

## BAB III

### LANDASAN TEORI

#### 3.1 Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. (Hartono, 1999:1) Informasi adalah data yang di olah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lanjut. (Hartono, 1999:8)

Informasi dapat menggambarkan kejadian nyata yang di gunakan untuk mengambil keputusan. Sumber dari informasi adalah data yang berbentuk huruf, simbol, alphabet dan sebagainya. Sistem informasi mempunyai elemen utama, yaitu data yang menyediakan informasi, prosedur yang memberitahu pengguna bagaimana mengoperasikan sistem informasi, menyelesaikan masalah, membuat keputusan dan menggunakan sistem informasi tersebut. Orang-orang dalam sistem informasi membuat prosedur untuk mengolah dan memanipulasi data sehingga menghasilkan informasi dan menyebarkan informasi tersebut ke lingkungannya.

Model dasar sistem adalah masukan, pengolahan dan pengeluaran. Fungsi pengolahan informasi sering membutuhkan data yang telah di kumpulkan dan diolah dalam waktu periode sebelumnya. Oleh karna itu, dalam model sistem informasi ditambahkan pula media penyimpanan data. Maka fungsi pengolahan informasi bukan lagi mengubah data menjadi informasi, tetapi juga menyimpan data untuk penggunaan lanjutan.

Sistem informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu untuk diolah lebih lanjut. Karena pada saat ini, para pembuat keputusan memahami bahwa informasi tidak hanya sekedar produk sampingan bisnis yang sedang dijalankan, namun juga sebagai bahan pengisi bisnis dan menjadi faktor kritis dalam menentukan kesuksesan atau kegagalan suatu usaha.

Untuk menghasilkan informasi yang berkualitas maka di buatlah sistem informasi. (Hartono, 1989:11). Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang di perlukan.

Informasi merupakan data yang telah diolah menjadi suatu bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau di masa

yang akan datang. Sumber dari informasi adalah data, merupakan bentuk yang masih mentah dan belum dapat bercerita banyak, sehingga membutuhkan pengolahan lebih lanjut. Kualitas dari sistem informasi bergantung pada dua hal, yaitu:

1. Informasi harus akurat, dimana informasi tersebut harus bebas dari kesalahan.
2. Informasi tersebut harus relevan, supaya informasi tersebut dapat memberikan masukan bagi penerimanya.

Ada beragam definisi sistem informasi, sebagaimana tercantum pada Tabel 3.1. Dari berbagai definisi tersebut, dapat di simpulkan bahwa sistem informasi mencakup sejumlah komponen (manusia, komputer, teknologi informasi, dan prosedur kerja), ada sesuatu yang di proses (data menjadi informasi), dan di maksudkan untuk mencapai suatu sasaran atau tujuan.

Tabel 3.1 Definisi Sistem Informasi

Sumber	Definisi
Alter (1992)	Sistem informasi adalah kombinasi antara prosedur kerja, informasi, orang dan teknologi informasi yang di organisasikan untuk mencapai tujuan dalam sebuah organisasi.
Hall (2001)	Sistem informasi adalah sebuah rangkaian prosedur formal dimana data di kelompokkan, diproses menjadi informasi,

	dan didistribusikan kepada pemakai.
Tuban, McLean, dan Wetherbe (1999)	Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuan yang spesifik.

Istilah sistem informasi juga sering di kacakaukan dengan sistem informasi manajemen (SIM). Kedua hal ini sebenarnya tidak sama. Sistem informasi manajemen merupakan salah satu jenis sistem informasi, yang secara khusus ditujukan untuk menghasilkan informasi bagi pihak manajemen dan untuk pengambilan keputusan.


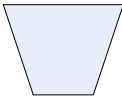



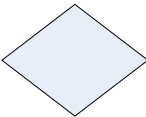
### **3.2 Sistem Flow**

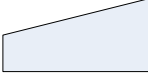
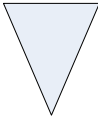

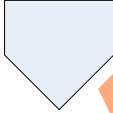

Sistem flow adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem di mana bagan ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem dan biasanya dalam membuat sistem flow sebaiknya ditentukan pula fungsi-fungsi yang melaksanakan atau bertanggung jawab terhadap sub-sistem yang ada (Hartono, 1998:10)

Terdapat berbagai macam bentuk simbol yang digunakan untuk merancang sebuah desain dari sistem, diantaranya adalah terminator, manual operation,

document, process, database, manual input, decision, off-line storage, on-page reference, dan off-page reference.

Tabel 3.2 Desain Simbol Sistem

Simbol	Keterangan
1. Simbol <i>Terminator</i> 	Bentuk simbol yang di gunakan sebagai tanda di mulainya jalan proses sistem ataupun tanda akhir dari sebuah pengerjaan suatu system
2. Simbol <i>Manual Operation</i> 	Menggambarkan sebuah proses kerja yang dilakukan tanpa menggunakan computer sebagai mediana (menggunakan proses manual)
3. Simbol <i>Document</i> 	Menunjukkan dokumen input dan output baik untuk proses manual atau komputer.
4. Simbol <i>Process</i> 	Sebuah bentuk kerja sistem yang dilakukan secara terkomputerisasi
5. Simbol <i>Database</i> 	Media penyimpanan data yang bersifat terkomputerisasi
6. Simbol <i>Decision</i> 	Operator logika yang digunakan sebagai penentu keputusan dari suatu permintaan atau proses dengan dua nilai, benar dan salah

<p>7. Simbol <i>Manual Input</i></p> 	<p>menunjukkan proses input kedalam database melalui keyboard</p>
<p>8. Simbol <i>Off – line Storage</i></p> 	<p>Media penyimpanan yang berbeda dengan database, dimana media penyimpanan ini menyimpan dokumen secara manual atau lebih dikenal dengan nama arsip</p>
<p>9. Simbol <i>On – page reference</i></p> 	<p>Simbol untuk menghubungkan bagan desain sebuah sistem apabila hubungan arus data yang ada terlalu jauh dalam permasalahan letaknya</p>
<p>10. Simbol <i>Off – page reference</i></p> 	<p>Simbol ini hanya digunakan apabila arus data yang ada dilanjutkan ke halaman yang berbeda.</p>
<p>11. Simbol <i>Paper Tape</i></p> 	<p>Sebuah simbol yang umumnya menggantikan bentuk penggambaran jenis pembayaran yang digunakan (missal : uang) dalam transaksi yang ada pada sistem yang di rancang</p>

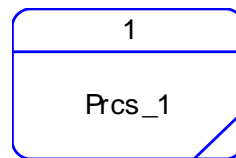
### 3.3 Data Flow Diagram (DFD)

Menurut (Kristanto, 2004:12), *Data Flow Diagram* (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan

kemana tujuan data yang keluar dari sistem, dimana data tersebut disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan, dan proses yang dikenakan pada data tersebut.

Data *flow* diagram merupakan suatu metode pengembangan sistem yang terstruktur (*structure analysis and design*). Penggunaan notasi dalam DFD sangat membantu untuk memahami suatu system pada semua tingkat kompleksitas. Pada tahap analisis, penggunaan notasi ini dapat membantu dalam berkomunikasi dengan pemakai sistem untuk memahami sistem secara logika.

Didalam DFD, terdapat empat simbol yang digunakan yaitu *process*, *external entity*, *data store*, dan *data flow*. Simbol *process* digunakan untuk melakukan suatu perubahan berdasarkan data yang diinputkan dan menghasilkan data dari perubahan tersebut. Simbol dari *process* dapat dilihat pada gambar 3.1.



Gambar 3.1 *Process*

Pada bentuk gambar *process*, bagian atas berisi nomor untuk identitas proses. Suatu proses dengan nomor 0 (nol atau kosong) menandakan bahwa proses tersebut adalah *context diagram*. Diagram ini merupakan level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya. Pembuatan *context*

*diagram* dapat dilakukan dengan terlebih dahulu menentukan nama sistemnya, menentukan batasan dari sistem, dan menentukan terminator yang diterima atau diberikan daripada sistem untuk kemudian dilakukan penggambaran.

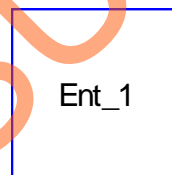
Nomor 1, 2, 3, dan seterusnya menandakan bahwa proses tersebut diartikan sebagai proses level-0 (nol) yang merupakan hasil turunan atau *decompose* dari proses *context diagram*. Proses level-0 membahas sistem secara lebih mendetil, baik dipandang dari segi kegiatan dari sebuah bagian, alur data yang ada, maupun database yang digunakan di dalamnya. Pembuatannya dapat dilakukan dengan cara menentukan proses utama yang ada dalam sistem, menentukan alur data yang diterima dan diberikan masing-masing proses dari pada sistem sambil memperhatikan konsep keseimbangan (alur data yang masuk atau keluar dari suatu level harus sama dengan alur data yang masuk dan keluar pada level berikutnya), memunculkan *data store* sebagai sumber maupun tujuan data (*optional*), menggambarkan diagram level-0, menghindari perpotoingan arus data, dan melakukan pemberian nomor pada proses utama (nomor tidak menunjukkan urutan proses).

Nomor 1.1, 1.2, 2.1, 2.2, dan seterusnya merupakan sebuah proses turunan atau *decompose* dari proses level-0 yang disebut sebagai proses level-1 (satu). Proses level-1 menggambarkan detil kerja dari sebuah bagian dalam sebuah sistem. Menggambarinya dilakukan dengan cara menentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level-0, menentukan apa yang diterima atau diberikan masing-masing sub-proses daripada sistem dan tetap memperhatikan



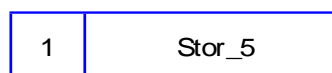
konsep keseimbangan, memunculkan data *store* sebagai sumber maupun tujuan alur data (*optional*), menggambar DFD level-1, dan berusaha untuk menghindari perpotongan arus data. Hasil turunan akhir disebut sebagai *the lowest level*, di mana hasil akhir ini tergantung dari kompleksitas sistem yang ada.

*External entity* disimbolkan dengan bentuk persegi yang digunakan untuk menggambarkan pelaku-pelaku sistem yang terkait, dapat berupa orang-orang, organisasi maupun instansi. *External entity* dapat memberikan masukan kepada process dan mendapatkan keluaran dari process. Simbol *external entity* dapat dilihat pada gambar 3.2.



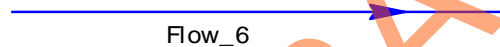
Gambar 3.2 *External Entity*

*Data store* digunakan sebagai media penyimpanan suatu data yang dapat berupa file atau *database*, arsip atau catatan manual, lemari file, dan tabel-tabel dalam *database*. Penamaan *data store* harus sesuai dengan bentuk data yang tersimpan pada *data store* tersebut, misalnya tabel pelanggan, tabel detail penjualan, tabel detail pembelian, dan lain-lain. Simbol *data store* dapat dilihat pada gambar 3.3.



Gambar 3.3 *Data Store*

*Data flow* merupakan penghubung antar *external entity* dengan *process* dan *process* dengan *data store*. *Data flow* menunjukkan aliran data dari satu titik ke titik lainnya dengan tanda anak panah mengarah ke tujuan data. Penamaan *data flow* harus menggunakan kata benda, karena didalam *data flow* mengandung sekumpulan data. Simbol *data flow* dapat dilihat pada gambar 3.4

Gambar 3.4 *Data Flow*

### 3.4 Entity Relationship Diagram (ERD)

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antar penyimpanan. ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks. Dengan ERD kita dapat 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*

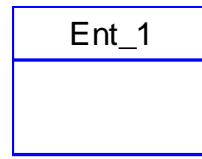
Jenis hubungan antar tabel dimana satu *record* pada satu tabel terhubung dengan beberapa *record* pada tabel lain. Jenis hubungan ini merupakan yang paling sering digunakan. Misalnya suatu pekerjaan hanya dikerjakan oleh satu departemen saja. Namun suatu departemen dapat mengerjakan beberapa macam pekerjaan 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.

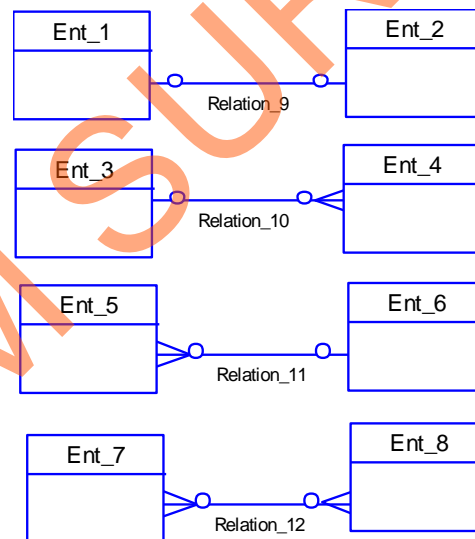
*Entity Relationship Diagram*, atau yang lebih dikenal dengan nama ERD, digunakan untuk mengimplementasikan, menentukan, dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan *database*. ERD menyediakan bentuk untuk menunjukkan struktur keseluruhan kebutuhan data dari pemakai. Adapun elemen-elemen yang terdapat pada ERD, adalah sebagai berikut:

1. *Entity* atau entitas, digambarkan dalam bentuk persegi seperti pada gambar 3.5



Gambar 3.5 Entity

2. *Relation* atau relasi merupakan penghubung antara entitas dengan entitas. Terdapat beberapa jenis relasi yang dapat digunakan, seperti *one-to-one*, *one-to-many*, *many-to-one*, dan *many-to-many*. Bentuk alur relasi secara detail dapat dilihat pada gambar 3.6.



Gambar 3.6 Relation Of Entity

### 3.5 Program Penunjang

Untuk membuat aplikasi absensi dan penggajian pada CV. Maha Meru, dibutuhkan beberapa perangkat lunak untuk memudahkan perancangan design maupun sistem. Perangkat lunak tersebut antara lain:

#### 3.5.1 *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* (CDM) : model yang di buat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (entity) serta hubungan (relationship) antara entitas-entitas itu.
- b. Model Relasional atau *Physical Data Model* (PDM) : 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.

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

### 3.5.3 *.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 *Microsoft*, 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*/penanganan kesalahan pada saat *runtime*. *Class Library* dan CLR ini merupakan komponen inti dari *.NET Framework*. kerangka kerja itupun dibuat sedemikian rupa agar para *programmer* dapat mengembangkan program komputer dengan jauh lebih mudah, dan juga untuk mengurangi kerawanan aplikasi dan juga computer dari beberapa ancaman keamanan.

CLR adalah turunan dari CLI (*Common Language Infrastructure*) yang saat ini merupakan standar ECMA. Untuk keterangan lebih lanjut, silahkan mengunjungi situs ECMA atau kunjungi sumber pranala dibawah artikel ini.

Solusi-solusi program pembentuk *Class Library* dari *.NET Framework* melindungi 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

#### **3.5.4 SQL Server 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.

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 2008* untuk merancang *database* yang digunakan pada sistem.

#### **3.5.5 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*.

### **3.6 Analisa dan Perancangan Sistem**

Analisis sistem didefinisikan sebagai uraian dari sistem informasi yang besar dan utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasikan permasalahan, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikannya.

Langkah-langkah dasar dalam melakukan analisa sistem :

1. Identifikasi masalah
2. Memahami kerja dari sistem
3. Menganalisa sistem
4. Membuat laporan hasil analisis

Perancang sistem dapat didefinisikan sebagai tahap setelah perancangan sistem secara umum dan perancangan sistem secara terinci. Perancangan sistem mempunyai dua tujuan utama yaitu memenuhi kebutuhan kepada pemakai dan untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram dan ahli teknik lainnya yang terlibat.

### **3.7 Presensi**

Presensi merupakan daftar hadir di tempat kerja pada hari kerja. Ketidakhadiran mempunyai hubungan yang negative dengan prestasi kerja. Sistem presensi adalah proses yang digunakan untuk melakukan presensi tersebut. Gaji merupakan kembalian-kembalian finansial yang diterima oleh para pegawai 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 Widiyasari, 2005:12).

### **3.8 Penggajian**

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 suatu imbalan jasa yang harus diberikan pengusaha (pemberi kerja) kepada pekerja sebagai balas jasa terhadap hasil kerja mereka. 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 tertentu, seperti pajak penghasilan karyawan, pension, astek dan sebagainya. 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).