

BAB III

LANDASAN TEORI

3.1 Aplikasi

Menurut Rizky (2009), aplikasi adalah *software* yang dibuat oleh perusahaan komputer untuk menunjang tugas-tugas tertentu, contohnya *Microsoft Word*, *Microsoft Excel*. Menurut Kadir (2005), aplikasi *software* merupakan program yang dipakai untuk melakukan atau mempermudah tugas-tugas yang spesifik; misalnya untuk membuat dokumen, membuat laporan atau memanipulasi foto.

Dari pengertian diatas, disimpulkan bahwa aplikasi merupakan perangkat lunak yang difungsikan sebagai alat bantu untuk mengerjakan hal-hal tertentu yang dapat memudahkan jalannya proses bisnis atau kegiatan yang dilakukan oleh perusahaan ataupun organisasi.

3.2 Pengertian Distribusi

Menurut Kotler (2009), saluran distribusi adalah sekelompok organisasi yang saling bergantung dan terlibat dalam proses pembuatan produk atau jasa yang disediakan untuk digunakan atau dikonsumsi.

Menurut Basu (2008), saluran distribusi merupakan suatu struktur organisasi dalam perusahaan dan luar perusahaan yang terdiri atas agen, dealer,

pedagang besar, dan pengecer, melalui sebuah komoditi, produk, atau jasa dipasarkan.

Untuk memperlancar arus barang atau jasa dari produsen ke konsumen, salah satu faktor yang penting adalah memilih saluran secara tepat saluran distribusi yang dipergunakan untuk menyampaikan barang atau jasa dari produsen ke konsumen. Dari kedua definisi di atas dapat ditarik kesimpulan bahwa saluran distribusi merupakan seperangkat alur yang diikuti produk atau jasa setelah produksi berakhir dalam pembelian dan digunakan oleh pengguna akhir.

3.3. Kartu Garansi

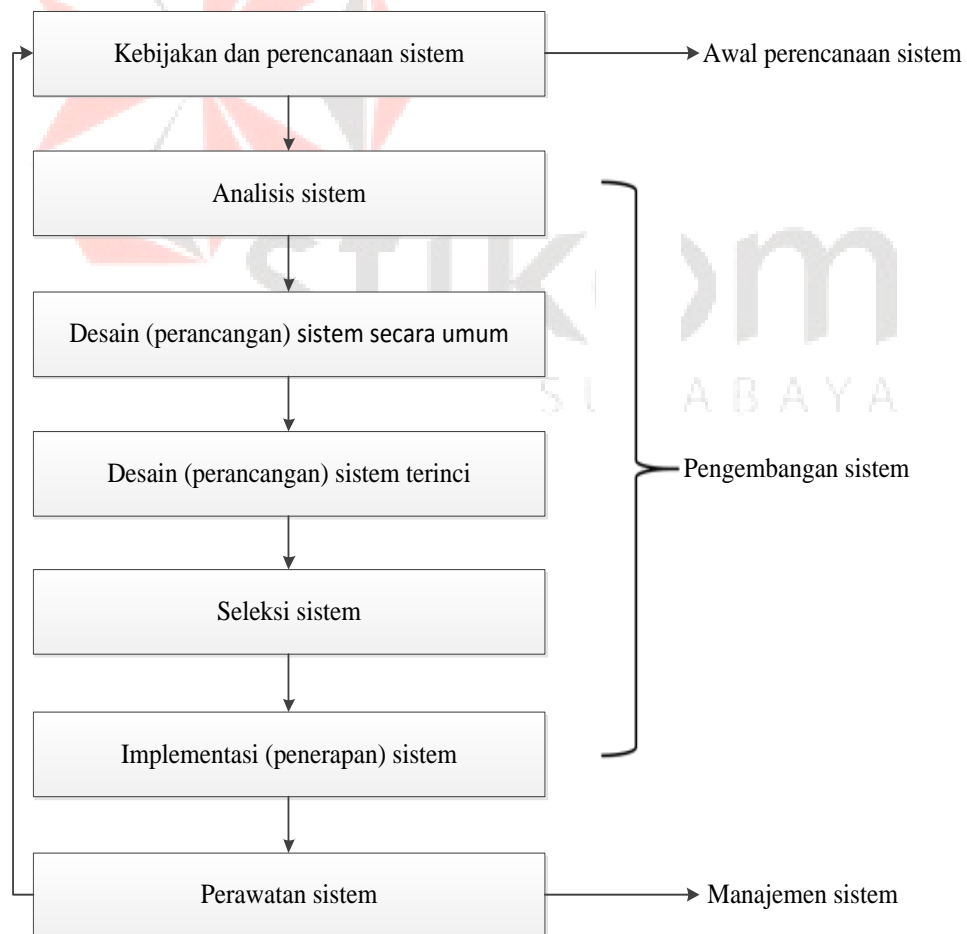
Kata garansi berasal dari bahasa Inggris yaitu *Guarantee* yang berarti jaminan atau tanggungan. Dalam kamus besar bahasa Indonesia, garansi mempunyai arti tanggungan atau bagian dari suatu perjanjian dalam jual beli, dimana penjual menanggung kebaikan atau keberesan barang yang dijual untuk jangka waktu yang ditentukan,

3.4 Systems Development Life Cycle (SDLC)

Pengembangan sistem informasi yang berbasis komputer dapat merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu berbulan-bulan bahkan bertahun-tahun untuk menyelesaikannya. Menurut Jogiyanto (2005) proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan dan dipelihara, siklus pengembangan sistem informasi ini disebut dengan siklus hidup suatu sistem (*systems life cycle*). Daur atau siklus dari pengembangan sistem merupakan suatu bentuk yang digunakan untuk

menggambarkan tahapan utama dan langkah–langkah di dalam tahapan tersebut dalam proses pengembangannya.

Pada *system life cycle*, tiap–tiap bagian dari pengembangan sistem dibagi menjadi beberapa tahapan kerja. Tiap–tiap tahapan ini mempunyai karakteristik tersendiri. Tahapan utama siklus hidup pengembangan siklus hidup pengembangan sistem dapat terdiri dari tahapan perencanaan sistem (*systems planning*), analisis sistem (*systems analysis*), desain sistem (*systems design*), seleksi sistem (*systems selection*), implementasi sistem (*systems implementation*), perawatan sistem (*systems maintenance*). *Systems Development Life Cycle* (SDLC) dapat dilihat pada gambar 3.1 *System Development Life Cycle* (SDLC).



Gambar 3.1 *System Development Life Cycle* (SDLC)

3.5 Diagram Alir Dokumen (*Document Flow Diagram*)

Menurut Jogiyanto (2005), *document flowchart* adalah diagram alir dokumen atau bisa disebut juga sebagai bagan alir formulir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Menurut James (2007), bagan alir dokumen digunakan untuk menggunakan elemen-elemen dari sebuah sistem manual, termasuk *record-record* akuntansi (dokumen, jurnal, buku besar, dan file), departemen organisasional yang terlibat dalam proses, dan kegiatan-kegiatan (baik klerikal maupun fisik) yang dilakukan dalam departemen tersebut.




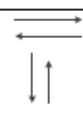







3.6 Diagram Alir System (*System Flowchat*)

Menurut Krismiaji (2005), bagan alir sistem menggambarkan hubungan antara input, pemrosesan dan output sebuah sistem informasi akuntansi. Bagan alir sistem ini dimulai dengan identifikasi input yang masuk ke dalam sistem dan sumbernya.

Menurut Jogiyanto (2005), bagan alir program (*system flowchart*) merupakan bagan alir yang mirip dengan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Bagan ini menjelaskan urutan-urutan dari prosedur-prosedur yang ada di dalam sistem. Baagan alir sistem menunjukkan apa yang dikerjakan di sistem. Pembuatan *system flowchart* memiliki aturan dan ketentuan yang harus diikuti. Seperti halnya dalam pembuatan *document flowchart* sebelumnya, *system flowchat* memiliki notasi-notasi sebagai representasi proses kerja suatu sistem. Sebagian notasi dalam *system flowchart* memiliki kesamaan dengan notasi yang ada pada *document flow* seperti, terminator (*start/end*), dan

notasi laporaan. Selaain kedua notasi tersebut terdapat perbedaan secara bentuk dan fungsinya.

System flowchart ini tidak digunakan untuk menggambarkan langkah-langkah pemecahan masalah, namun hanya digunakan untuk menggambarkan prosedur pada sistem yang dibuat. Terdapat symbol- symbol dalam system flowchart yaitu sebagai berikut :

	Operasi secara manual
	<i>Input output</i>
	Proses
	Arus informasi
	Keputusan
	Dokumen atau laporan
	Terminal
	Penyimpanan <i>file</i> secara sementara
	<i>Input manual</i>
	<i>Input secara manual</i>
	Penghubung ke halaman berikutnya

Tabel 3.1 Simbol Flowchart

3.7 Diagram Konteks (*Context Diagram*)

Menurut Jogiyantoo (2005), diagram konteks merupakan diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks adalah level tertinggi dari *Data Flow Diagram* (DFD) yang menggambarkan seluruh inputan ke sistem atau output dari sistem. Pada proses inilah gambaran mengenai keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digunakan dengan garis putus). Dalam konteks, diagram konteks hanya ada satu proses dan tidak ada store di dalam diagram konteks.

Menurut Oetomo (2002), terdapat beberapa hal yang diperhatikan dalam pembuatan diagram konteks, antara lain:

1. Kelompok pemakai, baik internal maupun eksternal perusahaan.
2. Identifikasi kejadian-kejadian yang mungkin terjadi dalam penggunaan sistem.
3. Arah anak panah yang menunjukkan aliran data.
4. Setiap proses digambarkan dalam bentuk yang sederhana dan mudah dipahami oleh pembuat sistem.

Proses dalam diagram konteks biasanya diberi nomor 0. Proses ini merupakan proses dari seluruh sistem dengan dunia luarnya.

3.8 *Data Flow Diagram* (DFD)

Data flow diagram adalah gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam objek kemudian melewati suatu proses yang mentransformasikan ke tujuan lain, yang ada pada objek lain.

Menurut Jogiyanto (2010) dalam bukunya Analisis & Desain, menjelaskan bahwa “Data Flow Diagram digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan. Data Flow Diagram juga digunakan pada metodologi pengembangan sistem yang terstruktur.

Dari kedua penjelasan diatas dapat disimpulkan bahwa, *Data Flow Diagram* (DFD) berfungsi untuk menggambarkan sistem baru ataupun sistem yang telah ada dimana arus datanya akan dikembangkan secara terstruktur. Tingkatan *Data Flow Diagram* (DFD) terdiri dari:

1. Diagram Konteks (*Context Diagram*).

Diagram konteks merupakan sebuah model proses yang digunakan untuk mendokumentasikan ruang lingkup dari sebuah sistem.

2. Diagram Level 0

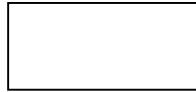
Diagram level 0 merupakan diagram alir data yang menggambarkan sebuah event konteks. Diagram ini menunjukkan interaksi antara input, output dan data store pada setiap proses yang ada.

3. Diagram Level 1

Diagram level 1 menggambarkan rincian dari diagram level 0. Setiap proses yang terdapat pada diagram level 0 akan dijelaskan pada proses ini.

Data Flow Diagram memiliki empat simbol, yaitu :

- a. *External Entity*, digunakan sebagai sumber dari inputan sistem atau tujuan dari *output* sistem.

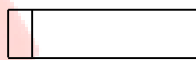
Gambar 3.2 Simbol *External Entity*

- b. Proses, dimana simbol proses ini sering digunakan untuk melakukan perubahan terhadap input yang masuk sehingga menghasilkan data dari perubahan input yang diolah.



Gambar 3.3 Simbol Proses

- c. *Data Store*, sering digunakan sebagai simpanan dari data yang dapat berupa suatu file atau data.

Gambar 3.4 Simbol *Data Store*

- d. Aliran Data, sering digunakan untuk menghubungkan antara proses dengan proses, proses dengan sumber proses dan proses dengan tujuan. Sedangkan anak panahnya menunjukkan arah aliran datanya.



Gambar 3.5 Simbol Aliran Data

3.9 Basis Data.

Menurut Adi Nugroho (2011), basis data didefinisikan sebagai kesimpulan data yang saling terhubung dan terorganisasi sehingga mudah disimpan, dimanipulasi, serta dipanggil oleh pengguna. Data disimpan dengan cara-cara tertentu sehingga mudah untuk digunakan atau ditampilkan kembali, data

disimpan sedemikian rupa sehingga proses penambahan, pengambian, dan modifikasi data dapat dilakukan dengan mudah dan terkontrol.

3.9.1 *Entity Relationship Model.*

Menurut Janner Simarmata (2007), *Entity Relationship Model* pertama kali diusulkan oleh Petter tahun 1976 sebagai cara untuk mempersatukan pandangan basis data jaringan dan relasional. *Entity Relationship Model* adalah model data konseptual yang memandang dunia nyata sebagai kesatuan (entitas) dan hubungan (relasi). Komponen dasar model merupakan diagram *Entity-Relationship* yang digunakan untuk menyajikan obyek data secara visual.

Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. Macam-macam jenis *attribute* yaitu:

a. *Key Attribute*

Attribute ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya,

b. *Partial Key Attribute*

Adalah *attribute* yang tidak menjadi atau merupakan anggota dari *Key Primer*.

c. *Single Value Attribute*

Attribute yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya umur (tanggal lahir).

d. *Multi Value Attribute*

Attribute yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya pendidikan (SD, SMP, SMA).

e. *Composite Attribute*

Attribute yang memiliki dua harga, misalnya nama besar (nama kerja) dan nama kecil (nama asli).

f. *Derived Attribute*

Attribute yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari *table attribute* atau tabel lain yang berhubungan.

ERD digunakan untuk menguji model dan mengabaikan proses apa yang harus dilakukan. ERD dapat dikategorikan menjadi tiga bagian, yaitu :

1. *One to one relationship*

Jenis hubungan antar tabel 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 antar tabel 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 antar tabel 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.

Entity Relationship Diagram dibagi menjadi dua jenis model, yaitu :

a. *Conceptual Data Model (CDM)*

Conceptual Data Model adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

b. *Physical Data Model (PDM)*

Physical Data Model adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

3.9.2 *Database Management System.*

Menurut Simarmata (2007), *Database Management System (DBMS)* adalah suatu sistem perangkat lunak yang mengatur permintaan dan penyimpanan data. DBMS menyediakan semua layanan dasar yang diperlukan untuk mengorganisir dan memelihara basis data, termasuk layanan:

1. Memindahkan data dari dan file-file data fisik jika dibutuhkan.
2. Mengelola akses data oleh berbagai pengguna secara bersamaan.
3. Mendukung bahasa *query*.
4. Ketentuan untuk membackup basis data dan pemulihan dari kegagalan.
5. Mekanisme keamanan untuk mencegah perubahan dan akses data yang tidak sah.

3.9.3 *Structured Query Language.*

SQL merupakan singkatan dari *Structure Query Language*. Menurut A, Ramon & Pauline Cushman, K. (2007), SQL merupakan bahasa *query* standart

yang digunakan untuk mengakses basis data relasional serta sebagai alat komunikasi untuk sistem basis data. Standarisasi internasional terhadap SQL pertama dilakukan oleh ANSI (*American National Standards Institution*) dan Internasional Standart Organisation (ISO). Salah satu karakteristik SQL adalah sebagai bahasa non procedural yang mengimplikasikan bahwa seorang *programmer* hanya menunjukkan kepada sistem manajemen database apa saja yang harus dipenuhi dan kemudian sistem menentukan bagaimana memperoleh hasil yang sesuai.

3.11 *Visual Basic*

Menurut yuswanto dan subari (2010), visual basic .NET 2010 adalah salah satu bahasa pemrograman yang tergabung dalam Microsoft Visual Studio 2010. Pada Visual basic .NET 2010 mempunyai suatu jendela yang luas sebagai ruangan kerjanya. Jendela- jendela tersebut yaitu sebagai berikut :

a. Menu Bar

Menu Bar merupakan kumpulan dari perintah- perintah yang dikelompokkan dalam kriteria operasinya. Contoh daftar pilihan menu yang terdapat dalam Visual basic .NET 2010 yaitu, File, Edit, View, Project, Build, Debug, Data, Format, Tools, Windows, dan Help

b. Toolbar

Toolbar merupakan sekumpulan tombol tau icon yang mewakili suatu perintah tertentu pada bahasa pemrograman berbasis windows dan juga bisa dikombinasikan dengan perintah yang dibuat sendiri dengan menggunakan logika pemikiran sendiri.

c. Toolbox

Toolbox merupakan control user interface yang digunakan untuk membentuk suatu program berbasis windows dan web. Control- control tersebut antara lain: all windows form, common controls, data, components, containers, menu & toolbar, printing, dialogs, WPF interoperability, reporting, dan visual basic powerpacks.

d. Form Windows

Form windows merupakan jendela form atau jendela desain yang merupakan pusat pengembangan visual basic .NET 2010 dimana pada obyek dari common controls pada toolbox ditempatkan.

e. Code Windows

Code windows atau jendela editor merupakan area yang dapat menuliskan kode-kode pemrograman visual basic .NET. Sedangkan kode-kode program merupakan kumpulan dari instruksi untuk menjalankan obyek berupa control maupun form serta logika program.

f. Soution Explore Windows

Soution Explore Windows merupakan jendela yang menampilkan daftar semua form, modul, class dan file lainnya untuk membuat aplikasi untuk membuat aplikasi.

g. Properti Window

Properti Window digunakan pada mode desain yang bertujuan untuk mengatur suatu nilai pada obyek. Pada bagian atas dari jendela properties terdapat kotak pilihan sebagai penunjuk dari nama obyek yang sedang aktif.

h. Jendela- jendela lain

Saat eksekusi program dijalankan, terdapat beberapa jendela yang menampilkan informasi dari efek proses program tersebut, antara lain :

1. Error list windows

Error list windows merupakan jendela yang menampilkan kesalahan- kesalahan yang ditemukan ketika menjalankan aplikasi.

2. Output Windows

Output Windows merupakan jendela yang menampilkan langkah- langkah dalam mengkompilasi program.

