



BAB II

DASAR TEORI

BAB II DASAR TEORI

2.1 Dasar Teori

2.1.1 Rekayasa Perangkat Lunak

Rekayasa Perangkat Lunak (RPL) adalah suatu disiplin ilmu yang mengulas semua aspek produksi perangkat lunak, dimulai dari tahap awal yaitu *communication*, *requirements capturing* (analisa kebutuhan pengguna), *specification* (menentukan spesifikasi dari kebutuhan pengguna), desain, *coding*, *testing* hingga *maintenance* (pemeliharaan sistem) setelah digunakan [1].

2.1.2 Object – Oriented Programming

Object-Oriented Programming merupakan metode pemrograman yang berorientasi kepada objek. Dalam OOP, programmer yang sama atau berbeda dapat mengembangkan sistem yang ada dimana class dan obyek dapat digunakan berulang-ulang atau menambahkan obyek baru pada sistem yang ada. Jika terjadi permasalahan, sistem lebih mudah diatasi. Untuk pengembangan sistem yang kompleks, bahasa pemrograman yang mendukung *Object Oriented Programming* sangat efektif digunakan [2].

2.1.3 Desktop Based

Desktop Based adalah aplikasi berbasis *desktop* yang berjalan pada komputer (terhubung dalam satu jaringan) secara independen dan tidak memerlukan *browser*[3]. Beberapa hal yang harus diperhatikan dalam membangun sebuah sistem berbasis desktop di antaranya :

- a. *Hardware* yang akan digunakan nantinya ketika sistem tersebut sudah menjadi program yang utuh haruslah sesuai dengan kebutuhan dan dengan spesifikasi yang lebih tangguh lagi.
- b. *Server* yang akan digunakan untuk penyimpanan data dari klien saat mengirimkan data penting harus dengan spesifikasi yang kuat agar tidak terjadi error/hang saat diakses
- c. Memiliki infrastruktur jaringan yang tidak terlalu besar karena aplikasi hanya diakses di jaringan dalam (intranet) saja sudah cukup.
- d. Tidak dapat diakses dari jaringan luar (internet) jika ingin menunjukkan laporan yang telah diinput dan dihasilkan oleh sistem/aplikasi tersebut.
- e. Jika aplikasi yang sudah diinstall di PC maupun *server*, sangat susah untuk melakukan *update* pada sistemnya jika masih ada kekurangan dalam pengaksesannya dan membutuhkan waktu yang cukup lama.

2.1.4 Bahasa Pemrograman *C Sharp*

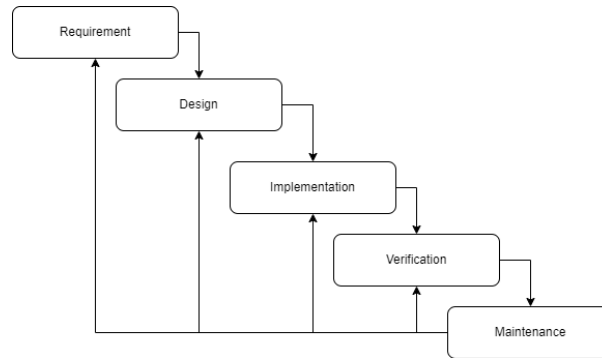
Bahasa Pemrograman *C Sharp* memiliki bentuk umum yang hampir sama dengan bahasa C, C++ dan Java. *C Sharp* merupakan jenis bahasa pemrograman yang berorientasi pada Objek (*Object Oriented Programming*). Bahasa pemrograman *C Sharp* sangat sederhana dan dapat digunakan untuk pembuatan aplikasi berbasis *desktop*, *mobile*, aplikasi *game* dan mendukung konsep Pemrograman Berorientasi Objek (PBO)[4].

A. Metode Pengembangan Perangkat Lunak

Metode adalah tahap-tahap ataupun aturan untuk melakukan sesuatu. *System Development Life Cycle* adalah sebuah proses logika yang digunakan oleh *system analyst* untuk mengembangkan sebuah sistem yang melibatkan *requirements*, *training*, dan pemilik sistem.

System Development Life Cycle (SDLC) atau siklus hidup pengembangan sistem dalam rekayasa sistem dan rekayasa perangkat lunak adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. SDLC juga merupakan pola untuk mengembangkan sistem perangkat lunak yang terdiri dari tahapan perencanaan (*planning*), analisis (*analyst*), desain (*design*), implementasi (*implementation*), uji coba (*testing*), dan pengelolaan (*maintenance*).

Model *Waterfall* merupakan salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model ini menggunakan pendekatan sistematis dan berurutan. Tahapan dalam model ini dimulai dari tahap perencanaan hingga tahap pengelolaan (*maintenance*) dan dilakukan secara bertahap. Pengembang perlu mengetahui lebih lanjut tentang bagaimana proses pengembangan sistem jika menggunakan model *waterfall* dan juga karakteristik dari model *waterfall* tersebut. Tahapan dari metode *waterfall* oleh pressman dapat dilihat pada gambar 2.1



Gambar 2. 1 Metode *Waterfall* Oleh Pressman

1. *Requirement*

Pada tahapan ini pengembang sistem memerlukan komunikasi yang memiliki tujuan untuk memahami perangkat lunak seperti apa yang diharapkan oleh pengguna. pengembang juga menentukan batasan pada perangkat lunak tersebut. Kemudian, informasi-informasi yang dibutuhkan dapat diperoleh dari pengumpulan data baik melalui wawancara, diskusi, maupun survei secara langsung. Informasi yang telah diperoleh dilakukan analisis untuk mendapatkan data yang benar-benar dibutuhkan oleh pengguna [5].

2. *Design*

Pada tahap ini, pengembang sistem akan membuat desain terkait sistem yang akan dibuat. Desain ini dapat digunakan untuk membantu menentukan perangkat keras (*hardware*) dan membantu mendefinisikan arsitektur sistem secara keseluruhan [5].

3. *Implementation*

Pada tahap ini, sistem pertama kali dikembangkan di program kecil yang disebut unit, yang terintegrasi dalam tahap selanjutnya. Setiap unit dikembangkan dan diuji untuk fungsionalitas yang disebut sebagai unit *testing* [5].

4. *Verification*

Pada tahap ini, akan dilakukan verifikasi pada sistem serta pengujian apakah sistem yang dibuat memenuhi persyaratan atau tidak. Pada tahap ini pula, akan dilakukan pengujian yang dikategorikan ke dalam unit *testing* (dilakukan pada modul tertentu dari kode), sistem pengujian (guna melihat cara kerja sistem dalam bereaksi ketika semua modul terintegrasi), dan penerimaan pengujian (dilakukan melalui pelanggan untuk melihat apakah seluruh kebutuhannya terpenuhi dan mendapat kepuasan) [5].

5. *Maintenance*

Pada tahap inilah akhir dari metode *waterfall*. Perangkat lunak yang telah jadi akan dijalankan dan selalu dilakukan pemeliharaan. Pemeliharaan yang ada salah satunya ialah melakukan perbaikan jika ada kesalahan yang sebelumnya tidak pernah terjadi [5].

B. Tools / Alat Bantu

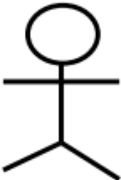
1. UML (*Unified Modeling Language*)






UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan *blueprint* sebuah *software*[6]. UML diharapkan mampu mempermudah pengembangan piranti lunak (RPL) serta memenuhi semua kebutuhan pengguna dengan efektif, lengkap, dan tepat.

a. *Use Case Diagram*

Dalam pemrograman berbasis OOP, untuk mendeskripsikan sistem dapat menggunakan diagram UML. Diagram tersebut terdiri dari 13 jenis diagram yaitu *activity*, *class*, *communication*, *component*, *composite structure*, *deployment*, *interaction overview*, *object*, *package*, *sequence*, *state machine*, *timing* dan *use case*. *Use case* merupakan deskripsi fungsi dari sebuah sistem dari perspektif atau sudut pandang para pengguna sistem. *Use case* mendefinisikan apa yang akan diproses oleh sistem dan komponen-komponennya. *Use case* bekerja dengan menggunakan scenario yang merupakan deskripsi dari urutan atau langkah-langkah yang menjelaskan apa yang dilakukan oleh user terhadap sistem maupun sebaliknya. *Use case* mengidentifikasi fungsionalitas yang dimiliki sistem, interaksi user dengan sistem dan keterhubungan antara user dengan fungsionalitas sistem[7]. Simbol-simbol *use case* dapat dilihat pada tabel 2.1.

Tabel 2. 1 Simbol - simbol Usecase

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer.
2.		<i>Extend</i>	Relase <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri

			sendiri walau tanpa <i>use case</i> tambahan itu.
3.		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan melalui <i>use case</i> ini untuk menjalankan fungsinya untuk sebagai syarat dijalankan <i>use case</i> ini.
4.		<i>Use Case</i>	Menjelaskan yang dilakukan actor dari sistem untuk mencapai tujuan tertentu.
5.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6.		<i>System</i>	Menspesifikasikan data paket yang menampilkan sistem secara terbatas

b. *Flowchart*




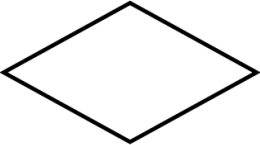
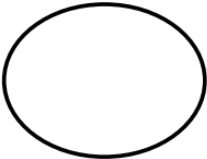
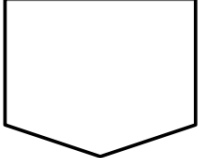
Flowchart dapat diartikan sebagai langkah langkah penyelesaian masalah yang di tuliskan dalam suatu simbol-simbol tertentu. Diagram alir ini akan menunjukkan alur di dalam program secara logika” [10]. *Flowchart* ini diperlukan tidak hanya sebagai alat komunikasi tetapi juga sebagai pedoman, dan sebelum komponen-komponennya dapat lebih dipahami, perlu dikomunikasikan aturan-aturan desain *org chart*, yaitu:

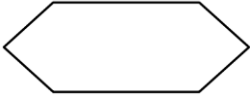

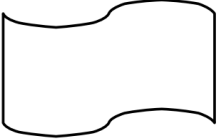

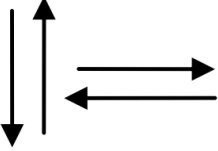
1. *Flowchart* digambarkan dengan top orientasi ke bawah dan kiri ke kanan.
2. Setiap aktivitas atau proses dalam bagan organisasi harus dinyatakan dengan jelas atau tidak ambigu.
3. Setiap diagram alur harus dimulai dari awalan atau status awal dan diakhiri dengan satu atau lebih status terminal/akhir/hals.
4. Gunakan konektor Status Halaman dan konektor Keluar halaman dengan label yang sama untuk menunjukkan bahwa koneksi antar algoritme terputus, misalnya karena perpindahan/perubahan halaman.

Flowchart digunakan untuk menggambarkan langkah atau pemecahan masalah secara sederhana, dapat dimengerti, rapi dan tidak ambigu dengan menggunakan beberapa simbol-simbol yang bisa dibidang standart merupakan tujuan dari *flowchart*. Simbol-

simbol *flowchart* dapat dilihat pada tabel 2.2.

Tabel 2. 2 Simbol-simbol Flowchart

No.	Simbol	Nama	Keterangan
1.		<i>Terminal</i>	Awal atau akhir suatu program (Prosedur).
2.		<i>Output/Input</i>	Proses input atau output terlepas dari jenis perangkat.
3.		<i>Process</i>	Proses operasional <i>computer</i> .
4.		<i>Decision</i>	Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan, ya/tidak
5.		<i>Connector</i>	Koneksi penghubung proses ke proses lain pada halaman yang sama
6.		<i>Offline Connector</i>	Koneksi penghubung dari satu proses ke proses lain di halaman lain

7.		<i>Predefined Process</i>	Mewakili ketentuan penyimpanan untuk diproses untuk memberikan awal harga
8.		<i>Punched Card</i>	Input berasal dari kartu atau <i>output</i> ditulis ke kartu.
9.		<i>Punch Tape</i>	-
10.		<i>Document</i>	Mencetak output dalam format dokumen (melalui <i>printer</i>)
11.		<i>Flow</i>	Menyatakan jalannya arus suatu proses

c. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram berbentuk notasi grafis yang berada dalam pembuatan database yang menghubungkan antara data satu dengan yang lain[8]. Fungsi ERD adalah sebagai alat bantu dalam pembuatan database dan memberikan gambaran bagaimana kerja database yang akan dibuat. Di dalam ERD terdapat 3 elemen dasar, yaitu entitas, atribut, dan relasi.

1. Entitas

Entitas merupakan objek yang akan menjadi perhatian dalam suatu *database*. Entitas dapat berupa manusia, tempat, benda, atau kondisi mengenai data yang dibutuhkan. Simbol dari entitas berbentuk persegi panjang.

2. Atribut

Atribut merupakan informasi yang terdapat dalam entitas. Sebuah entitas harus memiliki *primary key* sebagai ciri khas entitas dan atribut deskriptif. Atribut biasanya

terletak dalam tabel entitas atau dapat juga terpisah dari tabel. Simbol dari atribut berbentuk elips.

3. Relasi

Relasi di dalam ERD merupakan hubungan antara dua atau lebih entitas. Simbol dari relasi berbentuk belah ketupat.

Relasi yang dapat dimiliki oleh ERD ada beberapa macam, yaitu :

1. *One to One*

Satu anggota entitas dapat berelasi dengan satu anggota entitas lain.

2. *One to Many*


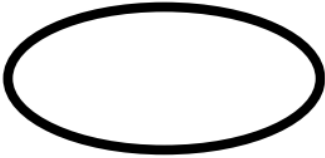
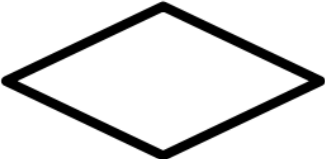

Satu anggota entitas dapat berelasi dengan satu anggota entitas lain.

3. *Many to Many*

Beberapa anggota entitas dapat berelasi dengan beberapa anggota entitas lain.

Simbol-simbol ERD dapat dilihat pada tabel 2.3 [9].

Tabel 2. 3 Simbol-simbol ERD

No.	Simbol	Nama	Keterangan
1.		Entitas	Sesuatu di dalam sistem atau di luar sistem yang saling berinteraksi
2.		Atribut	Sifat, atau elemen dari setiap entitas
3.		<i>Relationship</i>	Hubungan antara entitas 1 dengan entitas lainnya
4.		<i>Link</i>	Menghubungkan satu entitas dengan entitas lainnya

2.1.5 Database

Database merupakan sebuah sistem yang digunakan untuk membantu dalam mengorganisasi, menyimpan, dan mengambil data dengan lebih mudah. *Database* atau basis data meliputi sekumpulan data yang terorganisir untuk penggunaan dalam bentuk digital[10]. Basis data digital dalam fungsinya untuk menyimpan, membuat, memelihara serta melakukan pencarian dan akses data yang lainnya dilakukan dalam sebuah perangkat lunak yang disebut *Database Management System (DBMS)*. Basis data memiliki beberapa fungsi di antaranya :

1. Memudahkan dalam hal identifikasi data dengan adanya klasifikasi data dengan cara melakukan pembuatan tabel atau field yang berbeda-beda
2. Mencegah suatu data ganda
3. Memudahkan user pada proses *input* data baru
4. Melakukan penyimpanan secara digital
5. Sebagai alternatif bagi masalah penyimpanan ruang dalam suatu aplikasi

Basis data memiliki beberapa jenis di antaranya :

1. *Operational Database*

Basis data jenis ini melakukan operasi penyimpanan data yang sangat rinci agar dapat dengan mudah digunakan. Basis data ini biasa digunakan untuk basis data pelanggan.

2. *Relational Database*

Pada basis data ini, user dapat mengakses atau mencari informasi dalam tabel yang berbeda – beda.

3. *Distributed Database*

Untuk basis data jenis ini dapat mendistribusikan data – data secara tersebar namun saling berhubungan serta dapat diakses secara bersama – sama.

4. *External Database*

Database terakhir ini sering digunakan sebagai keperluan komersial karena kemudahan mengaksesnya yang memang dikhususkan untuk *public*.

2.1.6 Sistem

Sistem berasal dari bahasa Yunani “systema” yang memiliki arti sehimpunan bagian atau komponen yang saling berhubungan secara teratur dan merupakan suatu keseluruhan. Bisa dibayangkan sistem adalah istilah yang mempunyai makna sangat luas dan dapat dipakai sebagai sebutan yang melekat pada sesuatu. Sebuah perkumpulan atau organisasi bisa disebut sebagai sistem dan orang-orang menyebutnya sistem organisasi. Begitu juga dengan pendidikan sebagai sebuah sistem, pada akhirnya orang-orang menyebutnya sistem pendidikan.

Menurut (Sanjaya,2020) sistem merupakan sebagai satu kesatuan komponen yang satu sama lain saling berhubungan untuk mencapai suatu hasil yang diharapkan secara optimal sesuai dengan tujuan. Tatang M. Amirin (Adnan, 1018:101), menjelaskan pengertian sistem adalah sebagai berikut:

1. Sistem adalah suatu kebulatan keseluruhan yang kompleks atau terorganisir, suatu himpunan atau perpaduan hal-hal atau bagian-bagian yang membentuk suatu kebulatan atau keseluruhan yang kompleks atau utuh.
2. Sistem merupakan himpunan komponen yang saling berkaitan yang Bersama-sama berfungsi untuk mencapai suatu tujuan.
3. Sistem merupakan sehimpunan komponen atau subsistem yang terorganisasikan dan berkaitan sesuai dengan rencana untuk mencapai tujuan tertentu. Dari beberapa definisi di atas bisa ditarik kesimpulan bahwa sistem merupakan rangkaian keseluruhan kebulatan kesatuan dari komponen-komponen yang saling berinteraksi antara satu sama lainnya, yang semuanya itu sebagai satu kesatuan yang utuh dalam mencapai tujuan [11].

2.1.7 Monitoring

Monitoring adalah proses rutin pengumpulan data atau pemantauan untuk menemukan dan memperbaiki kesalahan dalam suatu program atau objek. *Monitoring* berfokus pada proses dan hasilnya. Pemantauan akan memberikan laporan tentang status dan kemajuan perubahan yang dilakukan secara teratur [12].

2.1.8 Barang Habis Pakai

Barang Habis Pakai (BHP) adalah barang yang hanya dapat digunakan satu kali. Benda ini tidak dapat meninggalkan bekas dalam pemakaiannya. Barang ini umumnya digunakan untuk kelangsungan kegiatan operasional atau suatu organisasi. Berdasarkan hal tersebut muncul istilah persediaan barang habis pakai yang diartikan persediaan yang mengelola barang dalam bentuk perlengkapan yang digunakan dalam operasional [13].

2.1.9 Laboratorium

Kata Laboratorium diambil dari Bahasa Latin yang memiliki arti “tempat bekerja”. Dalam perkembangannya, kata laboratorium menjaga arti aslinya yakni “tempat bekerja” khusus dalam lingkup kepentingan penelitian ilmiah. Laboratorium merupakan suatu ruangan atau kamar sebagai tempat untuk melaksanakan kegiatan praktikum atau penelitian yang didukung dengan adanya seperangkat peralatan serta adanya infrastruktur laboratorium

yang lengkap seperti fasilitas air, listrik, gas, dan lain sebagainya. Laboratorium juga merupakan tempat yang digunakan oleh sekelompok orang yang akan melaksanakan berbagai macam kegiatan penelitian, observasi, pelatihan, serta pengujian secara ilmiah sebagai suatu pendekatan antara teori dengan praktik dari berbagai ragam disiplin ilmu yang ada [14].

2.1.10 Ilmu Logistik

Logistik merupakan ilmu pengetahuan tentang seni yang terdiri dari perencanaan, pengajuan, pengadaan, penyaluran, dan pemeliharaan dalam hal penentuan kebutuhan penyediaan yang termasuk dalam pelayanan persediaan dalam kuantitas yang besar. Logistik mendeskripsikan manajemen dari suatu arus barang mentah menjadi barang jadi dalam sebuah organisasi dalam memenuhi persediaan pada kebutuhan [15].

2.1.11 Barcode

Barcode adalah tampilan dari data yang bisa dibaca dengan menggunakan mesin optik yang berkaitan dengan benda yang dilampirkan . Barcode digunakan untuk menyandikan informasi dalam pola visual, yang dapat digunakan untuk melacak produk, harga, dan tingkat stok untuk perekaman terpusat dalam sistem perangkat lunak [16]. Gambar 2.2 menunjukkan contoh dari barcode.



Gambar 2. 2 Contoh Barcode

(~~Halaman ini sengaja dikosongkan~~)