BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Yang dimaksud dengan sistem adalah “Jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.”, Andri Kristanto(2008:1).

Sistem juga merupakan kumpulan elemen-elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditunjukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (output) yang diinginkan.

2.1.1. Elemen Sistem

Elemen-elemen yang terdapat dalam sistem meliputi: tujuan sistem, batasan sitem, kontrol, *input*, proses, *output* dan umpan balik, Andri Kristanto (2008:2). Hubungan antara elemen-elemen dalam sistem dapat dilihat sebagai berikut:

Dari gambar 2.1, tujuan, batasan dan kontrol sistem akan berpengaruh pada *input*, proses dan *output*. *Input* yang masuk ke dalam sistem akan diproses dan diolah sehingga menghasilkan *ouput*. *Ouput* tersebut akan dianalisa dan akan menjadi umpan balik bagi penerima dan dari umpan balik tersebut akan muncul segala pertimbangan untuk input selanjutnya. Berikut penjelasan elemen-elemen yang membentuk sebuah sistem :

1. Tujuan Sistem

   Setiap sistem memiliki tujuan (*Goal*), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali dan tujuan antara satu sistem dengan sistem yang lain berbeda.

2. Batasan Sistem

   Batasan sistem merupakan sesuatu yang membatasi sistem dalam mencapai tujuan sistem dan pemisah antara sistem dengan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem.

3. Kontrol Sistem

   Kontrol atau pengawasan sistem merupakan pengawasan terhadap pelaksanaan pencapaian tujuan dari sistem tersebut. Kontrol tersebut dapat berupa kontrol terhadap pemasukan data (*input*), kontrol terhadap keluaran data (*ouput*), kontrol terhadap pengolahan data, kontrol terhadap umpan balik dan sebagainya.

4. Masukan (*Input*)

   Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa jenis data, frekuensi pemasukan data dan sebagainya.

5. Proses

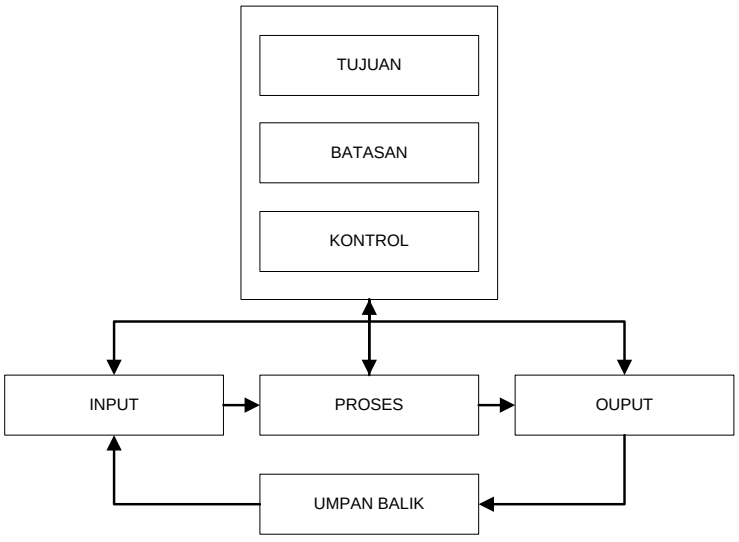
Proses merupakan elemen dari sistem yang bertugas untuk mengolah atau memroses seluruh masukan data menjadi suatu informasi yan lebih berguna.

6. Keluaran (*Output*)

Keluaran (*output*) merupakan hasil dari *input* yang telah diproses oleh bagian pengolah dan merupakan tujuan akhir sistem.

7. Umpan Balik

Umpan balik merupakan elemen dalam sistem yang bertugas mengevaluasi bagian dari *output* yang dikeluarkan, dimana elemen ini sangat penting demi kemajuan sebuah sistem. Umpan balik ini dapat merupakan perbaikan sistem, pemeliharaan dan sebagainya.



Gambar 2.1 Elemen-elemen Sistem

(Sumber :Perancangan Sistem Informasi dan Aplikasinya,

Andr Kristanto,2008:2)

2.1.2. Karakteristik Sistem

Menurut Andri Kristanto (2008:3) suatu sistem mempunyai karakteristik atau sifat tertentu yaitu :

a. Komponen(*Components*)

   Terdiri dari sejumlah komponen yang saling berinteraksi, dan bekerjasama membentuk satu kesatuan.

b. Batas Sistem(*Boundary*)

   Merupakan daerah yang membatasi antara sistem dengan system lainnya atau dengan lingkungan luarnya.

c. Lingkungan Luar Sistem(*Environments*)

   Adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

d. Penghubung(*Interface*)

   Merupakan media penghubung antara subsistem, yang memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

e. Masukkan(*Input*)

   Adalah energi yang dimasukkan kedalam sistem, yang dapat berupa masukkan perawatan(*Maintenance input*) dan masukkan signal(*signal input*).

f. Keluaran(*Output*)

   Adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

g. Pengolah(*process*)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukkan menjadi keluaran.

h. Sasaran dan Tujuan(*Objective*)

Suatu sistem dapat dikatakan berhasil menjalankan fungsinya jika berhasil mencapaisasaran dan tujuan sistem tersebut.

2.1.3. Klasifikasi Sistem

Dari berbagai sudut pandang,menurut Andri Kristanto (2008:5) sistem dapat diklasifikasikan menjadi beberapa bagian yaitu:

a. Sistem Abstrak dan Sistem Fisik

Sistem abstrak merupakan sistem yang tidak bisa dilihat secara mata biasa dan biasanya sistem ini merupakan ide-ide atau pemikiran. Sedangkan sistem fisik merupakan sistem yang bisa dilihat secara mata biasa dan biasanya sering digunakan oleh menusia.

b. Sistem Alamiah dan Sistem Buatan

Sistem alamiah merupakan sistem yang terjadi karena pengaruh alam. Sedangkan sistem buatan merupakan sistem yang dirancang dan dibuat oleh manusia.

c. Sistem Tertutup dan Sistem Terbuka

Sistem merupakan sistem yang tidak berhubungan dengan bagian luar sistem dan biasanya tidak berpengaruh oleh kondisi di luar sistem. Sedangkan sistem terbuka merupakan sistem yang berhubungan dengan bagian luar sistem.

2.1.4. Analisis Sistem

Menurut Andri Kristanto (2008:5) untuk mencapai tujuan dari suatu sistem yang dibuat, dibutuhkan 3 perangkat atau alat bantu yang dapat meningkatkan kinerja dari sebuah sistem sehinggatujuan dari sistem tersebut dapat tercapai.

Tiga perangkat tersebut meliputi: perangkat keras (*hardware*), perangkat lunak (*software*), dan perangkat manusia (*brainware*). Perangkat keras dapat berupa komputer, sedangkan perangkat lunak adalah program. Perangkat manusia berupa user.

Ada beberapa tentang pengertian analisis sistem, yaitu:

a. Seseorang yang mempunyai kemampuan untuk menganalisa sebuah sistem.
b. Seseorang yang mempunyai pengetahuan tentang aplikasi komputer yang digunakan untuk memecahkan masalah-masalah.
c. Seseorang yang mempunyai pengetahuan untuk merencanakan dan menerapkan rancangan sistemnya sesuai dengan permasalahan yang terjadi.

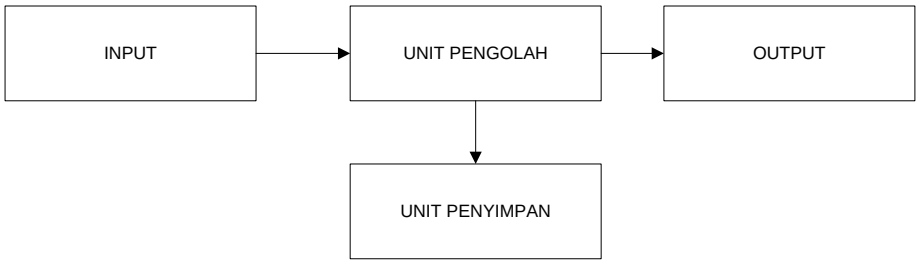
2.2.Konsep Dasar Informasi

Di dalam pengolahan sistem pada akhirnya menghasilkan suatu informasi, untuk itu pendefenisian informasi diperlukan untuk menunjang berhasilnya pengembangan sistem yang akan dirancang.

Menurut Edhy Sutanta(2004:4) “Informasi merupakan hasil pengolahan data sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai

kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat ini juga atau secara tidak langsung pada saat mendatang."

Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar, berikut transformasi data menjadi informasi :



Gambar 2.2 Transformasi data menjadi informasi

(Sumber: *Sistem Basis Data,*Edhy Sutanta (2004:5))

Dalam gambar tersebut input adalah data yang akan diolah oleh unit pengolah. Oleh karena itu, suatu data belum dapat berbicara banyak sebelum diolah lebih lanjut. dan output adalah informasi sebagai hasil pengolahan data yang telah diinputkan dan unit penyimpan diperlukan sebagai alat simpanan data, pengolah, maupun informasi.

Menurut Edhy Sutanta(2004:4), "Data dapat didefinisikan sebagai bahan keterangan tentang kejadian-kejadian nyata atau fakta-fakta yang dirumuskan dalam sekelompok lambang tertentu yang menunjukan jumlah, tindakan, atau hal."

Dengan kata lain sumber informasi adalah data. Data menggabarkan suatu kejadian yang sedang terjadi, dimana data tersebut akan diolah dan diterapkan dalam sistem menjadi input yang berguna dalam suatu sistem.

2.2.1. Kualitas Informasi

Kulaitas informasi tergantung dari 3 hal yang sangat dominan yaitu keakuratan informasi, ketepatan waktu dari informasi dan relevan. Ketiga hal tersebut dapat dijelaskan sebagai berikut :

1. Akurat

   Informasi yang tidak mengandung keraguan-keraguan, sama maksudnya yang disampaikan dengan yang menerima, bebas dari kesalahan-kesalahan dan tidak menyesatkan, harus menjelaskan dan mencerminkan maksudnya atau dengan kata lain tidak menimbulkan pertanyaan bagi penerima informasi tersebut.

2. Tepat Waktu

   Informasi yang diterima harus tepat pada waktunya yang dihasilkan tidak boleh terlambat karena informasi yang terlambat tidak mempunyai nilai yang baik, sehingga kalau digunakan sebagai dasar dalam pengambilan keputusan akan berakibat fatal atau kesalahan dalam tindakan.

3. Relevan

   Informasi harus mempunyai manfaat bagi penerima, karena informasi ini digunakan untuk pengambilan suatu keputusan dalam pemecahan suatu permasalahan.

4. Ekonomis, efesien dan dapat dipercaya

   Informasi yang dihasilkan mempunyai manfaat yang lebih besar dibandingkan dengan biaya mendapatkannya dan sebagian besar informasi tidak dapat ditaksir keuntungannya dengan satuan nilai. Selain itu informasi yang dihasilkan juga bisa dipercaya kebenarannya dan tidak mengada-ada.

2.3. Konsep Sistem Informasi

Sistem informasi merupakan aplikasi komputer untuk mendukung operasi dari suatu organisasi yaitu operasi, instalasi, dan perawatan komputer, perangkat lunak, dan data. Sistem Informasi Manajemen adalah kunci dari bidang yang menekankan finansial dan personal manajemen.

Menurut James A. O'Brien (2005:5) "Sistem informasi yaitu dapat merupakan kombinasi teratur apa pun dari orang-orang, *hardware*, *software*, jaringan komunikasi dan sumber daya data yang mengumpulkan, mengubah, menyebarkan suatu informasi dalam sebuah organisasi."

Sistem informasi dapat didefinisikan pula sebagai suatu kesatuan dari berbagai informasi yang saling berkaitan dan berinteraksi satu sama lainnya untuk keperluan dan tujuan tertentu.

2.3.1. Komponen Sistem Informasi

Sistem informasi merupakan sebuah susunan yang terdiri dari beberapa komponen seperti orang, aktivitas, data, perangkat keras, perangkat lunak, dan jaringan yang terintegrasi yang berfungsi untuk mendukung dan meningkatkan operasi sehari-hari sebuah bisnis, juga menyediakan kebutuhan informasi untuk pemecahan masalah dan pengambilan keputusan oleh manajer.

1. Orang

Semua pihak yang bertanggung jawab dalam hal penyokong atau sponsor sistem informasi (*system owner*), pengguna sistem (*system users*), perancang sistem (*system designer*) dan pengembang sistem informasi (*sistem development*).

2. Aktivitas

Sekumpulan aturan atau tahapan-tahapan untuk membuat, memakai, memproses dan mengolah sistem informasi ataupun hasil keluaran dari sistem informasi tersebut.

3. *Data*

Secara konseptual, data adalah deskripsi tentang benda, kejadian, aktivitas, dan transaksi yang tidak mempunyai makna dan tidak berpengaruh langsung secara langsung kepada pemakainya atau disebut juga sebagai sekumpulan fakta mentah dalam isolasi.

4. Perangkat Keras (*hardware*)

Perangkat keras yang meliputi piranti-piranti yang digunakan oleh sistem komputer untuk masukan dan keluaran yang terdiri dari komputer, printer, jaringan.

5. Perangkat Lunak (*sotfware*)

Sekumpulan instruksi-instruksi atau perintah-perintah yang memungkinkan perangkat keras bisa digunakan untuk memproses data, atau sering disebut sebagai program.

6. Jaringan (*network)*

Sistem penghubung yang memungkinkan suatu sumber dipakai secara bersama-sama, baik pada waktu dan tempat bersamaan ataupun berbeda.

2.4. Perancangan Sistem

Perancangan sistem adalah desain sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan, tahap ini menyangkut

mengkonfigurasikan dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir analisis sistem.(Sumber: http://id.shvoong.com/writing-and-speaking/presenting/2129410-definisi-perancangan-sistem/#ixzz1Q5uPRmlD)

Tahapan perancangan sistem mempunyai 2 maksud yaitu sebagai berikut :

a. Untuk memenuhi kebutuhan kepada pemakai sistem.
b. Untuk memberikan gambaran yang jelas dan rancangan bangunan yang lengkap kepada pemogram komputer dan ahli-ahli teknik lainnya yang terlibat.

2.4.1. Perancangan Proses

Perancangan proses merupakan tahapan yang sangat menentukan terciptanya sistem informasi yang baik untuk mendukung proses pembentukan database tersebut ada beberapa peralatan, yaitu :

2.4.1.1. Flowmap

Flowmap merupakan diagram alir data yang penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponen dengan maksud untuk mengidentifikasi serta dapat mengevakuasi suatu permasalahan yang diharapkan dapat diusulkan perbaikan-perbaikan.

2.4.1.2.Diagram Konteks

Diagram *konteks* menurut Andri Kristanto (2008:70) adalah, " Sebuah diagram yang menggambarkan hubungan antara entiti luar, masukan dan keluaran dari sistem". Diagram *konteks* menyoroti jumlah karakteristik sistem yaitu :

1. Kelompok pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi(sebagai *terminator*).
2. Data masuk, yaitu data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
3. Data keluar, yaitu data yang dihasilkan sistem dan diberikan ke dunia luar.
4. Penyimpanan data, yaitu digunakan secara bersamaan antara sisten dengan terminator. Data ini dibuat oleh sistem dan digunakan oleh atau sebalikanya dibuat oleh lingkungan dan digunakan oleh sistem. Hal ini berarti pembuatan simbol penyimpanan dalam diagram *Konteks* dibenarkan dengan syarat simbol tersebut merupakan bagian dari luar sistem.
5. Batasan antara sistem dan lingkungan. Simbol yang digunakan dalam diagram *konteks* antara lain :
    a. Persegi panjang, Untuk berkomunikasi langsung dengan sistem melalui aliran data.
    b. Lingkaran, Untuk menunjukan adanya kegiatan proses dalam sistem.

2.4.1.3.Data Flow Diagram (DFD)

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang dikembangkan secara logika tanpa memepertimbangkan lngkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan. DFD merupakan alat yang digunakan pada metdologi pengembangan sistem yang terstruktur.

Menurut Andri Kristanto (2008:61) DFD adalah, " Suatu model logika data atau proses yang dibuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi anatara data yang tersimpan."

DFD merupakan peralatan yang berfungsi untuk menggambarkan secara rinci mengenai sistem sebagai jaringna kerja antara fungsi yang berhubungan satu sama lain dengan menunjukan dari dan kemana data mengalir serta penyimpananya.

2.4.1.4.Kamus Data

Kamus data menurut Andri Kristanto (2008:72) yaitu, "Kumpulan elemen-elemen atau simbol-simbol yang digunakan untuk membantu dalam penggambaran atau pengidentifikasian setiap field atau file di dalam sistem."

Bisa dikatakan bahwa kamus data merupakan tempat penyimpanan semua struktur dan elemen datan yang ada pada sistem. Juga sebagai katalog untuk mengetahui detail data seperti sumber dan tujuan data, deskripsi, bentuk, dan

struktur dari data. Kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir pada sistem.

2.4.2. Perancangan Basis data

Menurut Edhy Sutanta (2004:204) “Perancangan basis data merupakan bagian dari kegiatan besar dalam rangka pengembangan Sistem Informasi Manajemen, khususnya pada tahap perancangan.”

Perancangan pada basis data (*database*) adalah perancangan yang digunakan pada pembuatan sistem informasi perangkat lunak (*software*) ini. Basis data sendiri dapat didefinisikan sebagai berikut:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudancy*) yang tidak perlu untuk memenuhi berbagai kebutuhan.
3. Kumpulan *file* atau table atau pun arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

   Perancangan basis data terdiri dari ERD (Entity Relationship Data), normalisasi, table relasi atau relasi *File*, struktur *File.*

2.4.2.1.Normalisasi

Menurut Edhy Sutanta (2004:172) “Normalisasi diartikan sebagai suatu teknik yang menstrukturkan data dalam cara-cara tertentu mencegah timbulnya permasalahan pengolahan data dalam basis data.”

Normalisasi merupakan peralatan yang digunakan untuk melakukan proses pengelompokan data menjadi tabel-tabel yang menunjukan entitas dan relasinya. Secara umum proses normalisasi dibagi dalam tiga tahap, yaitu bentuk tidak normal, bentuk Normal Pertama, Bentuk Normal Kedua, Bentuk Normal Ketiga.

Pada tahap ketiga biasanya sudah akan diperoleh tabel yang optimal. Berikut penjelasan tahap-tahap pemebentukan Normalisasi :

a. Bentuk tidak normal

   Bentuk tidak normal adalah suatu bentuk dimana semua data dikumpulkan apa adanya tanpa mengikuti aturan – aturan tertentu.

b. Bentuk normal pertama

   Bentuk normal pertama adalah suatu bentuk dimana setiap field dalam tabel memiliki nilai data.

c. Bentuk normal kedua

   Bentuk normal kedua harus memenuhi syarat :

   a) Sudah memenuhi kriteria normal pertama
   b) Setiap field bukan kunci tergantung secara fungsional pada kunci primer.

d. Bentuk normal ketiga

Bentuk normal ketiga adalah bentuk yang memenuhi syarat – syarat berikut :

a) Tabel sudah dalam bentuk normal kedua.
b) Field bukan kunci tergantung secara fungsional pada kunci primer.

2.4.2.2.ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram (ERD) menurut Edhy Sutanta (2004:79), "ERD merupan model yang membantu perancangan basis data karena model ini dapat menunjukan macam data yang dibutuhkan dan kerelasian antar data didalamnya."

Jadi *Entity Relationship Diagram* digunakan untuk pemodelan struktur hubungan antar data. Komponen-komponen yang digunakan yaitu:

1. Entitas (*Entity*), objek yang diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks yang dirancang.
2. Atribut, elemen dari entitas yang berfungsi mendeskripsikan karakteristik dari entitas.
3. Relasi antar entitas, lebih dari suatu entitas yang saling berelasi.

Kerelasian antar entitas dapat dikelompokan dalam tiga jenis, yaitu :

1. *One to one relationship* (satu ke satu)

   Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding satu. Hubungan tersebut dapat dibandingkan dengan tanda lingkaran untuk menunjukan tabel dan relasi antara keduanya diwakilkan

dengan tanda panah tunggal. Relasi satu ke satu terjadi bila satu *record* yang ada pada satu entity tabel hanya punya satu relasi pada *file* lain.

2. *One to Many* (satu ke banyak)

   Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik banyak lawan satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukan tabel dan relasi atara keduanya diwakilkan tanda panah ganda untuk menunjukan hubungan banyak tersebut. Relasi satu ke banyak terjadi bila *record* dengan kunci tertentu pada satu *file* mempunyai relasi banyak *record* pada *file* lain.

3. *Many to Many* (banyak ke banyak)

   Hubungan anatara *file* pertama dengan *file* kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukan tabel dan relasi anatara keduanya diwakilkan dengan tanda panah ganda untuk menunjukan hubungan banyak tersebut. Relasi banyak ke banyak terjadi bila kedua *file* saling mempunyai relasi banyak *record* pada *file* yang lain.

2.4.2.3.Relasi Tabel

Menurut Edhy Sutanta(2004:155) “ Relasi adalah menyatakan sebuah tabel dalam basis data, sedangkan kerelasian menyatakan hubungan antar relasi dalam basis data.”

Dalam satu database dalam satu entity atau tabel mempunyai sebuah *field* yang memiliki nilai unik setiap baris. Baris-baris yang berhubungan tabel mengulangi kunci primer dari baris yang dihubungkannya pada tabel lain. Salinan dari kunci primer dalam tabel lain disebut kunci asing (*foreign key*). Kunci asing tersebut tidak perlu bersifat unik, dan semua *field* bisa menjadi kunci asing jika sesuai dengan kunci primer pada tabel lain.

2.4.2.4.Struktur File

Struktur *file* merupakan penyusunan *file-file* berdasarkan kelas datanya agar dapat memudahkan dalam penyimpanan data. Dalam pembuatan program dibutuhkan struktur *file* yang dimaksudkan untuk dapat melakukan kegiatan pengaturan pencarian data dan pembuatan laporan yang dapat memudahkan kerja sistem komputer.

2.5 Perangkat lunak pendukung

Perangkat lunak pendukung adalah *software* yang diperlukan untuk mendukung sistem yang akan dibangun.

2.5.1. Sekilas tentang bahasa pemograman java

Bahasa pemrograman Java pertama lahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh Joe Palrang.

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program java tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram Java banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas Java, yang disebut dengan *Java Application Programming Interface* (API).

Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket *(package)*. Java API telah menyediakan fungsionalitas yang memadai untuk menciptakan *applet* dan aplikasi canggih. Jadi ada dua hal yang harus dipelajari dalam Java, yaitu mempelajari bahasa Java dan bagaimana mempergunakan kelas pada Java API. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada Java program javac untuk mengkompilasi file kode sumber Java menjadi kelas-kelas *bytecode*. File kode sumber mempunyai ekstensi *.java. Kompilator javac menghasilkan file *bytecode* kelas dengan ekstensi *.class. Interpreter merupakan modul utama sistem Java yang digunakan aplikasi Java dan menjalankan program *bytecode* Java.

Beberapa keunggulan java yaitu java merupakan bahasa yang sederhana. Java dirancang agar mudah dipelajari dan digunakan secara efektif. Java tidak menyediakan fitur-fitur rumit bahasa pemrograman tingkat tinggi, serta banyak pekerjaan pemrograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan Java secara otomatis seperti dealokasi memori. Bagi pemrogram yang sudah mengenal bahasa C++ akan cepat belajar susunan bahasa

Java namun harus waspada karena mungkin Java mengambil arah (semantiks) yang berbeda dibanding C++.

Java merupakan bahasa berorientasi objek *(OOP)* yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi ringkas OOP adalah mengorganisasikan program sebagai kumpulan komponen, disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu. Kelas bertindak sebagai modul sekaligus tipe. Sebagai tipe maka pada saat jalan, program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas dapat mewarisi kelas lain.

Seluruh objek diprogram harus dideklarasikan lebih dulu sebelum digunakan. Ini merupakan keunggulan Java yaitu *Statically Typed*. Pemaksaan ini memungkinkan kompilator Java menentukan dan melaporkan terjadinya pertentangan (ketidakkompatibelan) tipe yang merupakan barikade awal untuk mencegah kesalahan yang tidak perlu (seperti mengurangkan variabel bertipe integer dengan variabel bertipe string). Pencegahan sedini mungkin diharapkan menghasilkan program yang bersih. Kebaikan lain fitur ini adalah kode program lebih dapat dioptimasi untuk menghasilkan program berkinerja tinggi.

Java menggunakan model pengamanan tiga lapis *(three-layer security model)* untuk melindungi sistem dari *untrusted Java code*. Pertama, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi

aturan-aturan dasar bahasa Java. Kedua, *class loader* menangani pemuatan kelas Java ke *runtime interpreter*. Ketiga, manajer keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem file, port jaringan, proses eksternal dan sistem *window.*

Platform independence adalah kemampuan program bekerja di sistem operasi yang berbeda. Bahasa Java merupakan bahasa yang secara sempurna tidak bergantung platform.

Tipe variabel Java mempunyai ukuran sama di semua platform sehingga variabel bertipe integer berukuran sama tidak peduli dimana program java dikompilasi. Begitu telah tercipta file .class dengan menggunakan kompilator Java di platform manapun, maka file .class tersebut dapat dijalankan di platform manapun. Jadi "dimanapun dibuat, dimanapun dapat dijalankan". Slogan ini biasa diringkas sebagai *Write Once, Run Anywhere (WORA).*

Java termasuk bahasa *Multithreading*. *Thread* adalah untuk menyatakan program komputer melakukan lebih dari satu tugas di satu waktu yang sama. Java menyediakan kelas untuk menulis program *multithreaded*, program mempunyai lebih dari satu *thread* eksekusi pada saat yang sama sehingga memungkinkan program menangani beberapa tugas secara konkuren.

Program Java melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh pemrogram dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat di bahasa yang memungkinkan alokasi dinamis.

Java mempunyai mekanisme *exception-handling* yang ampuh. *Exception-handling* menyediakan cara untuk memisahkan antara bagian penanganan kesalahan dengan bagian kode normal sehingga menuntun ke struktur kode program yang lebih bersih dan menjadikan aplikasi lebih tegar. Ketika kesalahan yang serius ditemukan, program Java menciptakan *exception*. *Exception* dapat ditangkap dan dikelola program tanpa resiko membuat sistem menjadi turun.

Program Java mendukung *native method* yaitu fungsi ditulis di bahasa lain, biasanya C/C++. Dukungan *native method* memungkinkan pemrogram menulis fungsi yang dapat dieksekusi lebih cepat dibanding fungsi ekivalen di java. *Native method* secara dinamis akan di-*link* ke program java, yaitu diasosiasikan dengan program saat berjalan.

Selain itu keuntungan menggunakan bahasa pemrograman Java antara lain memori pada Java secara otomatis dilengkapi *garbage collector* yang berfungsi mendealokasi memori yang tidak diperlukan. Tidak ada lagi upaya yang dilakukan pemrogram untuk melakukan *dispose*(). Kita tidak lagi dibebani urusan korupsi memori. Java menerapkan array sebenarnya, menghilangkan keperluan aritmatika pointer yang berbahaya dan mudah menjadi salah. Menghilangkan pewarisan jamak *(multiple inheritance)* diganti fasilitas antarmuka. Dan mudah dijalankan diberbagai platform.

Grafical User Interface (GUI) adalah salah satu kemampuan Java dalam mendukung dan manajemen antarmuka berbasis grafis. Tampilan grafis yang akan ditampilkan terhubung dengan program serta tempat penyimpanan data. Elemen dasar di Java untuk penciptan tampilan berbasis grafis adalah dua paket yaitu

AWT dan Swing. *Abstract Windowing Toolkit* (AWT), atau disebut juga *"Another Windowing Toolkit"*, adalah pustaka *windowing* bertujuan umum dan *multiplatform* serta menyediakan sejumlah kelas untuk membuat GUI di Java. Dengan AWT, dapat membuat window, menggambar, manipulasi gambar, dan komponen seperti *Button, Scrollbar, Checkbox, TextField,* dan menu *pull-down*.

Penggunaan komponen AWT ditandai dengan adanya instruksi : *import java.awt.*;* Swing merupakan perbaikan kelemahan di AWT. Banyak kelas swing menyediakan komponen alternatif terhadap AWT. Contohnya kelas JButton swing menyediakan fungsionalitas lebih banyak dibanding kelas *Button*. Selain itu komponen swing umumnya diawali dengan huruf "J", misalnya JButton, JTextField, JFrame, JLabel, JTextArea, JPanel, dan sebagainya. Teknologi swing menggunakan dan memperluas gagasan-gagasan AWT. Sementara, penggunaan komponen Swing ditandai dengan adanya instruksi : *import javax.swing.*;*

Beberapa perbedaan AWT dan Swing, AWT merupakan komponen *heavyweight* (kelas berat) sedangkan Swing *lightweight* (kelas ringan). Swing memiliki lebih banyak komponen. Fasilitas Swing *Look and Feel* : Metal, Windows, Motif. Komponen Swing berdasar *model-view,* yaitu suatu cara pengembangan komponen dengan pemisahan penyimpanan dan penanganan data dari representasi visual data.

Bahasa pemrograman Java merupakan salah satu bahasa pemrograman yang umum digunakan untuk mengembangkan aplikasi basis data yang dibuat menggunakan MySQL.

2.5.2. MySQL

SQL merupakan bahasa query yang paling banyak dipilih oleh *DBMS* dan *Development Tolls* (seperti *Visual Basic, Delphi, PHP, Java,* dll) dalam menyediakan median bagi penggunanya untuk berinteraksi dengan basisdata (Fatansyah, 2004).

MySQL merupakan salah satu contoh produk DBMS yang sangat popular di lingkungan linux, tetapi juga tersedia pada windows. Banyak situs web yang menggunakan *MySQL* sebagai database *server* (*server* yang melayani permintaan akses *database*) (Abdul Kadir, 2003).

Pernyataan-pernyataan SQL digunakan untuk melakukan beberapa tugas seperti, *update* data pada *database*, atau menampilkan data dari *database*. Beberapa software RDBMS dan dapat menggunakan SQL.

2.5.3. NetBeans

NetBeans mengacu pada kedua platform kerangka untuk aplikasi desktop java, dan sebuah lingkungan pengembangan terpadu (IDE) untuk pengembangan dengan java, java script, php, python, ruby, groovy, c, c++, scala, clojure, dan lain-lain. NetBeans yang ditulis dalam java dan berjalan di mana-mana JVM diinstal, termasuk windows, Mac Os, Linux, solaris. Sebuah JDK diperlukan untuk pengembangan fungsionalitas jawa, tetapi tidak diperlukan untuk pembangunan di bahasa pemograman lain. Platform NetBeans memungkinkan aplikasi untuk dikembangkan dari satu set modular komponen software yang disebut modul. Aplikasi berbasis platform NeatBeans(termasuk IDE NetBeans)

dapat diperpanjang oleh pengembangan pihak ketiga. (Miftakhul Huda & Bunafit Komputer. 2010. *Membuat Aplikasi Database dengan Java, MySQL, dan NetBeans*. PT Elex Media Komputindo. Jakarta.)

2.5.4. Rational Rose

Dalam penelitian ini, *Rational Rose* berfungsi sebagai *tool* untuk pemodelan sistem yang menggambarkan proses-proses yang ada pada sistem ini. *Rational Rose* adalah kakas (*tools)* pemodelan visual untuk pengembangan system berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam melakukan analisis dan perancangan sistem. *Rational Rose* digunakan untuk melakukan pemodelan sistem sebelum pengembang menulis kode-kode dalam bahasa pemrograman tertentu. Ia juga membantu analisis system dengan cara pengembang membuat diagram *use case* untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna. Kemudian, ia juga menuntut pengembang untuk mengembangkan *Interaction* diagram untuk melihat bagaimana objek-objek saling bekerjasama dalam menyediakan fungsionalitas yang diperlukan (Adi Nugroho, 2005:20).

2.6 Siklus Informasi

Informasi menjadi penting, karena berdasarkan informasi itu para pengelola dapat mengetahui kondisi obyektif perusahaannya. Informasi tersebut merupakan hasil pengolahan data/fakta yang dikumpulkan dengan cara tertentu. Informasi

disajikan dalam bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan yang dibutuhkan untuk menambah wawasan bagi pemakainya guna mencapai suatu tujuan.

Pengelolaan data menjadi informasi itu merupakan siklus, yang terdiri dari tahapan sebagai berikut :

a. Pengumpulan data, tahap ini dilakukan tahap ini dilakukan agar proses pengumpulan data yang asli dengan cara tertentu seperti sampling, data transaksi, data *warehouse*, dan lain sebagainya yang biasanya merupakan proses pencatatan data kedalam suatu file.
b. Input, tahap ini merupakan proses pemasukan data dan prosedur pengolahan data kedalam komputer melalui alat input seperti *keyboard.*
c. Pengolahan data, tahap ini merupakan tahap dimana data diolah sesuai dengan prosedur yang telah dimasukkan.
d. Output, hasil pengolahan data akan ditampilkan pada alat output seperti monitor dan printer sebagai informasi.
e. Distribusi, setelah proses pengolahan data dilakukan, maka informasi yang dihasilkan harus segera didistribusikan. Proses pendistribusian ini tidak boleh terlambat dan harus diberikan kepada yang berkepentingan. Sebab hasil pengolahan tersebut akan menjadi bahan pertimbangan dalam pengambilan keputusan atau menjadi data dalam pengolahan data selanjutnya.

2.6.1. Pengertian Gaji

Menurut Kamus Besar Bahasa Indonesia Pengertian Gaji adalah upah kerja yang dibayar dalam waktu tetap atau bias juga balas jasa yang diterima pekerja dalam bentuk uang berdasarkan waktu tertentu.

Dengan pengertian gaji diatas, maka dapat ditari kesimpulan bahwa gaji adalah upah atau imbalan sebagai balas jasa atau penghargaan atas hasil kerja seseorang.

Imbal jasa yang diberikan kepada pegawai sering kali disebut dengan gaji upah atau konpensasi, akan tetapi bila ketiga nama itu ditelaah satu persatu, maka akan berbeda pengertian.

2.6.2. Pengertian Upah

Menurut Kamus Besar Bahasa Indonesia Ada beberapa Pengertian upah, diantaranya adalah :

1. Upah adalah pembayaran imbalan jasa yang diterima oleh pegawai untuk pekerjaan yang dilakukan.
2. Upah adalah suatu istilah yang menunjukan pembayaran yang diberikan kepada pegawai yang dihitung dari jumlah jam kerja.

Maka berdasarkan beberapa pengertian diatas, dapat diambil kesimpulan bahwa upah adalah balas jasa yang diberikan kepada pekerja atas pekerjaan yang dilakukan baik didasarkan atas jam kerja maupun dengan ketentuan lain.

Meskipun pengertian gaji dan upah hamper sama, tapi pengertian ini sedikit berbeda dalam penerapan. Gaji secara umum berate pembayaran tiap bulan atau minggu, sedangkan upah bersangkutan dengan pembayaran atas dasar jam kerja.

2.6.3. Peran Gaji

Setiap pegawai berhak mendapatkan gaji layak dan adil sesuai dengan pekerjaan dan tanggung jawab. Gaji mempunyai peran penting bagi pegawai diantaranya :

1. Dengan gaji yang cukup, pegawai akan bekerja dengan baik.
2. Gaji yang cukup dapat mendorong pegawai untuk mengembangkan jasa dan semua kemampuan yang dimilikki.
3. Dengan gaji yang cukup, pegawai dapat mencukupi kebutuhan hidupnya.

2.6.4. Pengertian Penggajian

Menurut Kamus Besar Bahasa Indonesia penggajian berasal dari kata gaji yang berarti balas jasa yang diberikan kepada pegawai karena pegawai tersebut telah meberikan jasa, tenaga, dan pikiran terhadap organisasi.

2.6.5. Fungsi penggajian

Adapun fungsi dari penggajian adalah :

1. Pengembangan SDM secara efisien.
2. Pemberian gaji yang baik bagi pegawai yang berprestasi mendorong pegawai lain untuk ikut berprestasi.

3. Penggajian mendorong perbaikan dan stabilitas ekonomi.

2.6.6. Ketentuan Penggajian

Ketentuan proses perhitungan gaji pegawai yang ada di CV. MANDIRI JAYA adalah :

1. Untuk Keseluruhan Gaji Honor

Gaji honor dihitung dari Penerimaan, potongan gaji. Berikut ini adalah proses perhitungannya:

Penerimaan :

1. Gaji Pokok + Tunjangan istri, anak.

Potongan :

2. Pinjaman

Total gaji = penerimaan – potongan.

Order(Bonus) :

3. Bonus(berdasarkan pemesanan)/Rp. 30.000.

Total gaji = penerimaan – potongan + bonus.