

## BAB III

### LANDASAN TEORI

#### 3.1 Gaji

Gaji merupakan salah satu hal yang mendorong atau memotivasi pegawai untuk bekerja atau mengabdikan secara menyeluruh terhadap perusahaan. Gaji sering disebut juga sebagai upah, tetapi kedua hal tersebut memiliki sedikit perbedaan. Adapun ada beberapa definisi gaji seperti berikut ini:

*Gaji adalah imbalan kepada pegawai yang diberi tugas-tugas administratif dan pimpinan yang jumlahnya, biasanya tetap secara bulanan/tahunan. Disamping gaji, pegawai mungkin memperoleh manfaat yang diberikan dalam bentuk tunjangan, misalnya tunjangan jabatan, tunjangan perumahan, tunjangan pengobatan, tunjangan hari raya, uang transport, uang makan, dan lain-lain. (Soemarso 2005:307)*

Dari paparan di atas dapat disimpulkan bahwa gaji merupakan suatu kompensasi yang dibayarkan oleh perusahaan kepada pegawai sebagai balas jasa atas kinerja yang telah diberikan terhadap perusahaan. Kompensasi tersebut biasanya diberikan bulanan kepada pegawai.

#### 3.2 Perhitungan Gaji dan Upah

Aplikasi dirancang untuk menangani suatu yang berulang kali atau secara rutin terjadi. Gaji dan upah adalah komponen biaya yang secara rutin terjadi dalam penyelenggaraan perusahaan dan sangat penting karena berkaitan dengan

motivasi karyawan. Untuk memudahkan pelaksanaan administrasinya maka diperlukan suatu sistem yaitu aplikasi penggajian.

Untuk mengatasi adanya kesalahan dan penyimpangan dalam perhitungan dan pembayaran gaji dan upah maka perlu dibuat suatu aplikasi penggajian. Aplikasi penggajian dirancang untuk memberikan gambaran yang jelas mengenai gaji karyawan sehingga mudah dipahami dan mudah digunakan.

Menurut Mulyadi (2008 : 382) fungsi-fungsi yang terkait dalam sistem penggajian adalah:

1. Fungsi kepegawaian dan penempatan pegawai
2. Fungsi pencatat waktu
3. Fungsi pembuat daftar gaji dan upah
4. Fungsi akuntansi
5. Fungsi keuangan

### 3.3 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. (Jogiyanto, 1999: 692)

Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang. (Gordon Davis, 1991: 28)

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya

yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan.

### **3.4 Software Aplikasi**

Proses dimana keperluan pengguna dirubah ke dalam bentuk paket perangkat lunak dan atau kedalam spesifikasi pada komputer yang berdasarkan pada sistem informasi. (Kristanto, 1994: 60)

Aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu. (Dhanta, 2009: 32)

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 Document Flowchart**

*Flowchart* (bagan alir) adalah bagan (*charts*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika dapat didefinisikan sebagai bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. *Document Flowchart* adalah Bagan alir dokumen atau bisa disebut juga sebagai bagan alir formulir yang merupakan bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Dalam pembuatannya *document flowchart* memiliki ketentuan-ketentuan yang harus diperhatikan. Salah satunya adalah notasi-notasi yang ada di dalamnya. (Jogiyanto, 2005:795)

Tabel 3.1 Simbol-simbol *Flowchart*

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.

No.	Simbol	Nama Simbol <i>Flowchart</i>	Fungsi
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.

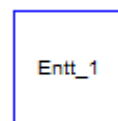
### 3.6 Data Flow Diagram (DFD)

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*

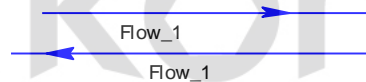
Suatu *external entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat.



Gambar 3.1 Simbol *External Entity*

### 2. *Data Flow*

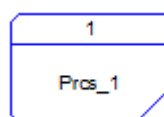
*Data Flow* atau aliran data disimbolkan dengan data tanda panah. Aliran data menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses.



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*

### 3.7 *Entity Relationship 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*

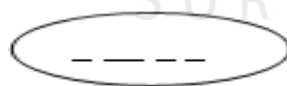
*Attribute* ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya, misalnya entity mahasiswa yang *attribute*-nya NIM.



Gambar 3.5 *Key Attribute*

### b. *Particial key Attribute*

Adalah *Attribute* yang tidak menjadi atau merupakan anggota dari *Key Primer*. Misalnya antara Cabang (toko) dan kode cabang.



Gambar 3.6 *Particial Key Attribute*

### c. *Single Vallue Attribute*

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

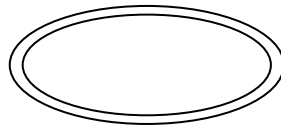


Gambar 3.7 *Single Value Attribut*



d. *Multi Value Attribute*

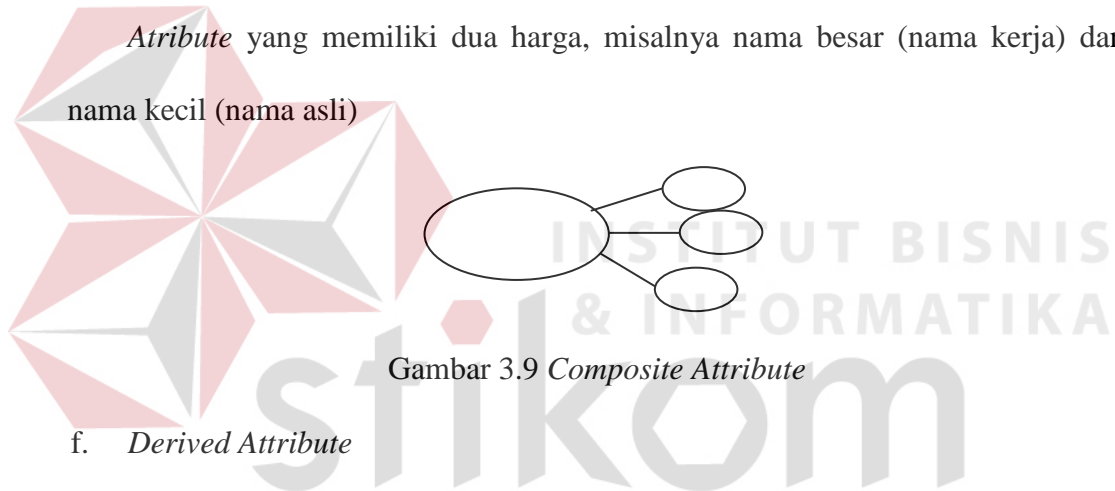
*Attribute* yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya pendidikan (SD, SMP, SMA).



Gambar 3.8 *Multi Value Attribute*

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

<b>Proses</b>	<b>Keterangan</b>
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 perancang *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.

### **3.8 Program Penunjang**

Untuk membuat rancang bangun aplikasi penggajian pada Kardi Putera Motor, dibutuhkan beberapa perangkat lunak untuk memudahkan perancangan *design* maupun sistem. Perangkat lunak tersebut antara lain:

#### **3.8.1 Microsoft Visual Basic. Net**

*Microsoft Visual Basic.NET* adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak diatas sistem *.NET Framework*, 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 Studio .NET* (Yuswanto, 2005).

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

#### **3.8.2 Database Management System**

Database Management System (DBMS) merupakan kumpulan file yang saling berkaitan dan program untuk pengelolanya. Basis data adalah kumpulan datanya, sedang program pengelolanya berdiri sendiri dalam suatu paket program

yang komersial untuk membaca data, menghapus data, dan melaporkan data dalam basis data. (Marlinda, 2004:6).

### 3.8.3 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 (Yuswanto, 2005).

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 basis data yang digunakan pada sistem.

### 3.8.4 Crystal Report

Definisi crystal report menurut Djoko Pramono (2003:16), Crystal Report merupakan program khusus untuk membuat laporan yang terpisah dengan program Visual Basic 6.0, tetapi keduanya dapat dihubungkan (linkage). Mencatat dengan Crystal Report hasilnya lebih baik dan mudah, karena pada Crystal Report banyak tersedia objek-objek maupun komponen yang mudah digunakan.