

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Teori Umum**

Tujuan dari tinjauan teoritis umum ini adalah untuk memberikan pemahaman tentang kerangka konseptual topik penelitian, konsep-konsep kunci, definisi, dan teori.

##### **2.1.2. Sistem**

Sistem adalah serangkaian yang terdiri dari komponen yang saling terkait yang digunakan untuk mencapai tujuan tertentu dan untuk mencatat persyaratan yang harus dipenuhi oleh sistem yang baru dibangun. Terlihat pada penelitian yang membangun sistem kepegawaian dimana hasil menunjukkan bahwasannya dengan adanya sistem kepegawaian tersebut dapat membantu pekerjaan lebih cepat dibandingkan sistem manual sebelumnya (Wijaya et al., 2022). Menurut (Tjiptabudi & Bernardino, 2019) adapun pengertian sistem keamanan informasi sistem prototipe ini menghasilkan sistem prototipe aplikasi yang aman, efisien, dan efektif. Terlihat pada penelitian keamanan sistem informasi pengawasan perbatasan darat Indonesia, menemukan bahwa penelitian tambahan diperlukan tentang bagaimana sistem berintegrasi dengan aplikasi bea cukai, imigrasi, dan karantina.

##### **2.1.3. Informasi**

Sebagaimana ketika kuantitas dan kualitas informasi meningkat, pengguna membuat keputusan yang lebih baik berdasarkan peran mereka. Terbukti bahwa data yang telah diolah oleh sistem kepegawaian menjadi lebih berguna dan

bermakna, terutama dalam mendukung proses pengambilan keputusan (Wijaya et al., 2022). Mendefinisikan informasi sebagai bentuk yang lebih bermakna dan berguna untuk pengambilan keputusan yang melibatkan banyak orang secara bersamaan (Septiani et al., 2019).

#### **2.1.4. Rancang bangun**

Rancang bangun adalah proses mengubah temuan analisis kerangka perencanaan menggunakan bahasa pemrograman. Proses ini adalah untuk menerapkan kerangka kerja yang telah diakui secara mendalam, perencanaan juga merupakan bagian dari desain dan konstruksi, dari awal hingga selesai, memastikan bahwa semua proses sesuai dengan pedoman yang telah ditetapkan (Putri & Wahyuni, 2024). Menurut (Novitasari et al., 2021) salah satu komponen utama pembuatan program adalah rancangan. Rancang bangun dibuat untuk memberi pemrograman dan ahli teknik gambaran yang jelas dan lengkap. Proses ini melibatkan komponen sistem harus diterjemahkan dalam bahasa pemrograman hasil analisis dan komponen sistem, menunjukkan bagaimana komponen-komponen tersebut dapat digunakan dalam konteks kegiatan ini, tujuannya adalah untuk mencapai tujuan tertentu yang dapat diimplementasikan secara praktis.

#### **2.1.5. Sistem Informasi**

Sistem informasi merupakan objek antara *software*, *hardware* dan infrastruktur yang merupakan informasi yang berfungsi sebagai bentuk sebuah laporan dari suatu organisasi atau perusahaan, hal ini juga telah diamati dalam penelitian mengenai sistem informasi rekam medis pukesmas kecamatan matraman jakarta yang berfungsi khususnya dalam pencatatan dan pencarian data rekam

medis pasien pada bagian rekam medis merupakan bagian integral dari proses untuk memastikan informasi kesehatan yang akurat dan dapat diakses dengan cepat. Hasil pengujian metode black-bok, yang sesuai dengan desain usecase, yang ditunjukkan oleh sistem. Sistem ini dapat menghemat tempat penyimpanan data rekam medis (Helling et al., 2019). Menurut (Novianti et al., 2024) sistem informasi ini merupakan sistem parkir elektronik berbasis *web* dapat diakses secara fleksibel melalui berbagai perangkat seperti handphone kapanpun dan dimanapun.

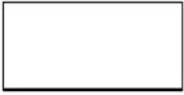


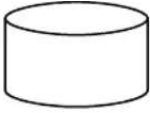
#### **2.1.6. Website (Web)**

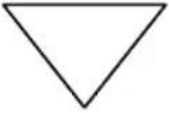
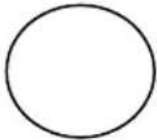

Dalam penelitian yang dilakukan oleh (Hambali & Samsumar, 2022) sebuah website merupakan koleksi untuk menyajikan informasi dalam bentuk teks, animasi, suara, atau kombinasi format tersebut dikenal dengan istilah website. Dimana sistem manajemen proyek berbasis website baik statis maupun dinamis, website merupakan struktur bangunan terdiri dari halaman yang terkait dan menghubungkan dalam jaringan. Menurut (Budiman et al., 2023) website atau situs web merupakan sekumpulan halaman bagian dari konten dan dapat diakses melalui internet atau jaringan lokal, yang berencana untuk membangun sebuah sistem manajemen proyek yang dapat membantu manajer proyek serta divisi yang berkaitan dalam proses pengejerjan sebuah proyek.

#### **2.1.7. Aliran Sistem Informasi**

Sebuah bagan diagram alir yang dimaksud untuk menjelaskan aliran sistem dari fase awal hingga selesai. Oleh karna itu, diperlukan pendoman untuk pembuatan aliran sistem informasi. Di bawah ini adalah simbol-simbol yang ditemukan dalam aliran sistem informasi (Alhamidi, 2020).

**Tabel 2. 1** Simbol Aliran Sistem informasi

Gambar	Nama	Keterangan
	Simbol proses	Menampilkan operasi yang dilakukan oleh proses operasi program komputer
	Simbol manual	Menampilkan pekerjaan manual.
	Simbol multi dokumen	Menampilkan masukan dan keluaran dokumen untuk proses komputer, mekanis, dan manual.
	Simbol <i>database</i>	Menampilkan <i>input</i> atau <i>output</i> yang menggunakan <i>database</i> .

	<p>Simbol simpanan offline</p>	<p>Menampilkan simpan data secara manual.</p>
	<p>Simbol Penghubung</p>	<p>Menampilkan penghubung dalam satu halaman.</p>
	<p>Simbol Aliran</p>	<p>Menampilkan arah aliran dan proses.</p>

### 2.1.8. *UML (Unified Modeling Language)*




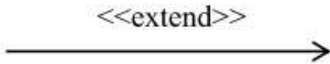
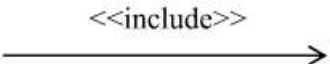
*UML* pada dasarnya merupakan sebuah bahasa visualisasi standar, *UML* menyediakan standar untuk merancang sistem, dan *UML syntax* mendefinisikan mendefinisikan bagaimana elemen-elemen tersebut dikombinasikan (Ridho Bestari et al., 2023). Menurut (Darmawan & Ratnasari, 2020) Tidak hanya *UML* dapat digunakan untuk membuat tata letak perangkat lunak, tetapi juga untuk menampilkan, mendefinisikan, membangun, dan mencatat sistem perangkat lunak. Dengan kata lain, pengembang perangkat lunak melakukan hal yang sama seperti arsitek konstruksi membuat rencana untuk perusahaan konstruksi juga dapat

membangun perangkat lunak.. Sedangkan menurut (Mawaddah & Devitra, 2023) *Unified Modeling Language (UML)* adalah bahasa visual yang digunakan dalam sistem pemodelan. Bahasa ini memungkinkan diagram dengan teks pendukung untuk berkomunikasi, merupakan alat yang bermanfaat untuk meningkatkan kualitas analisis dan desain sistem dan berkontribusi pada pengembangan sistem informasi yang lebih berkualitas. *UML* juga digunakan untuk menggambarkan konstruksi, kemampuan, dan hubungan bagian-bagian kerangka kerja dengan memberikan gambar dan bagian yang dapat dipahami oleh pemrograman. Hal ini memfasilitasi komunikasi yang jelas dan efektif dalam proses merancang dan mengembangkan kerangka kerja pemrograman. Beberapa diagram dalam *UML* seperti *Use case diagram*, *Class diagram*, *Activity diagram*, *Sequence diagram*, *Deployment diagram*, *State diagram* (ZA & Hadiwinata, 2024) maupun *component diagram* yang sering digunakan sebagai berikut:

#### **2.1.7.1. Diagram Kasus (*Use Case Diagram*)**

*Use Case Diagram* menunjukkan bagaimana sistem informasi baru akan berinteraksi dengan satu atau lebih aktor. Seringkali, diagram ini digunakan untuk mengetahui aktor mana yang berhak mengakses dan menggunakan fitur-fitur yang ditawarkan oleh suatu sistem informasi. Berikut *diagram Use Case* yang ada:



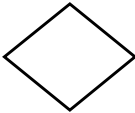


**Tabel 2. 2** Simbol *Use Case* Diagram

Simbol	Deskripsi
	<i>Use case</i> : Abstraksi sistem dan interaksi aktor sistem.
 Asosiasi / <i>Association</i>	Asosiasi/ <i>Association</i> Abstraksi dari hubungan antara <i>actor</i> dengan <i>use case</i> .
 Aktor/ <i>Actor</i>	Aktor/ <i>Actor</i> Mewakili bagaimana orang, sistem, dan alat berinteraksi dengan <i>use case</i> .
Ekstensi/ <i>extend</i> 	Untuk menunjukkan jika suatu kondisi terpenuhi, bahwa <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya.
Menggunakan/ <i>include/uses</i> 	Menunjukkan bahwa, dengan asumsi suatu kondisi terpenuhi, suatu <i>use case</i> dapat digunakan sebagai tambahan fungsional dari <i>use case</i> pemanfaatan lainnya.

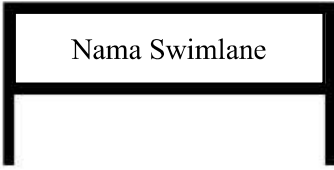
### 2.1.7.2. Diagram Aktivitas (*Activity Diagram*)

Diagram aktivitas adalah desain proses untuk bisnis. Setiap urutan aktivitas yang dijelaskan di sini merupakan urutan visualisasi atau kelompok sistem, sistem proses bisnis yang ditentukan, dan gambaran antarmuka pengguna untuk setiap aktivitas. Pengujian harus ditentukan didefinisikan Berikut diagram *Activity* yang ada:

**Tabel 2. 3** Simbol *Activity Diagram*

Simbol	Deskripsi
Status Awal / <i>Initial State</i> 	Status awal atau initial state adalah Saat sistem pertama kali mulai berfungsi, sistem berada dalam keadaan awal.
Aktivasi 	Suatu tindakan dalam sistem disebut aktivasi, dan biasanya dimulai dengan kata kerja.
Percabangan / <i>Decision</i> 	Percabangan adalah suatu Kegiatan percabangan mencakup berbagai macam kegiatan.
Penggabungan / <i>joint</i> 	Asosiasi yang menggabungkan lebih dari satu gerakan menjadi satu
Status Akhir 	Status Akhir, juga disebut sebagai Status Akhir, adalah keadaan akhir dari daur


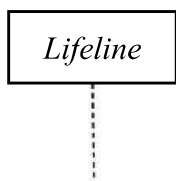


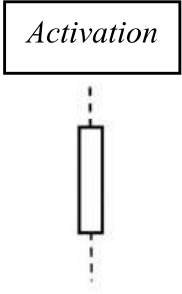


	hidup.
<p>Swimlane</p> 	Memisahkan organisasi bisnis yang masing-masing bertanggung jawab atas pelaksanaan.

### 2.1.7.3. Diagram Urutan (*Sequence Diagram*)

Diagram urutan dapat digunakan untuk menunjukkan bagaimana objek berperilaku selama operasi, diagram urutan dapat digunakan untuk menunjukkan siklus hidup terhubung dengan objek dan pesan yang dikirim dan diterima. Diagram skenario saat ini adalah sebagai berikut:

**Tabel 2. 4** Simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Actor</p> 	Aktornya adalah orang proses, sistem yang berjalan saat ini yang berinteraksi dengan sistem informasi yang akan dibangun, tetapi tidak selalu.
	Komponen yang digambarkan garis putus terhubung dengan objek.

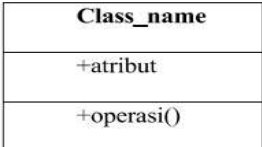






	Mewakili proses durasi aktivitas sebuah operasi.
<p style="text-align: center;">Call Message</p> 	Operasi yang ada pada objek lain atau dirinya sendiri disebut sebagai menyatakan objek.
<p style="text-align: center;">Return Message</p> 	Menunjukkan bahwa suatu entitas telah melakukan suatu operasi yang menghasilkan suatu kepada entitas tertentu.

#### 2.1.7.4. Diagram Kelas (*Class Diagram*)

Untuk mengetahui bagaimana semua kelas yang akan digunakan untuk memahami bagaimana setiap struktur pemrograman akan dibangun. Diagram kelas berguna untuk memastikan bahwa desain visual dan logika pemrograman yang akan dikodekan dalam bahasa pemrograman sinkron. Berikut diagram *class*.

**Tabel 2. 5** *Class Diagram*

Simbol	Deskripsi
--------	-----------

<p>Class</p> 	<p><i>Class</i> adalah Struktur sistem yang akan ditampilkan dalam sistem informasi.</p>
<p><i>Interface</i> </p> <p><b>Interface name</b></p>	<p>Konsep interface dalam pemrograman berorientasi objek mirip.</p>
<p><i>association</i></p> 	<p>Asosiasi biasanya disertai dengan <i>multiplicity</i> dalam hubungan kelas-makna umum.</p>
<p>Asosiasi berarah / <i>directed association</i></p> 	<p>Perhubungan antara kelas dengan makna kelas yang digunakan oleh kelas lain, biasanya disertai dengan <i>multiplicity</i>.</p>
<p>Generalisasi</p> 	<p>Hubungan antara kelas dengan arti generalisasi-spesialisasi.</p>
<p>Kebergantungan / <i>dependency</i></p> 	<p>Kelas memiliki hubungan yang sama dengan kebergantungan satu sama lain.</p>
<p><i>aggregation</i></p> 	<p>Hubungan antara kelas dengan arti semua bagian.</p>

## **2.2 Tinjauan Teori Khusus**

Teori khusus didefinisikan sebagai teori yang dimasukkan sebagai bagian dari penelitian dan dibahas dalam literatur yang secara khusus berkaitan dengan topik penelitian.

### **2.2.1. Manajemen Proyek**

Manajemen proyek adalah disiplin ilmu yang menyelidiki bagaimana merancang, menyusun, mengendalikan dan mengawasi sumber daya hingga pencapaian tujuan proyek. Terlihat pada penelitian ini sistem pembangunan dapat membantu dalam pengelolaan manajemen proyek sampai pada tahapan akhir (Wanty Eka Jayanti<sup>1</sup>), Eva Meilinda<sup>2</sup>), 2021). Manajemen proyek biasanya digunakan untuk mengelola proyek besar, dengan tujuan utama mencapai tujuan proyek dengan mempertimbangkan waktu dan anggaran yang tersedia (Darmawan & Ratnasari, 2020).

### **2.2.2. PHP**

*PHP*, juga dikenal sebagai *Preprocessor Hypertext*, adalah bahasa pemrograman yang dapat memproses semua jenis data dan terintegrasi dengan berbagai basis data seperti *MySQL*, *PostgreSQL*, *SQLite*. Selain mendukung pengembangan sistem, mendukung bahasa tambahan seperti *PHP* (Teguh, 2019). Menurut (Mudjahidin & Dita Pahang Putra, 2020). *PHP* adalah bahasa pemrograman yang digunakan untuk membuat aplikasi *web* menggunakan fungsionalitas server. Oleh karena itu, bahasa pemrograman *web* ini perlu menggunakan *web server* untuk menjalankannya, tanpa *PHP* server tidak dapat dijalankan (Firmansyah et al., 2022).

### 2.2.3. *Visual Studio Code*

*Visual Studio Code*, editor kode sumber yang dibuat oleh Microsoft, tersedia untuk sistem operasi *Windows*, *Linux*, dan *MacOS*, Pengguna dapat mengubah tema, *shortcut keyboard*, dan preferensi mereka. Meskipun unduhan resmi *Visual Studio Code* memerlukan lisensi *proprietary*, itu gratis dan *open-source* (Kelvin Adha Bilqis Ibrahim, 2021). Menurut (Nur Ariesanto Ramdhan, 2019). Aplikasi editor *code open-source* *Visual Studio Code* dibuat oleh Microsoft untuk Mendukung berbagai jenis bahasa pemrograman yang umum digunakan meliputi *Bahasa pemrograman seperti C++, C#, Java, PHP, Python, dan GO* dapat dikenali oleh *Visual Studio Code* dan secara otomatis menyesuaikan warna sintaks sesuai dengan bahasa tersebut.

### 2.2.4. *PostgreSQL*

*PostgreSQL* memiliki banyak fitur yang menarik bagi pengembang perangkat lunak dan merupakan salah satu sistem manajemen basis data relasional *open-source* yang paling banyak digunakan dan paling cepat berkembang. Dalam situasi seperti ini, *PostgreSQL* dapat berfungsi sebagai dasar untuk pembuatan aplikasi dan analisis data (Putri & Wahyuni, 2024). Ini dikenal sebagai *PostgreSQL* dan merupakan salah satu *database* terbaik karena kinerja, skalabilitas, dan kemampuan beradaptasinya yang luar biasa. *SQL* didukung oleh *PostgreSQL* (Nugroho et al., 2023). Penulis memilih *PostgreSQL* karena merupakan database objek relasional yang terpercaya dan stabil, memiliki tipe data geometri, menyediakan semua fitur *database*, kemampuan konektivitas dengan *database* lain, memiliki rute, mampu menampung data spasial, dan dapat diperluas dengan mudah.

Kekurangan *PostgreSQL* adalah kurangnya popularitas dan belum tersedianya replikasi (Wijayanti et al., 2023).

#### **2.2.5. *Framework Laravel***

*Framework* adalah kerangka kerja yang membantu pengembang dalam menghasilkan kode secara konsisten untuk membangun website atau aplikasi. *Framework* memungkinkan pengembang untuk mengurangi jumlah bug dalam aplikasi yang dibuat, karena menyediakan struktur dan alat yang teruji untuk pengembangan yang konsisten dan terorganisir. *Laravel* sendiri merupakan *framework open source* berbasis *PHP* yang bersifat *open source* (Darmawan & Ratnasari, 2020). Sedangkan menurut *Framework Laravel* merupakan *framework* yang paling populer digunakan oleh para programmer (Ridho Bestari et al., 2023).