



BAB II

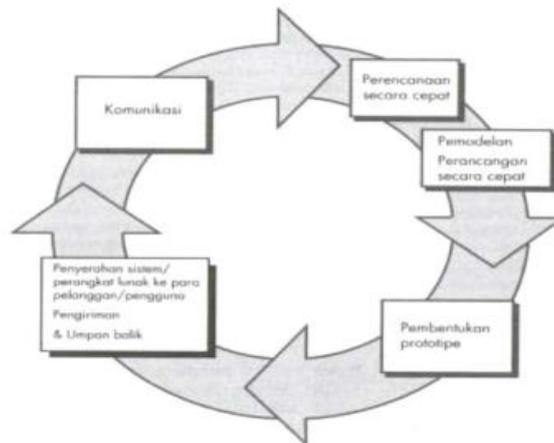
DASAR TEORI

BAB II LANDASAN TEORI

2.1 Landasan Teori

2.1.1 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak (*software engineering*) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Rekayasa perangkat lunak dibutuhkan agar perangkat lunak yang dibuat tidak hanya menjadi perangkat lunak yang tidak terpakai. Rekayasa perangkat lunak lebih fokus pada praktik pengembangan perangkat lunak dan mengirimkan perangkat lunak yang bermanfaat kepada pelanggan [3]. Rekayasa Perangkat Lunak (*software engineering*) adalah aplikasi dari pendekatan rekayasa (*engineering*) yang sistematis, bersifat keilmuan dan terukur terhadap pengembangan, operasi dan perawatan perangkat lunak



Gambar 2.1 Metode *Prototype* Menurut Pressman

1. Metode

Metode yang digunakan untuk sistem ini adalah metode *prototype*. Metode pengembangan sistem dengan model *prototype*, dimana metode *prototype* adalah sebuah metode pengembangan software yang banyak digunakan pengembang agar dapat saling berinteraksi dengan *user* selama proses pembuatan sistem [4]. Dalam model *Prototype*, sebuah *prototype* perangkat lunak dibuat dan kemudian dipresentasikan kepada pelanggan. Pelanggan diberi kesempatan untuk memberikan masukan, sehingga perangkat lunak yang dihasilkan sesuai dengan keinginan dan kebutuhan pelanggan [5]. Pada tahapan metode *Prototype* antara lain :

a. Komunikasi (*Communication*)

Dalam tahapan komunikasi, penting untuk memperoleh pemahaman yang jelas tentang kebutuhan sistem dari pengguna. Proses ini melibatkan interaksi antara pengembang dan pengguna untuk menggambarkan dengan tepat alur sistem yang diinginkan. Analisis menyeluruh terhadap kebutuhan pengguna menjadi landasan penting dalam tahapan awal pembangunan sistem.

b. Perencanaan (*Quick Plan*)

Dalam perencanaan awal, fokusnya adalah menganalisis kebutuhan sistem dengan mempertimbangkan aspek teknologi dan analisis pengguna. Langkah pertama yang dilakukan adalah mengidentifikasi kebutuhan yang spesifik dan menganalisis kebutuhan dalam merancang sistem.

c. Permodelan Perancangan (*Modelling Quick Design*)

Dalam tahapan permodelan perancangan, dilakukan perancangan terhadap alur kerja sistem yang akan dibuat. Selain itu, juga merancang aktor - aktor dan proses-proses yang akan

berinteraksi pada sistem, yang sesuai dengan analisis kebutuhan yang telah dianalisis sebelumnya. Langkah ini penting untuk memastikan bahwa rancangan sistem akan sesuai dengan kebutuhan pengguna.

d. Pembentukan *Prototype*

Pada pembentukan *Prototype*, dilakukan implementasi rancangan prototipe dalam bentuk penulisan program. Selanjutnya, sistem yang telah dibuat akan dilakukan pengujian unit.

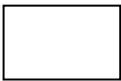
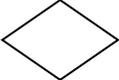
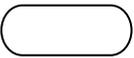
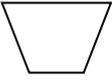
e. Penyerahan sistem (*Deployment Delivery & Feedback*)

Penyerahan sistem dilakukan untuk memperoleh umpan balik dari pengguna sistem. Jika sistem telah terbukti baik dan sesuai dengan kebutuhan pengguna serta nilai *prototype* telah dilakukan dengan memperhatikan kebutuhan pengguna, maka sistem akan diserahkan secara penuh kepada pengguna untuk digunakan. Langkah ini pastikan bahwa implementasi sistem dilakukan sesuai dengan kebutuhan pengguna.

2. *Flowchart*

Flowchart atau diagram alir adalah sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing-masing langkah tersebut dengan menggunakan panah. Diagram ini bisa memberikan solusi selangkah demi selangkah untuk penyelesaian masalah yang ada didalam proses atau algoritma tersebut [3]. Simbol-simbol yang digunakan dalam *Flowchart* dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol *Flowchart*

No	Nama	Simbol	Deskripsi
1.	<i>Input / Output</i>		Simbol ini menyatakan proses <i>input</i> data atau <i>output</i> data yang diproses atau informasi.
2.	Proses		Simbol ini menyatakan proses perhitungan atau proses pengolahan data.
3.	Garis Aliran(<i>Flow Line</i>)		Simbol ini menyatakan arah alur kerja dalam konsep (prosedur).
4.	Keputusan (<i>Decision</i>)		Simbol ini menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban yaitu ya atau tidak.
5.	<i>Document</i>		Simbol ini menunjukkan <i>Input</i> berasal dari dokumen dalam bentuk kertas / <i>Output</i> dalam format yang dicetak ke kertas.
6.		Simbol ini menyatakan awal atau akhir suatu program.	
7.	<i>Manual Operation</i>		Simbol ini menunjukkan pengolahan operasi manual atau tidak dilakukan oleh komputer.
8.	<i>Predefined process</i> (Sub Program)		Simbol ini menunjukkan permulaan sub program atau proses menjalankan sub program.

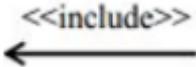
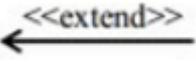
3. UML

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan gambar untuk bahasa yang berdasarkan gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasikan dari sebuah sistem pengembangan perangkat lunak berbasis objek. UML (*Unified Modeling Language*) bukanlah merupakan bahasa pemrograman tetapi model-model yang dicipta berhubungan langsung berbagai macam bahasa pemrograman [6]. UML adalah sesuatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. Dengan ini, UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi [7].

a. Use Case Diagram

Use case atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar *use case* digunakan digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu [3]. *Use case diagram* adalah mendeskripsikan fungsi-fungsi yang dibutuhkan dari sebuah sistem yang dibangun dan juga berfungsi sebagai sistem untuk bertukar pesan antar unit dan aktor [8]. Notasi-notasi simbolik di dalam *diagram use case* yang digunakan ada pada Tabel 2.2.

Tabel 2.2 Simbol *Use case Diagram*

No.	Nama	Simbol	Deskripsi
1.	<i>Actor</i>		simbol ini mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i> .
2.	<i>Use case</i>		simbol ini menyatakan abstraksi dan interaksi antara sistem dan aktor.
3.	<i>Include</i>		simbol ini menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.
4.	<i>Extend</i>		simbol ini menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi.
5.	<i>Association</i>		simbol ini menunjukkan abstrak dari penghubung antara aktor dengan <i>use case</i> .
6.	<i>Generalization</i>		simbol ini menunjukkan bahwa spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i> .

2.1.2 Pemrograman Berorientasi Objek

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana suatu sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas [9]. Berikut ini adalah beberapa konsep dasar yang harus dipahami tentang metodologi berorientasi objek :

- a) Kelas (*class*)
Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai (atribut). Kelakuan (operasi/metode), hubungan (*relationship*) dan arti suatu kelas dapat diturunkan dengan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.
- b) Objek (*object*)
Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya, dimanipulasi dan dihancurkan.
- c) Metode (*method*)
Operasi atau metode atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.
- d) Atribut (*attribute*)
Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek, misalnya berat, jenis, nama, dan sebagainya. Atribut sebaliknya bersifat privat untuk menjaga konsep enkapsulasi.
- e) *Generalisasi* dan Spesialisasi
Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus. Misalkan kelas yang lebih umum (*generalisasi*) adalah kendaraan darat dan kelas khususnya (*spesialisasi*) adalah mobil, motor, dan kereta.

2.1.3 Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, berupa file teks ataupun *Database Management System (DBMS)* [3]. Basis data adalah kumpulan data yang terorganisir dengan baik dan terstruktur di dalam suatu sistem komputer. Basis data terdiri dari beberapa tabel yang terhubung dengan relasi atau hubungan tertentu. Basis data digunakan untuk menyimpan, mengelola, dan mengakses data dengan efisien [10].

a. ERD

ERD adalah suatu *diagram* untuk menggambarkan desain konseptual dari model konseptual suatu basis data relasional. ERD juga merupakan gambaran yang merelasikan antara objek yang satu dengan objek yang lain dari objek di dunia nyata yang sering dikenal dengan hubungan antar entitas [11]. Simbol-simbol yang digunakan dalam pembuatan ERD dapat dilihat pada Tabel 2.3.

Tabel 2.3 Simbol ERD

No.	Nama	Simbol	Keterangan
1.	Entitas (<i>Entity</i>)		Entitas berfungsi untuk memberikan identitas pada entitas yang memiliki label dan nama. Entitas memiliki bentuk persegi panjang.
2.	Relasi		Relasi, simbol ini menunjukkan nama relasi antar satu entitas dengan entitas lainnya
3.	Atribut		Digunakan untuk menggambarkan elemen-elemen dari suatu <i>entity</i> , yang menggambarkan karakter <i>entity</i> .
4.	Atribut <i>Multivalued</i>		Suatu entitas yang dapat memiliki lebih dari satu
5.	Garis		Digunakan untuk menghubungkan <i>entity</i> dengan relasi/hubungan, maupun <i>entity</i> dengan atribut.
6.	Kebergantungan / <i>dependency</i>		Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi / <i>Aggregation</i>		Relasi antar kelas dengan makna semua bagian (<i>whole part</i>).

2.1.4 Sistem Informasi

Sistem informasi adalah suatu sistem organisasi yang mempertemukan kebutuhan pengelolaan transaksi harian, mendukung operasi, bersifat manajemen, dan kegiatan startegi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan [12]. Sistem informasi juga merupakan kumpulan antara sistem yang saling berhubungan yang membentuk suatu komponen yang didalamnya mencakup *input-proses-output* yang berhubungan dengan pengolahan data.

2.1.5 Monitoring

Monitoring merupakan suatu kegiatan mengamati secara seksama suatu keadaan atau kondisi, termasuk juga perilaku atau kegiatan tertentu. Tujuan adalah agar semua data masukan atau informasi yang diperoleh dari hasil pengamatan tersebut dapat menjadi landasan dalam mengambil keputusan tindakan selanjutnya yang diperlukan [13]. Dan juga *monitoring* adalah proses pengamatan, pengukuran, dan menilai terus-menerus atau kondisi, sistem, atau ke atau kegiatan tersebut sesuai dengan tujuan yang telah diterapkan.

2.1.6 Posko PB (Penanggulangan Bencana)

Posko merupakan ruang pusat yang mengendalikan kegiatan PMI dalam menyebarkan informasi kepada pemerintah dan masyarakat. Sementara itu, Posko PB (Penanggulangan Bencana) berfungsi sebagai pusat pengendalian kegiatan penanggulangan bencana yang meliputi kegiatan yang merencanakan, mengkoordinasikan dan pemantauan (*monitoring*). Posko darurat lapangan adalah pusat pengendalian operasional kegiatan tanggap darurat di lokasi bencana.

2.1.7 Bencana Alam

Menurut Undang-undang No.24 Tahun 2007 tentang Penanggulan Bencana menyebutkan bahwa pengertian Bencana Alam adalah rangkaian peristiwa yang mengancam dan mengganggu kehidupan dan penghidupan masyarakat yang disebabkan, baik oleh faktor alam atau faktor non-alam maupun faktor manusia sehingga mengakibatkan timbulnya korban jiwa manusia, kerusakan lingkungan, kerugian harta benda dan bencana sosial [14]. Bencana alam berupa banjir, tsunami, gunung meletus, longsor.