

BAB II

LANDASAN TEORI

2.1 Aplikasi

Menurut kamus besar bahasa Indonesia (2002 : 52) Aplikasi adalah suatu penerapan dari rancangan sistem pengolahan data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu.

Aplikasi adalah suatu sistem yang dirancang dan disusun sedemikian rupa untuk menghasilkan informasi yang terpadu dengan menggunakan komputer sebagai sarana penunjang. (Jogiyanto, 2000 : 13). Dalam penerapannya aplikasi terdiri dari beberapa bagian, yaitu:

1. *Input*, menggambarkan suatu kegiatan untuk menyediakan data yang akan diproses.
2. *Proses*, menggambarkan bagaimana suatu data diolah untuk menghasilkan suatu informasi yang bernilai tambah.
3. *Output*, kegiatan untuk menghasilkan laporan dari proses di atas.
4. *Penyimpanan*, kegiatan untuk memelihara dan menyimpan data.
5. *Kontrol*, aktifitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

2.2 Koperasi

Menurut UU No. 25/1992 Pasal 1 Ayat 1 tentang Perkoperasian, Koperasi adalah badan usaha yang beranggotakan orang-seorang atau badan hukum

koperasi, dengan melandaskan kegiatannya berdasarkan prinsip koperasi sekaligus sebagai gerakan ekonomi rakyat yang berdasar atas asas kekeluargaan.

Menurut Anoraga (1995:23), koperasi berasal dari kata *co* dan *operation*, yang mengandung arti bekerja sama untuk mencapai tujuan. Karena itu, koperasi adalah suatu perkumpulan yang beranggotakan orang-orang atau badan-badan yang memberikan kebebasan masuk dan keluar sebagai anggota dengan bekerja sama secara kekeluargaan menjalankan usaha untuk mempertinggi kesejahteraan jasmaniah para anggotanya.

2.3 Manajemen Koperasi

Menurut Hendrojogi (1998:25), manajemen koperasi dalam hal ini merupakan proses perencanaan, pengorganisasian, kepemimpinan, dan pengendalian upaya anggota organisasi untuk tercapainya tujuan organisasi yang telah ditetapkan. Untuk mencapai tujuan organisasi tersebut tidak lepas dari prinsip-prinsip koperasi, salah satunya adalah prinsip : “ Partisipasi Anggota dalam Kegiatan Ekonomi”, dimana para anggota memberikan kontribusi pemodalan koperasi secara adil dan melakukan pengawasan secara demokratis. Para anggota mengalokasikan sisa hasil usaha untuk beberapa atau semua dengan tujuan berikut ini:

1. Mengembangkan koperasi mereka, mungkin dengan membentuk dana cadangan, sebagian daripadanya tidak dapat dibagikan.
2. Membagikan kepada anggota seimbang dengan transaksi mereka dengan koperasi.

2.4 Teori Simpan Pinjam

Teori simpan pinjam menjelaskan tentang pengertian sendiri dari simpan pinjam dan juga menjelaskan hal lainnya yang berkaitan dengan koperasi simpan pinjam pada umumnya. Keterangan lebih lengkapnya adalah sebagai berikut :

a. Pengertian simpan pinjam

Menurut Tohar (2000:160), simpan pinjam adalah kegiatan menghimpun dan menyalurkan dana melalui usaha simpan pinjam dari dan untuk anggota koperasi maupun kepada koperasi dan anggota lainnya. Kegiatan usaha simpan pinjam biasanya dilaksanakan oleh Koperasi Simpan Pinjam (KSP) atau Unit Simpan Pinjam (USP) pada sebuah koperasi.

b. Simpanan

Setiap anggota koperasi diwajibkan untuk membayar uang simpanan pokok dan wajib. Kedua iuran simpanan tersebut tidak bisa diambil selama menjadi anggota koperasi dan hanya bisa diambil jika anggota sudah keluar dari keanggotaan, sedangkan simpanan sukarela boleh diambil sewaktu-waktu.

Jenis simpanan koperasi pada umumnya adalah sebagai berikut :

1. Simpanan pokok adalah iuran yang dibayar sewaktu pertama kali mendaftarkan diri menjadi anggota koperasi, dimana besarnya iuran ditentukan oleh pihak koperasi. Pembayaran iuran simpanan pokok hanya dilakukan 1 (satu) kali selama menjadi anggota.
2. Simpanan wajib adalah iuran yang wajib dibayar setiap bulan selama menjadi anggota koperasi, dimana besarnya iuran ditentukan oleh pihak koperasi. Besar iuran wajib ditentukan oleh keputusan dan kebijakan dari

pihak koperasi tersebut sesuai dengan prosedur dan ketentuan yang berlaku.

Simpanan sukarela adalah iuran yang dibayar sesuai keinginan selama menjadi anggota koperasi, dimana besarnya iuran sesuai dengan kemampuan anggotanya (bersifat sukarela).

c. Pinjaman

Pemberian kredit pinjaman merupakan jasa atau bisnis yang beresiko, karena ada kemungkinan kredit yang diberikan tidak dapat tertagih (macet).

Sehubungan hal tersebut, sudah menjadi keharusan bagi koperasi hanya memberikan pinjaman kepada anggota yang layak dengan melakukan seleksi setiap usulan kredit.

2.5 *Flowchart*

Flowchart adalah teknik analisis yang dipergunakan untuk mendeskripsikan beberapa aspek dari sistem informasi secara jelas, ringkas dan logis. *Flowchart* menggunakan serangkaian symbol standar yang menggambarkan prosedur yang digunakan oleh perusahaan dan arus data melalui sistem. Menurut Romney (2003), ada dua macam *flowchart* yang menggambarkan proses dengan menggunakan komputer, yaitu:

a. *Flowchart System*





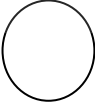
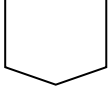
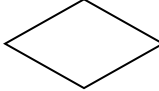

Flowchart system adalah suatu gambar yang menunjukkan hubungan antara input, pemrosesan dan output dari suatu sistem informasi.

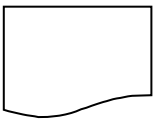



b. *Flowchart Program*

Flowchart program adalah bagan yang memperlihatkan urutan proses secara logis yang dilaksanakan oleh komputer dalam menjalankan sebuah program. *Flowchart program* ini merupakan langkah awal pembuatan program. Dengan adanya *flowchart* program maka urutan proses di program menjadi lebih jelas.

Flowchart disusun dengan symbol. Simbol-simbol yang digunakan dalam flowchart dapat dilihat pada Tabel 2.1.

Tabel 2.1. Simbol-simbol *Flowchart*

Simbol	Nama	Keterangan
	Terminal	Simbol untuk menunjukkan awal, akhir dan pemberhentian dalam suatu proses atau program
	Proses manual	Simbol yang menunjukkan pengolahan yang dilakukan secara manual
	Proses dengan komputer	Simbol untuk menunjukkan pengolahan yang dilakukan dengan komputer
	Arah dokumen atau proses	Simbol menunjukkan arah aliran dokumen atau proses
	<i>On-page connector</i>	Simbol untuk menggambarkan proses pada halaman yang sama
	<i>Off-page connector</i>	Simbol untuk menggambarkan proses pada halama yang berbeda
	Keputusan	Simbol untuk kondisi yang menggambarkan langkah pengambilan keputusan
	<i>On-line keying</i>	Simbol yang menyatakan input dengan menggunakan mesin yang mempunyai keyboard

	Dokumen	Simbol yang menyatakan input berasal dari dokumen dalam bentuk tulisan tangan atau output dicetak dengan computer
	Beberapa tembusan dari satu dokumen	Simbol yang menyatakan dokumen yang sama dicetak beberapa kali dengan computer
	<i>Input/output</i>	Simbol yang menyatakan proses input dan output di dalam bagan alir program
	Penyimpanan online	Simbol yang menggambarkan data yang disimpan dalam file on-line yang dapat diakses secara langsung

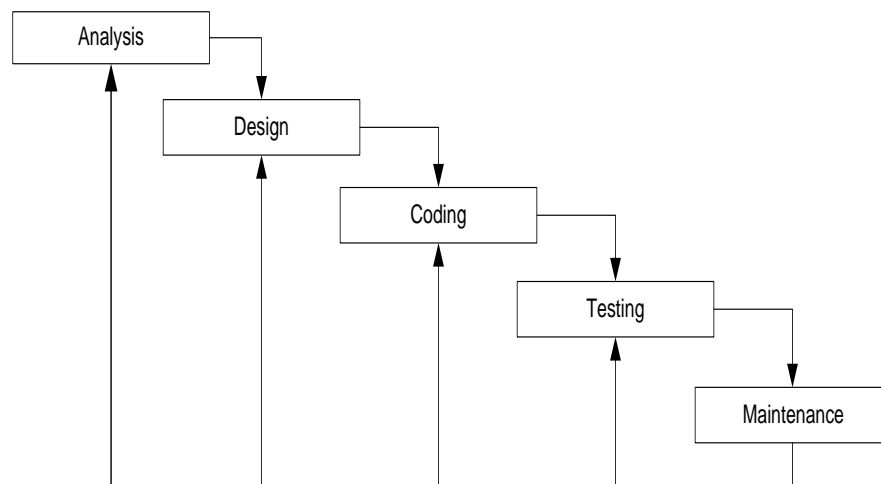
2.6 Rekayasa Perangkat Lunak

2.6.1 Definisi Rekayasa Perangkat Lunak

Menurut Jogiyanto (2005), rekayasa perangkat lunak adalah suatu disiplin rekayasa yang berkonsentrasi terhadap seluruh aspek produksi perangkat lunak, mengadopsi pendekatan yang sistematis dan terorganisir terhadap pekerjaannya dan menggunakan *tool* yang sesuai serta teknik yang ditentukan berdasarkan masalah yang akan dipecahkan, kendala pengembangan, dan sumber daya yang tersedia.

2.6.2 Proses Perangkat Lunak

Proses perangkat lunak merupakan aktivitas yang saling terkait (koheren) untuk menspesifikasikan, merancang, mengimplementasikan, dan menguji sistem perangkat lunak. Proses perangkat lunak model air terjun dapat dilihat pada gambar dibawah ini:



Gambar 2.1 Siklus Model Air Terjun.

1. *Analysis* adalah tahap menganalisa hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan atau pengembangan *software*.
2. *Design* adalah tahap penerjemah dari keperluan-keperluan yang dianalisis ke dalam bentuk yang lebih mudah dimengerti oleh pemakai. Yaitu dengan cara menampilkan ke dalam Diagram Konteks, *Data Flow Diagram* (Diagram Aliran Data), *Entity Relationship Diagram*, Struktur Tabel, dan Struktur Menu.
3. *Coding* adalah tahap penerjemah data/pemecahan masalah *software* yang telah dirancang ke dalam bahasa pemrograman yang telah ditentukan.
4. *Testing* adalah tahap pengujian terhadap program yang telah dibuat. Pengujian ini dimulai dengan membuat suatu uji kasus untuk setiap fungsi pada perangkat lunak, kemudian dilanjutkan dengan pengujian terhadap modul-modul dan terakhir pada tampilan antar muka untuk memastikan tidak ada kesalahan dan semua berjalan dengan baik dan *input* yang diberikan hasilnya sesuai dengan yang diinginkan.

5. *Maintenance* adalah perangkat lunak yang telah dibuat dapat mengalami perubahan sesuai permintaan pemakai. Pemeliharaan dapat dilakukan jika ada permintaan tambahan fungsi sesuai dengan keinginan pemakai ataupun adanya pertumbuhan dan perkembangan baik perangkat lunak maupun perangkat keras.

2.7 Data Flow Diagram

DFD merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut (Kendall 2003:204). Diagram ini digunakan untuk menggambarkan arus data di dalam sistem secara terstruktur dan jelas. Selain itu, DFD merupakan gambaran dari sistem yang baik.

DFD ini salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks daripada data yang digunakan untuk menjelaskan aliran informasi dan transformasi data yang bergerak dari pemasukan data hingga keluaran.

DFD fokus pada aliran data dari dan ke dalam sistem serta memproses data tersebut (Kendall, 2003:241). Untuk memudahkan pembacaan DFD, maka penggambaran DFD disusun berdasarkan tingkatan atau level dari atas ke bawah, yaitu :

a. Context Diagram

Context Diagram merupakan langkah pertama dalam pembuatan *Data Flow Diagram*. Pada *Context Diagram* dijelaskan sistem apa yang dibuat dan *entity* apa saja yang digunakan. Dalam *Context Diagram* harus ada arus data yang masuk dan arus data yang keluar.

b. Diagram Zero (Level 0)

Diagram Zero adalah langkah selanjutnya setelah *Context Diagram*. Hal yang digambarkan dalam *diagram zero* ini adalah proses utama dari sistem serta hubungan *Entity*, *Proses*, *Alur Data* dan *Data Store*.

c. Diagram Detail (Primitif)

1) *External Entity*

Suatu *External Entity* atau entitas merupakan orang, kelompok, *departement*, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat.

2) *Data Flow*

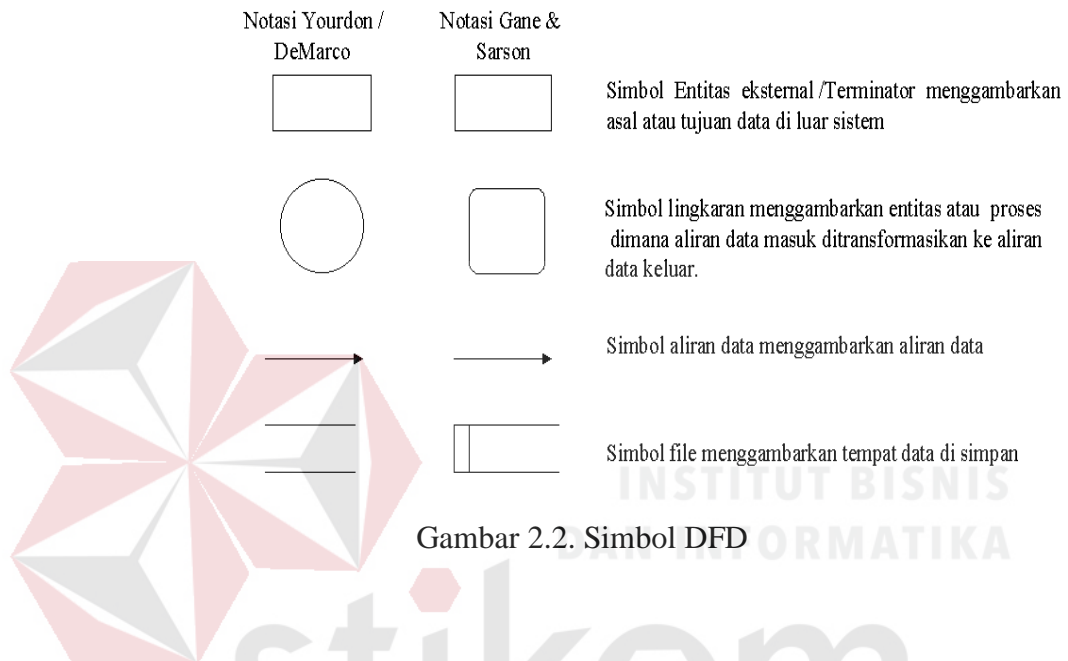
Data Flow atau aliran data disimbolkan dengan tanda panah. *Data Flow* menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses.

3) *Proses*

Suatu proses dimana beberapa tindakan atau sekelompok tindakan dari arus data yang masuk untuk dijalankan atau diproses agar menghasilkan arus data yang akan keluar dari proses.

4) Data Store

Data Store adalah simbol yang digunakan untuk melambangkan proses penyimpanan data. Suatu nama perlu diberikan pada data store untuk menunjukkan nama dari filenya.



2.8 Entity Relationship Diagram (ERD)

ERD merupakan notasi grafis dalam permodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antardata, karena hal ini relatif kompleks. Dengan ERD kita dapat menguji model dan mengabaikan proses apa yang harus dilakukan (Al-barha bin jadmudin, 2005 : 130).

ERD dapat dikategorikan menjadi tiga bagian, yaitu:

1. One To One Relationship

Jenis hubungan antartabel yang menggunakan bersama sebuah kolom *primary key*. Jenis hubungan ini tergolong jarang digunakan, kecuali untuk

alasan keamanan atau kecepatan akses data. Misalnya satu departemen hanya mengerjakan satu jenis pekerjaan saja dan satu pekerjaan hanya dikerjakan oleh satu departemen saja.

2. *One To Many Relationship*

Jenis hubungan antartabel dimana satu *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain. Jenis hubungan ini merupakan yang paling sering digunakan. Misalnya suatu pekerjaan hanya dikerjakan oleh satu departemen saja, namun suatu departemen dapat mengerjakan beberapa macam pekerjaan sekaligus.

3. *Many To Many Relationship*

Jenis hubungan ini merupakan hubungan antartabel dimana beberapa *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain. Misalnya satu departemen mampu mengerjakan banyak pekerjaan, juga satu pekerjaan dapat ditangani oleh banyak departemen.