

BAB II

LANDASAN TEORI

2.1. Konsep Sistem Informasi

2.1.1. Pengertian Sistem

Sistem adalah kumpulan/grup dari sub sistem/bagian/komponen apapun baik fisik ataupun non fisik yang saling berhubungan satu sama lain dan bekerjasama secara harmonis untuk mencapai satu tujuan tertentu (Rusdianto, 2024). Beberapa prinsip umum sistem ini adalah sebagai berikut :

1. Sistem selalu merupakan bagian sistem lebih besar. Sistem dapat dipartisi menjadi sub sistem yang lebih kecil.
2. Sistem lebih terspesialisasi dan kurang dapat beradaptasi untuk menghadapi keadaan-keadaan yang berbeda.
3. Lebih besar ukuran sistem maka akan memerlukan lebih banyak sumberdaya untuk operasi dan pemeliharaan.
4. Sistem senantiasa mengalami perubahan, tumbuh dan berkembang

Batas sistem kadang sering tidak jelas terutama untuk sistem perangkat lunak. Jika *user* memandang manusia sebagai sistem, batasnya jelas karena badan manusia terpisah dari manusia lain dan lingkungan secara jelas. Serupa itu adalah mobil, rumah, dan sistem fisik lain. Tidak demikian untuk sistem perangkat lunak atau sistem berbasis komputer.

2.1.2. Pengertian Informasi

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat

ini atau saat mendatang, (Muhajir et al., 2024).

Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Bahan bakunya adalah data yaitu sekumpulan fakta-fakta dari suatu peristiwa atau kejadian yang belum mempunyai arti., (Lusiana & Salam, 2024), pendapat lain mengatakan Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan, (Sujarwo et al., 2023).

Sistem informasi adalah suatu kombinasi teratur apapun baik dari *people*, *hardware*, *software*, maupun *database* yang mengumpulkan, mengubah, dan menyebarkan informasi di dalam suatu bentuk organisasi, (Firman et al., 2025).

2.1.3. Pengertian Sistem Informasi

Sistem Informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, sistem informasi istilah yang sering digunakan untuk merujuk pada interaksi antara orang, proses algoritmik, data, dan teknologi, (Anwar & Padiku, 2024).

Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya untuk penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi dalam mendukung proses bisnis.

Dengan demikian, saling terkait sistem informasi dengan sistem data pada satu sisi dan pada aktivitas sistem lainnya. Sistem informasi adalah suatu bentuk sistem komunikasi di mana data direpresentasikan dan diproses sebagai bentuk memori sosial. Sistem informasi juga dapat dianggap sebagai bahasa semi formal yang mendukung orang dalam pengambilan keputusan dan tindakan, (Chaniago et al., 2024) :

1. Sistem informasi merupakan fokus utama dari studi untuk disiplin sistem informasi dan organisasi informatika.
2. Sistem ini diselenggarakan informasi dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi dan sumber data untuk mengumpulkan, mengubah, dan mendistribusikan informasi dalam sebuah organisasi.
3. Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, kegiatan manajerial dan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

2.2. Perancangan

Perancangan adalah proses merencanakan, merancang, atau membuat suatu sistem, produk, atau kegiatan secara terstruktur dan rinci sebelum diwujudkan dalam bentuk nyata, bertujuan untuk memecahkan masalah, memenuhi kebutuhan, dan mencapai tujuan tertentu dengan melibatkan analisis, evaluasi, serta deskripsi komponen dan batasan yang diperlukan, seringkali setelah tahap analisis (Terezian & Fatmawati, 2024).

Menurut (Sumaryanto et al., 2024) perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya.

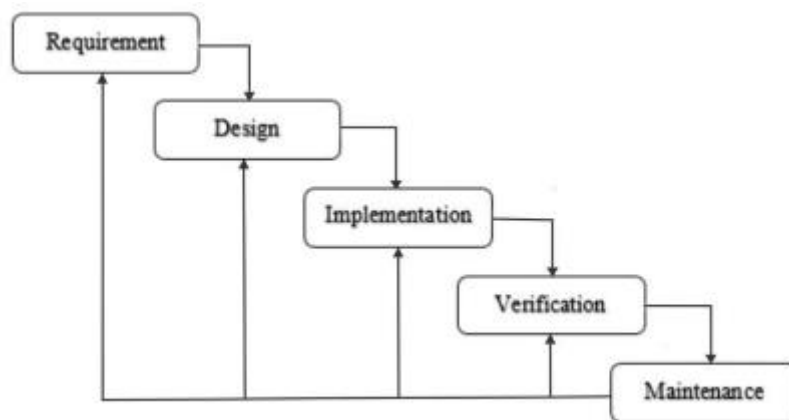
Berdasarkan beberapa pendapat diatas dapat disimpulkan bahwa perancangan adalah tahapan setelah analisis sistem yang tujuannya untuk menghasilkan rancangan yang memenuhi kebutuhan yang ditentukan selama tahap analisis.

2.2.1. Metode *Waterfall*

Penelitian yang dilakukan dalam pengembangan website praktik dr. Dewi Marbau, Kabupaten Labuhanbatu menggunakan *framework Laravel* mengadopsi metode *SDLC (Software Development Life Cycle)* dengan model *Waterfall*. Model ini, sering disebut sebagai pendekatan linier berurutan atau siklus hidup klasik, menerapkan proses pengembangan perangkat lunak secara bertahap. Dimulai dari tahap analisis, diikuti oleh desain sistem, pembuatan kode, pengujian, hingga tahap pemeliharaan atau dukungan sistem, (Azizah et al., 2024).

System Development Life Cycle (SDLC) yang berfokus pada serangkaian langkah sistematis untuk menciptakan dan memelihara perangkat lunak. Proses ini dimulai dengan fase perencanaan, di mana tujuan dan kebutuhan sistem diidentifikasi secara mendetail. Selanjutnya, fase analisis dilakukan untuk mengevaluasi dan mendokumentasikan kebutuhan

tersebut, diikuti oleh fase desain yang melibatkan pembuatan arsitektur sistem dan spesifikasi teknis. Setelah desain disetujui, fase implementasi dimulai dengan pengkodean dan pembangunan sistem. Fase pengujian dilakukan untuk memastikan bahwa sistem berfungsi sesuai harapan dan bebas dari cacat. Akhirnya, fase pemeliharaan mengaddress masalah yang mungkin timbul setelah sistem dioperasikan, serta melakukan pembaruan atau perbaikan jika diperlukan. Model *SDLC* yang digunakan dalam penelitian ini adalah model *Waterfall*, yang mengikuti pendekatan linier dan bertahap, memastikan bahwa setiap tahap diselesaikan secara menyeluruh sebelum melanjutkan ke tahap berikutnya. Tahapan dalam perencanaan berbasis web dengan metode *SDLC* diantaranya sebagai berikut : (Suraharta, 2024).



Gambar 2.1. Ilustrasi Pengembangan Model *Waterfall*

Penelitian yang dilakukan dalam pengembangan model *waterfall* dapat dijelaskan sebagai berikut :

1. Tahapan Requirement

Tahap pertama adalah Requirement, yaitu proses pengumpulan dan analisis kebutuhan sistem dari pengguna. Pada tahap ini

pengembang berusaha memahami apa saja yang dibutuhkan oleh pengguna, seperti fungsi sistem, data yang akan dikelola, dan tujuan penggunaan sistem, sehingga dihasilkan dokumen kebutuhan yang menjadi dasar pengembangan.

2. Tahapan *Design*

Pada tahap ini kebutuhan yang sudah dikumpulkan diterjemahkan ke dalam bentuk rancangan sistem. Rancangan ini meliputi struktur database, alur proses, arsitektur sistem, serta desain tampilan antarmuka. Tujuan dari tahap desain adalah memberikan gambaran yang jelas tentang bagaimana sistem akan dibangun sehingga memudahkan proses pengembangan pada tahap selanjutnya.

3. Tahapan *Implementation*

Tahap di mana rancangan sistem mulai diwujudkan menjadi program nyata. Pada fase ini programmer menulis kode program, membuat database, dan mengembangkan seluruh fungsi sesuai dengan desain yang telah dibuat. Hasil dari tahap ini adalah sebuah sistem yang sudah dapat dijalankan meskipun belum sepenuhnya diuji.

4. Tahapan *Verification* atau Pengujian

Pada tahap ini sistem diuji untuk memastikan bahwa semua fungsi berjalan sesuai dengan kebutuhan yang telah ditentukan pada tahap awal. Pengujian dilakukan untuk menemukan kesalahan, memastikan keakuratan data, dan mengecek apakah sistem sudah siap digunakan oleh pengguna.

5. Tahapan terakhir adalah *Maintenance*

Tahap pemeliharaan sistem setelah digunakan. Pada fase ini dilakukan perbaikan jika ditemukan kesalahan, penyesuaian jika ada perubahan kebutuhan, serta pengembangan tambahan agar sistem tetap berjalan dengan baik dan relevan. Dengan adanya tahap ini, sistem dapat terus digunakan dalam jangka panjang dan menyesuaikan dengan kebutuhan pengguna.

2.3. Data Pasien

Data pasien adalah kumpulan informasi lengkap yang berkaitan dengan identitas, kondisi kesehatan, serta riwayat pelayanan medis yang dimiliki oleh seseorang sebagai pengguna layanan kesehatan. Data ini tidak hanya mencakup data pribadi seperti nama, nomor rekam medis, tanggal lahir, jenis kelamin, alamat, dan nomor kontak, tetapi juga mencakup informasi medis seperti keluhan pasien, hasil pemeriksaan, diagnosis penyakit, tindakan medis yang diberikan, resep obat, hasil laboratorium, serta riwayat kunjungan ke fasilitas kesehatan (Ilman, 2024).

Data pasien memiliki peranan yang sangat penting dalam dunia pelayanan kesehatan karena menjadi dasar utama bagi dokter dan tenaga medis dalam menentukan diagnosis dan pengobatan yang tepat. Dengan adanya data pasien yang lengkap dan akurat, tenaga kesehatan dapat memantau perkembangan kondisi pasien dari waktu ke waktu, mengetahui riwayat penyakit yang pernah diderita, serta menghindari kesalahan dalam pemberian tindakan atau obat. Selain itu, data pasien juga berfungsi sebagai dokumentasi resmi yang mencatat seluruh proses pelayanan kesehatan yang

telah diberikan kepada pasien.

Dalam sistem informasi kesehatan, data pasien disimpan dan dikelola secara terstruktur sehingga mudah diakses, diperbarui, dan dilindungi keamanannya. Pengelolaan data pasien yang baik membantu meningkatkan kualitas pelayanan, mempercepat proses administrasi, serta mendukung pengambilan keputusan medis dan manajerial. Oleh karena itu, data pasien merupakan elemen penting yang harus dijaga kerahasiaannya dan keakuratannya demi keselamatan dan kenyamanan pasien, (Maharani, 2025).

2.3.1. Pemahaman Sistem Informasi Data Pasien

Pemahaman sistem informasi data pasien adalah pemahaman tentang bagaimana suatu sistem berbasis teknologi digunakan untuk mengelola, menyimpan, mengolah, dan menyajikan seluruh data pasien secara terintegrasi dan terstruktur. Sistem ini dirancang untuk membantu fasilitas kesehatan, seperti klinik atau praktik dokter, dalam mencatat identitas pasien, riwayat pemeriksaan, diagnosis, tindakan medis, resep obat, hingga hasil layanan yang diberikan kepada pasien secara digital.

Melalui sistem informasi data pasien, proses pencatatan yang sebelumnya dilakukan secara manual dapat digantikan dengan sistem komputerisasi sehingga data menjadi lebih rapi, mudah dicari, dan lebih aman. Setiap kali pasien datang berobat, petugas atau dokter dapat dengan cepat mengakses rekam medis pasien, melihat riwayat penyakit, serta mengetahui pengobatan yang pernah diberikan. Hal ini sangat membantu dalam meningkatkan ketepatan diagnosis dan kualitas pelayanan kesehatan.

Selain mendukung pelayanan medis, sistem informasi data pasien juga berperan penting dalam kegiatan administrasi dan manajemen. Data yang tersimpan dapat digunakan untuk membuat laporan jumlah pasien, jenis penyakit yang sering muncul, penggunaan obat, dan kinerja pelayanan. Dengan demikian, sistem ini tidak hanya membantu dokter dalam memberikan perawatan, tetapi juga membantu pihak pengelola dalam mengambil keputusan dan meningkatkan efisiensi operasional.

2.3.2. Kegunaan Sistem Informasi Data Pasien

Kegunaan sistem informasi data pasien sangat penting dalam menunjang pelayanan dan pengelolaan fasilitas kesehatan. Sistem ini digunakan untuk menyimpan dan mengelola seluruh data pasien secara terkomputerisasi sehingga informasi seperti identitas pasien, riwayat pemeriksaan, diagnosis, tindakan medis, dan resep obat dapat dicatat dengan rapi, aman, dan mudah diakses. Dengan adanya sistem ini, tenaga medis tidak perlu lagi mencari berkas secara manual karena data dapat ditampilkan dengan cepat melalui komputer (Hasrul & Siregar, 2025).

Selain itu, sistem informasi data pasien membantu meningkatkan kualitas pelayanan kesehatan. Dokter dan perawat dapat melihat riwayat kesehatan pasien secara lengkap sehingga dapat memberikan diagnosis dan pengobatan yang lebih tepat. Sistem ini juga mengurangi risiko kesalahan pencatatan dan kehilangan data, yang sering terjadi pada sistem manual berbasis kertas.

Dari sisi manajemen, sistem informasi data pasien sangat berguna untuk menghasilkan laporan dan informasi yang dibutuhkan oleh pihak pengelola,

seperti jumlah kunjungan pasien, jenis penyakit yang paling sering muncul, serta penggunaan obat. Data tersebut dapat digunakan sebagai dasar dalam pengambilan keputusan, perencanaan layanan, dan peningkatan mutu pelayanan. Dengan demikian, sistem informasi data pasien tidak hanya mendukung kegiatan medis, tetapi juga membantu meningkatkan efisiensi dan profesionalisme pengelolaan layanan kesehatan (Pranolo, 2023).

2.4. *Unified Modeling Language (UML)*

Menurut (Ginting, 2025), menjelaskan bahwa “*UML (Unified Modeling Language)* adalah salah standar Bahasa yang banyak digunakan didunia industry untuk mendefinisikan requirement, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”. Berikut *UML* yang akan digunakan dalam penulisan berikut :

2.4.1. *Usecase Diagram*

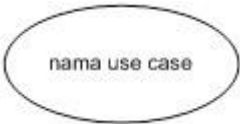
Usecase adalah interaksi antar aktor eksternal dan sistem hasil yang diamati oleh aktor berorientasi pada tujuan, dideskripsikan ada diagram *usecase* dan teks. Menurut (Ginting, 2025). *Usecase Diagram* merupakan salah satu diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan interaksi antara pengguna (aktor) dengan sistem yang akan dibangun. Diagram ini berfungsi untuk menunjukkan fungsi-fungsi utama yang disediakan oleh sistem serta bagaimana aktor berinteraksi dengan fungsi tersebut. Dalam penelitian ini, *Usecase Diagram* digunakan untuk memodelkan proses yang terjadi dalam Sistem Informasi Manajemen Data Pasien pada praktik dokter umum dr. Dewi Marbau, seperti pengelolaan data pasien, pencatatan rekam medis, pengaturan jadwal

pemeriksaan, serta pembuatan laporan.

Usecase Diagram membantu pengembang sistem dan pihak pengguna, dalam hal ini dokter, petugas administrasi, dan pasien, untuk memahami kebutuhan sistem secara menyeluruh. Melalui diagram ini, setiap fungsi sistem dapat diidentifikasi dengan jelas sehingga memudahkan proses perancangan, pengembangan, dan pengujian sistem.

Untuk memudahkan pemahaman dalam membaca *Usecase Diagram*, digunakan beberapa simbol standar *UML* yang memiliki makna tertentu. Simbol-simbol ini berfungsi untuk menunjukkan aktor, aktivitas yang dilakukan sistem, serta hubungan antar komponen dalam diagram. Simbol-simbol yang digunakan dalam *Usecase Diagram* pada penelitian ini dapat dilihat pada Tabel 2.1 berikut :

Tabel 2.1. *Symbol Usecase Diagram*

Simbol	Deskripsi
<i>Use Case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
<i>Aktor/actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase

	nama aktor.
Asosiasi/ <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi/ <i>extend</i> <<extend>> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
Menggunakan/ <i>include</i> <<include>> 	<i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan

Sumber : (Saputra et al., 2024).

2.4.2. Activity Diagram

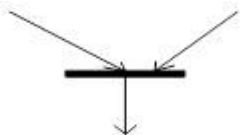
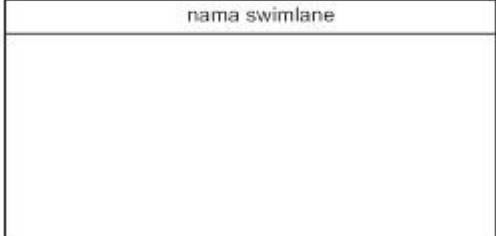
Activity Diagram merupakan salah satu diagram *UML* yang digunakan untuk menggambarkan alur aktivitas atau proses kerja yang terjadi di dalam sistem (Ginting, 2025). Diagram ini menunjukkan urutan kegiatan dari awal hingga akhir, termasuk proses percabangan dan aktivitas yang berjalan secara paralel. Dalam Sistem Informasi Manajemen Data Pasien pada praktik dokter umum dr. Dewi Marbau, *Activity Diagram* digunakan untuk memodelkan alur kerja seperti proses pendaftaran pasien, pemeriksaan pasien, penginputan rekam medis, hingga pembuatan laporan.

Penggunaan *Activity Diagram* sangat penting untuk memahami bagaimana suatu proses dijalankan di dalam sistem secara berurutan dan sistematis. Dengan adanya diagram ini, pengembang dan pengguna sistem dapat melihat dengan jelas bagaimana suatu aktivitas dimulai, diproses, hingga menghasilkan *output* akhir. Selain itu, *Activity Diagram* juga membantu dalam mengidentifikasi potensi hambatan atau kesalahan dalam alur proses yang sedang berjalan.

Untuk menggambarkan alur aktivitas tersebut secara visual, digunakan berbagai simbol standar *UML* yang masing-masing memiliki fungsi tertentu. Simbol-simbol ini menunjukkan status awal dan akhir aktivitas, jenis kegiatan yang dilakukan, alur perpindahan aktivitas, serta pihak yang bertanggung jawab terhadap setiap proses. Simbol-simbol yang digunakan dalam *Activity Diagram* pada penelitian ini ditampilkan pada Tabel 2.2 berikut :

Tabel 2.2. *Symbol Activity Diagram*

Simbol	Deskripsi
<i>Initial State</i> 	Status awal aktivitas sistem
<i>Activity</i> 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
<i>Control Flow</i> 	Urutan perpindahan suatu aktivitas
<i>Decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu

<i>Final State</i> 	Status akhir yang dilakukan sistem
<i>Fork</i> 	Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel
<i>Join</i> 	Digunakan untuk menunjukkan kegiatan yang digabungkan
<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber : (Saputra et al., 2024).

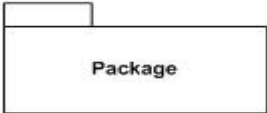
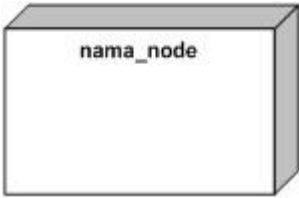
2.4.3. Deployment Diagram

Deployment Diagram merupakan salah satu diagram dalam *UML* yang digunakan untuk menggambarkan bagaimana komponen perangkat lunak ditempatkan dan dijalankan pada infrastruktur perangkat keras. Diagram ini menunjukkan hubungan antara perangkat keras (*server*, komputer, dan perangkat lain) dengan perangkat lunak yang digunakan oleh sistem (Ginting, 2025). Dalam Sistem Informasi Manajemen Data Pasien pada praktik dokter umum dr. Dewi Marbau, *Deployment Diagram* berfungsi untuk memodelkan bagaimana aplikasi berbasis *PHP* dan database *MySQL* diinstal dan dijalankan pada *server* serta diakses oleh pengguna melalui komputer atau perangkat lain.

Deployment Diagram membantu dalam memahami arsitektur fisik sistem, termasuk lokasi penyimpanan data, jalur komunikasi, dan ketergantungan antar perangkat. Dengan adanya diagram ini, dapat diketahui bagaimana sistem bekerja secara teknis mulai dari pengguna yang mengakses aplikasi hingga data tersimpan di dalam basis data. Hal ini penting untuk memastikan sistem berjalan dengan stabil, aman, dan efisien dalam mendukung kegiatan pelayanan kesehatan di praktik dr. Dewi Marbau.

Untuk merepresentasikan struktur fisik sistem tersebut, digunakan beberapa simbol standar *UML* yang masing-masing memiliki makna tertentu. Simbol-simbol ini digunakan untuk menggambarkan node, hubungan antar node, serta paket komponen yang dijalankan pada setiap node. Simbol-simbol yang digunakan dalam *Deployment Diagram* pada penelitian ini disajikan pada Tabel 2.3 berikut :

Tabel 2.3 *Symbol Deployment Diagram*

Simbol	Deskripsi
<i>Package</i> 	<i>Package</i> merupakan sebuah bungkusan dari satu atau lebih <i>node</i>
<i>Node</i> 	Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak (<i>software</i>) yang tidak dibuat sendiri, jika didalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen

Kebergantungan/ <i>dependency</i> 	Kebergantungan antara <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
<i>Link</i> 	Relasi antar <i>node</i>

Sumber : (Saputra et al., 2024).

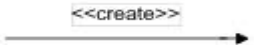
2.4.4. Sequence Diagram

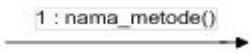
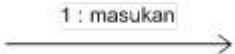
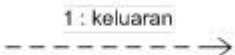
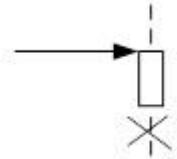
Sequence Diagram merupakan salah satu diagram dalam *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan urutan interaksi antar objek atau aktor dalam suatu sistem berdasarkan alur waktu (Ginting, 2025). Diagram ini menunjukkan bagaimana pesan atau data dikirim dan diterima oleh setiap entitas yang terlibat, mulai dari proses awal hingga proses akhir. Dalam Sistem Informasi Manajemen Data Pasien pada praktik dokter umum dr. Dewi Marbau, *Sequence Diagram* digunakan untuk memodelkan interaksi antara pasien, admin, sistem, dan dokter dalam proses pelayanan, seperti pendaftaran pasien, penyampaian keluhan, pemeriksaan dokter, hingga penyimpanan rekam medis.

Penggunaan *Sequence Diagram* sangat penting untuk memahami bagaimana komunikasi antar aktor dan sistem berlangsung secara berurutan dan terstruktur. Dengan adanya diagram ini, pengembang dan pengguna sistem dapat melihat secara jelas siapa yang memulai suatu proses, bagaimana data diproses oleh sistem, serta bagaimana hasilnya dikirim kembali kepada pihak yang membutuhkan. *Sequence Diagram* juga membantu dalam memastikan bahwa setiap proses berjalan sesuai dengan urutan yang benar sehingga dapat meminimalkan kesalahan dalam alur pelayanan pasien.

Untuk menggambarkan interaksi tersebut secara visual, Sequence Diagram menggunakan berbagai simbol standar UML yang masing-masing memiliki fungsi tertentu. Simbol-simbol ini menunjukkan aktor atau objek yang terlibat, garis waktu (lifeline), pesan yang dikirim, serta urutan proses yang terjadi di dalam sistem. Simbol-simbol yang digunakan dalam *Sequence Diagram* pada penelitian ini disajikan dalam Tabel 2.4 :

Tabel 2.4 *Symbol Sequence Diagram*

Simbol	Deskripsi
Aktor  nama aktor Atau  nama aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor
Garis Hidup/<i>Lifeline</i> 	Menyatakan kehidupan suatu objek
Objek  nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya
Pesan Tipe <i>Create</i> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat

Pesan Tipe <i>Call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
Pesan Tipe <i>Send</i> 	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan Tipe <i>Return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima
Pesan Tipe <i>Destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destory</i>

Sumber : (Saputra et al., 2024).

2.4.5. Entity Relationship Diagram

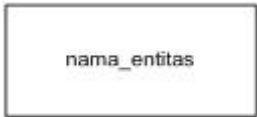
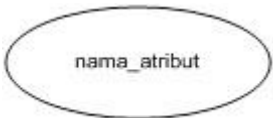
Entity Relationship Diagram (ERD) merupakan salah satu model konseptual dalam perancangan basis data yang digunakan untuk menggambarkan struktur data dan hubungan antar entitas dalam suatu sistem (Ginting, 2025). *ERD* menunjukkan bagaimana data disusun, disimpan, serta saling berhubungan satu sama lain melalui relasi tertentu. Dalam Sistem Informasi Manajemen Data Pasien pada praktik dokter umum dr. Dewi Marbau, *ERD* digunakan untuk memodelkan hubungan antara entitas seperti pasien, dokter, rekam medis, pemeriksaan, dan pengguna sistem.

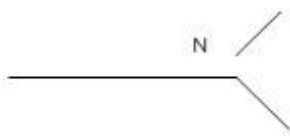
Penggunaan *ERD* sangat penting untuk memastikan bahwa

perancangan basis data sesuai dengan kebutuhan sistem dan mampu menyimpan informasi secara terstruktur dan konsisten. Dengan adanya *ERD*, pengembang dapat memahami bagaimana data pasien disimpan, bagaimana data pemeriksaan dicatat, serta bagaimana rekam medis terhubung dengan pasien dan dokter. Selain itu, *ERD* juga membantu menghindari terjadinya duplikasi data dan kesalahan dalam penyimpanan informasi, sehingga kualitas data dalam sistem dapat terjaga.

Untuk menggambarkan struktur data tersebut secara visual, *ERD* menggunakan simbol-simbol khusus seperti entitas, atribut, *primary key*, dan relasi. Simbol-simbol ini menunjukkan jenis data yang disimpan, kunci utama setiap entitas, serta hubungan antar tabel dalam basis data. Simbol-simbol yang digunakan dalam *ERD* pada penelitian ini disajikan pada Tabel 2.5 :

Tabel 2.5 *Symbol Entity Relationship Diagram*

Simbol	Deskripsi
Entitas/<i>Entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas

Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi/association 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian

Sumber : (Saputra et al., 2024).

2.5. Pengertian PHP (*Personal Home Page*)

PHP adalah bahasa *scripting server-side* bagi pemograman *website*. Secara sederhana, *PHP* merupakan tool bagi pengembang website dinamis. *PHP* sangat populer karena fungsi built-in lengkap, cepat, mudah dipelajari, dan bersifat gratis. Script *PHP* cukup disisipkan pada kode *HTML* agar dapat bekerja. *PHP* dapat berjalan di berbagai web server dan sistem informasi yang berbeda. *HTTP* (*hypertext transfer protocol*) merupakan protokol yang digunakan untuk mentransfer data antara *web server* ke *web browser*. Protokol ini mentransfer data- data *web* yang ditulis atau berformat *HTML*.

HTML berfungsi untuk memformat *file* dokumen teks biasa untuk bisa ditampilkan pada *web browser* sesuai keinginan. Hal tersebut dapat dilakukan dengan menambah elemen atau sering disebut sebagai *tag-tag*. Elemen *HTML* biasanya berupa *tag-tag* yang berpasangan dan setiap *tag* ditandai dengan simbol “<” dan “>” tanpa tanda kutip. Pasangan dari sebuah *tag* ditandai dengan tanda “/” dan ini menunjukkan penutupan suatu *tag*.

2.6. Pengertian *Database*

Database merupakan kumpulan dari banyak data yang saling berkaitan. *Database* memiliki relasi yang berbeda antar tabel dan memiliki fungsi yang berbeda-beda. Dengan adanya *database*, sistem informasi akan mudah dalam mengklasterkan data sesuai fungsinya. Dengan adanya *database*, sistem informasi dapat dengan mudah mencari informasi terkait apapun secara cepat. *Database* juga dapat dimodifikasi dengan mudah sesuai perubahan yang mungkin dapat terjadi suatu waktu dengan cepat dan mudah, (Firman et al., 2025)

Basis data (*database*) adalah kumpulan data yang terstruktur secara sistematis dan tersimpan di suatu tempat agar dapat diakses, dimanipulasi, dan dikelola dengan mudah. Basis data digunakan untuk menyimpan informasi yang terkait satu sama lain dan dapat diakses oleh sistem atau aplikasi perangkat lunak (Hasrul & Siregar, 2025).

2.6.1. Pengertian *MySQL*

MySQL adalah perangkat lunak *database server* atau disebut *Database Smart. Database* ini semakin lama semakin populer. Dengan menggunakan *database* ini, data semakin aman dan berdaya guna. *Database* ini juga banyak dipakai pada *web database* sehingga data semakin terintegrasi antara *database desktop* dengan *database web*.

MySQL adalah sebuah perangkat lunak system manajemen basis data *SQL* (bahasa Inggris: *database management system*) atau *DBMS* yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi di seluruh dunia. *MySQL AB* membuat *MySQL* tersedia sebagai perangkat lunak gratis

dibawah lisensi *GNU General Public License (GPL)*, tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan *GPL*.

MySQL adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis dibawah lisensi *GPL (General Public License)*. Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu *SQL (Structured Query Language)*. *SQL* adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis, (Rafi et al., 2024). Keandalan suatu sistem database (*DBMS*) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah *SQL*, yang dibuat oleh user maupun program-program aplikasinya.

Sebagai *database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam query data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase*.

2.7. XAMPP

Pengertian *XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Fungsi *XAMPP* adalah sebagai server yang berdiri sendiri (*localhost*), terdiri

atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang (Felmidi et al., 2025).



Gambar 2.2. XAMPP
Sumber : (Budiana et al., 2023)

2.8. Tools yang digunakan

2.8.1. Pemrograman Web dengan Framework Laravel

Laravel adalah sebuah *framework PHP* yang dirilis dibawah lisensi *MIT*, dibangun dengan konsep *MVC (model view controller)*. *Laravel* adalah pengembangan *website* berbasis *MVP* yang ditulis dalam *PHP* yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu. *MVC* adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. *MVC* memisahkan aplikasi berdasarkan komponen-komponen aplikasi, seperti : manipulasi data, controller, dan *user interface*, (Januarisman, 2025). Framework adalah kerangka kerja.

Framework digunakan oleh *developer* untuk memudahkan pembangunan aplikasi *web* yang dapat berupa sekumpulan *library* yang berisi fungsi, tools, ataupun class-class, dan digunakan sebagai kerangka dalam pembangunan aplikasi *web*. Umumnya didalam *framework* tersebut telah menyediakan solusi untuk akses *database*, *authentication*, *templating*, *controls*, dan fungsi-fungsi lainnya. Penggunaan *framework* diharapkan membuat pengembangan aplikasi menjadi rapi dan bersih, memiliki struktur yang optimal, dan *reusable*. Struktur aplikasi yang lebih rapi dan teratur, biasanya menggunakan struktur *MVC* yang melakukan pemisahan antara *business logic* dengan *presentation*. Pemisahan dilakukan dengan tujuan agar setiap perubahan yang terjadi pada *presentation logic* atau *business logic* tidak memberikan pengaruh satu sama lainnya yang kompleks.

2.9. Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai bahan referensi dalam penyusunan penelitian yang mendorong penulis untuk melakukan penelitian yang sama. Penelitian terdahulu yang digunakan adalah penelitian yang memiliki relevansi dengan tema yang diangkat yaitu mengenai Perancangan Dan Implementasi Sistem Informasi Manajemen Data Pasien Menggunakan *PHP* Dan *MYSQL* Pada Praktek Dokter Umum dr. Dewi Marbau. Beberapa bahan yang penulis ambil dari penelitian terdahulu adalah sebagai berikut :

1. Penelitian yang dilakukan oleh (M.A.Abdillah, 2025), yang berjudul “Sistem Informasi Rawat Jalan Klinik Cahaya Putera Siti”. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan dapat mempercepat proses registrasi, meningkatkan akurasi data, serta

menghasilkan laporan yang lebih terstruktur. Dengan adanya otomatisasi alur administrasi dan integrasi data pasien, sistem ini dapat mengurangi beban kerja staf klinik dan meningkatkan kualitas layanan. Secara keseluruhan, sistem informasi rawat jalan ini diharapkan menjadi solusi praktis untuk mengoptimalkan efisiensi operasional dan mendukung pelayanan kesehatan yang lebih baik di Klinik Cahaya Putera Siti.

2. Penelitian yang dilakukan oleh (Indarto et al., 2023), yang berjudul “Perancangan Sistem Informasi Pengelolaan Administrasi Pada Klinik Pratama Mardi Waluyo”, dengan hasil penelitian endekatan prototipe digunakan untuk membuat aplikasi ini, dengan bahasa pemrograman *PHP CI* dan *database MySQL*. Suatu aplikasi yang dapat mengatur data pasien, data dokter, serta laporan operasional lainnya merupakan hasil dari penelitian ini. Dengan mempercepat proses pencarian informasi dan meningkatkan akurasi data, aplikasi ini memungkinkan layanan masyarakat yang lebih efektif dan berkualitas.
3. Penelitian yang dilakukan (Soepomo, 2023), yang berjudul “Sistem Informasi Manajemen Klinik Basmallah Jambi Berbasis Web” dengan hasil penelitian berupa rancangan prototype Sistem informasi manajemen klinik yang dapat diimplementasikan nantinya sebagai solusi untuk permasalahan yang ada pada Klinik Basmallah Jambi.