

BAB II TINJAUAN PUSTAKA

2. Landasan Teori

Untuk mendukung penulisan laporan ini selain memaparkan informasi umum yang di tulis pada Bab I, juga perlu di kemukakan beberapa teori yang berkaitan dengan masalah dan ruang lingkup pembahasan sebagai landasan dalam penulisan laporan ini.

2.1 Pengembangan

Menurut (seels & richey Alim Sumarno, 2012) pengembangan berarti proses menerjemahkan atau menjabarkan rancangan kedalan bentuk fitur fisik. Pengembangan secara khusus berarti proses menghasilkan bahan bahan pembelajaran.

2.2 Konsep Dasar Sistem Informasi

2.2.1 Sistem

Sistem menurut (Tata Sutabri,2016) merupakan suatu bentuk integrasi antara suatu komponen dengan komponen lain karena sistem memiliki sasaran yang berbeda untuk setiap kasus yang terjadi di dalam sistem tersebut.

2.2.2 Informasi

Menurut (Tata Sutarbi, 2016) Informasi adalah data yang telah di klarifikasi atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah data dari bentuk tak berguna menjadi bentuk yang berguna bagi penerimanya. Nilai informasi berhubungan dengan keputusan maka informasi maka informasi menjadi tidak diperlukan keputusan dapat berkisar dari keputusan berulang sederhana sampai keputusan strategis jangka panjang, nilai informasi dilukiskan paling berarti dalam konteks sebuah keputusan.

2.2.3 Sistem Informasi

Menurut (Tata Sutarbi, 2016) Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan- laporan yang diperlukan.

2.3 Monitoring

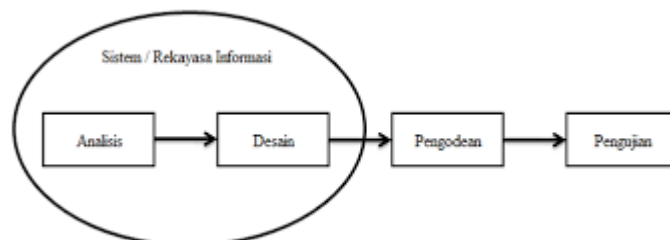
Menurut (Tata Sutabri, 2016) Monitoring didefinisikan sebagai langkah untuk mengkaji apakah kegiatan yang dilaksanakan telah sesuai dengan rencana, mengidentifikasi masalah yang timbul agar langsung dapat diatasi, melakukan penilaian apakah pola kerja dan manajemen yang digunakan sudah tepat untuk mencapai tujuan, mengetahui kaitan antara kegiatan dengan tujuan untuk memperoleh ukuran kemajuan.

2.4. Web Mobile

(Indonesian Journal on Networking and Security, 2016) web mobile adalah lanjutan dari aplikasi tradisional menggunakan smartphone, tablet, atau komputer, desainnya akan beradaptasi dengan perangkat yang digunakan tersebut, dengan desain yang fleksibel ini memaksimalkan kemampuan web browser yang digunakan di perangkat mobile.

2.5 Metode Pengembangan Waterfal

Menurut (Rosa, A.S.M. Shalahuddin, 2019) Model SDLC air terjun (waterfall) sering juga disebut model sequential linier (sequential linier) atau alur hidup klasik (classic life cycle). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sequential atau terturut dimulai dari analisis, desain, pengodean, dan tahap pendukung (support). Berikut adalah gambar model air terjun:



Gambar 2.1 Model Waterfal

1. Analisis kebutuhan perangkat lunak
Proses pengumpulan kebutuhan dilakukan secara insensif untuk mengspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user.
2. Desain
Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan seperti desain antarmuka. Tahapan

perancangan sistem mengalokasikan kebutuhan-kebutuhan sistem baik perangkat keras maupun perangkat lunak dengan membentuk arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan penggambaran abstraksi sistem dasar perangkat lunak dan hubungannya.

3. Pembuatan kode program

Desain harus ditranslasikan kedalam program perangkat lunak hasil dari tahap ini adalah aplikasi mobile yang sesuai dengan desain yang telah dibuat pada tahapan desain.

4. Pendukung *support* atau pemeliharaan *maintenance*

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.5.1 Kelebihan Metode Waterfall

Kelebihan menggunakan metode air terjun (waterfall) adalah metode ini memungkinkan untuk departementalisasi dan kontrol proses pengembangan *model fase one by one*, sehingga meminimalis kesalahan yang mungkin akan terjadi. Pengembangan bergerak dari konsep, yaitu melalui desain, implementasi, pengujian, instalasi, penyelesaian masalah, dan berakhir di operasi dan pemeliharaan.

2.5.2 Kekurangan Metode Waterfall

Kekurangan menggunakan metode *waterfall* adalah metode ini tidak memungkinkan untuk banyak revisi jika terjadi kesalahan dalam prosesnya. Karena setelah aplikasi ini dalam tahap pengujian, sulit untuk kembali lagi dan mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep sebelumnya.

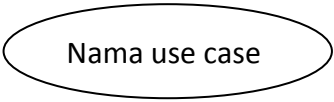
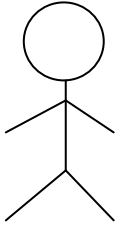

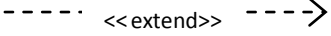
2.6 Definisi *Unified Modeling Language* (UML)

Menurut (Rosa A.S M. Shalahudin, 2019) UML *Definisi Unified Modeling Language* adalah setandar bahasa yang banyak digunakan didunia industri untuk mendefinisikan *requerement*, membuat analisis, & desain, serta menggambarkan arsitektur dalam pemograman berorientasi objek. UML hanya berfungsi untuk melakukan pemodelan, jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek. Perkembangan penggunaan UML bergantung pada level abstraksi penggunaannya, dalam penggunaan UML adalah suatu yang salah, tapi perlu ditelaah dimanakah UML digunakan dan hal apa yang ingin divisualkan. Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori.

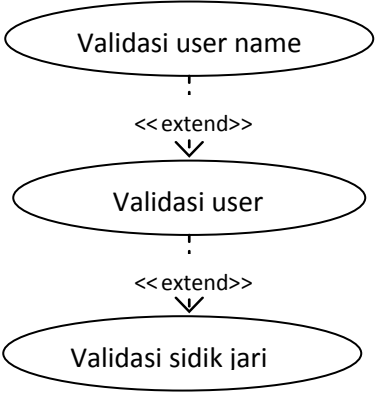

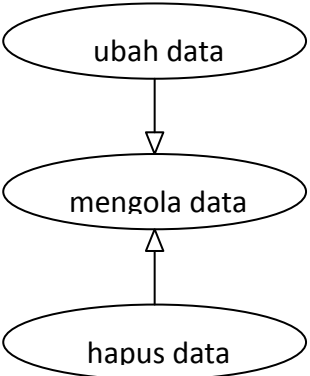
a. *Use Case Diagram*

Merupakan pemodelan untuk melakukan (behavior) sistem informasi yang akan dibuat. Usecase mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. secara kasar usecase digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Simbol dan keterangan use case diagram seperti pada tabel 2.1

Tabel 2.1 : Simbol dan Keterangan *Use case* Diagram.

No.	Simbol	Deskripsi
1.	<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>.</p>
2.	<p>Aktor/<i>actor</i></p> 	<p>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 di awal frase nama aktor.</p>
3.	<p>Assosiasi/<i>association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
4.	<p>Exstensi/<i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan kesebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu, mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>

Tabel 2.1 : Simbol dan Keterangan *Use case* Diagram (Lanjutan)

		 <p>Arah panah mengarah pada use case yang ditambahkan, biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5	<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua <i>buah use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>

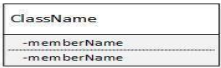



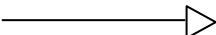
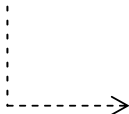
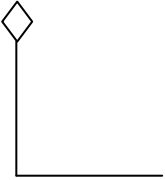
Tabel 2.1 : Simbol dan Keterangan *Use case* Diagram (Lanjutan).

		Kedua interpretasi di atas dapat dianut salah satu atau keduanya atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.
--	--	---

b. Class Diagram

Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Diagram kelas dibuat agar pembuatan perogram membuat kelas-kelas sesuai rancangan didalam diagram kelas antara dokumentasi perancangan dan perangkat lunak sinkron. Berikut ini adalah simbol-simbol yang ada pada *Class* Diagram. Simbol dan keterangan use case diagram seperti pada tabel 2.2






Tabel 2.2 : Simbol dan Keterangan *class* Diagram.

No.	Simbol	Deskripsi
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka/ <i>interface</i> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	Asosiasi/ <i>association</i> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah/ <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umumkhusus)
6.	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7.	Agrpasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

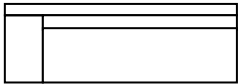
c. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol dan keterangan use case diagram seperti pada tabel 2.3

Tabel 2.3 : Simbol dan Keterangan *Activity Diagram*.

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

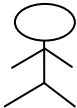
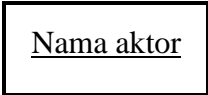

Tabel 2.3 : Simbol dan Keterangan Activity Diagram (Lanjutan).

6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
----	---	--



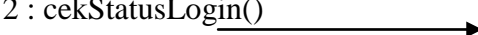



d. Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek atau message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlibat dalam sebuah usecase beserta metode-metode yang dimiliki kelas yang di instansiasi menjadi objek itu. Simbol dan keterangan use case diagram seperti pada tabel 2.4


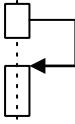

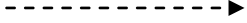
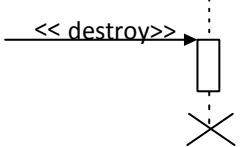
Tabel 2.4 : Simbol dan Keterangan Sequence Diagram.

No.	Simbol	Deskripsi
1.	Aktor  Atau  Tanpa waktu aktif	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 dalam menggunakan kata benda diawal frase nama aktor.
2.	Garis hidup/ <i>lifeline</i> 	Menyatakan kehidupan suatu objek

Tabel 2.4 : Simbol dan Keterangan *Sequence* Diagram (Lanjutan).

3.	<p>Objek</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <u>Nama objek : nama kelas</u> </div>	<p>Menyatakan objek yang berinteraksi pesan</p>
4.	<p>Waktu aktif</p> <div style="text-align: center; margin: 10px 0;">  </div>	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> <div style="margin-left: 40px;">  </div> <div style="margin-left: 40px;">  <p>2 : cekStatusLogin()</p> </div> <div style="margin-left: 40px;">  <p>1: login()</p> </div> <div style="margin-left: 40px;">  <p>3 : open()</p> </div> <p>Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif.</p>
5.	<p>Pesan tipe <i>create</i></p> <div style="text-align: center; margin: 10px 0;"> <p><<create>></p>  </div>	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>

Tabel 2.4 : Simbol dan Keterangan *Sequence* Diagram (Lanjutan).

6.	<p>Pesan tipe <i>call</i></p> <p>1 : nama_metode()</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>1 : nama_metode()</p> <p>Arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p>
7.	<p>Pesan tipe <i>send</i></p> <p>1 : masukkan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
8.	<p>Pesan tipe <i>return</i></p> <p>1 : keluaran</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian</p>
9.	<p>Pesan tipe <i>destroy</i></p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i></p>

2.7 Teori Program

2.7.1 Basis Data

Menurut (Rosa A.S M, Shalahudin, 2019) Basis Data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah *file text* ataupun *Database Management System* (DBMS).

2.7.2 MySQL

Menurut (Rosa A.S M. Shalahuddin, 2019) SQL (Structured Query Language) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

2.7.3 PHP (Hypertext Prerocessor)

Menurut (Saputra, 2016) menyatakan bahwa "pada dasarnya PHP dapat di definisikan sebagai suatu bahasa pemograman yang difungsikan untuk rnebangun suatu website dinamis". PHP menyatu dengan kode HTML maksudnya adalah beda kondisi. HTML digunakan sebagai pernbangun atau pondasi dari kerangka layout web, sedangkan php difungsikan sebagai prosesnya sehingga dengan adanya php tersebut, sebuah web akan sangat mudah di- maintance.

2.7.4 HTML (*Hyper Text Markup Language*)

(Rosa A.S, M. Shalahuddin, 2016) HTML singkatan dari *Hyper Text Markup Language* yaitu skrip yang mengatur berupa tag-tag untuk membuat dan mengatur struktur website. Beberapa tugas utama HTML dalam membangun website diantaranya sebagai berikut:

- Menentukan layout website.
- Memformat text dasar seperti pengaturan paragraf, dan format font.
- Membuat list.
- Membuat table.
- Menyisipkan gambar, video, dan audio.
- Membuat link.
- Membuat formulir

2.7.5 XAMPP

Menurut (Ginting, 2016) XAMPP adalah perangkat lunak gratis yang mendukung banyak sistem operasi dan merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang terdiri dari (localhost), yang terdiri atas program Apache HTTP server, MySQL database, dan penterjemah bahasa yang ditulis dengan 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. Xampp merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis.

2.7.6 CSS (Cascading Style Sheet)

Menurut (Kustiyaningsih dan Anamisa, 2016) menyatakan bahwa "pada dasarnya CSS dapat didefinisikan sebagai kumpulan kode-kode yang berurutan dan saling berhubungan untuk mengatur format atau tampilan suatu halaman HTML".