

## **BAB III**

### **LANDASAN TEORI**

Landasan teori digunakan untuk menyelesaikan masalah secara sistematis. Pada bab ini akan membahas landasan teori yang meliputi landasan teori mengenai hal-hal dari permasalahan yang ada dan landasan teori yang membahas tentang ilmu yang terkait dalam permasalahan tersebut.

#### **3.1 Sumber Daya Manusia**

Manurut Nawawi (2001) ada tiga pengertian Sumber daya manusia yaitu:

1. Sumber daya manusia adalah manusia yang bekerja dilingkungan suatu organisasi (disebut juga personil, tenaga kerja, pekerja atau karyawan).
2. Sumber daya manusia adalah potensi manusiawi sebagai penggerak organisasi dalam mewujudkan eksistensinya.
3. Sumber daya manusia adalah potensi yang merupakan asset dan berfungsi sebagai modal (non materia/non finansial) di dalam organisasi bisnis, yang dapat mewujudkan menjadi potensi nyata secara fisik dan non fisik dalam mewujudkan eksistensi organisasi.

#### **3.2 Absensi Pegawai**

Absensi Pegawai adalah pencatatan dan pengolahan kehadiran pegawai yaitu dilakukan secara terus menerus untuk menunjang peningkatan kinerja pegawai, pencatatan dilakukan setiap jam kerja.

### 3.3 Penggajian

Menurut Warren, Reeve, dan Fess (2006, p.552) dalam akuntansi, istilah gaji diartikan sebagai jumlah tertentu yang dibayarkan kepada karyawan untuk jasa yang diberikan selama periode tertentu.

Menurut Horngren, Harrison dan Bamber (2006, p.430) gaji merupakan pendapatan yang jumlahnya dihitung per tahun, per bulan, atau per minggu, sedangkan upah merupakan pendapatan yang dihitung berdasarkan tarif per jam. Gaji merupakan hal yang penting karena:

1. Para karyawan sangat sensitif terhadap kesalahan atau ketidakwajaran dalam gaji.
2. Untuk menjaga moral karyawan dengan cara membayar gaji tepat waktu dan dengan jumlah yang akurat.
3. Merupakan hal yang diatur dengan berbagai peraturan pemerintah federal atau negara bagian.
4. Mempunyai efek yang signifikan terhadap besar laba bersih pada sebagian besar usaha.

### 3.4 Aplikasi

Aplikasi (*application*) adalah penerapan, penggunaan atau penambahan pada suatu software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel.

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.

### **3.5 Data**

Data adalah kumpulan dari angka - angka maupun karakter - karakter yang tidak memiliki arti namun merupakan keterangan yang benar dan nyata. Data dapat diolah sehingga menghasilkan informasi.

### **3.6 Informasi**

Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian – kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan.

### **3.7 Analisis Sistem**

Menurut (Jogiyanto, 2006) analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan dan pengembangannya. Tahap analisis dilakukan setelah tahap perencanaan sistem dan sebelum tahap desain sistem.

### 3.8 Desain Sistem

Setelah tahap analisa sistem selesai dilakukan, maka analisis sistem telah mendapatkan gambaran yang jelas apa yang harus dikerjakan. Kemudian memikirkan bagaimana membentuk sistem tersebut. Desain sistem dapat diartikan sebagai berikut:

- a. Tahap setelah analisis dari siklus pengembangan sistem.
- b. Pendefinisian dari kebutuhan – kebutuhan fungsional.
- c. Persiapan untuk rancang bangun implementasi.
- d. Menggambarkan bagaimana suatu sistem dibentuk.
- e. Berupa gambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.
- f. Menyangkut konfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem.

#### 3.8.1 *Document Flow*

Document Flow adalah bagan yang menunjukkan alur dalam program ataupun prosedur sistem secara fisik. Bagan alur digunakan terutama untuk alat bantu komunikasi dan dokumentasi. Bagan alur sistem digambar dengan menggunakan simbol – simbol antara lain sebagai berikut :

Tabel 3.1 *Simbol*

No.	Simbol	Nama Simbol <i>Flowchart</i>	Fungsi
1.		Dokumen	Untuk menunjukkan dokumen input dan output baik untuk proses manual, mekanik atau komputer.
2.		Proses Komputerisasi	Menunjukkan kegiatan dari operasi program komputer.
3.		Database	Untuk menyimpan data.
4.		Penghubung	Menunjukkan hubungan di halaman yang sama.
5.		Penghubung Halaman Lain	Menunjukkan hubungan di halaman lain.
6.		Terminator	Menandakan awal/akhir dari suatu sistem.
7.		Decision	Menggambarkan logika keputusan dengan nilai <i>true</i> atau <i>false</i> .
8.		Kegiatan Manual	Untuk menunjukkan pekerjaan yang dilakukan secara manual.
9.		Simpanan Offline	Untuk menunjukkan file non-komputer yang diarsip urut angka.

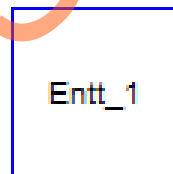
### 3.8.2 *Data Flow Diagram ( DFD )*

Menurut Kendall (2003: 241), *Data Flow Diagram* menggambarkan pandangan sejauh mungkin mengenai masukan, proses dan keluaran sistem, yang berhubungan dengan masukan, proses, dan keluaran dari model sistem yang dibahas. Serangkaian

diagram aliran data berlapis juga bisa digunakan untuk merepresentasikan dan menganalisis prosedur-prosedur mendetail dalam sistem. Prosedur-prosedur tersebut yaitu konseptualisasi bagaimana data-data berpindah di dalam organisasi, proses-proses atau transformasi dimana data-data melalui, dan apa keluarannya. Jadi, melalui suatu teknik analisa data terstruktur yang disebut *Data Flow Diagram*, penganalisis sistem dapat merepresentasi proses-proses data di dalam organisasi. Menurut Kendall (2003: 265), dalam memetakan *Data Flow Diagram*, terdapat beberapa simbol yang digunakan antara lain:

1. *External entity*

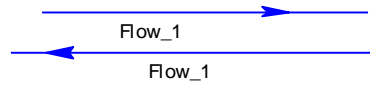
Suatu *external entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat.



Gambar 3.1 Simbol *External Entity*

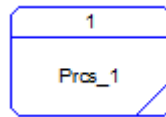
2. *Data Flow*

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

Gambar 3.2 Simbol *Data Flow*

### 3. *Process*

Suatu proses dimana beberapa tindakan atau sekelompok tindakan dijalankan.

Gambar 3.3 Simbol *Process*

### 4. *Data Store*

*Data store* adalah simbol yang digunakan untuk melambangkan proses penyimpanan data.

Gambar 3.4 Simbol *Data Store*

### 3.8.3 *Entity Relational Diagram ( ERD )*

*Entity relationship diagram (ERD)* adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. *Attribute* yaitu uraian dari entitas dimana mereka dihubungkan atau dapat dikatakan sebagai *identifier* atau *descriptors* dari entitas.

Entitas digolongkan menjadi *independent* atau *dependent entity*. *Independent entity* adalah apa yang tidak bersandar pada yang lain sebagai identifikasi. Suatu *dependent entity* adalah apa yang bersandar pada yang lain sebagai identifikasi. Selain digolongkan menjadi *independent* atau *dependent entity*, terdapat jenis- jenis entitas khusus yaitu:

1. *Associative Entity*

*Associative Entity* (juga dikenal sebagai *intersection entity*) adalah entitas yang digunakan oleh rekanan dua entitas atau lebih untuk menyatukan suatu hubungan banyak - ke - banyak (*Many to Many*)

2. *Subtypes Entity*

*Subtypes Entity* digunakan di dalam hierarki generalisasi (*generalization hierarchies*) untuk menyajikan suatu subset kejadian dari entitas orangtua, yang disebut *supertype*, tetapi yang memiliki atribut atau hubungan yang berlaku hanya untuk *subset*.

Menurut Marlinda (2004: 28), *attribute* sebagai kolom di sebuah relasi mempunyai macam-macam jenis *attribute* yaitu :

a. *Key Attribute*

*Attribute* ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya, misalnya entity mahasiswa yang *attribute*-nya NIM.



Gambar 3.5 *Key Attribute*



b. *Particial key Atribute*

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

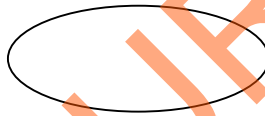
Misalnya antara Cabang (toko) dan kode cabang.



Gambar 3.6 *Particial Key Atribute*

c. *Single Vallue Atribute*

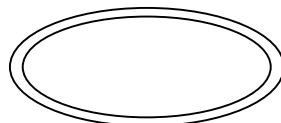
*Atribute* yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *atribute*-nya Umur (Tanggal lahir).



Gambar 3.7 *Single Value Atribute*

d. *Multi Vallue Atribute*

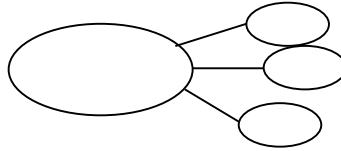
*Atribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *atribute*-nya pendidikan (SD, SMP, SMA).



Gambar 3.8 *Multi Value Atribute*

e. *Composite Attribute*

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



Gambar 3.9 *Composite Attribute*

f. *Derived Attribute*

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



Gambar 3.10 *Derived Attribute*

Model *Entity - Relationship* (ER) mula-mula diusulkan oleh Peter pada tahun 1976 sebagai cara untuk mempersatukan pandangan basis data jaringan dan relasional. Langkah sederhana dari model ER adalah model data konseptual yang memandang dunia nyata sebagai kesatuan (*entities*) dan hubungan (*relationship*).

Komponen dasar model merupakan diagram *entity-relationship* yang digunakan untuk menyajikan objek data secara *visual*. *Entity Relationship Diagram* mengilustrasikan struktur logis dari basis data yang mempunyai metodologi sebagai berikut:

Tabel 3.2 Ilustrasi Pembuatan ERD

Proses	Keterangan
1. Menentukan Entitas	Menentukan peran, kejadian, lokasi, hal nyata, dan konsep dimana pengguna akan menyimpan data.
2. Menentukan Relasi	Tentukan hubungan antara pasangan entitas menggunakan matriks relasi.
3. Gambar ERD Sementara	Entitas digambarkan dengan kotak dan relasi dengan garis yang menghubungkan entitas.
4. Isi Kardinalitas	Tentukan jumlah kejadian dari satu entitas untuk sebuah kejadian pada entitas yang berhubungan.
5. Tentukan Kunci Utama	Tentukan atribut yang mengidentifikasi satu dan hanya satu kejadian pada masing-masing entitas.
6. Gambar ERD berdasar Kunci	Hilangkan relasi <i>Many-to-Many</i> dan masukkan <i>primary</i> dan kunci tamu pada masing-masing entitas.
7. Menentukan Atribut	Tuliskan <i>field-field</i> yang diperlukan oleh sistem.
8. Pemetaan Atribut	Pasangkan atribut dengan satu entitas yang sesuai pada masing-masing atribut.
9. Gambar ERD dengan Atribut	Aturlah ERD dari langkah 6 dengan menambahkan entitas atau relasi yang ditemukan pada langkah 8.
10. Periksa Hasil	Apakah ERD sudah menggambar sistem yang akan dibangun.

*Entity Relationship Diagram* ini diperlukan agar dapat menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah *entity* dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh perandang *database*. Untuk itu, *entity relationship diagram* dibagi menjadi dua jenis model, yaitu:

1. *Conceptual Data model*

*Conceptual Data model* (CDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

2. *Physical Data Model*

*Physical Data Model* (PDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

### **3.9 Teori Yang Terkait Dengan Penyelesaian Masalah**

#### **3.9.1 Microsoft Visual Basic. Net**

*Microsoft Visual Basic .NET* adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem *.NET Fremawork*, dengan menggunakan bahasa *BASIC*. Dengan menggunakan alat ini, para pembuat program dapat membangun aplikasi *Windows Forms*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (seperti *Microsoft Visual C++*, *Visual C#*, atau *visual j#*) atau juga dapat diperoleh secara terpadu dalam *Microsoft visual Studion .NET* (Yuwanto, 2005).

Bahasa *Visual Basic .NET* sendiri menganut paradigam bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari *Microsot Visual Basic* versi sebelumnya yang dimplementasikan diatas *.NET Framework*. Peluncurannya mengundang kontrovensi, mengingat banyak sekali perubahan yang dilakukan oleh *Microsoft*, dan versi baru ini tidak kompatibel dengan versi terdahulu.

### 3.9.2 SQL Sever 2008

*Microsoft SQL Server* adalah sebuah sistem manajemen basis data relasional (RDBMS) produk *Microsoft*. Bahasa kueri utamanya adalah *transact-SQL* yang merupakan implementasi dari *SQL standar ANSI/ISO* yang digunakan oleh *Microsoft* dan *Sybase*. *SQL (Structured Query Language)* adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional (Yuwanto, 2007).

Umumnya *SQL Server* digunakan di dunia bisnis yang memiliki basis data berskala kecil sampai dengan menengah, tetapi kemudian berkembang dengan digunakannya *SQL Server* pada basis data besar. Penulis menggunakan *SQL Server 2010* untuk merancang database yang digunakan pada sistem.

### 3.9.3 Crystal Report

Merupakan *software* yang digunakan untuk pembuatan laporan. Dengan cara mengoneksi nama tabel yang akan dibuatkan laporannya. Setelah tampilan data ada maka klik dan *drag* semua *field* yang ada sesuai dengan tampilan yang diinginkan. Biasanya *crystal report* adalah komponen dari *VB.NET*.