real-time. Selain itu, pengelolaan *Database* menjadi lebih terstruktur dan terkontrol, sehingga data laporan kerusakan ruangan dapat disimpan, diperbarui, dan ditampilkan secara akurat.

2.10. Penelitian Terdahulu

No	Penulis	Judul	Tahun	Hasil
1	(Sistem et al., 2021)	Perancangan Sistem Informasi Pelaporan Rujukan Pasien Di Puskesmas Cicalengka Dtp	2021	Hasil penelitian menunjukkan bahwa sistem informasi pelaporan yang dirancang mampu mengatasi permasalahan pelaporan rujukan yang sebelumnya dilakukan secara manual dan sering menimbulkan keterlambatan serta ketidakakuratan data. Implementasi aplikasi tersebut mempermudah petugas dalam menginput, mengolah, dan menghasilkan laporan rujukan secara lebih cepat, terstruktur, dan efisien.
2	(Mustika Sari & Prihartono, 2021)	PENDAMPINGAN PENGGUNAAN APLIKASI SIPEDAS BERANI (SISTEM INFORMASI PELAPORAN DASA WISMA) KECAMATAN BATUNUNGGAL KOTA BANDUNG	2023	Hasil penelitian menunjukkan bahwa sistem informasi pelaporan yang dikembangkan mampu menggantikan proses pengumpulan data warga yang sebelumnya dilakukan secara manual, sehingga mengurangi penumpukan kertas dan mempercepat alur pelaporan. Sistem ini juga terbukti mudah digunakan oleh anggota PKK, dengan tingkat keberhasilan implementasi aplikasi mencapai

				100% serta efektivitas pelatihan pengguna sebesar 80,4%.
3	(Alfonsius et al., 2023)	Sistem Informasi Pelaporan Pekerjaan Proyek Berbasis SDLC Modelling (Studi Kasus: PT Vertikal Tiara Manunggal)	2023	Hasil penelitian tersebut menunjukkan bahwa sistem informasi pelaporan yang dibangun mampu menyajikan informasi perusahaan secara lengkap dan dapat diakses pengguna dengan mudah melalui website yang telah dirancang. Pengujian menggunakan metode BlackBox membuktikan bahwa seluruh fitur pelaporan pada sistem berjalan sesuai fungsinya dengan tingkat keberhasilan 100%.
4	(Iqbal et al., 2022)	SISTEM INFORMASI PELAPORAN DAN MONITORING PEKERJAAN DI BADAN PUSAT STATISTIK KABUPATEN SUBANG	2022	Sistem informasi pelaporan yang dibangun mampu mengatasi permasalahan keterlambatan dan ketidakteraturan pelaporan tugas harian dengan menyediakan fitur pengiriman laporan yang disertai batas waktu serta notifikasi, sehingga proses pelaporan menjadi lebih tertib dan terpantau. Hasil pengujian menunjukkan bahwa sistem ini berjalan sangat baik dan diterima oleh pengguna, terbukti dari nilai keberhasilan Blackbox sebesar 100% serta tingkat kepuasan User Acceptance Test yang tinggi dari tiap kategori pengguna.
5	(Mutmainah et al., 2024)	Rancang Bangun Sistem Informasi Pelaporan Penyelenggaraan Pemerintahan Daerah (SIPPDA) Provinsi Nusa Tenggara Barat Berbasis Website	2024	Sistem informasi pelaporan yang dikembangkan mampu mempercepat proses pengumpulan data dan perhitungan capaian kinerja pemerintah daerah sehingga penyusunan laporan yang sebelumnya dilakukan secara manual dapat berjalan lebih efisien. Hasil pengujian menunjukkan bahwa sistem mampu berfungsi dengan baik tanpa error dan diterima oleh pengguna dengan tingkat penerimaan yang meningkat pada setiap iterasi UAT. Saya lebih suka respons ini

2.11. Kerangka Kerja Penelitian



Gambar 2. 7. Kerangka Kerja Penelitian

Untuk mencapai tujuan penelitian dan menghasilkan sistem informasi yang sesuai kebutuhan, diperlukan rangkaian langkah yang sistematis dan terstruktur agar proses pengembangan dapat berjalan dengan jelas dan terarah. Tahapan penelitian berikut menggambarkan alur kerja yang digunakan mulai dari identifikasi masalah hingga penyusunan laporan akhir.

1) Identifikasi Masalah

Tahap awal dimulai dengan mengamati masalah yang terjadi di Universitas Labuhanbatu, yaitu proses pelaporan kerusakan fasilitas ruangan yang belum terstruktur sehingga laporan sering terlambat, tidak terdokumentasi, dan sulit dipantau.

2) Studi Literatur

Mengumpulkan teori dan referensi terkait sistem informasi, PHP, MySQL, perancangan sistem, serta penelitian terdahulu yang relevan untuk menjadi dasar dalam mengembangkan sistem.

3) Pengumpulan Data Lapangan

Mengambil data melalui observasi, wawancara, atau dokumentasi untuk mengetahui jenis kerusakan, alur pelaporan yang berjalan, kebutuhan pengguna, serta struktur organisasi yang terlibat dalam proses pelaporan dan perbaikan.

4) Analisis Kebutuhan Sistem

Menentukan kebutuhan fungsional dan non-fungsional sistem, seperti fitur pelaporan, verifikasi, status perbaikan, manajemen pengguna, serta kebutuhan penyimpanan data pada basis data MySQL.

5) Perancangan Sistem (Design)

Membuat desain proses dan struktur sistem, seperti use case diagram, flowchart, diagram basis data, rancangan tabel MySQL, dan desain antarmuka pengguna (UI).

6) Pengembangan Sistem (Implementasi)

Membangun sistem informasi menggunakan bahasa pemrograman PHP dengan basis data MySQL sesuai hasil perancangan, sehingga menghasilkan aplikasi pelaporan kerusakan fasilitas yang dapat digunakan.

7) Pengujian Sistem

Melakukan pengujian fungsional untuk memastikan setiap fitur berjalan dengan benar, seperti form pelaporan, verifikasi, pembaruan status, dan

penyimpanan data. Pengujian juga dilakukan kepada pengguna untuk menilai kemudahan penggunaan dan keakuratan sistem.

8) Evaluasi dan Perbaikan

Mengevaluasi hasil pengujian, menerima masukan dari pengguna, kemudian melakukan perbaikan terhadap fitur atau tampilan sistem agar lebih sesuai kebutuhan.

9) Penyusunan Laporan Penelitian

Tahap akhir adalah menyusun seluruh proses penelitian mulai dari pendahuluan hingga kesimpulan untuk menjadi laporan atau skripsi yang lengkap.