BAB III

LANDASAN TEORI

3.1 Gaji

Gaji dan upah merupakan bagian dari kompensasi-kompensasi yang paling besar yang diberikan perusahaan sebagai balas jasa kepada karyawannya dan bagi karyawan ini merupakan nilai hak dari prestasi mereka, juga sebagai motivator dalam bekerja. Mulyadi (2001: 373) mengemukakan bahwa: "Gaji umumnya merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan yang mempunyai jenjang jabatan manajer, sedangkan upah umumnya merupakan pembayaran atas penyeraha jasa yang dilakukan oleh karyawan pelaksana (buruh). Umumnya gaji dibayarkan secara tetap perbulan, sedangkan upah dibayarkan berdasarkan hari kerja, jam kerja atau jumlah satuan produk yang di hasilkan".

Gaji adalah salah satu hal yang penting bagi setiap karyawan yang bekerja dalam suatu perusahaan, karena dengan gaji yang diperoleh seseorang dapat memenuhi kebutuhan hidupnya. Gaji juga merupakan balas jasa yang dibayar secara periodik kepada karyawan tetap serta mempunyai jaminan yang pasti. Gaji adalah pemberian pembayaran finansial kepada karyawan sebagai balas jasa untuk pekerjaan yang dilaksanakan dan sebagai motivasi pelaksanaan kegiatan di waktu yang akan datang. Gaji merupakan salah satu unsur yang penting yang dapat mempengaruhi kinerja karyawan, sebab gaji adalah alat untuk memenuhi berbagai kebutuhan pegawai, sehingga dengan gaji yang diberikan pegawai akan termotivasi untuk bekerja lebih giat.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

Teori tentang gaji dapat berperan dalam meningkatkan motivasi karyawan untuk bekerja lebih efektif, meningkatkan kinerja, meningkatkan produktifitas dalam perusahaan, serta mengimbangi kekurangan dan keterlibatan komitmen yang menjadi ciri angkatan kerja masa kini. Perusahaan yang tergolong modern, saat ini banyak mengaitkan gaji dengan kinerja.

3.1.1 Peranan Gaji

Peranan gaji dapat ditinjau dari dua pihak, yaitu:

a. Aspek pemberi kerja adalah manager

Gaji merupakan unsur pokok dalam menghitung biaya produksi dan komponen dalam menentukan harga pokok yang dapat menentukan kelangsungan hidup perusahaan. Apabila suatu perusahaan memberikan gaji terlalu tinggi maka, akan mengakibatkan harga pokok tinggi pula dan bila gaji yang diberikan terlalu rendah akan mengakibatkan perusahaan kesulitan mencari tenaga kerja.

b. Aspek penerima kerja

Gaji merupakan penghasilan yang diterima oleh seseorang dan digunakan untuk memenuhi kebutuhannya. Gaji merupakan salah satu motivasi penting yang ikut mendorong karyawan untuk berprestasi, sehingga tinggi rendahnya gaji yang diberikan akan mempengaruhi kinerja dan kesetiaan karyawan.

3.1.2 Tujuan Penggajian

a. Ikatan kerjasama

Dengan pemberian gaji terjalinlah ikatan kerja sama formal antara majikan dengan karyawan. Karyawan harus mengerjakan tugas-tugasnya dengan baik, sedangkan pengusaha atau majikan wajib membayar gaji sesuai dengan perjanjian yang disepakati.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

b. Kepuasan kerja

Dengan balas jasa, karyawan akan dapat memenuhi kebutuhan-kebutuhan fisik, status sosial, dan egoistiknya sehingga memperoleh kepuasan kerja dari jabatannya.

c. Pengadaan efektif

Jika program gaji ditetapkan cukup besar, pengadaan karyawan yang *qualified* untuk perusahaan akan lebih mudah.

d. Motivasi

Jika balas jasa yang diberikan cukup besar, manajer akan mudah memotivasi bawahannya.

e. Stabilitas karyawan

Dengan program kompensasi atas prinsip adil dan layak serta eksternal konsistensi yang kompentatif maka stabilitas karyawan lebih terjamin karena turnover relatif kecil.

f. Disiplin

Dengan pemberian balas jasa yang cukup besar maka disiplin karyawan semakin baik. Karyawan akan menyadari serta mentaati peraturan-peraturan yang berlaku.

g. Pengaruh serikat buruh

Dengan program kompensasi yang baik pengaruh serikat buruh dapat dihindarkan dan karyawan akan berkonsentrasi pada pekerjaannya.

h. Pengaruh pemerintah

Jika program gaji sesuai dengan undang-undang yang berlaku (seperti batas gaji minimum) maka *intervensi* pemerintah dapat dihindarkan.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

3.1.3 Fungsi Penggajian

Fungsi gaji bukan hanya membantu manajer personalia dalam menentukan gaji yang adil dan layak saja, tetapi masih ada fungsi-fungsi yang lain, yaitu:

a. Untuk menarik pekerja yang mempunyai kemampuan ke dalam organisasi.
b. Untuk mendorong pekerja agar menunjukkan prestasi yang tinggi.
c. Untuk memelihara prestasi pekerja selama periode yang panjang.

3.2 Presensi dan Penggajian

Presensi merupakan daftar hadir ditempat kerja pada hari kerja. Ketidakhadiran mempunyai hubungan yang negatif dengan prestasi kerja. Sistem presensi adalah proses yang digunakan untuk melakukan presensi tersebut. Gaji merupakan kembalian-kembalian finansial yang diterima oleh para karyawan sebagai ganti kontribusi mereka terhadap organisasi.

Sistem penggajian adalah proses yang menentukan tingkat penggajian, memantau atau mengawasi, mengembangkan serta mengendalikan gaji pegawai. Sistem penggajian merupakan fungsi penting yang menjadi tanggung jawab Manajemen Sumber Daya Manusia (Kampilnastuti dan Widiasari, 2005: 12)

Penggajian menurut Peraturan Pemerintah No.8 Tahun 1981 adalah suatu penerimaan sebagai imbalan dari pengusaha kepada tenaga kerja untuk suatu pekerjaan atau jasa yang telah atau akan dilakukan, dinyatakan atau dinilai dalam bentuk uang yang ditetapkan menurut suatu persetujuan atau peraturan perundang-undangan dan dibayarkan atas dasar suatu perjanjian kerja antara pengusaha (pemberi kerja) dan pekerja termasuk tujangan, baik untuk pekerja sendiri maupun keluarganya.

Gaji merupakan sesuatu sumber penghidupan yang penting bagi setiap orang yang telah mengikatkan dirinya dalam perjanjian kerja, yang berarti bahwa tidak ada manusia yang mau mengerahkan tenaga atau jasanya untuk mengerjakan sesuatu secara terus-menerus atau dalam waktu tertentu demi kepentingan orang lain atau pihak lain tanpa memperoleh gaji atau imbalan yang memadai.

Penghasilan bruto meliputi gaji normal ditambah dengan bonus, lembur dan tunjangan-tunjangan lainnya. Penghasilan bersih adalah penghasilan bruto setelah dikurangi potongan-potongan. Ini merupakan jumlah yang dibayarkan oleh perusahaan atau pemberi kerja kepada karyawan.

Sistem penggajian memegang peranan penting karena sistem ini akan menentukan berapa gaji yang semestinya diterima karyawan. Namun demikian, sistem ini harus mampu memberikan tingkat gaji sebagaimana mestinya. Perusahaan memberikan gaji untuk merangsang karyawan agar bersedia bekerja dengan baik. (Heidjrachman dan Husnan, 1996: 8).

3.3 Analisa dan Perancangan Sistem

Analisa sistem dilakukan dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya. Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun sebuah aplikasi.

Menurut Kendall (2003: 7), analisa dan perancangan sistem dipergunakan untuk menganalisis, merancang dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.

3.4 *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:

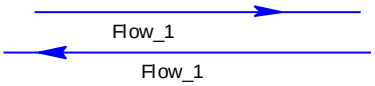
1. *External entity*

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*

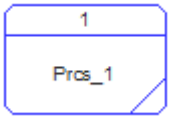
Data Flow atau aliran data disimbolkan dengan data tanda panah. Aliran data menunjukkan arus data atau aliran data yang menghubungkan dua proses atau *entitas* dengan proses.



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*

3.5 *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 terdefinisikan 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 benyak - 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), *atribute* sebagai kolom di sebuah relasi mempunyai macam-macam jenis *atribute* yaitu :

a. *Key Atribute*

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

Gambar 3.5 *Key Attribute*

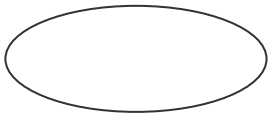
b. *Particial key Atribute*

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

Gambar 3.6 *Particial Key Attribute*

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 Attribute*

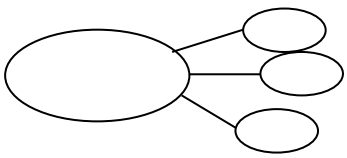
d. *Multi Vallue Atribute*

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

Gambar 3.8 *Multi Value Attribute*

e. *Composite Atribute*

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



Gambar 3.9 *Composite Attribute*

f. *Derived Attribute*

Attribute yang 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 3.1 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 *Many-to-Many* dan masukkan *primary* dan kunci tamu pada masing-masing entitas.
7. Menentukan Atribut	Tuliskan *field-field* 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:

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

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.

3.6 Program Penunjang

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

3.6.1 *Power Designer*

Power designer merupakan suatu *tool* berupa *software* untuk mendesain sistem dan rancangan *Entity Relationship Diagram* (ERD) yang dikembangkan oleh *Sybase Inc.* Ada dua model data, yaitu: *Entity Relationship Diagram* (ERD) dan model relasional. Keduanya menyediakan cara untuk mendeskripsikan perancangan basis data pada peringkat logika.

a. Model ERD atau *Conceptual Data Model* : model yang di buat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas serta hubungan (*relationship*) antara entitas-entitas itu.
b. Model Relasional atau *Physical Data Model* : model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom dimana setiap kolom memiliki nama yang unik.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA

3.6.2 *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*.

Bahasa *Visual Basic .NET* sendiri menganut paradigma 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.

3.6.3 *.NET Framework*

Microsoft .NET Framework (di baca *Microsoft dot Net Framework*) adalah sebuah komponen yang dapat ditambahkan ke sistem operasi *Microsoft Windows* atau telah terintegrasi ke dalam *Windows* (mulai dari *Windows server* 2003 dan versi-versi *Windows* Terbaru). Kerangka kerja ini menyediakan sejumlah besar solusi-solusi program untuk memenuhi kebutuhan-kebutuhan umum suatu program baru, dan mengatur eksekusi program-program yang ditulis secara khusus untuk framework ini. *.NET Framework* adalah kunci penawaran utama dari *Microsof*t, dan dimaksudkan untuk digunakan oleh sebagian besar aplikasi-aplikasi baru yang dibuat untuk *platform Windows*.

Pada dasarnya, *.NET framework* memiliki 2 komponen utama: *CLR* dan *.NET Framework Class Library*. Program-program yang ditulis untuk *.NET Framework* dijalankan pada suatu lingkungan *software* yang mengatur persyaratan-persyaratan *runtime* program. *Runtime environment* ini, yang juga merupakan suatu bagian dari *.NET Framework*, dikenal sebagai *Common Language Runtime* (CLR). CLR menyediakan penampilan dari *application virtual machine*, sehingga para *programmer* tidak perlu mengetahui kemampuan CPU tertentu yang akan menjalankan program. CLR juga menyediakan layanan-layanan penting lainnya seperti jaminan keamanan, pengaturan *memori*, *garbage collection* dan *exception handling* atau penanganan kesalahan pada saat *runtime*.

Class Library dan CLR ini merupakan komponen inti dari *.NET Framework*. Kerangka kerja itu pun dibuat sedemikian rupa agar para *programmer* dapat mengembangkan program komputer dengan jauh lebih mudah, dan juga untuk mengurangi kerawanan aplikasi dan juga komputer dari beberapa ancaman keamanan. CLR adalah turunan dari CLI (*Common Language Infrastructure*) yang saat ini merupakan standar ECMA.

Solusi-solusi program pembentuk *Class Library* dari *.NET Framework* mengcover area yang luas dari kebutuhan program pada bidang *user interface*, pengaksesan data, koneksi basis data, *kriptografi*, pembuatan aplikasi berbasis *web*, *algoritma numerik*, dan komunikasi jaringan. Fungsi-fungsi yang ada dalam *class library* dapat digabungkan oleh *programmer* dengan kodenya sendiri untuk membuat suatu program aplikasi baru

3.6.4 *SQL Server 2010*

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.

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.

3.6.5 *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*.

INSTITUT BISNIS & INFORMATIKA stikom SURABAYA