

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Aplikasi**

Menurut Jogiyanto (1999) adalah penggunaan dalam suatu komputer, instruksi (instruction) atau pernyataan (statement) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.

Sedangkan menurut Anisyah (2000), aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan software yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.

#### **2.2 Penjualan**

Kotler (2000) mengatakan bahwa "Penjualan adalah proses sosial manajerial dimana individu dan kelompok mendapatkan apa yang mereka butuhkan dan inginkan, menciptakan, menawarkan, dan mempertukarkan produk yang bernilai dengan pihak lain".

Sedangkan menurut Swastha (2004) penjualan adalah interaksi antara individu saling bertemu muka yang ditujukan untuk menciptakan, memperbaiki, menguasai atau mempertahankan hubungan pertukaran sehingga menguntungkan bagi pihak lain. Penjualan dapat diartikan juga sebagai usaha yang dilakukan manusia untuk menyampaikan barang bagi mereka yang memerlukan dengan imbalan uang menurut harga yang telah ditentukan atas persetujuan bersama.

### 2.3 Tujuan Penjualan

Kemampuan perusahaan dalam menjual produknya menentukan keberhasilan dalam mencari keuntungan, apabila perusahaan tidak mampu menjual maka perusahaan akan mengalami kerugian. Menurut Swastha (2004) tujuan umum penjualan dalam perusahaan yaitu: Mencapai volume penjualan, mendapatkan laba tertentu, menunjang pertumbuhan perusahaan.

### 2.4 Rumah Makan

Definisi Rumah Makan dan Restoran: Menurut SK Menteri Pariwisata, Pos dan Telekomunikasi No. KM 73/PW 105/MPPT-85 menjelaskan bahwa Rumah Makan adalah setiap tempat usaha komersial yang ruang lingkup kegiatannya menyediakan hidangan dan minuman untuk umum.

Dalam SK tersebut juga ditegaskan bahwa setiap rumah makan harus memiliki seseorang yang bertindak sebagai pemimpin rumah makan yang sehari-hari mengelola dan bertanggungjawab atas perusahaan Rumah makan tersebut.

Masakan Padang merupakan makanan khas Sumatra Barat yang bumbunya terdiri dari berbagai macam rempah-rempah, seperti: cengkeh, pala, kunyit, jahe, lengkuas dan masih banyak lagi. Masakan Padang terkenal karena cita rasanya yang menggugah selera. Salah satu makanan khas Sumatra Barat yang paling terkenal adalah rendang.

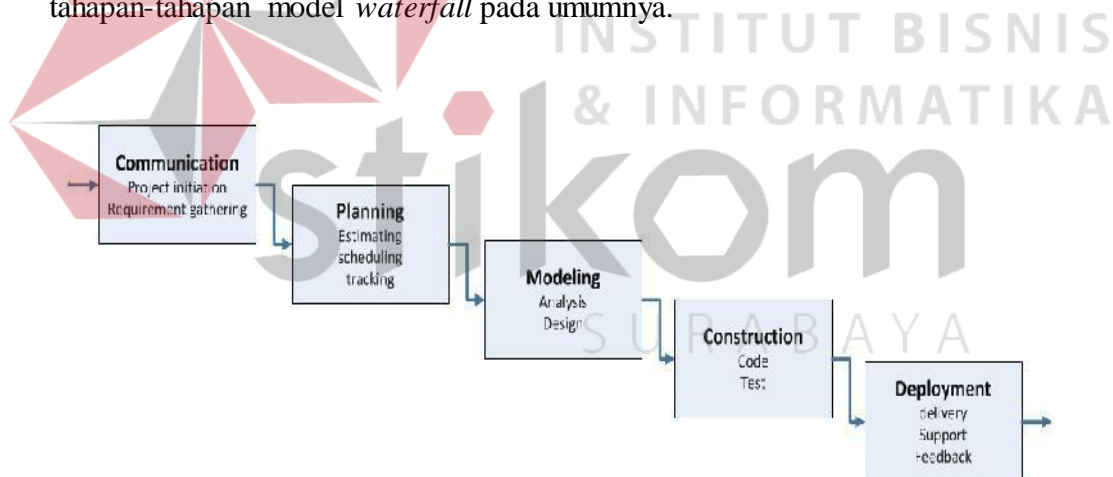
### 2.5 Metode *Waterfall*

Menurut Pressman (2015), nama lain dari Model *Waterfall* adalah Model Air Terjun kadang dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menyiratkan pendekatan yang sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak, yang dimulai dengan spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*),

pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem, perangkat lunak ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak lengkap yang dihasilkan.

Model ini merupakan model yang paling banyak dipakai dalam *Software Engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap *Communication*, *Planning*, *Modeling*, *Construction*, dan *Deployment*.

Gambar 2.1 menunjukkan tahapan umum dari model proses *waterfall*. Model ini disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Akan tetapi, Pressman (2015) memecah model ini meskipun secara garis besar sama dengan tahapan-tahapan model *waterfall* pada umumnya.



Gambar 2.1 Model waterfall menurut pressman (2015)

Berikut ini adalah penjelasan dari tahap-tahap yang dilakukan di dalam

Model *Waterfall* menurut Pressman (2015):

a. *Communication*

Langkah pertama diawali dengan komunikasi kepada konsumen/pengguna. Langkah awal ini merupakan langkah penting karena menyangkut pengumpulan informasi tentang kebutuhan konsumen/pengguna.

b. *Planning*

Setelah proses *communication* ini, kemudian menetapkan rencana untuk pengerjaan *software* yang meliputi tugas-tugas teknis yang akan dilakukan, risiko yang mungkin terjadi, sumber yang dibutuhkan, hasil yang akan dibuat, dan jadwal pengerjaan.

c. *Modeling*

Pada proses *modeling* ini menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat *coding*. Proses ini berfokus pada rancangan struktur data, arsitektur *software*, representasi *interface*, dan detail (algoritma) prosedural.

d. *Construction*

*Construction* merupakan proses membuat kode (*code generation*). *Coding* atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. *Programmer* akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat. Tujuan *testing* adalah menemukan kesalahan-kesalahan terhadap sistem tersebut untuk kemudian bisa diperbaiki.

#### e. *Deployment*

Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan *user*. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala.

### 2.6 Testing

Menurut Romeo (2003), testing adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan. *Testing Software* adalah proses pengoperasian *software* dalam suatu kondisi yang dikendalikan untuk verifikasi, mendeteksi *error* dan validasi. Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah melihat kebenaran sistem apakah proses yang telah dituliskan sudah sesuai dengan apa yang dibutuhkan oleh pengguna. Deteksi *error* adalah testing yang berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut tidak terjadi. *Test case* merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun hasil yang telah ditentukan sebelumnya. Adapun kegunaan dari *test case* ini, adalah sebagai berikut:

1. Untuk melakukan testing kesesuaian suatu komponen terhadap desain *White Box Testing*.
2. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi *Black Box Testing*.

### 2.6.1 White Box Testing

Menurut Romeo (2003), *white box testing* adalah suatu metode desain *test case* yang menggunakan struktur kendali dari desain prosedural. Seringkali *white box testing* diasosiasikan dengan pengukuran cakupan tes, yang mengukur persentase jalur-jalur dari tipe yang dipilih untuk dieksekusi oleh *test cases*. *White box testing* dapat menjamin semua struktur *internal* data dapat dites untuk memastikan validitasnya.

Cakupan pernyataan, cabang dan jalur adalah suatu teknik *white box testing* yang menggunakan alur logika dari program untuk membuat *test cases* alur logika adalah cara dimana suatu bagian dari program tertentu dieksekusi saat menjalankan program. Alur logika suatu program dapat direpresentasikan dengan *flow graph*.

### 2.6.2 Black Box Testing

Menurut Romeo (2003), *Black box testing* dilakukan tanpa adanya suatu pengetahuan tentang detail struktur internal dari sistem atau komponen yang dites, juga disebut sebagai *functional testing*. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.

Dengan adanya *black box testing*, perancang *software* dapat menggunakan kebutuhan fungsional pada suatu program. *Black box testing* dilakukan untuk melakukan pengecekan apakah sebuah *software* telah bebas dari *error* dan fungsi-fungsi yang diperlukan telah berjalan sesuai dengan yang diharapkan. Dengan adanya *black box testing*, perancang *software* dapat menggunakan kebutuhan fungsional pada suatu program. *Black box testing*

dilakukan untuk melakukan pengecekan apakah sebuah *software* telah bebas dari *error* dan fungsi-fungsi yang diperlukan telah berjalan sesuai dengan yang diharapkan.

## **2.7 Basis Data (Database)**

Basis data menurut Hengky (2002) adalah sebuah kumpulan dari sekelompok informasi yang diorganisasikan dengan beberapa cara logik dan saling berhubungan.

Menurut Aryanto (2001) menerangkan database merupakan keterangan mengenai kumpulan sejumlah tabel, prosedur tersimpan (*stored procedure*) dan hubungan relasi antar tabel yang saling berhubungan dalam membentuk suatu program aplikasi.

### **2.7.1 Normalisasi**

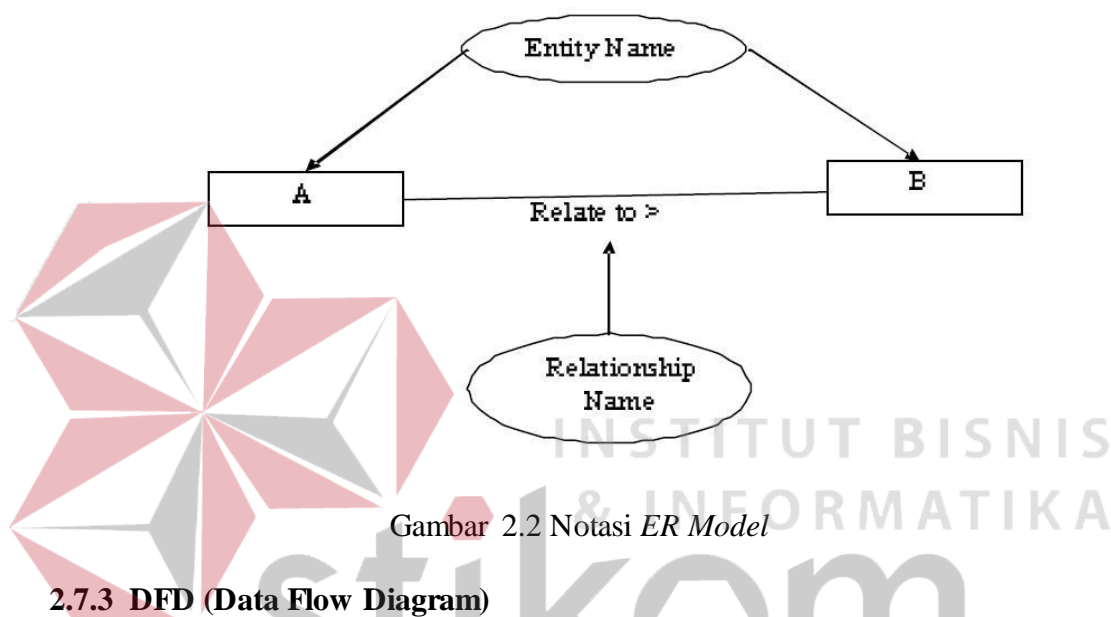
Normalisasi menurut Kadir (2003): “Proses untuk mengubah suatu tabel yang memiliki masalah tertentu kedalam dua buah tabel atau lebih yang tak lagi memiliki masalah tersebut”.

Pada proses normalisasi selalu diuji beberapa kondisi, yaitu kesulitan pada saat menambah, menghapus, mengubah dan membaca pada suatu database. Bila ada kesulitan pada pengujian tersebut maka relasi tersebut dipecah lagi menjadi beberapa tabel atau dengan kata lain perancangan database belum mencapai optimal.

### **2.7.2 Entity Relationship Diagram (ERD)**

Menurut Connolly dan Begg (2010), *entity relationship model* merupakan salah satu model yang dapat memastikan pemahaman yang tepat terhadap data dan bagaimana penggunaannya di dalam suatu organisasi.

Model ini menggunakan pendekatan *Top-Down* dalam merancang *database*, di mulai dengan mengidentifikasi data penting yang di sebut *entity* dan *relationship* antara data yang harus direpresentasikan ke dalam model, kemudian ditambahkan beberapa *attribute* dan *constraint* pada *entity*, *atributte* dan *relationship*.



Gambar 2.2 Notasi ER Model

### 2.7.3 DFD (Data Flow Diagram)

Data Flow Diagram (DFD) menurut Jogiyanto (2005) adalah: “Diagram yang menggunakan notasi simbol untuk menggambarkan arus data system”. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem yang baru yang akan dikembangkan secara logika dan menjelaskan arus data dari mulai pemasukan sampai dengan keluaran data tingkatan diagram arus data mulai dari diagram konteks yang menjelaskan secara umum suatu system atau batasan system dari level 0 dikembangkan menjadi level 1 sampai system tergambar secara rinci. Gambaran ini tidak tergantung pada perangkat keras, perangkat lunak, struktur data atau organisasi file.



Menurut Jogiyanto (2005) ada beberapa simbol yang digunakan pada DFD untuk mewakili:

1. Kesatuan Luar (*External Entity*)

Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lain yang berada pada lingkungan luarnya yang memberikan *input* atau menerima *output* dari sistem.

2. Arus Data (*Data Flow*)

Arus Data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir di antara proses, simpan data dan kesatuan luar. Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. Proses (*Process*)

Proses (*process*) menunjukkan pada bagian yang mengubah input menjadi output, yaitu menunjukkan bagaimana satu atau lebih input diubah menjadi beberapa output. Setiap proses mempunyai nama, nama dari proses ini menunjukkan apa yang dikerjakan proses.

4. Simpanan Data (*Data Store*)

*Data Store* merupakan simpanan dari data yang dapat berupa suatu file atau database pada sistem komputer.