

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Teori Umum

2.1.1 Sistem

Teknik untuk mengumpulkan, memasukkan, mengolah, dan menyimpan data, ini juga mencakup teknik untuk menyimpan, mengelola, mengawasi, dan melaporkan data untuk membantu organisasi mencapai tujuan mereka (Wati et al., 2021)

Atribut Sitem adalah berikut:

1. Kumpulan bagian yang saling terkait satu sama lain.
2. Area Sistem yang membatasi luas operasi sistem lain.
3. Subsistem adalah bagian sistem yang saling berhubungan satu sama lain dan melakukan tugas yang sama.

Kita dapat membuat kesimpulan bahwa sistem terdiri dari sekelompok orang yang saling berinteraksi satu sama lain dan bahwa sekelompok individu memengaruhi proses bisnis tertentu yang dilakukan oleh individu tersebut.

2.1.2 Informasi

Data yang diproses untuk digunakan dalam penentuan kebijakan dan bermanfaat bagi pengguna. Salah satu tujuan analisis informasi adalah untuk mengubah data yang tidak dapat digunakan menjadi data yang dapat digunakan, mengubah data yang berguna menjadi data yang sesuai dengan situasi, atau mengubah fakta yang diperoleh

menjadi cara yang bermanfaat bagi mereka yang membutuhkannya (Silalahi & Saragih, 2022).

Peneliti menemukan bahwa data diperoleh melalui penilaian yang didasarkan pada teori sebelumnya.

2.1.3 Sistem Informasi

Mengumpulkan, memasukkan, mengolah, dan menyimpan data, serta cara untuk menyimpan, mengelola, mengawasi, dan melaporkan informasi sehingga organisasi dapat mencapai tujuan mereka (Wati et al., 2021)

Dan menurut (Ramdani, 2022) Sistem informasi adalah perangkat prosedur yang saling berhubungan, terarah, dan sistematis dengan tujuan Untuk proses pembuatan kebijakan atau keputusan

2.1.4 Implementasi

Proses membuat sistem baru yang dapat diandalkan untuk menggantikan sistem yang sudah ketinggalan zaman dengan menggunakan data dan saran penelitian yang dapat diandalkan. Orang berusaha menggunakan alat bantu pengembangan sistem untuk memenuhi kebutuhan pengguna akhir selama berbagai tahap proses desain. Sebuah spreadsheet yang berisi temuan desain lengkap yang mencakup berbagai data tentang proses, input, dan output sistem yang dimaksud digunakan untuk mendukung pengembangan ini (Toruan & Saragih, 2020).

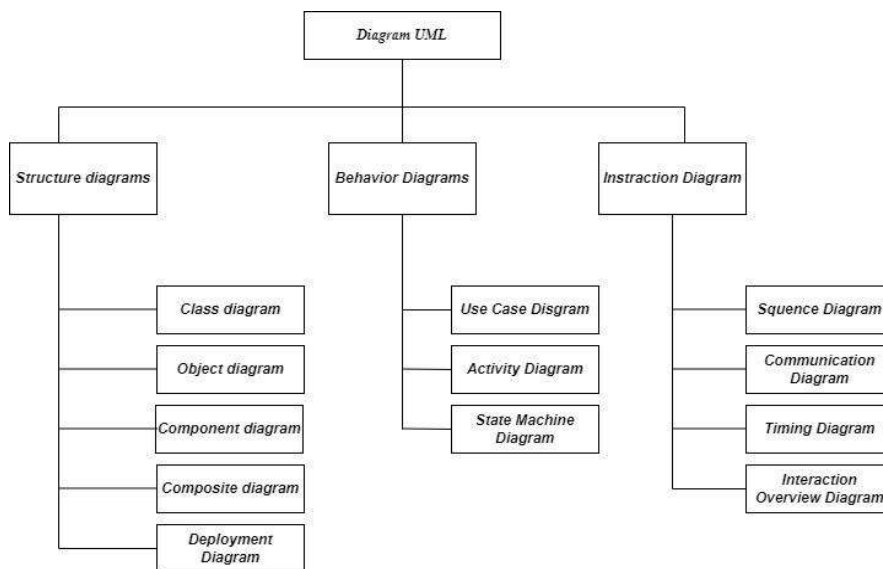
Proses dan data yang dibutuhkan oleh sistem baru disebut desain. Salah satu keuntungan dari tahap arsitektur sistem ini adalah bahwa itu memberikan programmer

desain lengkap untuk membantu mereka menulis aplikasi. Saat ini, aplikasi, basis data, dan peralatan harus dikembangkan dengan mempertimbangkan berbagai bagian kerangka kerja terkomputerisasi.

Berdasarkan penjelasan teori di atas, peneliti mencapai kesimpulan bahwa desain adalah proses membuat segala sesuatu yang memiliki aktualitas fisik atau konstruksi untuk mencapai hasil akhir dengan tindakan yang tepat.

2.1.5 Unified Modeling Language (UML)

Sekitar akhir tahun 80-an dan awal tahun 90-an, Unified Modeling Language (UML) muncul sebagai pengganti metode analisis berorientasi objek dan desain berorientasi objek (OOAD&D, atau analisis dan desain berorientasi objek). UML menggabungkan metode Booch, Rumbaugh (OMT), dan Jacobson, tetapi UML meliputi lebih banyak daripada OOAD. Dalam upaya untuk menjadi bahasa pemodelan standar di masa depan, standarisasi proses dilakukan pada pertengahan proses pengembangan UML (Prasetya et al., 2022)

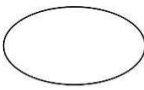


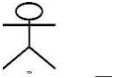

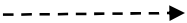
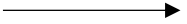
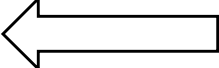

Gambar 2.1 Diagram UML
(Sumber: Penelitian 2024)

1. *Use Case Diagram*

Dipakai untuk menunjukkan komunikasi antara sistem dan aktor. Ini memvisualisasikan bagaimana aktor (seperti pengguna manusia atau sistem lain) menggunakan sistem untuk mencapai tujuan mereka melalui fungsi-fungsi atau kasus penggunaan yang disediakan oleh sistem itu sendiri.

Tabel 2. 1 Simbol *Use Case Diagram*

Simbol	Keterangan
<i>Use case</i> 	memberikan penjelasan tentang bagaimana evolusi operasi sistem dapat menghasilkan aktor yang dapat diukur. Ini mencakup pengembangan dan perubahan sistem dari waktu ke waktu

	yang dapat diamati dan diukur dalam hal kinerja, efisiensi, dan efektivitas
Aktor / actor 	mengimplementasikan tindakan pengguna sesuai dengan kasus penggunaan.
Asosiasi 	Hubungan antara benda-benda yang berbeda dan cara mereka berinteraksi satu sama lain.
<i>Include/include</i> 	Sumber daya digunakan dengan jelas.
<i>Generalisasi</i> 	Hubungan objek (inheritance) membagi perilaku dan struktur data dari objek-objek dalam objek utama (kelas induk atau super kelas). Perilaku dari use case target.
<i>Extensi/extend</i> 	Mengidentifikasi perilaku use case target untuk meningkatkan kemungkinan menggunakan sumber daya pada saat tertentu.
<i>System</i> 	Paket dari sistem khusus diaktifkan

(Sumber: Penelitian 2024)


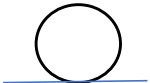
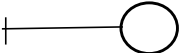

2. *Sequence diagram*

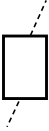
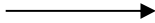
Sequence diagram berdasarkan (Prasetya et al., 2022) *Sequence diagram*, juga disebut sebagai diagram urutan, digunakan untuk menjelaskan dan memvisualisasikan interaksi antar objek dalam sebuah sistem. Diagram ini menunjukkan seperti apa objek berinteraksi satu sama lain melalui pertukaran

pesan atau perintah, serta urutan waktu pelaksanaan. Dalam kebanyakan kasus, objektif diurutkan dari kiri ke kanan sesuai dengan alur kerja operasi.

- a. Indikator a pada diagram urutan proses aliran, dimulai dengan menerima pesan
- b. *Messages* adalah Pesan terdiri dari sejumlah objek yang terkait dan instruksi untuk tindakan yang segera dilakukan.
- c. *Sync messages* adalah Pesan sinkronisasi yang terjadi ketika seseorang mengirim pesan kepada teman terdekatnya dan teman tersebut menerima pesan itu.
- d. *Synchronous message* menggunakan *return value* Mendapat pesan dan mengirim pesan kembali untuk pengiriman pertama yang digambarkan titik-titik.

Tabel 2. 2 Simbol *Sequence Diagram*


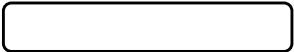


Simbol	Keterangan
Aktor 	Pelanggan yang dapat diidentifikasi dengan sistem harus ditarik
<i>Entity class</i> 	Gambaran sistem sebagai dasar untuk pembuatan basis data.
<i>Boundary class</i> 	Mengatasi komunikasi yang terjadi dilingkungan sistem
<i>control class</i> 	Bertanggungjawab terhadap kelas objek logika.


<p><i>A focus of control & a life line</i></p> 	sebutkan sumber dan tujuan komunikasi
<p><i>A message</i></p> 	Jelaskan bagaimana mengirim pesan

3. Activity diagram

menurut syarif & Nugraha, 2020 aktivitas diagram menunjukkan proses kerja atau alur bisnis yang telah diimplementasikan dalam software.

Tabel 2. 3 Simbol Activity diagram


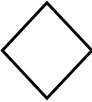
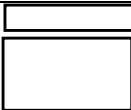
Simbol	Keterangan
<p>Aktivitas</p> 	Menggambarkan cara antarmuka kelas berinteraksi
<p>Aksi</p> 	Bagaimana suatu tindakan dilakukan dapat dilihat melalui status sistem.
<p><i>Initial node</i></p> 	Proses dimulai.
<p><i>Activity final Node</i></p> 	Pada titik saat ini, ia akan

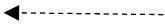



<i>Fork Node</i> 	Terbagi menjadi banyak alur
---	-----------------------------

4. *Class Diagram*

Salah satu jenis diagram struktur yang tersedia dalam UML adalah diagram kelas, yang memberikan deskripsi tentang struktur, atribut, metode, dan hubungan dari setiap kelas dalam sistem. Diagram kelas ini tidak bersifat dinamis atau menjelaskan perilaku saat kelas-kelas berinteraksi, tetapi fokus pada menggambarkan hubungan dan struktur yang ada di antara kelas-kelas tersebut (Prasetya et al., 2022).

Tabel 2. 4 Simbol *Class Diagram*




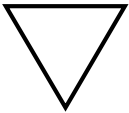

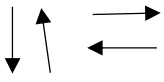



Simbol	penjelasan
<i>generalization</i> 	Ketika objek (turunan) berada di atas objek induk (pengakses), struktur file objek menjelaskan bagaimana dua keputusan berhubungan satu sama lain
<i>Nary association</i> 	menjelaskan bagaimana dua keputusan berhubungan satu sama lain.
<i>Class</i> 	instruksi untuk sekelompok barang yang sama.

<i>Realization</i> 	rincian tentang proses pengembalian objek
<i>Dependency</i> 	Perubahan pada elemen independen dapat berdampak pada elemen yang terkait dengan elemen dependen.
Association 	Menguraikan hubungan antara dua hal.
Collaboration 	Rincian tentang proses yang dilakukan sistem untuk mencapai hasil.

2.1.6 Aliran Sistem Informasi

Urutan data yang harus diproses dari awal program hingga akhir ditunjukkan dalam diagram aliran alur sistem informasi. Simbol-simbol yang ada di diagram alir akan menjadi fokus perhatian (Sutanto, 2019). Diagram ini membantu menentukan apakah sistem informasi otomatis atau manual. Jika sistem data tidak lagi berguna, perubahan pada cara pemrosesan data dibutuhkan untuk penilaian yang lebih baik dan pengumpulan data yang cepat dan tepat.

Tabel 2. 5 Simbol Aliran Sistem Informasi

No	Nama	Simbol	Keterangan
1	Proses		untuk memproses data secara digital.
2	Media penghubung		Digunakan untuk menyambungkan pola.
3	Multi Dokumen		Proses input
4	Simpan		Data yang dibuat disimpan.
5	Proses Manual		Memproses data dengan manual.
6	Aliran sistem		Untuk mengarahkan proses aliran data
7	Penyimpanan		Perangkat penyimpanan digital.
8	Pita kertas		menggunakan pita kertas.
9	Display		untuk menampilkan output ke monitor

2.2 Tinjauan Teori Khusus

2.2.1 Kesehatan Ibu Hamil

Ibu Hamil adalah seorang wanita hamil yang mengandung sejak konsepsi sampai lahirnya janin. Kehamilan adalah masa dimana seorang wanita membawa embrio atau janin didalam tubuhnya. Kehamilan pada wanita terjadi selama 40 minggu antara periode menstruasi terakhir kelahiran atau 38 minggu dari pembuahan (Wati et al., 2021).

Tanda awal bahaya kehamilan yang dapat dialami setiap ibu hamil adalah risiko kehamilan. Sangat penting untuk menjaga keselamatan ibu hamil dan janin dalam kandungan hingga proses persalinan.

Tidak mengetahui gejala dini dari risiko kehamilan dapat menyebabkan banyak konsekuensi yang merugikan (Setyawan et al., 2023).

2.2.2 Website

Kumpulan halaman web dan file digital seperti gambar, video, dan file digital lainnya yang disimpan pada *web server* yang dapat diakses melalui internet. Dengan kata lain, website juga memiliki berbagai folder dan file yang dapat melakukan berbagai hal, seperti menampilkan konten, menyimpan data, dll (Wahyudin & Rahayu, 2020).

2.2.3 Framework Node JS

Platform yang dibangun di atas *runtime JavaScript Chrome* untuk membuat aplikasi dengan cepat dan mudah untuk diskalakan. Keunggulannya terletak pada

teknik *non-blocking*, yang memungkinkan banyak request diselesaikan secara paralel tanpa harus menunggu operasi sebelumnya selesai (Mubariz et al., 2020).

Node Package Manager (NPM), sebuah kumpulan kode *open-source*, membantu pengembang menggunakan Node JS dan menginstal paket untuk digunakan dalam proyek yang sedang dikerjakan. NPM juga mempercepat proses pengembangan. Banyak perusahaan terkemuka seperti Netflix, LinkedIn, NASA, Paypal, dan Medium telah menggunakan Node.JS karena kecepatan dan penggunaan sumber daya yang kuat (Nasution, 2021).

2.2.4 React.JS

JavaScript deklaratif *open source*, efektif dan fleksibel dalam pembuatan antarmuka pengguna. React memungkinkan pembuatan antarmuka pengguna yang kompleks dengan kumpulan kode kecil dan terisolasi yang disebut “komponen”. *React.js* digunakan untuk menangani lapisan tampilan dalam aplikasi satu halaman dengan pengembangan aplikasi seluler. *React.JS* dikelola facebook, Instagram, komunitas pengembang, dan perusahaan. Yang berupaya memberikan kecepatan, kesederhanaan, dan skalabilitas (Iswari & Nasution, 2021).

Beberapa fitur yang paling mencolok adalah JSX, Stateful components, virtual documents, virtual documents object model.

2.2.5 Express.JS

Framework yang cukup minimalis, dibuat oleh pengembang Paket Middleware yang Kompatibel untuk menyelesaikan hampir semua masalah dalam proses

pengembangan web. Selain itu, terdapat beberapa perpustakaan untuk membantu pengembang bekerja dengan cookie, sesi, login pengguna, parameter URL, data POST, header keamanan, dan sebagainya. Selain hal-hal tersebut, alasan penggunaan Express dalam pengembangan web Ubaform adalah popularitas Express itu sendiri (Nasution, 2021).

2.2.6 Mongo.DB

Database NoSQL, yang termasuk non-SQL dan tidak hanya SQL, adalah database yang menyimpan data dalam format yang berbeda daripada tabel relasional atau Relational Database Management System (RDBMS). Dibangun pada akhir tahun 2000-an, database NoSQL berfokus pada skalabilitas, kecepatan pertanyaan, kompatibilitas dengan perubahan aplikasi yang sering terjadi dan kemudahan pemrograman untuk pembangun. Namun, mengingat peningkatan pesat dalam penggunaan data digital dalam beberapa tahun terakhir, RDBMS telah menunjukkan dirinya sebagai solusi yang kuat untuk penyimpanan data selama beberapa dekade. maka hal ini juga sejalan dengan peningkatan permintaan untuk jenis data yang disimpan dan frekuensi operasi data yang dibutuhkan oleh perangkat lunak database (Nasution, 2021).

2.2.8 Visual Studio Code

Perangkat lunak desktop yang kuat dan ringan untuk editor kode. Program yang membutuhkan aplikasi yang kompeten dibuat dengan Visual Studio Code. Visual studio code bekerja dengan berbagai sistem operasi seperti Windows, macOS, dan

Linux. Ini mendukung berbagai bahasa pemrograman seperti JavaScript, HTML, CSS, PHP, Python, dan C++. Selain itu, fitur *live share* Visual Studio Code memungkinkan banyak pengembang bekerja pada proyek yang sama dari berbagai tempat (Nyaris, 2023).