BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Suatu sistem sangatlah dibutuhkan dalam suatu perusahaan atau instansi pemerintahan, karena sistem sangatlah menunjang terhadap kinerja perusahaan atau instansi pemerintah, baik yang berskala kecil maupun besar. Supaya dapat berjalan dengan baik diperlukan kerjasama diantara unsur-unsur yang terkait dalam sistem tersebut.

Konsep dasar sistem ada dua pendekatan yaitu penekanan pada prosedurnya dan penekanan pada komponennya. Menurut Jerry Fitz Gerald pengertian sistem yang menakankan pada prosedurnya adalah

sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. (Jogiyanto,2005 : 1).

Adapun menurut Jogiyanto dalam bukunya "Analisis dan Desian Sistem Informasi" pengertian system yang menggunakan pendekatan pada komponennya yaitu.

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu".(Jogiyanto,2005:2)

2.1.1 Karakteristik Sistem

Suatu system mempunyai karakteristik atau sifat – sifat yang tertentu, yaitu mempunyai komponen – komponen *(components)*, Batasan sistem *(boundary)*, lingkungan luar sistem *(environments)*, penghubung *(interface)*, masukan *(input)*, keluaran *(output)*, pengolahan *(process)*, dan sasaran *(objectives)* atau tujuan *(goal)*.

Adapun sistem mempunyai beberapa katekteristik atau sifat-sifat tertentu antara lain yaitu:

a. Komponen sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak perduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut supra sistem, misalnya suatu perusahaan dapat disebut dengan suatu sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya.

b. Batas sistem (*boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu system dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan luar sistem (*environment*)

Adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar system dapat menguntungkan dan merugikan.

1. Keuntungan : Merupakan energi dari sistem yang harus tetap dijaga dan dipelihara karena bersifat menguntungkan.
2. Kerugian : Merupakan energi dari sistem yang harus ditahan dan dikendalikan, energi ini bisa merusak kelangsungan hidup dari sistem.

d. Penghubung sistem (*interface*)

Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Adapun kegunaan penghubung lainnya adalah.

1. Memungkinkan sumber-sumber dapat mengalir dari subsistem yang satu ke subsistem yang lainnya.
2. Keluaran dari subsistem menjadi masukan untuk subsistem yang lainnya melalui penghubung.
3. Satu subsistem dapat berintegrasi dengan subsistem yang lainnya untuk membentuk satu kesatuan.

e. Masukan sistem *(input)*

Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam system komputer, program adalah maintanance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

f. Keluaran sistem *(Output)*

Merupakan hasil dari energi yang diolah oleh sistem.

g. Pengolah sistem *(Process)*

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

h. Sasaran sistem

Kalau sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

2.1.2 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya adalah sebagai berikut ini:

1. Sistem Abstrak dan Sistem Fisik Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sedangkan sistem fisik adalah sistem yang tampak secara fisik.

2. Sistem Alamiah *(natural system)* dan Sistem Buatan Manusia *(human made system).*

   Sistem alamiah *(natural system)* adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Sistem buatan manusia *(human made system)* merupakan sistem yang melibatkan hubungan manusia dengan mesin yang disebut dengan *human machine system.*

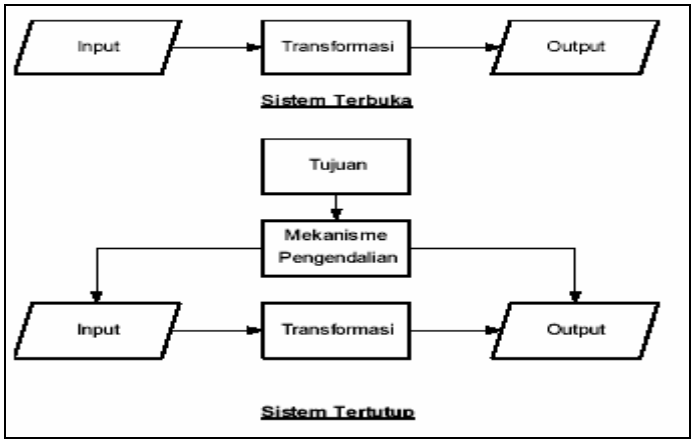
3. Sistem Deterministik (system tertentu) dan Sistem Probabilistik (sistem tak tentu).

   Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem Tertutup *(closed system)* dan Sistem Terbuka *(open system).*

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

   Klasifikasi sistem terbuka *(open system* .dan system tertutup *(closed system)* dapat digambarkan sebagai berikut :



Gambar 2.1 Klasifikasi sistem terbuka dan tertutup

sumber : http://www.unhas.ac.id/

2.2 Konsep Dasar Informasi

Suatu sistem yang kurang mendapatkan informasi akan menjadi luruh, kerdil dan akhirnya berakhir. Informasi yang berguna bagi sistem akan menghindari proses kematian sebuah sistem (*Robert N. Anthony dan John Dearden*).

Menurut Jogiyanto dalam buku *'Analisis dan desain sistem informasi'* adalah :

"Informasi diartikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya" . (Jogiyanto,2005:8).

Sumber dari informasi adalah data. Sedangkan data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu. Didalam dunia bisnis, kejadian nyata yang sering terjadi adalah perubahan dari suatu nilai yang disebut dengan transaksi. Misalnya penjualan adalah transaksi perubahan nilai barang menjadi nilai uang atau nilai piutang dagang. Satuan nyata (*fact dan entity*)

adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul – betul ada dan terjadi.

Menurut Robert N.antony dan John Dearden, Data adalah :

‘ Bentuk jamak dari bentuk tunggal datum atau data-item’.

dan

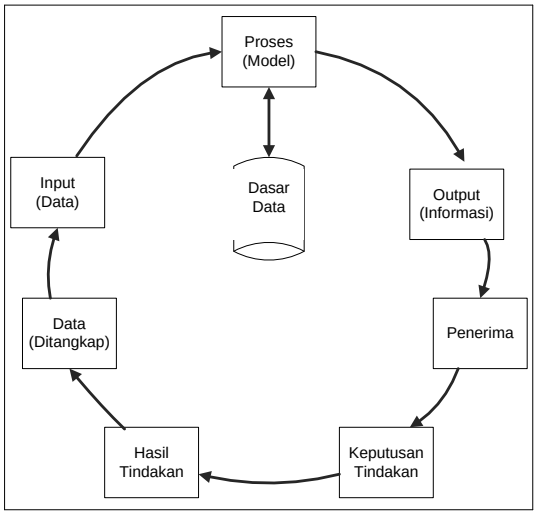
“ Data Merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.” (Jogiyanto, 2005:8).

Pengertian Informasi selalu dikaitkan dengan data, namun arti dari masing-masing kata dalam pengertian tersebut berbeda. Keberadaan suatu data sangat menunjang terhadap informasi, karena data merupakan bahan mentah yang diperlukan untuk mengambil keputusan.

2.2.1 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi.

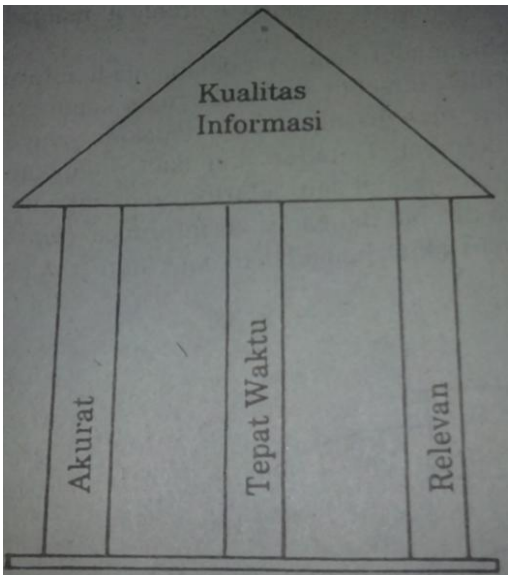
Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, siklus ini oleh John Burch disebut dengan siklus informasi. Siklus ini disebut juga dengan siklus pengolahan data.



Gambar 2.2 Siklus Informasi
Sumber: Jogiyanto HM (2005 : 9)

2.2.2 Kualitas Informasi

Kualitas dari suatu informasi tergantung dari tiga hal, yaitu informasi harus akurat, tepat pada waktunya dan relevan.



Gambar 2.3 Pilar Kualitas Informasi
Sumber:Jogiyanto HM (2005: 10)

1. Akurat

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.

2. Tepat pada waktunya

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan didalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka berakibat fatal untuk organisasi.

3. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.2.3 Nilai Informasi

Nilai dari informasi (*value of information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak

memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah yang tertentu untuk memperolehnya, karena sebagian besar informasi dinikmati tidak hanya oleh satu pihak tetapi oleh banyak perusahaan.

2.3 Konsep Dasar Sistem Informasi

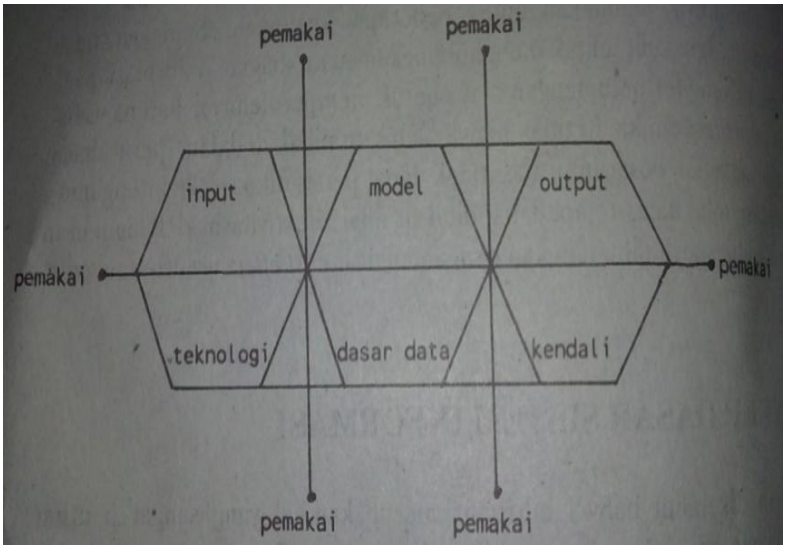
Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen didalam pengambilan keputusan. Informasi dapat diperoleh dari sistem informasi (*information system*) atau disebut juga dengan *processing system* atau *Information processing system* atau *information – generating system.*

Sistem Informasi menurut Jogiyanto (2005: 11) adalah :

"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 diperlukan".

2.3.1 Komponen Sistem Informasi

John Burch Gary Grudnitski mengemukakan bahwa:
"Sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan, yaitu blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya."



Gambar 2.4 Blok system informasi yang berinteraksi
Sumber: Jogiyanto HM (2005 : 12)

Adapun komponen-komponen dari suatu system informasi meliputi:

1. Blok Masukan

Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

Teknologi merupakan “kotak alat” dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*) dan perangkat keras *(hardware)*.

5. Blok Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat lunak untuk memanipulasinya. Data perlu disimpan didalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisensi kapasitas penyimpanannya. Basis data di akses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan BBMS (*data base management system*).

6. Blok Kendali.

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, air, kegagalan–kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur kesalahan-kesalahan dapat langsung cepat diatasi.

2.3.2 Komponen Sistem Informasi Berbasis Komputer

Pengelompokan komponen-komponen sistem informasi adalah sebagai berikut:

1. Perangkat Keras *(Hardware)*

Perangkat keras *(hardware)* mencakup piranti-piranti fisik seperti komputer dan printer.

2. Perangkat Lunak *(Software)* atau Program

Perangkat lunak *(software)* merupakan sekumpulan instruksi yang memungkinkan perangkat keras untuk memproses data atau merupakan program-program yang digunakan untuk menjalankan komputer.

3. Prosedur *(Procedure)*

Prosedur *(procedure)* merupakan sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

4. Orang *(Brainware)*

Orang *(brainware)* merupakan semua pihak yang bertanggungjawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.

5. Basis Data *(Database)*

Basis data *(database)* merupakan sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.

6. Jaringan Komputer dan Komunikasi Data

Jaringan komunikasi merupakan sistem penghubungan yang memungkinan sesumber (*resource*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

2.4 Metode Pengembangan Sistem

Adapun metode pengembangan yang digunakan adalah prototype, dalam buku Kendal (2003:221) prototyping sistem informasi yaitu suatu teknik yang sangat berguna untuk mengumpulkan informasi tertentu mengenai syarat-syarat informasi pengguna secara cepat. Prototyping yang efektif seharusnya dilakukan pada awal-awal siklus pengembangan sistem, yakni selama fase penetapan syarat-syarat. Prototyping adalah suatu teknik yang sangat rumitkarena memerlukan pengetahuan mengenai siklus hidup pengembangan system secara keseluruhan.

Sistem prototype merupakan bagian operasional dari sistem secara keseluruhan yang akan dibangun. Prototype merupakan suatu cara yang baik untuk mendapatkan umpan balik mengenai sistem yang diajukan dan mengenai bagaimana sistem tersebut tersedia untuk memenuhi kebutuhan informasi pengguna.

Ada empat petunjuk yang harus diamati saat mengintegrasikan prototype kedalam fase penetapan siklus hidup pengembangan sistem, yakni:

1. Bekerja sesuai modul
2. Membangun prototype dengan cepat
3. Memodifikasi prototype dengan iterasi yang berurutan

4. Menekankan antar muka pengguna.

2.5 Pemodelan UML

Pada inti peran UML dalam pengembangan perangkat lunak, orang – orang memiliki cara – cara yang berbeda dalam penggunaannya, perbedaan – perbedaan yang masih dibawa dari bahasa – bahasa grafis lain. Perbedaan – perbedaan ini mengakibatkan perslisihan yang panjang dank eras tentang bagaimana UML seharusnya digunakan.

2.5.1 Pengertian UML

Unified Modeling Language (*UML*) adalah keluarga notasi grafis yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang *UML* berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif (Martin Fowler 2005:1).

Pada tahap analisis, meliputi usaha untuk mengetahui apa kemampuan sebuah sistem yang diinginkan pengguna dan pelanggan dari sebuah perangkat lunak.

2.5.2 Diagram-Diagram Dalam UML

Adapun diagram – diagram dalam UML mempunyai beberapa diagram antara lain yaitu:

2.5.2.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng- *create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di- *include* akan dipanggil setiap kali *use case* yang meng- *include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common* . Sebuah *use case* juga dapat meng- *extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2.5.2.2 Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)

2. Atribut

3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung

diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.* Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

2.5.2.3 Hubungan Antar *Class*

1) Asosiasi, yaitu hubungan statis antar *class* . Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class* .
2) Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").
3) Pewarisan, yaitu hubungan hirarkis antar *class* . *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4) Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di- *passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

2.5.2.4 Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*). Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

2.5.2.5 Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di- *trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem)

secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state* , standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

2.5.2.6 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* , dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men- *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message*

digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class* . *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*

2.5.2.7 Collaboration Diagram

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram,* tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number* , di mana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

2.5.2.8 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code* , baik berisi *source code* maupun *binary code* , baik *library* maupun *executable* , baik yang muncul pada *compile time, link time* , maupun *run time* . Umumnya komponen terbentuk dari

beberapa *class* dan/atau *package* , tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

2.5.2.9 Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di- *deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server, *workstation* , atau piranti keras lain yang digunakan untuk men- *deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

2.6 Konfigurasi Jaringan Komputer

Menurut Abdul Kadir (2003:346) Jaringan komputer adalah hubungan dua buah simpul (umumnya berupa komputer) atau lebih yang tujuan utamanya adalah untuk melakukan pertukaran data. Jaringan komputer juga dapat diartikan sebagai kumpulan sebuah terminal komunikasi yang berada diberbagai lokasi yang terdiri dari lebih satu komputer yang saling berhubungan. Dalam sebuah jaringan komputer biasanya terhubung sejumlah komputer ke sebuah/beberapa *server*. *Server* adalah komputer yang difungsikan sebagai “pelayanan” pengiriman data dan penerimaan data diantara komputer-komputer yang tersambung. Tujuan

dibangunnya suatu jaringan komputer adalah membawa informasi secara tepat tanpa adanya kesalahan dari sisi pengirim *(transmisi)* menuju ke sisi penerima *(receiver)* melalui media komunikasi.

2.6.1 Jenis-jenis Jaringan Komputer

Jenis-jenis jaringan ada dua, yaitu:

1. *Model peer to peer*

Menurut model ini, setiap *host* dapat menawarkan layanan ke *peer* lain dan juga mengambil layanan dari *peer* lain. Model ini cocok untuk jaringan kecil.

2. *Model Client Server*

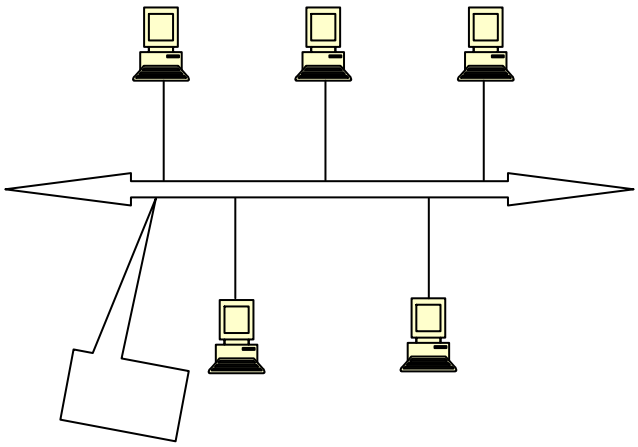
Model ini memisahkan secara jelas mana yang dapat memberikan layanan jaringan *(server)* dan mana yang hanya menerima layanan *(client).* Beberapa komputer di *set up* sebagai *server* yang memberikan sumberdaya *(resource)* dari jaringan : printer, modem dan saluran lainnya kepada komputer lain yang dikoneksi kejaringan yang berfungsi sebagai *client.* Dalam perancangan sistem infomasi Pengolahan data penduduk ini, jenis jaringan yang digunakan adalah *client server.*

2.6.2 Topologi Jaringan Komputer

Menurut Abdul Kadir (2003 :352) Yang dimaksud topologi jaringan menyatakan susunan komputer secara fisik dalam suatu jaringan. Ada tiga topologi jaringan komputer yaitu :

2.6.2.1 Topologi Bus

Pada topologi ini semua simpul (umumnya komputer) dihubungkan melalui kabel yang disebut bus. Kabel yang digunakan adalah kabel koaksial. Jika seorang pemakai mengirimkan pesan ke seorang pemakai lain maka pesan tersebut akan melalui bus. Setiap komputer perlu membaca alamat dalam pesan. Sekiranya alamat pesan cocok dengan alamat komputer pembaca, komputer tersebut segera mengambil pesan tersebut.

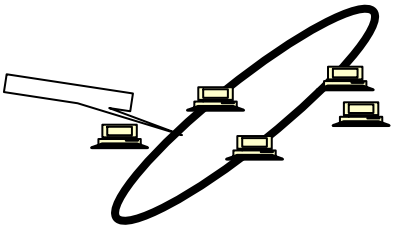


Gambar 2.5 Topologi Bus

Sumber: Abdul Kadir (2003 : 353)

2.6.2.2 Topologi Cincin (Ring)

Topologi cincin mirip dengan topologi bus. Informasi dikirim oleh sebuah komputer akan dilewatkan ke media transmisi, melewati satu komputer ke komputer berikutnya.

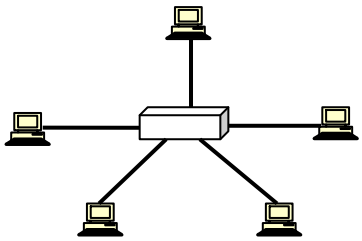


Gambar 2.6 Topologi Cincin

Sumber: Abdul Kadir (2003 : 354)

2.6.2.3 Topologi Bintang (Star)

Pada topologi ini terdapat komponen yang bertindak sebagai pusat pengontrol. Semua simpul yang hendak berkomunikasi selalu melalui pusat pengontrol tersebut. Dalam hal ini, pusat pengontrol berupa *hub* atau *switch.*



Gambar 2.7 Topologi Bintang

Sumber: Abdul Kadir (2003 : 355)

2.6.3 Manfaat Jaringan Komputer

Manfaat yang didapat dari membangun jaringan komputer adalah sebagai berikut:

1. Sharing Resources

Sharing resources bertujuan agar seluruh program, peralatan/periperal lainnya dapat dimanfaatkan oleh setiap orang yang ada pada jaringan komputer tanpa terpengaruh oleh lokasi maupun pengaruh dari pemakai. Dengan kata lain,

seorang pemakai yang letaknya jauh sekalipun dapat memanfaatkan data maupun informasi yang lainnya tanpa mengalami kesulitan.

2. Media komunikasi

Jaringan komputer memungkinkan terjadinya komunikasi antara pengguna, baik untuk *teleconference* maupun untuk mengirim pesan/informasi yang penting lainnya

3. Integrasi Data

Pembangunan jaringan komunikasi dapat mencegah ketergantungan pada komputer pusat. Setiap proses data tidak harus dilakukan satu komputer saja, melainkan dapat didistribusikan ketempat lainnya, oleh sebab inilah maka dapat terbentuk data yang terintegrasi sehingga dengan demikian memudahkan pemakai untuk memperoleh dan mengolah informasi setiap saat.

4. Pengembangan dan Pemeliharan

Dengan adanya jaringan komunikasi ini, maka perkembangan peralatan dapat dilakukan dengan mudah dan menghemat biaya. Jaringan komputer bisa memudahkan pemakai dalam merawat *harddisk* dan peralatan lainnya. Misalnya untuk memberikan perindungan serangan virus maka pemakai cukup memusatkan perhatian pada *harddisk* yang ada pada komputer pusat.

5. Keamanan Data

Sistem jaringan komputer memberikan perlindungan terhadap data jaminan keamanan, data tersebut diberikan melalui pengaturan hak akses para

pemakai dan *password*, serta teknik perlindungan terhadap *harddisk* sehingga data mendapatkan perlindungan yang efektif.

6. Sumber Daya Lebih Efisien dan Informasi Terkini

Dengan adanya pemakaian sumberdaya secara bersama-sama, maka pemakai bisa mendapatkan hasil dengan maksimal dan kualitas yang tinggi. Selain itu data atau informasi yang di akses selalu terbaru. Karena ada perubahan yang terjadi dapat secara langsung diketahui oleh setiap pemakai.

2.7 Sistem *Client Server*

Sistem *client server* mempunyai dua komponen utama yaitu komputer *client* dan komputer *server. Server* merupakan komputer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari komputer *client* dan bertindak sebagai *server database* yang menyimpan data. *Client* yaitu komputer atau *workstation* yang melakukan pengiriman permintaan-permintaan data pada server kemudian menampilkan data tersebut pada *interface* aplikasi yang dimiliki.Selain itu *client* juga mempunyai kemampuan untuk mengubah atau menghapus data. Sistem *client server* merupakan suatu sistem *client* komputer yang melibatkan proses-proses *client* yang meminta suatu pelayanan data kepada komputer *server* yang menyediakan layanan data tersebut. Sehingga *client* maupun *server* sama-sama melakukan pekerjaan. Dengan adanya kombinasi *client (front end*) dan *server (back end)* ini, maka kumpulan dari modul-modul program

tidak dieksekusi dalam memory yang sama namun terbagi dalam komputer *client server.*

Hal ini menjadikan konfigurasi bagi komputer *client* dan komputer *server* bisa berbeda seperti kapasitas memory, kecepatan *processor* atau alat masukan dan keluaran yang disesuaikan dengan fungsi kerja dari elemen-elemen tersebut. Bagi *server* yang menjalankan tugas pengelolaan suatu *database* digunakan suatu konfigurasi yang khusus menangani tugasnya tersebut dengan sistem operasi yang dikhususkan bagi *server* seperti *Windows NT Server, Windows 2000 Server,* sedangkan komputer *client* menggunakan konfigurasi yang umum bagi sebuah komputer *dekstop* yang terhubung ke jaringan dengan sistem operasi seperti *Windows 98, Windows ME, Windows XP* dan lain-lain.

2.7.1 Cara Kerja *Client Server*

Sistem *client server* berjalan seutuhnya pada dua sistem komputer yang berbeda. Biasanya sebuah *server* melayani beberapa komputer *client,* walaupun mungkin ada yang hanya melayani satu *client* saja. Apabila pemakai ingin mengakses informasi bagian aplikasi, *client* mengeluarkan permintaan yang dikirimkan melalui jaringan kepada *server* kemudian menjalankan permintaan dan mengirimkan informasi kembali kepada client. Proses *server* berperan sebagai aplikasi yang mengelola sumber daya nilai bersama *(shared resource)* seperti *database,* printer atau jalur komunikasi selain menjalankan tugasnya sebagai *back end*. Sistem *client server* yaitu pusat pemrosesan data, sedangkan proses *client* meliputi program-program untuk mengirimkan pesan permintaan pada *server*

serta melakukan pengaksesan pada data seperti mengedit, menghapus atau menambah data. Karena itu program *client* adalah aplikasi *front end* yang digunakan sebagai antar muka *(interface)* bagi pemakai untuk berinteraksi dengan *server* selain itu *client* menangani pemakaian sumberdaya lokal seperti monitor, keyboard dan perangkat lokal lainnya.

2.8 Perangkat Lunak Pendukung

Untuk perangkat lunak pendukung, penulis mengembangkannya melalui dua macam aplikasi yang berbeda. Karena komputer *server* dan *client* memerlukan aplikasi yang berbeda. *SQL Server 2000* digunakan untuk pengelolaan *database server* dan *Visual Basic 6.0*. Sebagai aplikasi pemograman untuk mengembangkan aplikasi sistem *client server* pada komputer *client*. Kedua perangkat lunak ini dinilai merupakan perangkat lunak yang lebih baik dibanding yang lainnya sebagai alat yang mengembangkan aplikasi bagi sistem *client server* ini.

2.8.1 Sekilas Tentang *Visual Basic 6.0*

Pemrograman *visual basic 6.0* adalah pemrograman yang bekerja dalam lingkup *MS.Windows*. *Microsoft Visual Basic 6.0* dapat memanfaatkan kemampuan *MS.Windows* secara optimal. Visual Basic menunjukan cara yang digunakan untuk membuat *graphical user interface* dan lingkungan pemrograman *Microsoft Visual Basic* mengandung beberapa *tool* yang bermanfaat untuk menjalankan dan mengelola program yang di buat. *Microsoft Visual Basic* juga

mengandung segala sesuatu yang dibutuhkan untuk membangun aplikasi untuk *Windows* dari awal.

Tool yang terdapat dalam *Microsoft visual basic*.

1. *Form* adalah jendela yang bisa di ubah-ubah untuk membuat antar muka program. Pada program *set up*, *Form* bisa mengandung menu, tombol, kotak daftar, baris penggulung dan item-item lan yang bisa di lihat pada program berbasis Windows lainnya.
2. *Toolbox* untuk menambahkan elemen antarmuka program ke dalam *form* digunakan *Tool* atau kontrol yang terdapat pada *Toolbox*. *Toolbox* juga mengandung objek yang bisa melakukan operasi " di balik layer" pada program *Visual Basic*. Objek ini melakukan pekerjaan yang bermanfaat tetapi tidak nampak oleh pemakai apabila program dijalankan. Objek ini diantaranya adalah objek untuk memanipulasi informasi pada *database*.
3. Jendela *Properties* digunakan untuk mengubah karakteristik atau *setting property* dari elemen yang terdapat pada *form*. *Setting property* adalah kualitas objek-objek yang terdapat pada antar muka.

2.8.2 Sekilas Tentang *Microsoft SQL Server 2000*

Menurut Ketut Darmayuda (2007: 7) SQL *server* merupakan salah satu *database relational* yang banyak digunakan oleh dunia usaha. SQL *server* mendukung penggunaan SQL *(Strukture Query Language).* SQL *Server* merupakan salah satu contoh produk RDMBS*(Relational DataBase Management System)* yang banyak digunakan di perusahaan – perusahaan.

Database sistem pemakai file secara bersama-sama bergantung pada sebuah proses *client* tunggal per *user* untuk memanipulasi data pada file yang dipakai pada *server* jaringan. *SQL Server 2000* memberikan bahasa dan antarmuka yang baik untuk pemograman dan komunikasi pada *server.*

Transact-SQL adalah bahasa pemograman *server* yang merupakan superset dari *ANSI-SQL. ANSI-SQL* mendefinisikan empat perintah dasar untuk memanipulasi data yaitu **SELECT, INSERT, UPDATE, DELETE** dan sejumlah perintah untuk mendefinisikan struktur *database.*

SQL Server 2000 memakai konsep transaksi dan *logging* transaksi untuk menjamin konsistensi dan kemampuan *recover* informasi yang disimpan didalam *database SQL Server* mendukung *backup* secara *online.* File-file data yang dipakai pada aplikasi *database* dibuka secara ekslusif oleh proses *server*. Oleh sebab itu melakukan *backup* pada file secara individu tidak memungkinkan, kecuali *server* diberhentikan dulu. *Backup* database dapat dilakukan meskipun ada transaksi yang aktif.

SQL Server merupakan server database yang "cerdik". *SQL Server* mempunyai fungsi utama untuk memproses *query* dan transaksi *database* yang diberikan oleh aplikasi client dengan performansi yang baik. *SQL Server* akan menentukan cara yang paling baik dan efisien untuk memproses perintah-perintah tersebut dengan sebuah teknik yang dinamakan *Cost Based Query Optimization.*

2.8.3 Crystal Report

Crystal reports merupakam program khusus untuk membuat laporan yang terpisah dari program *Microsoft Visual Basic* tetapi keduanya dapat dihubungkan (*linkage*). Hasil cetak dengan menggunakan *Crystal Report* lebih baik dan lebih mudah, karena pada *Crystal Report* banyak tersedia objek maupun komponen yang mudah digunakan.

2.9 Penggajian Pegawai

Adapun penggajian pegawai yang ada di CV. Citra Retrofita Bandung antara lain yaitu:

2.9.1 Penggajian

Banyak definisi gaji dikemukakan oleh para ahli,namun pada dasarnya gaji bukan merupakan kata atau istilah yang baru untuk mempermudah kita dalammelakukan sebuah pekerjaan. Berikut ini penulis kemukakan definisi dari system menurut beberapa ahli asalah sebagai berikut:

Menurut Soemarso S.R. (2005: 307) Untuk menjalankan kegiatan, suatu perusahaan harus memperkerjakan orang-orang yang disebut karyawan. Setiap karyawan memberikan tenaga dan pikirannya kepada perusahaan akan menuntut timbal balik yang berupa gaji.

Gaji adalah "imbalan kepada pegawai yang diberi tugas-tugas administratif dari pimpinan yang jumlahnya, biasanya tetap secara bulanan atau tahunan.

Disamping gaji, pegawai mungkin memperoleh manfaat - manfaat yang diberikan dalam bentuk tunjangan, misalnya tunjangan jabatan, tunjangan

perumahan, tunjangan pengobatan, tunjangan hari raya, uang transport, uang makan dan lain-lain.

Menurut Mulyadi (2001: 373) *"Gaji umumnya merupakan pembayaran atau penyerahan jasa yang dilakukan oleh karyawan yang mempunyai jenjang jabatan manajer, umumnya gaji dibayarkan secara tetap per bulan.*

Sedangkan upah merupakan pembayaran atas penyerahan jasa yang dilakukan oleh karyawan pelaksanaan/buruh".

Dari pengertian di atas dapat disimpulkan bahwa gaji adalah imbalan yang diberikan kepada pegawai yang mempunyai jabatan lebih tinggi daripada pegawai yang menerima upah. Gaji merupakan balas jasa yang diserahkan kepada pegawai dan biasanya diberikan secara bulanan yang tidak bergantung dari jumlah jam atau hari kerja serta jumlah periode yang dihasilkan.

2.9.2 Pegawai

pegawai menurut kamus besar Bahasa Indonesia adalah : *"Orang atau seseorang yang bekerja pada suatu lembaga (perusahaan/ kantor) dan mendapatkan gaji sesuai dengan tingkatan pekerjaannya".*

2.9.3 Absensi

Pengertian absensi adalah pencatatan kehadiran sering dikenal dengan istilah absensi. Ketidakhadiran (*absenteism*) adalah kegagalan untuk melapor pada waktu kerja (Panggabean : 2002). Dengan kata lain ketidakhadiran merupakan kegagalan seorang karyawan untuk hadir di tempat kerja pada hari kerja.

Ketidakhadiran berbeda dengan terlambat (*lateness*) yang menunjukkan kegagalan untuk datang tepat waktu. Cara menghitung ketidakhadiran dengan membagi *time loss* yaitu jumlah hari-hari yang hilang dengan frekuensi, yaitu jumlah kehadiran selama satu periode. Menurut Julius (1991 : 490): Absensi merupakan kegagalan, pekerja pegawai untuk melaporkan pekerjaan ketika mereka dijadwalkan bekerja.

2.9.3.1 Jenis-Jenis Absensi

Kita mengenal beberapa jenis absensi. Yang membedakan jenis-jenis absensi tersebut adalah cara penggunaannya, dan tingkat daya gunanya Secara umum jenis-jenis absensi dapat di kelompokkan menjadi dua, yaitu:

2.9.3.1.1 Absensi manual

Absensi manual adalah cara pengentrian kehadiran dengan cara menggunakan pena (tanda tangan)

2.9.3.1.2 Absensi non manual (dengan menggunakan alat)

Absensi non manual adalah suatu cara pengentrian kehadiran dengan menggunakan system terkomputerisasi, bisa menggunakan kartu dengan barcode, finger print ataupun dengan mengentrikan nip dan sebagainya.

2.9.4 Tunjangan

Tunjangan merupakan kompensasi tambahan yang bertujuan untuk mengikat karyawan agar tetap bekerja pada perusahaan Handoko (1994:42).Tujuan utama dari tunjangan adalah untuk membuat karyawan mengabdikan hidupnya pada organisasi dalam jangka panjang Flippo (1990:23).

Dengan pemberian tunjangan kepada pegawai yang diterapkan dengan tepat dalam suatu perusahan dapat memberikan manfaat yang besar bagi perusahaan.

2.9.5 Prinsip-prinsip Pemberian Gaji Kepada Pegawai

Agar pegawai yang menerima gaji atau upah merasa puas, maka perlu diperhatikan perinsip-prinsip pemberian gaji sebagai berikut :

1. Gaji yang diberikan harus cukup untuk hidup pegawai dan keluarganya. Dengan kata lain besarnya gaji harus memenuhi kebutuhan pokok minimum.
2. Pemberian gaji harus adil, artinya besar kecilnya gaji tergantung kepada berat ringanya kewajiban dan tanggung jawab yang dibebankan kepada pegawai yang bersangkutan. Pegawai yang pekerjaannya sulit, tanggung jawabnya berat, harus diberi gaji yang lebih banyak dari pegawai lain yang kewajiban dan tanggung jawabnya lebih ringan. Salah satu cara untuk menyusun sekala gaji yang adil adalah dengan membuat skala klasifikasi golongan jabatan.
3. Gaji harus diberikan tepat pada waktunya. Gaji yang terlambat diberikan dapat mengakibatkan kekecewan dan rasa tidak puas pegawai, yang pada gilirannya akan dapat merugikan produktivitas pegawai.
4. Besar kecilnya gaji dan upah harus mengikuti perkembangan harga pasar. Hal ini perlu diperhatikan, karena yang penting bagi para pegawai bukan banyaknya uang yang diterima,tetapi berapa banyak barang atau jasa yang diperoleh dengan gaji tersebut.
5. Sistem pembayaran gaji harus mudah dipahami dan dilaksanakan, sehingga pembayaran dapat dilakukan dalam waktu yang relatip singkat.

6. Perbedaan dalam tingkat gaji harus didasarkan evaluasi jabatan yang objektif.
7. Stuktur gaji harus ditinjau kembali dan mungkin harus diperbaiki apabila kondisi berubah.

2.9.6 Aturan Penggajian Pegawai

1. Absensi

   Absensi harian dilakukan tiap jam masuk dimulai pukul 08:00 WIB. Toleransi keterlambatan sampai pukul 09:00 dan untuk absensi harian proyek dilakukan pada saat sebelum keberangkatan ketempat proyek.

2. Penggajian

   Mekanisme perhitungan gaji berdasarkan absensi, tunjangan transportasi, tunjangan makan dikurangi potongan (mangkir atau tidak masuk kerja).

   1) Gaji pokok diisi sesuai dengan jabatan, dimana gaji pokok Direktur sebesar Rp.4000000, Manajer Rp.3000000 dan pegawai kontark (pekerja) 1 hari Rp.50000.
   2) Tunjangan Makan disesuaikan dengan jabatan ada pada data jabatan.
   3) Tunjangan Transportasi disesuaikan dengan jabatan ada pada data jabatan.
   4) Potongan karena mangkir atau tidak masuk masing-masing sebesar Rp.50000 per hari untuk pegawai tetap dan pegawai kontrak potongan nya sama dengan gaji proyek perhari apabila tidak masuk yaitu Rp.50000.
   5) Gaji kotor untuk pegawai tetap diperoleh dari Gaji pokok + Tunjangan transportasi + Tunjangan Makan – potongan (50000 * total_hari (jumlah bulan) – jumlahharikerja (jumlah masuk kerja) * 50000).

Gaji kotor untuk pegawai kontrak diperoleh dari tunjangan trasportasi + tunjangan makan.

6) Gaji bersih untuk pegawai tetap atau kontrak diperoleh dari gaji kotor + gaji proyek (jumlahharikerja (jumlah masuk kerja proyek) * gaji proyek).