

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Rancang Bangun

Menurut (Putri, 2022), kata "rancang" berasal dari istilah "perancangan" dan menggambarkan serangkaian metode yang digunakan untuk menerjemahkan analisis suatu sistem ke dalam bahasa pemrograman. Hal ini bertujuan untuk menyajikan deskripsi yang terperinci tentang setiap aspek dalam sistem yang akan diimplementasikan. Oleh karena itu, dalam konteks penelitian tersebut, istilah "rancang bangun" merujuk pada proses yang diperlukan untuk menciptakan atau mengembangkan sistem baru. Sistem ini diharapkan dapat membantu peneliti dalam mengatasi masalah yang ada pada objek penelitian.

Menurut (Ibrahim, Agus, dan Wulansari, 2021), dalam rancang bangun sistem informasi, terdapat dua pendekatan utama yang dapat diambil, yakni metode berorientasi objek dan metode terstruktur. Metode berorientasi objek fokus pada konsep objek yang dapat mencakup data dan fungsionalitas dalam suatu entitas tunggal, sementara metode terstruktur berfokus pada hierarki, urutan, dan pemisahan tugas dalam pengembangan sistem. Pemilihan antara keduanya bergantung pada kebutuhan dan karakteristik khusus dari proyek pengembangan sistem informasi tersebut. Metode berorientasi objek sering kali memberikan fleksibilitas dan modularitas, sementara metode terstruktur dapat menonjolkan kontrol dan keteraturan dalam proses pengembangan.

##### 2.1.2. Pengertian Aplikasi

Menurut Achmad (2016), aplikasi diartikan sebagai salah satu unit perangkat lunak yang dirancang untuk memenuhi kebutuhan berbagai aktivitas. Aplikasi ini merupakan sebuah program yang mengandung perintah-perintah untuk melakukan pengolahan data dan membangun sistem atau program untuk mengolah data tersebut.

Dalam konteks sistem komputer, aplikasi atau perangkat lunak (software) memegang peran krusial sebagai komponen penting. Aplikasi ini juga dapat diklasifikasikan menjadi beberapa jenis, seperti berbasis desktop, web, dan mobile, sesuai dengan lingkup dan platform penggunaannya.

### 2.1.3. Pengertian Advokasi

Menurut (Wiro, 2021), “Advokasi adalah usaha yang melibatkan berbagai bentuk komunikasi persuasif untuk memperjuangkan suatu tujuan atau pandangan. Ini mencakup tindakan pembelaan, dukungan, dan rekomendasi yang bertujuan memengaruhi pendapat dan tindakan orang lain demi kepentingan tertentu”.

peneliti menyimpulkan bahwa pentingnya advokasi sebagai alat yang kuat untuk membawa perubahan positif dalam berbagai aspek kehidupan sosial dan masyarakat.

## 2.2. Teori Khusus

### 2.2.1. *System Development Life Cycle (SDLC)*

Menurut (Dennis, 2015) SDLC memiliki empat tahapan utama yang serupa, yaitu perencanaan, analisis, desain, dan implementasi. Meskipun proyek yang berbeda mungkin menekankan aspek tertentu dari SDLC atau mendekati tahapan SDLC dengan pendekatan yang berbeda, semua proyek melibatkan elemen dari keempat tahap tersebut. Setiap tahap terdiri dari serangkaian langkah yang mengandalkan teknik tertentu untuk menghasilkan hasil yang dapat dicapai. 4 Fase dalam SDLC yaitu:

#### A. Perencanaan (*Planning*)

Merupakan suatu proses dasar yang bertujuan untuk memahami aplikasi atau sistem yang akan dikembangkan, sekaligus mempertimbangkan bagaimana tim proyek akan mengimplementasikannya.

#### B. Analisis (*Analysis*)

Analisis dilakukan untuk menghasilkan informasi terkait identifikasi proses yang berkaitan dengan pengguna sistem, alur sistem, dan lokasi penggunaan sistem.

#### C. Perancangan (*Design*)

Perancangan bertujuan untuk mengetahui metode atau proses yang digunakan oleh aplikasi pada perangkat keras, perangkat lunak, dan jaringan yang tersedia. Ini juga melibatkan spesifikasi antarmuka pengguna, formulir, laporan, database, dan file yang diperlukan.

#### D. Implementasi (*Implementation*)

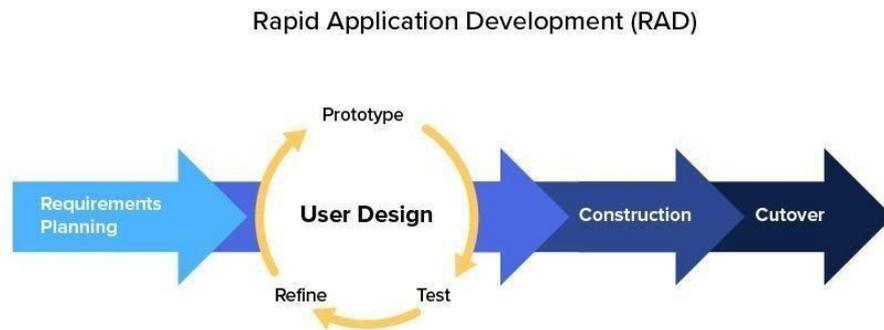
Implementasi menandai tahap akhir dalam SDLC. Pada tahap ini dilakukan pembangunan sistem atau aplikasi, meliputi proses pemrograman aplikasi dan pengujian.

### 2.2.2. *Rapid Application Development (RAD)*

Menurut (Dicky, Ricki, dan Ferina, 2021), *Rapid Application Development (RAD)* merupakan model proses pengembangan perangkat lunak yang menekankan pada

perbaikan bertahap, terutama untuk proyek dengan batasan waktu yang singkat. RAD merupakan model proses pengembangan perangkat lunak yang berfokus pada siklus pengembangan yang pendek, dan sangat berbeda dengan metode Waterfall.

Menurut (Husain, 2023), proses pengembangan dengan metode RAD melibatkan empat fase utama yaitu perencanaan kebutuhan, desain pengguna, konstruksi cepat dan custover.



Gambar 2.1. Fase RAD  
Sumber : (Husain, 2023)

### 1. *Requirements Planning*

Pada tahap ini, *developer*, *client*, dan tim berkolaborasi untuk mengoptimalkan tujuan dan harapan proyek. Selain itu, mereka juga bekerja sama untuk mengatasi tantangan yang ada dan potensi permasalahan yang mungkin timbul selama proses rancang bangun.

### 2. *User Design*

Pada tahap ini, *client* bekerja sama dengan *developer* untuk memastikan persyaratan mereka terpenuhi pada setiap tahap proses desain. Demikian pula dalam pengembangan perangkat lunak yang dapat disesuaikan, ini memungkinkan *client* untuk menguji prototipe produk pada tahap mana pun untuk memastikan produk sesuai dengan harapan mereka. Semua kesalahan dan masalah diidentifikasi dan diperbaiki melalui proses berulang. Pengembang merancang prototipe, *client* melakukan uji coba, dan setelah itu mereka berkumpul untuk mendiskusikan metode tersebut.

### 3. *Rapid Construction*

Pada tahap ketiga diperoleh prototipe dan sistem Beta dari proses perancangan yang kemudian diubah menjadi model kerja. Sebagian besar masalah dan perubahan besar diselesaikan selama iterasi lengkap pada tahap desain, sehingga memungkinkan *developer* untuk lebih cepat menghasilkan model kerja akhir dibandingkan dengan metode manajemen proyek tradisional.

### 4. *Cutover*

Proses perakitan terjadi ketika produk akhir diimplementasikan. Tahapan ini meliputi transfer data, pengujian, transisi ke sistem baru, dan pelatihan pengguna. Semua modifikasi akhir dilakukan sementara pengembang dan pelanggan terus memeriksa kesalahan sistem.

### **2.2.3. Object Oriented Analysis and Design (OOAD)**

Menurut (Purwaningtias, 2018), *Object-Oriented Analysis and Design* (OOAD) adalah suatu pendekatan analisis dan perancangan perangkat lunak yang berfokus pada konsep berorientasi objek. Paradigma berorientasi objek dianggap sebagai cara baru dalam mengembangkan sistem, dimana sistem dianggap sebagai kumpulan objek yang mempunyai perilaku dan berinteraksi satu sama lain. OOAD adalah metode analisis yang mengkaji kebutuhan sistem dari perspektif kelas dan objek dalam konteks masalah sistem dan subsistem. Dalam perancangan OOAD diperlukan teknik khusus yang menggunakan Unified Modeling Language (UML) sebagai alat standar dalam pengembangan sistem perangkat lunak berorientasi objek

Menurut (Dennis, 2015), dasar dari *Object-Oriented Analysis and Design* (OOAD) adalah bahwa semua hal seharusnya dianggap sebagai objek yang memiliki data (atribut) dan proses (perilaku). Sebuah objek mengubah atau mengakses atributnya sendiri hanya melalui perilakunya. Objek dapat berkomunikasi satu sama lain untuk pertukaran informasi atau pelaksanaan tindakan tertentu. Perubahan pada satu objek tidak berdampak pada objek lain karena atribut dan perilaku bersifat mandiri atau terenkapsulasi dalam masing-masing objek. Enkapsulasi ini memungkinkan objek digunakan kembali untuk membangun berbagai sistem, karena objek dapat dimasukkan dan dihapus dari aplikasi dengan sedikit efek berantai.

### **2.2.4. Unified Modelling Language (UML)**

Menurut (Dede Wira dan Rahmi Andriani, 2019), UML merupakan standar bahasa yang umum digunakan dalam industri untuk menentukan persyaratan, melakukan analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berbasis objek.

Menurut (Dennis, Wixom, dan Tegarden, 2015), UML bertujuan untuk memberikan istilah umum dan teknik diagram yang kaya guna memfasilitasi pemodelan proyek pengembangan sistem dari tahap analisis hingga implementasi, khususnya dalam konteks istilah berorientasi objek.

peneliti menyimpulkan bahwa UML adalah bahasa standar yang digunakan dalam industri untuk mengukur persyaratan, melakukan analisis, merancang, dan mendeskripsikan arsitektur dalam pemrograman berbasis objek, dengan tujuan menyediakan istilah umum dan teknik diagram untuk memfasilitasi pemodelan proyek

pengembangan sistem.

Tabel 2.1. Jenis UML

<b>Nama Diagram</b>	<b>Fungsi</b>	<b>Fase Primer</b>
<b><i>Structure Diagrams</i></b>		
Class	Mengilustrasikan hubungan antar kelas yang dimodelkan di dalam sistem	Analysis, Design
<b><i>Behavioral Diagrams</i></b>		
Activity	Mengilustrasikan alur kerja bisnis yang tidak bergantung pada <i>class</i> , alur aktivitas dalam <i>use case</i> , atau desain metode secara mendetail	Analysis, Design
Sequence	Menggambarkan model perilaku objek dalam <i>use case</i> <u>berfokus</u> pada pengurutan aktivitas berdasarkan waktu	Analysis, Design
Use-Case	Menangkap kebutuhan bisnis untuk sistem dan menggambarkan interaksi antara sistem dengan <u>lingkungannya</u>	Analysis

Sumber : (Dennis, Wixom, dan Tegarden, 2015)

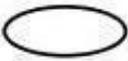
### 2.2.5. Use Case Diagram

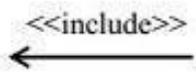
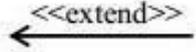
Menurut (Dennis, Wixom, dan Tegarden, 2015), *Use Case Diagram* mencerminkan cara suatu sistem berperilaku atau berfungsi. Diagram ini terdiri dari serangkaian kemungkinan urutan interaksi antara sistem dan pengguna dalam konteks tertentu, terkait dengan tujuan tertentu. Dalam *Use Case Diagram* ini, sistem menunjukkan perilakunya dalam berbagai kondisi sambil merespons permintaan dari aktor-aktor utama.

Permintaan pertama kali datang dari aktor utama, terkait dengan tujuan tertentu, dan sistem memberikan respons. *Use Case Diagram* penggunaan dapat dinyatakan sebagai frase kata kerja dalam present tense yang mencakup kata kerja (tindakan yang dilakukan sistem) dan objek kata kerja (hal yang dilakukan sistem).

*Use Case Diagram* membantu dalam memahami persyaratan fungsional suatu sistem. Pada tahap analisis kebutuhan, analis bekerja sama dengan pengguna sistem yang bersangkutan untuk memahami secara komprehensif fungsi-fungsi yang diinginkan dari sistem. Saat proses pengembangan persyaratan dimulai, fungsi sistem yang dijelaskan digambarkan sebagai kasus penggunaan.

Tabel 2.2. Notasi Use Case

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 use case seluruhnya merupakan fungsionalitas dari use case lainnya
	Menunjukkan bahwa suatu use case merupakan tambahan fungsional dari use case lainnya jika suatu kondisi terpenuhi

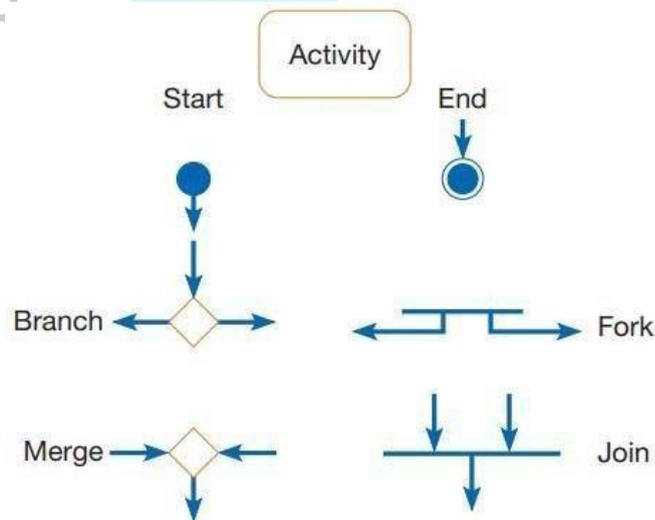
Sumber : (Dennis, Wixom, dan Tegarden, 2015)

### 2.2.6. Activity Diagram

Menurut (Dennis, Wixom, dan Tegarden, 2015), *Activity Diagram* menunjukkan logika kondisional untuk urutan aktivitas sistem yang diperlukan untuk mencapai proses bisnis. Aktivitas individu mungkin manual atau otomatis. Selanjutnya setiap kegiatan menjadi tanggung jawab unit organisasi tertentu.

Notasi *Activity Diagram* dasar hanya berisi beberapa simbol. Setiap aktivitas diwakili oleh persegi panjang bulat, dengan tindakan yang dilakukan oleh aktivitas tersebut tertulis di dalamnya. *Activity Diagram* sendiri mewakili keseluruhan proses yang terdiri dari serangkaian kegiatan.

peneliti menyimpulkan bahwa *Activity Diagram* merupakan representasi yang menunjukkan urutan aktivitas sistem untuk mencapai suatu proses bisnis, dengan setiap aktivitas bertanggung jawab pada unit organisasi tertentu.



Gambar 2.2. Notasi Activity Diagram

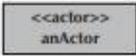
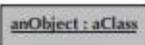
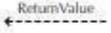
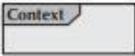
Sumber : (Dennis, Wixom, dan Tegarden, 2015)

## 2.2.7. Sequence Diagram

Menurut (Dennis, Wixom, dan Tegarden, 2015), *Sequence Diagram* menyajikan objek-objek yang terlibat dalam suatu *Use Case* dan pesan-pesan yang saling dikirim di antara mereka sepanjang jalannya suatu skenario. Diagram urutan merupakan model dinamis yang menampilkan urutan pesan eksplisit yang dikirim antar objek selama interaksi tertentu. Karena diagram ini menitikberatkan pada urutan aktivitas di antara kelompok objek berdasarkan waktu, sangat bermanfaat untuk memahami spesifikasi waktu nyata dan *Use Case* yang rumit.

Notasi *Sequence Diagram* Dasar mencakup sejumlah simbol yang digunakan untuk merepresentasikan interaksi antar objek atau entitas dalam suatu sistem dalam suatu urutan waktu. Simbol-simbol tersebut antara lain:

Tabel 2.3. Notasi Sequence Diagram

Term and Definition	Symbol
<p><b>An actor:</b></p> <ul style="list-style-type: none"> <li>Is a person or system that derives benefit from and is external to the system.</li> <li>Participates in a sequence by sending and/or receiving messages.</li> <li>Is placed across the top of the diagram.</li> <li>Is depicted either as a stick figure (default) or, if a nonhuman actor is involved, as a rectangle with &lt;&lt;actor&gt;&gt; in it (alternative).</li> </ul>	 <p>anActor</p> 
<p><b>An object:</b></p> <ul style="list-style-type: none"> <li>Participates in a sequence by sending and/or receiving messages.</li> <li>Is placed across the top of the diagram.</li> </ul>	
<p><b>A lifeline:</b></p> <ul style="list-style-type: none"> <li>Denotes the life of an object during a sequence.</li> <li>Contains an X at the point at which the class no longer interacts.</li> </ul>	
<p><b>An execution occurrence:</b></p> <ul style="list-style-type: none"> <li>Is a long narrow rectangle placed atop a lifeline.</li> <li>Denotes when an object is sending or receiving messages.</li> </ul>	
<p><b>A message:</b></p> <ul style="list-style-type: none"> <li>Conveys information from one object to another one.</li> <li>A operation call is labeled with the message being sent and a solid arrow, whereas a return is labeled with the value being returned and shown as a dashed arrow.</li> </ul>	 
<p><b>A guard condition:</b></p> <ul style="list-style-type: none"> <li>Represents a test that must be met for the message to be sent.</li> </ul>	
<p><b>For object destruction:</b></p> <ul style="list-style-type: none"> <li>An X is placed at the end of an object's lifeline to show that it is going out of existence.</li> </ul>	
<p><b>A frame:</b></p> <ul style="list-style-type: none"> <li>Indicates the context of the sequence diagram.</li> </ul>	

Sumber : (Dennis, Wixom, dan Tegarden, 2015)

### 2.2.8. Pelacakan Aduan

Menurut (Mutiah, 2020), "Pelacakan pengaduan adalah suatu sistem atau mekanisme yang digunakan untuk melacak atau mengikuti perkembangan pengaduan atau pengaduan yang disampaikan oleh individu atau pihak tertentu kepada suatu lembaga atau organisasi. Sistem ini biasanya digunakan oleh lembaga pemerintah, perusahaan, atau entitas lain untuk mengumpulkan dan mengelola pengaduan mulai dari awal hingga penyelesaian.

(Yunanto, 2023) menjabarkan contoh isi yang biasanya ada dalam buku catatan pengaduan:

- A. Nomor Referensi Pengaduan: Setiap pengaduan diberi nomor referensi unik atau nomor tiket untuk pelacakan yang mudah.
- B. Tanggal dan Waktu Penerimaan: Catatan tanggal dan waktu pengaduan diterima.
- C. Nama Pelapor: Tuliskan nama lengkap atau identifikasi pelapor, jika memungkinkan.
- D. Kontak Pelapor: Sertakan nomor telepon, alamat email, atau alamat fisik pelapor.
- E. Deskripsi Pengaduan: Jelaskan dengan detail apa yang menjadi pokok pengaduan terkait dengan pelayanan publik.
- F. Lokasi Kejadian: Catat lokasi atau wilayah di mana pengaduan terkait dengan pelayanan tersebut terjadi.
- G. Jenis Pelayanan: Tentukan jenis pelayanan yang terkait dengan pengaduan, seperti layanan kesehatan, pendidikan, transportasi, dan sebagainya.
- H. Status Pengaduan: Perbarui status pengaduan, seperti "sedang diselidiki," "ditutup," atau "selesai," sesuai dengan perkembangan.

Berdasarkan penelitian Mutiah (2020) tentang pelacakan pengaduan, sistem ini didefinisikan sebagai mekanisme penting untuk melacak dan mengelola pengaduan dari awal hingga penyelesaian. Yunanto (2023) menjabarkan unsur-unsur penting dalam catatan pengaduan, termasuk nomor referensi, tanggal dan waktu penerimaan, informasi pelapor, deskripsi pengaduan, lokasi kejadian, jenis pelayanan, dan status pengaduan. Keseluruhan, pelacakan pengaduan menjadi kunci untuk efisiensi, transparansi, dan peningkatan layanan.

### 2.2.9. Visualisasi Data

Menurut (Donny Fernando, 2018), “Visualisasi data adalah konsep yang luas yang mencakup semua usaha untuk membantu individu memahami makna dari data dengan menghadirkannya dalam bentuk visual. Nilai data dalam ekonomi saat ini telah mengalami perubahan yang signifikan. Data kini menjadi pendorong utama dalam mempercepat kegiatan bisnis, memberikan organisasi kemampuan untuk membuat keputusan yang lebih cerdas, dan mendekati kerja dalam waktu nyata.”

Menurut (Sosulski, 2018), komponen visualisasi data terdiri dari:

a) Cerita

Cerita mencerminkan maksud di balik representasi visualisasi data. Para ahli data berinteraksi dengan berbagai pihak yang berkepentingan untuk menyampaikan tujuan mereka dalam menganalisis data. Sebagai contoh, mereka mungkin ingin menilai indikator kinerja utama atau meramalkan volume penjualan. Kolaborasi terjadi antara ahli data dan pengguna bisnis untuk menentukan narasi yang ingin mereka peroleh dari informasi yang terkandung dalam data tersebut.

b) Data

Selanjutnya, analisis data dilakukan untuk mengenali himpunan data yang dapat mendukung mereka dalam menggambarkan cerita data tersebut. Mereka akan mengubah struktur data yang ada, membersihkan data, menghilangkan nilai yang tidak biasa, dan melakukan analisis mendalam. Setelah persiapan data selesai, mereka akan merancang berbagai metode eksplorasi visual.

c) Visual

Ilmuwan data selanjutnya akan memilih teknik visualisasi yang paling cocok untuk mengkomunikasikan wawasan terbaru. Mereka akan menciptakan diagram dan grafik yang menyoroti titik-titik kunci dalam data dan menyederhanakan set data yang kompleks. Mereka secara efisien merancang cara untuk menyajikan data secara sistematis agar relevan bagi kecerdasan bisnis.

Menurut Donny Fernando (2018), visualisasi data adalah usaha menyajikan data dalam bentuk visual untuk membantu pemahaman makna data, menjadikan data sebagai pendorong utama dalam bisnis. Sosulski (2018) menyebutkan tiga komponen visualisasi data, yaitu cerita yang mencerminkan maksud di balik representasi data, analisis data untuk mendukung cerita, dan pilihan visualisasi yang sesuai untuk menyampaikan wawasan terbaru. Dalam keseluruhan, visualisasi data

bukan hanya alat bantu, melainkan kunci untuk memahami dan memaksimalkan nilai data dalam konteks kecerdasan bisnis.

#### 2.2.10. Basis Data

Menurut (Conger, 2015), istilah “database” dapat diartikan sebagai kumpulan data yang mempunyai keterkaitan antar informasi. Perlu diketahui bahwa konsep dasar data tidak hanya terbatas pada format elektronik saja, namun dapat juga mencakup pencatatan manual, seperti buku catatan. Namun di era modern ini, istilah “database” lebih sering merujuk pada kumpulan data yang diuraikan secara elektronik dan diproses oleh komputer.

(Solihin, 2016) memandang database sebagai kombinasi informasi yang tersimpan. Informasi ini disusun dalam suatu sistem komputer sehingga dapat dikelola dan dikendalikan dengan menggunakan pemrograman komputer. Pembuatan database melibatkan entitas dan atribut. Entitas menurut (Conger, 2015) meliputi objek atau hal yang menjadi fokus perhatian pada data dasar, seperti siswa, transaksi, dan inventaris. Sedangkan atribut adalah nilai-nilai yang terkandung dalam suatu entitas.

Proses dasar pengelolaan data melibatkan sistem yang dikenal sebagai *Database Management System* (DBMS). DBMS memberi pengguna sarana untuk mengelola dan menggunakan database dengan mudah. Berbagai program atau perangkat lunak DBMS digunakan dalam industri perangkat lunak, dan salah satunya adalah MySQL.

Komponen pokok dalam sistem basis data digunakan untuk mendukung kemudahan pembuatan dan pengelolaan basis data (Lubis, 2016), dan ini mencakup:

a) Data

Data dalam sistem data memiliki dua karakteristik utama, yakni terintegrasi dan dapat digunakan bersama-sama.

- i. Terintegrasi mengacu pada penggabungan berbagai file dari aplikasi-aplikasi yang berbeda, dengan menghilangkan bagian yang redundan.
- ii. Dapat digunakan bersama-sama menunjukkan bahwa setiap bagian dalam database dapat diakses oleh pengguna secara simultan, terlepas dari aplikasi yang digunakan.

b) Perangkat Keras

Perangkat keras ini mencakup semua komponen perangkat keras komputer yang digunakan dalam pengelolaan sistem database, termasuk:

- i. Peralatan untuk penyimpanan seperti disk, drum, dan perangkat keras lainnya.
- ii. Peralatan input dan output.
- iii. Peralatan komunikasi data.

c) Perangkat Lunak

Perangkat lunak berperan sebagai antarmuka antara pengguna dan data fisik dalam

database, bisa berupa Database Management System (DBMS) atau aplikasi perangkat lunak beserta prosedurnya.

d) Pengguna

Pengguna dapat dibagi menjadi dua kelompok, yakni:

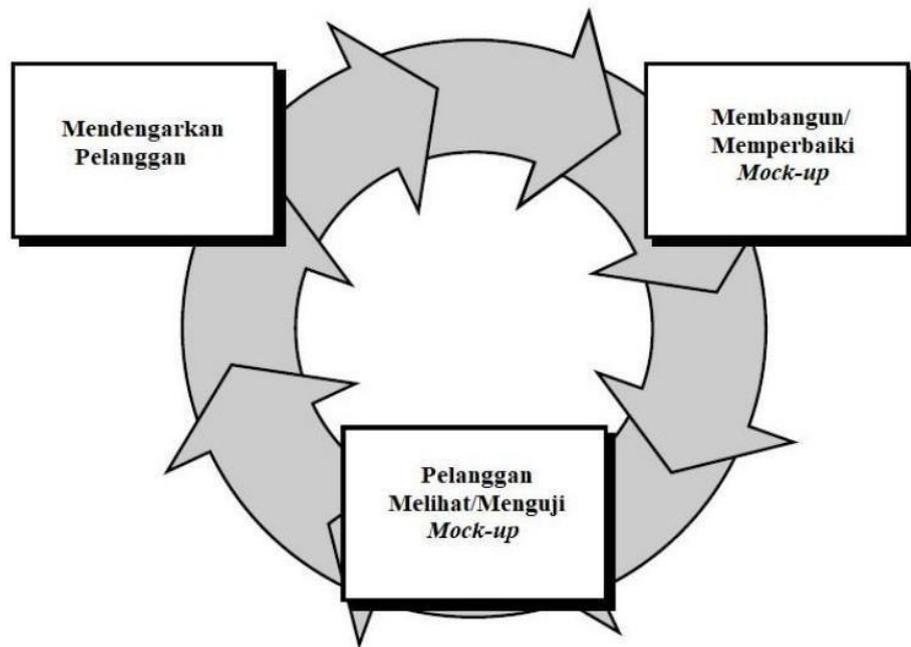
- i. Programmer, individu atau tim yang bertanggung jawab untuk membuat program aplikasi yang berinteraksi dengan database menggunakan bahasa pemrograman.
- ii. End user, individu yang mengakses database melalui terminal dengan menggunakan bahasa kueri atau program aplikasi yang telah dibuat oleh programmer.

### 2.2.11. Prototipe

Model prototipe digunakan dalam perancangan sistem informasi, memberikan kesempatan bagi pengembang program dan objek penelitian untuk saling berinteraksi selama proses perancangan (Sukanto & Shalahuddin, 2015). Yurindra (2017) menyatakan bahwa model prototipe adalah suatu proses yang memungkinkan pengembang membuat model perangkat lunak, terutama ketika klien tidak dapat memberikan informasi yang maksimal mengenai kebutuhan yang diinginkan.

Langkah-langkah penggunaan model prototipe melibatkan user, dimana pengembang program dan objek penelitian bertemu untuk menetapkan tujuan umum dan kebutuhan dasar, dengan detail kebutuhan yang mungkin dikumpulkan pada awal pengumpulan informasi. Kemudian, pembangunan atau perbaikan mock-up dilakukan setelah data terkait dikumpulkan selama tahap pengumpulan kebutuhan. Rancangan sistem ini menjadi dasar untuk pembuatan prototipe, yang merupakan tahapan perealisasi rancangan menggunakan bahasa pemrograman.

Selanjutnya, user memiliki kesempatan untuk melihat dan menguji mock-up atau prototipe yang telah dibuat. Objek penelitian dapat mengevaluasi prototipe tersebut untuk memperjelas kebutuhan perangkat lunak yang diinginkan. Proses ini memungkinkan iterasi dan penyesuaian sepanjang pengembangan sistem, sehingga menciptakan hasil akhir yang lebih sesuai dengan kebutuhan dan harapan pelanggan.



Gambar 2.3. Alur Prototipe

Sumber : (Sukanto & Shalahuddin, 2015)

### 2.2.12. *User Requirement*

Menurut (Maniah dan Hamidin, 2017), User Requirement atau Analisis Kebutuhan merupakan fase di mana interaksi terjadi antara pengguna akhir (end user) dan analis atau pengembang sistem. Proses pengumpulan dokumen kebutuhan pengguna melibatkan diskusi mendalam dengan tujuan mencapai kesepakatan antara pihak pengguna sistem dan pihak pengembang untuk menetapkan batasan-batasan sistem yang akan dibangun. Hasil dari proses ini diwujudkan dalam dokumen kebutuhan pengguna.

Menurut (Martono, Sucipto, dan Maulana, 2019), salah satu metode yang digunakan dalam menganalisis kebutuhan untuk membangun sistem atau perangkat lunak adalah metode elisitasi. Proses elisitasi sendiri melibatkan wawancara atau diskusi antara pengguna dan pengembang, dan terdiri dari tiga tahap sebagai berikut:

- A. Tahap elisitasi pertama mencakup usulan-usulan desain untuk sistem atau aplikasi yang akan dikembangkan oleh pihak pengguna.
- B. Tahap elisitasi kedua berfokus pada hasil pengelompokan dari tahap pertama menggunakan metode MDI (Mandatory, Desirable, Inessential). MDI bertujuan untuk memilah kebutuhan desain yang penting dan harus ada (Mandatory) dari kebutuhan lainnya.
- C. Tahap elisitasi ketiga merupakan hasil penyempurnaan dari tahap kedua

dengan menghilangkan semua user requirement dengan kategori Inessential (I) pada tahap MDI. Setelah itu, klasifikasi dilakukan menggunakan metode Technical, Operational, Economic (TOE).

D. Tahap elisitasi terakhir adalah hasil akhir dari seluruh tahap elisitasi yang akan menjadi pedoman dalam pengembangan sistem atau perangkat lunak oleh pihak pengembang. Ini mencakup kebutuhan fungsional dan non-fungsional, di mana kebutuhan fungsional melibatkan fasilitas internal atau di dalam sistem, sedangkan kebutuhan non-fungsional melibatkan fasilitas di luar sistem.

### 2.3. *Literature Review*

Literatur review dilakukan peneliti sebagai pendukung penelitian riset penelitian, literature review ini berkaitan dengan topik yang dibahas peneliti. Berikut beberapa referensi literature review :

Jurnal berjudul “**Analisis dan Perancangan Sistem Pengaduan Mahasiswa Berbasis Web (Studi Kasus: Universitas Mercu Buana Kranggan)**” oleh Melati Naomi dan Handrie Noprisson, diterbitkan dalam Jurnal Sistem Informasi dan E-Business (Jusibi), Volume 1, Edisi 5 pada bulan September 2019 membahas tentang pentingnya pengembangan sistem pengaduan berbasis web di Universitas Mercu Buana Kranggan.

Pada saat penelitian jurnal ini, pihak universitas masih menggunakan metode pengaduan manual atau tertulis, sehingga mengakibatkan proses yang panjang dan tidak efisien bagi mahasiswa. Mahasiswa harus datang langsung ke ruang humas untuk meminta formulir pengaduan, dan staf humas seringkali tidak selalu ada saat dibutuhkan. Kualitas layanan yang tidak memadai merupakan masalah serius yang perlu diatasi.

Pengembangan sistem pengaduan berbasis web merupakan solusi yang optimal. Sistem ini memungkinkan siswa untuk menyampaikan keluhan secara online, sehingga tidak perlu repot mengambil formulir manual. Selain itu, staf humas dapat dengan mudah mengakses dan menangani keluhan melalui platform web ini, sehingga meningkatkan daya tanggap dan efisiensi layanan.

Salah satu aspek yang diperhatikan dalam jurnal ini adalah tampilan aplikasi yang dibuat. Penelitian ini memberikan detail mendalam mengenai fitur-fitur aplikasi yang disediakan, termasuk tampilan setiap menu. Hal ini turut membuktikan keberhasilan pengembangan sistem dalam memenuhi kebutuhan Universitas Mercu Buana Kranggan.

Dengan sistem pengaduan berbasis web ini, perguruan tinggi dapat

mengelola pengaduan mahasiswa dengan lebih baik, meningkatkan transparansi, dan meningkatkan kualitas layanan. Selain itu, sistem ini juga memudahkan mahasiswa menyampaikan keluhan dan memantau perkembangannya secara online. Jurnal ini memberikan kontribusi yang berharga dalam pengembangan sistem informasi di lingkungan pendidikan dan dapat menjadi referensi bagi lembaga lain yang ingin meningkatkan layanan pengaduannya.

Jurnal berjudul **“Rancang Bangun Sistem Informasi Pengaduan Pelayanan Sarpras Berbasis Web di Universitas Teknologi Sumbawa”** yang ditulis oleh Herfandi Herfandi, Yuliadi Yuliadi, Sultan Naufal Abdillah, dan Eri Sasmita Susanto dan diterbitkan pada Jurnal Sains dan Teknologi Informatika (JINTEKS) Vol. . 3 Nomor 1 Februari 2021 membahas tentang pengembangan sistem informasi pengaduan layanan sarana dan prasarana berbasis web di Universitas Teknologi Sumbawa (UTS).

Sistem informasi yang dikembangkan dalam jurnal ini bertujuan untuk menyediakan sarana yang mudah diakses oleh pengguna untuk melaporkan permasalahan atau kebutuhan terkait sarana dan prasarana di UTS. Berbasis web, pengguna dapat mengakses sistem ini kapan saja dan dari mana saja, sehingga meningkatkan efisiensi dalam penanganan pengaduan.

Salah satu poin penting yang dicatat dalam jurnal ini adalah sistem telah melalui pengujian black box, yaitu metode pengujian perangkat lunak untuk menilai fungsionalitasnya tanpa mengetahui struktur internalnya. Hasil pengujian menunjukkan bahwa sistem ini berhasil dengan tingkat keberhasilan uji coba mencapai 100%. Ini merupakan indikasi yang sangat baik mengenai kualitas sistem yang sedang dikembangkan.

Selain itu, sistem informasi ini telah diterapkan di Biro Sarana dan Prasarana dan digunakan oleh civitas akademika di UTS. Keberhasilan implementasi menunjukkan bahwa sistem ini tidak hanya sekedar teori, namun juga dapat memberikan manfaat nyata dalam meningkatkan pelayanan sarana dan prasarana di UTS.

Dalam dunia pendidikan, pelayanan sarana dan prasarana yang baik sangatlah penting untuk menunjang proses belajar mengajar dan kegiatan kampus pada umumnya. Sistem informasi seperti yang dikembangkan dalam jurnal ini dapat menjadi alat yang berharga dalam meningkatkan transparansi, efisiensi dan daya tanggap dalam menangani pengaduan terkait sarana dan prasarana.

Dengan demikian, jurnal ini tidak hanya menyajikan konsep sistem informasi pengaduan layanan sarana dan prasarana berbasis web, namun juga membuktikan keberhasilannya dalam praktik, bermanfaat bagi UTS dan memberikan wawasan berharga dalam pengembangan sistem serupa di lembaga pendidikan lainnya.

Jurnal berjudul “**Survey Paper: Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)**” yang ditulis oleh Titania Pricillia dan Zulfachmi dari Jurnal Bangkit Indonesia pada Maret 2021, ditekankan pada pentingnya pemilihan Siklus Hidup Pengembangan Perangkat Lunak (SDLC) model yang cocok untuk pengembangan sistem. Salah satu pendekatan yang dibahas dalam jurnal ini adalah *Rapid Application Development (RAD)*.

RAD merupakan pendekatan yang menekankan kecepatan dan keterlibatan aktif pemangku kepentingan dalam pengembangan perangkat lunak. Di RAD, pengembang dengan cepat membuat prototipe perangkat lunak yang memungkinkan pemangku kepentingan melihat hasilnya terlebih dahulu dan memberikan umpan balik. Keunggulan RAD yang dibahas dalam jurnal ini mencakup beberapa aspek penting.

Pertama, RAD memungkinkan perkembangan yang cepat. Dengan pembuatan prototipe yang relatif singkat, tim pengembangan dapat mencapai tahap yang dapat dilihat oleh pemangku kepentingan lebih awal dibandingkan dengan metode pengembangan lainnya. Hal ini sangat berguna dalam menangani proyek-proyek yang memiliki batasan waktu yang ketat.

Kedua, RAD mengutamakan kolaborasi erat dengan pemangku kepentingan. Dalam proses pengembangan, pemangku kepentingan terlibat aktif dalam memberikan umpan balik, sehingga memastikan perangkat lunak yang dihasilkan benar-benar memenuhi kebutuhan dan harapannya.

Ketiga, RAD telah mengatur untuk mengakomodasi perubahan. Dengan prototipe yang dapat diubah dengan cepat, RAD memungkinkan perubahan kebutuhan atau persyaratan proyek tanpa mengganggu jadwal pengembangan secara signifikan.

Keempat, RAD memungkinkan optimalisasi pengembangan. Dengan penggunaan prototipe interaktif, pengguna akhir dapat merasakan dan mengidentifikasi masalah sebelum mencapai tahap produksi. Hal ini membantu mengurangi risiko proyek dan mendorong inovasi.

Dengan pendekatan RAD, perangkat lunak dapat dikembangkan lebih adaptif sesuai kebutuhan dan pemangku kepentingan. Hal ini membantu proyek perangkat lunak menjadi lebih responsif terhadap perubahan pasar dan kebutuhan pengguna. Oleh karena itu, dalam pemilihan model SDLC, penting untuk mempertimbangkan karakteristik RAD, terutama pada proyek yang memerlukan pengembangan cepat, kolaborasi aktif dengan pemangku kepentingan, persaudaraan, dan optimalisasi dalam pembangunan.

Jurnal berjudul **“Sistem Informasi Pengaduan Kekerasan Terhadap Perempuan dan Anak di DP3A Kota Ternate Berbasis Website Menggunakan PHP dan MySQL”** yang ditulis oleh Melissa Von Emster, Muksin Hi. Abdullah, dan Junaidi Sabtu dari Jurnal Aikom Ternate, Vol.1, No.1, Maret 2021 membahas tentang pengembangan sistem informasi yang bertujuan untuk meningkatkan efisiensi dalam mengelola dan mengakses data pengaduan kekerasan terhadap perempuan dan anak di Kota Ternate.

Latar belakang dikembangkannya sistem ini adalah adanya tuntutan masyarakat dan DP3A (Departemen Pemberdayaan Perempuan dan Perlindungan Anak) terhadap efisiensi dalam proses pengaduan. Sebelumnya, pengaduan kekerasan terhadap perempuan dan anak dilakukan secara konvensional sehingga seringkali memakan banyak waktu dan tenaga. Oleh karena itu, diperlukan sistem informasi yang lebih efisien untuk menggantikan sistem pengaduan yang konvensional.

Sistem informasi ini dirancang berbasis website dan menggunakan model DFD (*Data Flow Diagram*), ERD (*Entity Relationship Diagram*), serta bahasa pemrograman PHP dan *JavaScript*, dengan *database* MySQL. Pendekatan berbasis website memudahkan masyarakat mengakses informasi dan data terkait pengaduan kekerasan terhadap perempuan dan anak.

Implementasi rencana sistem pengaduan ini telah berhasil memberikan manfaat yang signifikan bagi masyarakat dan DP3A Kota Ternate. Sistem ini memudahkan pelaporan kejadian kekerasan dalam rumah tangga (KDRT) atau kasus kekerasan terhadap perempuan dan anak di luar lingkungan rumah tangga. Dengan sistem ini, pengaduan dapat disampaikan dengan lebih cepat dan efisien, sehingga memungkinkan pihak berwenang memberikan respons yang lebih baik terhadap kasus-kasus kekerasan.

Selain itu, sistem ini juga membantu pengumpulan dan analisis data yang lebih baik, yang dapat digunakan untuk merencanakan kebijakan dan intervensi yang lebih efektif dalam upaya melindungi perempuan dan anak dari

kekerasan. Oleh karena itu, jurnal ini memberikan kontribusi penting dalam upaya peningkatan perlindungan perempuan dan anak di Kota Ternate melalui pemanfaatan teknologi informasi yang canggih.

Jurnal berjudul **“Perancangan Sistem Informasi Aplikasi Kilat (Keluhan, Kritik dan Saran) Bagi Mahasiswa Berbasis Digital Studi Kasus: Mahasiswa Teknik Industri Universitas Muria Kudus”** oleh Dian Erliana Febriyanti, Nofia Mukarromah, dan Dina Tauhida, diterbitkan dalam Journal Of Industrial Teknik Dan Teknologi (Jointtech) Universitas Muria Kudus Jilid 1 Nomor 1 Desember 2020 membahas tentang pengembangan sistem informasi yang inovatif di lingkungan akademik.

Latar belakang pengembangan sistem ini berasal dari pengalaman yang terdapat pada Program Studi Teknik Industri Universitas Muria Kudus. Sebelumnya mahasiswa kesulitan dalam menyampaikan masukan, kritik, atau saran mengenai program studinya karena harus melalui Ketua Program Studi. Hal ini menghambat partisipasi mahasiswa dalam memberikan masukan berharga untuk meningkatkan mutu program studi.

Untuk mengatasi kendala tersebut Universitas Muria Kudus merasa perlu mengembangkan sistem yang memungkinkan pengumpulan masukan secara digital. Inisiatif ini menghasilkan aplikasi “Kotak Keluhan, Kritik, dan Saran” (PETIR) sebagai wadah bagi mahasiswa Teknik Industri untuk menyampaikan aspirasinya mengenai program studi. Meskipun telah memberikan manfaat, namun masih diperlukan perbaikan pada desain tampilan dan struktur aplikasi.

Dengan PETIR, mahasiswa mempunyai kemampuan lebih besar dalam menyampaikan pandangannya tanpa harus melalui Ketua Program Studi sebagai perantara. Inisiatif ini telah membantu mengungkap informasi berharga, yang pada gilirannya memungkinkan peningkatan kualitas program studi melalui solusi yang lebih inklusif dan responsif terhadap masukan mahasiswa.

Sistem ini merupakan contoh yang baik penerapan teknologi dalam pendidikan untuk meningkatkan partisipasi mahasiswa dalam proses pengembangan program studinya. Dengan PETIR, Universitas Muria Kudus membuka jalur komunikasi yang lebih efektif antara mahasiswa dan tenaga kependidikan, memastikan setiap suara mahasiswa didengar dan dipertimbangkan dalam upaya peningkatan mutu pendidikan secara berkelanjutan.