

BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Teori Umum

Pada bagian ini akan dibahas teori-teori umum yang relevan dengan penelitian ini, mencakup konsep digitalisasi, sistem informasi, dan aspek-aspek terkait lainnya, serta pemeliharaan fasilitas industri. Penjelasan teori umum ini bertujuan untuk memberikan landasan konseptual yang kuat dalam pengembangan aplikasi web untuk digitalisasi pelaporan dan pengecekan fasilitas .

2.1.1 Digitalisasi Dalam Industri

Digitalisasi adalah proses mengubah informasi atau proses bisnis yang sebelumnya dilakukan secara memanfaatkan kertas dan catatan tulisan tangan. menjadi format digital, dengan tujuan meningkatkan , akurasi, dan aksesibilitas. Dalam konteks industri, digitalisasi memainkan peran penting dalam mengoptimalkan operasional, meminimalkan kesalahan, serta mempermudah analisis data secara real-time. Transformasi digital di industri mencakup berbagai aspek seperti pemantauan produksi, manajemen rantai pasok, hingga pemeliharaan fasilitas (Tarsini, 2015).

Digitalisasi pelaporan dan pengecekan fasilitas di bagian *plant activity* PT Philips Industri Batam bertujuan untuk meminimalkan ketergantungan pada sistem memanfaatkan kertas dan catatan tulisan tangan. dan meningkatkan operasional.

2.1.2 Manufaktur Elektronik

Manufaktur adalah proses transformasi bahan mentah menjadi produk jadi melalui proses fisik, kimia, atau mekanis yang sistematis. Manufaktur menjadi inti dari sektor industri karena mencakup perencanaan, produksi, pengendalian mutu, dan distribusi produk. Menurut Heizer & Render (2015), manufaktur merupakan kegiatan yang mencakup penciptaan produk fisik melalui penggunaan sumber daya seperti tenaga kerja, mesin, energi, dan informasi. Sektor ini sangat berperan dalam meningkatkan produktivitas, nilai tambah, dan daya saing suatu negara.

Secara fundamental, manufaktur elektronik adalah proses produksi yang melibatkan perancangan, pembuatan, dan perakitan komponen-komponen elektronik untuk membentuk perangkat atau sistem elektronik yang berfungsi (Serdar, 2021). Ini adalah disiplin ilmu multidisiplin yang memadukan teknik elektro, teknik material, teknik mesin, dan ilmu komputer. Manufaktur elektronik lebih dari sekadar perakitan; ia mencakup seluruh siklus hidup produk, yang meliputi:

1. Desain Produk dan Sirkuit: Tahap awal di mana spesifikasi fungsional, arsitektur sirkuit, dan tata letak komponen dirancang. Ini seringkali melibatkan penggunaan software Computer-Aided Design (CAD).
2. Manufaktur Komponen: Produksi individual komponen elektronik seperti resistor, kapasitor, dioda, transistor, dan sirkuit terpadu (Integrated Circuit - IC). Proses ini sering dilakukan oleh produsen komponen spesialis.

3. Manufaktur Papan Sirkuit Tercetak (PCB): Produksi Printed Circuit Board (PCB), yaitu papan non-konduktif dengan jalur konduktif yang terukir di permukaannya untuk menghubungkan komponen elektronik.
4. Perakitan Komponen pada PCB: Proses menempatkan dan menyolder komponen elektronik ke PCB. Ini dapat dilakukan secara manual atau, yang lebih umum, menggunakan mesin pick-and-place otomatis (Surface Mount Technology - SMT) untuk volume tinggi (Handoko & Suryanto, 2018).
5. Pengujian dan Kontrol Kualitas: Memverifikasi fungsionalitas dan kinerja perangkat yang telah dirakit, serta memastikan produk memenuhi standar kualitas yang ditetapkan.

Pengemasan dan Distribusi: Tahap akhir sebelum produk sampai ke tangan konsumen.

2.1.3 *E-Facility Plant*

Konsep E-Facility Plant mengacu pada penerapan teknologi digital untuk mengelola dan mengoptimalkan berbagai aspek operasional fasilitas industri atau plant. Ini melibatkan transformasi proses pemeliharaan, inspeksi, dan pelaporan yang sebelumnya manual menjadi sistem terintegrasi yang berbasis elektronik dan data. Tujuannya adalah untuk meningkatkan efisiensi operasional, akurasi data, kecepatan respons, dan transparansi dalam manajemen fasilitas secara keseluruhan (Chandra & Singh, 2018).

Dalam konteks industri 4.0, *E-Facility Plant* bukan hanya sekadar mendigitalisasi dokumen, tetapi juga menciptakan ekosistem di mana data dari berbagai fasilitas dapat dikumpulkan secara *real-time*, dianalisis, dan

divisualisasikan untuk mendukung pengambilan keputusan yang lebih baik. Pendekatan ini memungkinkan manajemen untuk memiliki visibilitas penuh terhadap kondisi aset, jadwal pemeliharaan, serta kinerja tim di lapangan (Lee et al., 2020). Digitalisasi ini juga berkontribusi pada peningkatan keamanan kerja karena potensi masalah pada fasilitas dapat teridentifikasi lebih dini dan ditindaklanjuti dengan cepat (Wang & Chen, 2019). Sistem *E-Facility Plant* seringkali memanfaatkan teknologi seperti aplikasi web, *mobile*, sensor (IoT), dan *big data analytics* untuk menciptakan lingkungan operasional yang lebih cerdas dan responsif.

E-Facility Plant adalah sebuah sistem terintegrasi yang memanfaatkan teknologi digital untuk mengelola inspeksi fasilitas di suatu pabrik atau area industri. Sistem ini memungkinkan otomatisasi proses pengecekan, pelaporan, dan pemeliharaan fasilitas, dari yang sebelumnya manual menjadi berbasis data elektronik. Tujuannya adalah untuk menyediakan informasi kondisi fasilitas secara *real-time*, meningkatkan kinerja tim lapangan, dan mempermudah pengambilan keputusan manajemen melalui visualisasi data yang komprehensif, sehingga operasional *plant* menjadi lebih mudah dan aman.

2.1.4 Sistem Informasi

Sistem informasi didefinisikan sebagai suatu kerangka terorganisir yang terdiri dari komponen-komponen interdependen yang bekerja sama untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi. Komponen-komponen utama ini meliputi perangkat keras (hardware), perangkat lunak (software), basis data (data), prosedur operasional, serta sumber daya manusia yang

mengoperasikannya (Laudon and Laudon 2022). Dalam konteks pengelolaan fasilitas industri, sistem informasi memiliki peran krusial dalam mengatur data pengecekan dan pelaporan secara sistematis, sehingga memungkinkan akses informasi yang mudah dan pemantauan kondisi secara berkala.

Transformasi digital melalui sistem informasi memungkinkan organisasi untuk beralih dari praktik manual berbasis kertas menuju platform digital yang lebih efisien, terintegrasi, dan mudah diakses. Hal ini tidak hanya mengurangi ketergantungan pada dokumen fisik dan meminimalkan risiko kehilangan data, tetapi juga memfasilitasi analisis data secara real-time untuk mendukung pengambilan keputusan yang lebih cepat dan akurat (Mcgraw-Hill and O'brien 2011). Efektivitas sistem informasi sangat bergantung pada bagaimana elemen-elemennya dirancang untuk saling melengkapi dan memenuhi kebutuhan spesifik pengguna serta tujuan bisnis organisasi.

2.1.5 Pemeliharaan Fasilitas (*Facility Management*)

Pemeliharaan fasilitas, atau Facility Management (FM), merupakan disiplin ilmu yang mengintegrasikan orang, tempat, proses, dan teknologi untuk memastikan fungsionalitas lingkungan yang dibangun (IFMA, 2020). Tujuannya adalah untuk mendukung produktivitas inti organisasi dengan mengelola aset fisik dan layanan pendukung secara efektif. Ini mencakup berbagai aktivitas mulai dari perawatan preventif dan korektif, manajemen ruang, keselamatan dan keamanan, hingga pengelolaan energi dan keberlanjutan (Atkin & Brooks, 2015). Dalam konteks industri seperti plant atau pabrik, pemeliharaan fasilitas sangat krusial

untuk menjaga kelancaran operasional, meminimalkan downtime, dan memperpanjang umur aset.

Penerapan Facility Management yang strategis tidak hanya berfokus pada perbaikan ketika terjadi kerusakan, tetapi juga pada perencanaan proaktif untuk mencegah masalah. Hal ini melibatkan penjadwalan inspeksi rutin, pemantauan kondisi aset, dan implementasi sistem pelaporan yang efisien untuk mendeteksi potensi masalah sebelum menjadi kritis (Rondeau et al., 2012). Dengan adanya digitalisasi dalam pemeliharaan fasilitas, seperti yang diusung oleh konsep E-Facility Plant, proses-proses ini dapat dilakukan dengan lebih akurat, cepat, dan berbasis data, sehingga mendukung keputusan yang lebih informatif dan respons yang lebih sigap terhadap kebutuhan pemeliharaan.

2.1.6 Web Development

Pengembangan web (web development) merupakan proses komprehensif yang melibatkan perancangan, pembangunan, dan pemeliharaan aplikasi berbasis internet, mulai dari situs web statis hingga sistem interaktif yang kompleks (Kiran Gandikota et al. 2023). Disiplin ini secara fundamental terbagi menjadi dua area utama: pengembangan sisi klien (frontend) yang berfokus pada antarmuka pengguna dan pengalaman visual, serta pengembangan sisi server (backend) yang menangani logika bisnis, pengelolaan basis data, dan interaksi dengan server. Teknologi frontend umumnya mencakup HTML untuk struktur, CSS untuk gaya, dan JavaScript untuk fungsionalitas interaktif, yang semuanya dieksekusi di peramban pengguna (Li and Zhang 2021).

Sementara itu, backend development bertanggung jawab atas fungsionalitas di balik layar, termasuk manajemen data, otentikasi pengguna, dan pemrosesan permintaan dari frontend. Bahasa pemrograman seperti PHP, Python, atau Node.js sering digunakan di sisi backend, didukung oleh kerangka kerja (framework) yang memfasilitasi pengembangan yang efisien dan terstruktur (Sharma and Sharma 2021). Keamanan siber menjadi aspek krusial dalam pengembangan web modern untuk melindungi aplikasi dari berbagai ancaman seperti injeksi SQL dan cross-site scripting (XSS), memastikan integritas data dan privasi pengguna (Steven, Rifaldi, and Nugraha 2023). Dengan demikian, pengembangan web melibatkan integrasi berbagai teknologi untuk menciptakan solusi digital yang fungsional, responsif, dan aman.

2.1.7 *Web*

World Wide Web (WWW), atau yang lebih dikenal sebagai *Web*, merupakan sebuah sistem informasi global yang memfasilitasi pengguna untuk mengakses beragam sumber daya digital seperti teks, gambar, video, dan konten lainnya melalui jaringan internet (Akhil Krishna and Professor n.d.). Web tersusun dari kumpulan halaman-halaman yang saling terhubung melalui tautan (*hyperlink*) dan dapat diakses menggunakan peramban web (*web browser*) seperti Google Chrome, Mozilla Firefox, atau Microsoft Edge. Dalam lingkup pengembangan aplikasi web, terdapat dua komponen esensial: frontend dan backend. Frontend adalah bagian yang secara langsung berinteraksi dengan pengguna, dibangun menggunakan teknologi standar seperti HTML untuk struktur konten, CSS untuk tata letak dan

gaya visual, serta JavaScript untuk menambahkan interaktivitas (Akhil Krishna and Professor n.d.).

Sementara itu, backend berpusat pada pengelolaan logika bisnis aplikasi, pemrosesan data, dan komunikasi dengan basis data. Teknologi populer yang digunakan di backend mencakup bahasa pemrograman seperti PHP, Node.js, Python (dengan framework seperti Django), dan Ruby on Rails. Untuk penyimpanan data, umumnya digunakan sistem manajemen basis data seperti MySQL, PostgreSQL, atau MongoDB (Akhil Krishna and Professor n.d.). Dalam pembangunan aplikasi web yang kompleks, pemanfaatan framework menjadi sangat penting untuk mempercepat proses pengembangan sekaligus memastikan struktur kode yang terorganisir dengan baik. Laravel, sebagai salah satu framework PHP terkemuka, dikenal luas karena mengadopsi arsitektur Model-View-Controller (MVC) yang secara efektif memisahkan logika aplikasi, tampilan, dan data, sehingga meningkatkan modularitas dan pemeliharaan kode (Uyun, Rifqi, and Arif 2010). Laravel juga didukung oleh berbagai package tambahan, termasuk Filament, yang mempermudah pembentukan antarmuka administrasi dengan fitur-fitur yang interaktif dan modern, memungkinkan integrasi komponen siap pakai untuk pengembangan aplikasi yang lebih cepat dan efisien.

2.1.8 Aplikasi

Aplikasi adalah perangkat lunak yang dirancang untuk menjalankan fungsi tertentu pada perangkat elektronik seperti komputer, smartphone, atau tablet. Aplikasi secara umum terbagi menjadi dua jenis utama, yaitu aplikasi desktop dan aplikasi berbasis web. Dalam konteks ini, aplikasi berbasis web menjadi fokus

utama karena keunggulannya dalam hal aksesibilitas dan efisiensi. Aplikasi berbasis web adalah program yang dijalankan melalui browser dan diproses di server, sehingga tidak memerlukan instalasi langsung di perangkat pengguna. Keunggulan utamanya adalah kemudahan akses dari berbagai perangkat dan sistem operasi selama terhubung ke internet (Zhang 2013).

Web application dibangun dengan arsitektur *client-server*, di mana sisi client (*frontend*) berfungsi untuk menampilkan antarmuka pengguna, sementara sisi server (backend) bertugas mengelola logika bisnis dan pengolahan data, termasuk komunikasi dengan basis data (Zuhdi, Subiyanto, and Sukamta 2017). Dalam pengembangannya, penggunaan framework seperti Laravel sangat membantu mempercepat proses pembangunan aplikasi, menjaga struktur kode agar tetap rapi dan modular, serta meningkatkan aspek keamanan dan efisiensi sistem (He 2015).

Laravel merupakan framework PHP berbasis open-source yang mendukung pola arsitektur *Model-View-Controller* (MVC). Arsitektur ini memisahkan antara logika aplikasi, tampilan pengguna, dan pengolahan data, sehingga pengembangan menjadi lebih terorganisir dan mudah dipelihara (Nguyen et al. 2022). Laravel juga menyediakan berbagai fitur built-in seperti routing, sistem otentikasi, manajemen sesi, *middleware*, dan *Eloquent ORM*, yang sangat membantu dalam membangun aplikasi yang skalabel dan fleksibel (Ramadhani 2024).

Studi yang dilakukan oleh (Zuhdi, Subiyanto, and Sukamta 2017) menunjukkan bahwa implementasi Laravel dalam sistem informasi laboratorium berhasil meningkatkan efektivitas manajemen data dan jadwal operasional laboratorium. Selain itu, penelitian oleh (Ramadhani 2024) pada aplikasi

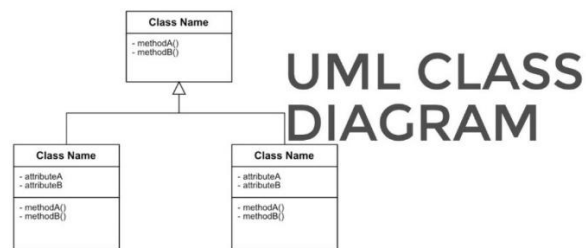
pengarsipan menunjukkan efisiensi pencarian data meningkat hingga 45% dibandingkan metode manual. Hal ini menunjukkan bahwa pemanfaatan Laravel tidak hanya mendukung pengembangan sistem modern, tetapi juga memberikan dampak nyata terhadap performa dan kemudahan penggunaan sistem berbasis web.

2.1.9 *Unified Modelling Language (UML)*

Unified Modeling Language (UML) adalah suatu standar visualisasi yang digunakan untuk memodelkan, mendokumentasikan, dan merancang sistem perangkat lunak berbasis objek. UML menyediakan berbagai jenis diagram seperti diagram use case, class, activity, sequence, dan lainnya yang membantu pengembang dalam memahami kebutuhan sistem, mendesain arsitektur, serta mempermudah proses komunikasi antar tim pengembang. Dengan UML, proses pengembangan perangkat lunak menjadi lebih terstruktur dan terdokumentasi dengan baik (Kurniawan, Immasari, and Sianipar 2022). Menurut Nasution dan Siregar (2021) dalam jurnal mereka yang berjudul "Analisis dan Perancangan Sistem Informasi Akademik Menggunakan *Unified Modeling Language (UML)*", penggunaan UML mampu membantu perancang sistem dalam menggambarkan alur proses bisnis dan struktur sistem secara sistematis dan logis, sehingga sistem yang dibangun lebih efisien dan sesuai dengan kebutuhan pengguna. Selain itu, dalam penelitian oleh Wibowo dan Lestari (2022), UML dipakai dalam merancang sistem informasi pengarsipan untuk menggambarkan proses bisnis secara menyeluruh dan meminimalkan kesalahan dalam tahap implementasi (Kurniawan, Immasari, and Sianipar 2022), yang membuktikan bahwa penerapan UML berperan besar dalam menunjang keberhasilan pengembangan sistem informasi.

Dalam proses rancang bangun, peneliti memanfaatkan class diagram, use case diagram, dan activity diagram. Berikut adalah penjelasan singkat mengenai masing-masing diagram.

1. *Class Diagram*



Gambar 2.1 *Class Diagram*

(Sumber: Penelitian, 2025)

Class diagram adalah diagram struktur statis yang menggambarkan kelas, atribut, metode, dan hubungan antar kelas seperti asosiasi, agregasi, dan generalisasi. Diagram ini krusial dalam tahap desain karena memvisualisasikan struktur data dan entitas sistem sebelum implementasi. Pembentukan diagram kelas juga sering menjadi dasar untuk menghasilkan basis data atau kerangka kode (Herchi and Abdessalem 2012).

2. *Use Case Diagram*

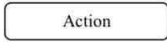
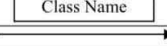
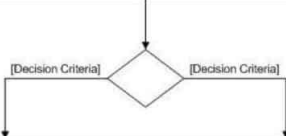
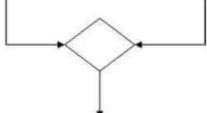
Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Gambar 2.2 *Use Case* Diagram

(Sumber: Penelitian, 2025)

Use case diagram memodelkan interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem, menggambarkan fungsionalitas sistem dari perspektif pengguna. Diagram ini berfungsi untuk mengidentifikasi kebutuhan dan batasan sistem melalui skenario penggunaan (Alturas 2023).

3. *Activity* Diagram

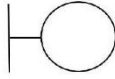
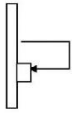
No.	Nama elemen	Fungsi	Notasi
1.	<i>Action</i>	Untuk menggambarkan perilaku yang sederhana dan bersifat <i>non-decomposable</i> .	
2.	<i>Activity</i>	Untuk mewakili kumpulan aksi (<i>action</i>)	
3.	<i>Object Node</i>	Untuk mewakili objek yang terhubung dengan kumpulan <i>object flow</i> .	
4.	<i>Control Flow</i>	Menunjukkan rangkaian pelaksanaan	
5.	<i>Object Flow</i>	Menunjukkan aliran sebuah objek dari sebuah aktivitas atau aksi ke aktivitas atau aksi lainnya.	
6.	<i>Initial Node</i>	Menandakan awal dari kumpulan aksi atau aktivitas.	
7.	<i>Final-Activity Node</i>	Untuk menghentikan seluruh <i>control flows</i> atau <i>object flows</i> pada sebuah aktivitas (atau aksi).	
8.	<i>Final-Flow Node</i>	Untuk menghentikan <i>control flow</i> atau <i>object flow</i> tertentu.	
9.	<i>Decision Node</i>	Untuk mewakili suatu kondisi pengujian yang bertujuan untuk memastikan bahwa <i>control flow</i> atau <i>object flow</i> hanya menuju ke satu arah.	
10.	<i>Merge Node</i>	Untuk menyatukan kembali <i>decision path</i> yang dibuat dengan menggunakan <i>decision node</i> .	

Gambar 2.3 Activity Diagram

(Sumber: Penelitian, 2025)

Activity diagram adalah representasi grafis aliran kerja yang menekankan urutan aktivitas, kondisi percabangan, paralelo, dan aliran data. Digunakan baik untuk modeling tingkat algoritmik maupun proses bisnis (Grönniger, Reiß, and Rumpe 2010).

4. Sequence Diagram

Gambar	Nama	Keterangan
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data
	Boundary Class	Menangani komunikasi antar lingkungan sistem
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika
	Recursive	Pesan untuk dirinya
	Activation	Mewakili proses durasi aktivasi sebuah operasi
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek

Gambar 2.4 *Sequence Diagram*

(Sumber: Penelitian, 2025)

Sequence diagram menggambarkan interaksi antar objek berdasarkan urutan waktu, menampilkan objek, pesan, dan lifeline. Diagram ini penting untuk memodelkan dinamika sistem, menjelaskan bagaimana objek berkomunikasi dalam suatu skenario (Yang et al. 2019).

2.2. Tinjauan Teori Khusus

Pada bagian ini akan dibahas teori-teori khusus yang relevan dengan pengembangan aplikasi web untuk digitalisasi pelaporan dan pengecekan fasilitas

di plant activity, termasuk konsep-konsep terkait *framework* Laravel, Filament sebagai *package* Laravel, serta penerapan digitalisasi dalam manajemen fasilitas.

2.2.1 Laravel

Laravel adalah framework PHP yang berorientasi pada pengembangan aplikasi web dengan arsitektur Model-View-Controller (MVC). Framework ini sangat populer karena kemudahan penggunaannya, fitur lengkap, dan dokumentasi yang baik. Laravel menyediakan berbagai fitur bawaan seperti routing, sistem autentikasi, manajemen basis data dengan Eloquent ORM, dan integrasi API, yang sangat mendukung pengembangan aplikasi web skala besar (Sinlae et al. 2024). Dalam pengembangan aplikasi pelaporan dan pengecekan fasilitas, Laravel digunakan sebagai fondasi untuk mengelola logika backend, komunikasi dengan database, serta penerapan arsitektur MVC. Penerapan MVC memastikan pemisahan yang jelas antara logika, data, dan tampilan, sehingga memudahkan pengembangan dan pemeliharaan kode (Prakarsya et al., 2023). Penggunaan Laravel dengan pola MVC terbukti efektif dalam membangun sistem monitoring dan pelaporan, seperti yang ditunjukkan oleh Surono, Suhandi, dan Alfiah (2022), yang menyatakan bahwa implementasi Laravel dengan arsitektur MVC memberikan kemudahan dalam proses monitoring produksi dan pengelolaan data (Surono, Suhandi, and Alfiah 2022). Hal serupa juga ditemukan dalam penelitian oleh (Sabna, Yasriady, and Giatman 2021), di mana penerapan Laravel berbasis MVC pada sistem manajemen kehadiran menghasilkan laporan otomatis serta mempermudah proses debugging karena pemisahan antara logika dan tampilan.

Beberapa keunggulan Laravel dalam konteks pengembangan aplikasi ini antara lain:

1. *Routing* yang Fleksibel: Laravel memudahkan pembuatan rute dinamis untuk mengatur jalur akses data dalam aplikasi.
2. *Blade Templating Engine*: Laravel menggunakan Blade sebagai mesin templating yang mempermudah integrasi logika bisnis dengan tampilan frontend.
3. *Eloquent ORM*: *Eloquent ORM* memfasilitasi pengelolaan database dengan menyediakan cara interaksi yang lebih mudah dan intuitif terhadap tabel-tabel dalam basis data.

2.2.2 Laravel Filament

Filament adalah sebuah *package* Laravel yang digunakan untuk membangun panel admin dengan tampilan modern dan interaktif. Filament menawarkan toolkit berbasis TALL stack (Tailwind CSS, Alpine.js, Laravel, Livewire) yang memungkinkan pengembang membuat antarmuka admin yang bersih, responsif, dan terstruktur tanpa banyak boilerplate—sehingga mempercepat pengembangan dengan komponen seperti form builder dan table builder (Den Harrin, 2021). Filament menyediakan komponen antarmuka siap pakai dan dapat dikustomisasi, memudahkan pengembang membuat dashboard untuk pengelolaan data, laporan, dan pengaturan aplikasi. Sebagai contoh, penggunaan form builder untuk input data, table builder untuk menampilkan data dengan filter dan aksi inline, serta sistem resource modular memungkinkan implementasi CRUD yang cepat dan efisien (Popoola, 2025).

Dalam konteks digitalisasi pelaporan dan pengecekan fasilitas, Filament memfasilitasi pembuatan dashboard interaktif yang memungkinkan pengguna melakukan pengecekan fasilitas, input laporan, dan akses data historis secara mudah. Paket ini menyederhanakan manajemen relasi antar entitas serta filtering data melalui API deklaratif yang intuitif (Den Harrin, 2021).

Komponen Filament antara lain:

1. *CRUD (Create, Read, Update, Delete) Generator*: Secara otomatis memproduksi modul manajemen data lengkap (index, create, edit, delete, filter) hanya melalui deklarasi model-resource sederhana (Popoola, 2025).
2. *Form dan Table Builder*: Komponen visual interaktif yang mendukung validasi, pagination, filter, dan aksi inline, memungkinkan pembuatan antarmuka admin dinamis tanpa perlu coding UI manual (Den Harrin, 2021).
3. *Relasi Data dan Filtering*: Mendukung relasi kompleks (HasMany, BelongsTo, Morphs) sekaligus filter deklaratif untuk membantu pengguna menavigasi data sesuai kebutuhan (Popoola, 2025).

2.2.3 MySQL

MySQL adalah salah satu sistem manajemen basis data relasional (RDBMS) yang paling populer dan banyak digunakan di dunia. MySQL berbasis *open-source* dan menggunakan bahasa SQL (*Structured Query Language*) sebagai bahasa utama untuk mengelola data. MySQL dikenal karena keandalannya, performanya yang tinggi, dan skalabilitasnya, sehingga sering digunakan dalam

pengembangan aplikasi *web*, termasuk yang berbasis Laravel (Šušter and Ranisavljević 2023).

Dalam pengembangan aplikasi, MySQL berfungsi sebagai tempat penyimpanan data seperti laporan dan hasil pengecekan fasilitas. Data disimpan dalam bentuk tabel yang saling berelasi melalui *primary key* dan *foreign key*. Keuntungan utama penggunaan MySQL meliputi kecepatan dan skalabilitas untuk menangani beban data besar, keamanan melalui enkripsi dan pengaturan akses pengguna, serta kompatibilitas tinggi dengan PHP dan Laravel (Yadav et al. 2024).

Fitur utama MySQL meliputi:

1. Sistem manajemen basis data relasional yang mendukung desain tabel berbasis relasi.
2. Kontrol transaksi seperti commit, rollback, dan savepoint untuk menjaga integritas data.
3. Kemampuan skalabilitas dan fleksibilitas yang dapat dioptimalkan untuk berbagai kebutuhan aplikasi (Yadav et al. 2024).

MySQL digunakan untuk menyimpan data fasilitas seperti laporan pengecekan Fire Alarm System Panel dan Fire Extinguisher Tube. Keunggulan integrasi antara Laravel dan MySQL melalui Eloquent ORM memungkinkan pengelolaan data menjadi lebih efisien dan intuitif, sehingga meningkatkan responsivitas dan stabilitas aplikasi (Singh, 2025).

2.2.4 Database

Database adalah kumpulan data yang disusun dan diorganisir secara sistematis untuk memudahkan penyimpanan, pengelolaan, dan pengambilan data.

Dalam sistem informasi, *database* memainkan peran penting sebagai fondasi yang menyimpan semua informasi yang dibutuhkan oleh aplikasi, termasuk data pengguna, laporan, dan transaksi. *Database* dapat diakses, dimodifikasi, dan dikelola melalui sistem manajemen basis data (DBMS), seperti MySQL, PostgreSQL, atau MongoDB.

Database dalam aplikasi berbasis web biasanya dikelola menggunakan database relasional (RDBMS), di mana data disimpan dalam tabel-tabel yang berelasi. Setiap tabel dalam database terdiri dari kolom (atribut) dan baris (*record*), yang menyimpan informasi tertentu. Relasi antar tabel didefinisikan melalui kunci primer (*primary key*) dan kunci asing (*foreign key*) yang menghubungkan tabel satu dengan yang lainnya.

Terdapat beberapa komponen penting dalam database, di antaranya:

1. **Tabel:** Struktur dasar untuk menyimpan data dalam bentuk baris dan kolom. Setiap tabel biasanya mewakili entitas tertentu, seperti pengguna, laporan, atau fasilitas.
2. **Primary Key:** Kunci unik yang mengidentifikasi setiap record dalam tabel. Setiap tabel memiliki satu primary key yang berbeda untuk mencegah duplikasi data.
3. **Foreign Key:** Kunci yang digunakan untuk menghubungkan satu tabel dengan tabel lainnya, menciptakan relasi antar data.
4. **Query:** Instruksi yang digunakan untuk mengakses, mengambil, atau memodifikasi data dalam database. *Query* dalam bahasa SQL

(Structured Query Language) mencakup perintah-perintah seperti *SELECT*, *INSERT*, *UPDATE*, dan *DELETE*.

Dalam konteks pengembangan aplikasi pelaporan dan pengecekan fasilitas di PT Philips Industri Batam, database digunakan untuk menyimpan berbagai data yang relevan seperti laporan pengecekan, kondisi fasilitas, data pengguna, dan histori kegiatan. Database memungkinkan data disimpan secara terstruktur dan terpusat, sehingga akses, pengelolaan, dan analisis data dapat dilakukan dengan lebih cepat dan akurat.

Manfaat utama menggunakan database dalam pengembangan aplikasi ini antara lain:

1. Penyimpanan Data Terstruktur: Database memastikan data tersimpan dengan format yang teratur, memudahkan pencarian dan manipulasi data.
2. Keamanan dan Integritas Data: Database dilengkapi dengan fitur keamanan seperti autentikasi pengguna, enkripsi, dan kontrol akses, yang menjaga kerahasiaan dan integritas data.
3. Pengambilan Data: Dengan *query* SQL yang efisien, data dapat diambil dan diproses dengan cepat, mendukung kebutuhan analisis dan pelaporan.

Dalam pengembangan aplikasi berbasis Laravel, database sering diintegrasikan dengan *framework* tersebut melalui *Eloquent ORM*, yang menyederhanakan interaksi dengan database, seperti pembuatan dan pengelolaan tabel, pengaturan relasi, dan penanganan *query*.

2.2.5 *Server*

Server adalah perangkat keras atau perangkat lunak yang menyediakan layanan kepada klien melalui jaringan. Dalam konteks aplikasi *web*, *server* adalah pusat pengelolaan yang menangani permintaan dari klien (biasanya *browser*) dan mengirimkan respons yang sesuai. *Server* menjalankan berbagai fungsi penting, seperti mengelola database, menjalankan logika bisnis, dan menyajikan konten web kepada pengguna. *Server* dapat dikategorikan menjadi beberapa jenis, antara lain:

1. *Web Server*: *Web server* bertanggung jawab untuk menangani permintaan HTTP dari klien dan mengirimkan konten seperti halaman HTML, file gambar, dan data lainnya. Contoh *web server* yang umum digunakan adalah Apache, Nginx, dan Microsoft IIS. Dalam aplikasi Laravel, *web server* seperti Apache atau Nginx digunakan untuk menjalankan aplikasi, menerima permintaan, dan mengirimkan respon ke klien.
2. *Database Server*: *Server* ini berfungsi untuk mengelola dan menyediakan akses ke database yang digunakan oleh aplikasi. MySQL, PostgreSQL, dan MongoDB adalah contoh database server yang sering digunakan dalam aplikasi web.
3. *Application Server*: *Application server* menangani logika bisnis dan menjalankan kode backend aplikasi. *Server* ini menerima permintaan dari web server dan memprosesnya menggunakan kode yang ditulis dalam berbagai bahasa pemrograman, seperti PHP, Python, atau Node.js. Dalam aplikasi berbasis Laravel, PHP berfungsi sebagai aplikasi server yang menjalankan logika bisnis.

2.2.6 Fungsi dan Peran Server dalam Aplikasi Web

Dalam arsitektur *client-server*, server berperan penting dalam menyediakan layanan dan sumber daya yang diperlukan oleh klien (Nyabuto 2023). Berikut adalah beberapa fungsi utama server dalam aplikasi web:

1. **Hosting Aplikasi:** Server menyimpan file dan kode aplikasi serta menyajikannya kepada klien yang mengaksesnya melalui browser.
2. **Pengolahan Data:** Server menangani proses logika bisnis, seperti autentikasi pengguna, pengelolaan data, dan pengolahan laporan.
3. **Manajemen Basis Data:** Server mengelola dan menyimpan data dalam database, serta memastikan integritas dan ketersediaan data.
4. **Keamanan dan Akses:** Server dilengkapi dengan fitur keamanan seperti firewall, SSL/TLS, dan pengelolaan izin untuk melindungi data dari akses yang tidak sah.

Dalam proyek digitalisasi plant activities di PT Philips Industri Batam, peran *server* menjadi fundamental dalam mengoperasikan aplikasi web berbasis Laravel. Unit *server* ini secara efisien memproses permintaan pengguna, berinteraksi dengan basis data MySQL untuk mengelola laporan serta data pengecekan fasilitas, dan menyajikan antarmuka administratif melalui *package* Filament. Umumnya, hosting aplikasi ini dilakukan pada lingkungan *server* yang berlandaskan sistem operasi Linux, seperti CentOS atau Ubuntu, dengan konfigurasi yang diatur sedemikian rupa untuk optimalisasi kinerja PHP dan *framework* Laravel.

2.2.7 QR Code

QR Code (Quick Response Code) adalah jenis kode matriks (barcode dua dimensi) yang dapat menyimpan informasi dalam bentuk teks, URL, atau data lain yang dapat diakses dengan cepat oleh pengguna (Alkhayyat et al. 2024). QR Code terdiri dari pola hitam putih yang diatur dalam kotak-kotak kecil, yang dapat dibaca menggunakan pemindai khusus atau kamera *smartphone*. Teknologi ini sangat populer karena kemudahan dan kecepatan aksesnya, terutama dalam aplikasi *mobile* dan *web* (Haridas Atmaram Tekale 2024).

2.3. Penelitian Terdahulu

Tabel 1.2 Penelitian Terdahulu

No	Nama	Thn	Judul	Metode	Hasil
1	Hasibuan, A., Sutrisno, S. P., & Nasution, S.	2024	SMART Manufacturi ng System: Sebuah Solusi Teknologi Manufaktur Proses Menuju Industri 4.0.	Deskriptif kualitatif melalui studi literatur	Jurnal ini membahas penerapan teknologi digital di lini produksi pengolahan, sejalan dengan digitalisasi sistem inspeksi fasilitas .

No	Nama	Thn	Judul	Metode	Hasil
2	Alvi Yuniar, Dadang Yusup & Purwantoro Purwantoro	2024	Rancang Bangun Smart Maintenance Menggunaka n Framework Laravel (Studi Kasus: PT Inti Ganda Perdana Karawang)	SDLC	Sistem berhasil dibangun dan diuji secara menyeluruh. Aplikasi ini terbukti dapat meningkatkan efisiensi, transparansi, dan kecepatan pelaporan maintenance fasilitas. Oleh karena itu, sistem direkomendasikan untuk diimplementasika n secara penuh di perusahaan.
3	Raptis, T. P., Passarella,	2019	Data Management in Industry	Systematic Literature Review (SLR)	Memaparkan tantangan manajemen data

No	Nama	Thn	Judul	Metode	Hasil
	A., & Conti, SsM.		4.0: State of the Art and Open Challenges.		di lingkungan Industri 4.0, penting bagi sistem monitoring real-time.
4	Faturrohman, F.	2023	Rancang Bangun Aplikasi Sistem Informasi Pengelola Inventaris Berbasis Laravel	Waterfall.	Aplikasi dapat mencatat seluruh barang inventaris secara digital, menampilkan data dalam bentuk tabel interaktif, serta membantu proses audit dengan laporan digital.
5	Muharam, Y. & Hanifah, W.	2024	Rancang Bangun Aplikasi Tracking Sample Berbasis	SDLC Waterfall	Aplikasi memudahkan pelacakan status sample pakaian dari proses awal hingga

No	Nama	Thn	Judul	Metode	Hasil
			Laravel di PT Kahatex		pengiriman, serta mengurangi kehilangan data dan mempercepat pelaporan kepada klien.
6	Robbani, A. A., Wiyono, B. H., & Haromain, I.	2025	Rancang Bangun Aplikasi Sistem Gudang Berbasis Web Menggunaka n Laravel dengan Agile	Agile	Aplikasi berhasil mengelola stok barang secara efisien, memudahkan pemantauan barang masuk dan keluar, serta menghasilkan laporan otomatis berbasis data real-time.
7	Lai, A. & Ramadhan, I.	2024	Rancang Bangun Aplikasi Manajemen	Waterfall	Aplikasi berhasil meningkatkan koordinasi antar tim proyek,

No	Nama	Thn	Judul	Metode	Hasil
			Proyek Berbasis Web Menggunakan Laravel dan Livewire		menyediakan fitur pencatatan progres dan pengelolaan tugas, serta mempercepat alur persetujuan proyek.
8	Ramadani, S.	2023	Sistem Informasi Manajemen Aset Berbasis Web Menggunakan Framework Laravel di IT Support PT Biznetnetworks	Pengembangan sistem berbasis web menggunakan Laravel.	Sistem informasi manajemen aset berbasis web yang lebih efisien dan akurat.
9	Fariz, S.	2022	Rancang Bangun	Research and Development	Sistem manajemen aset

No	Nama	Thn	Judul	Metode	Hasil
			Sistem Informasi Manajemen Aset Alat Kantor Berbasis Website menggunakan Framework Laravel dan Metode LIFO	(R&D), wawancara, dokumentasi, desain database, UML, implementasi metode LIFO.	berbasis web dengan Laravel dan LIFO, meningkatkan efisiensi dan akurasi.
10	AWALUDIN, M. I., ARIFIN, R. W., & SETIYADI, D.	2020	Implementasi Framework Laravel Pada Sistem Informasi Pengelolaan Aset Laboratorium Komputer	RAD (Requirements Planning, Design Workshop, Implementation), PHP, Laravel, Bootstrap, MySQL.	Sistem informasi pengelolaan aset yang akurat, rapi, mengurangi kehilangan data, pencarian efisien, pengelolaan mudah.

No	Nama	Thn	Judul	Metode	Hasil
11	Mohammad Roshandelpoor.	2024	Monitoring Laravel Pulse	Alat sumber terbuka dalam ekosistem Laravel, konfigurasi melalui file, <i>custom recorders/wid gets.</i>	Alat pemantauan kinerja aplikasi Laravel (CPU, memori, permintaan, kueri lambat, pengecualian, dll.) untuk produksi.
12	Muhammad Iqbal Awaludin, Rita Wahyuni Arifin.	2020	PERANCA NGAN APLIKASI ASET MANAJEM EN MENGGUN AKAN FRAMEWO RK LARAVEL DI PT DIRGANTA	Waterfall (Requirements , Design, Coding, Testing, Maintenance), UML, Black Box Testing.	Sistem manajemen aset berbasis web dengan fitur pencatatan, peminjaman, pemantauan, laporan digital (PDF/Excel).

No	Nama	Thn	Judul	Metode	Hasil
			RA INDONESIA A (IAe)		
13	Said Muhammad , Riswan Yunida	2024	Implementasi Sistem Informasi Manajemen Inventaris Berbasis WEB Menggunakan Framework Laravel (Studi Kasus Pada UPT. ICT Pomosda, Tanjunganm, Nganjuk)	Waterfall (analisis, desain, pembuatan, pengujian, pemeliharaan)	Aplikasi manajemen inventaris berbasis web dengan fasilitas input, cari, laporan; berjalan baik di berbagai browser.
14	Ria, A. N. P. L. & Hesinto, S.	2023	Rancang Bangun Aplikasi	Rancang Bangun	Aplikasi inventaris manajemen aset

No	Nama	Thn	Judul	Metode	Hasil
			Inventory IT Manajemen Aset Berbasis Web pada PT. Titis Sampurna Prabumulih	aplikasi berbasis web.	TI berbasis web untuk efisiensi pengelolaan aset.
15	Naibaho, S., Maulani, M. R., & Hamidin, D.	2024	Rancang Bangun Sistem Manajemen Informasi Aset Menggunaka n Laravel (Studi Kasus: Universitas XYZ)	Rancang Bangun sistem manajemen informasi aset.	Implementasi sistem diharapkan berkontribusi signifikan pada pengelolaan aset.

(Sumber: Penelitian, 2025)