BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Definisi sistem menurut Jogianto HM (2001 : 12) *Analisis Dan Desain Sistem Informasi*. Andi Offset. Yogyakarta.

"Sistem adalah kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu

menurut Dr. Azhar Susanto (2007 : 18) *system informasi manajemen: konsep dan pengembangannya*. Lingga Jaya Bandung.

"Sistem adalah kumpulan dari komponen apapun baik fisik atau non fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu"

Berdasarkan definisi-definisi diatas, dapat disimpulkan bahwa sebuah sistem bukanlah seperangkat unsur yang tersusun secara tak teratur, tetapi terdiri dari unsur yang dapat dikenal dan saling melengkapi karena satunya maksud, tujuan atau sasaran serta sistem yang baik adalah sistem yang mempunyai komponen-komponen yang saling berhubungan secara harmonis.

2.1.1 Elemen Sistem

Elemen – elemen yang terdapat dalam sistem ditandai dengan adanya :

1. Tujuan Sistem

   Tujuan Sistem merupakan tujuan dari sistem tersebut dibuat, dapat berupa tujuan organisasi, kebutuhan organisasi, permasalahan yang ada

dalam suatu organisasi maupun urutan prosedur untuk mencapai tujuan organisasi.

2. Batasan Sistem

   Batasan sistem merupakan sesuatu yang membatasi sistem dalam mencapai tujuan sistem, dapat berupa peraturan-peraturan yang ada dalam suatu organisasi, biaya-biaya yang dikeluarkan, orang-orang yang ada dalam organisasi, fasilitas baik itu sarana dan prasarana maupun batasan yang lain.

   a. Kontrol system

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

   b. *Input*

      Input merupakan elemen dari sistem yang bertugas untuk menerima seluruh masukan data, dimana masukan tersebut dapat berupa jenis data, frekuensi pemasukan data dan sebagainya.

   c. Proses

      Proses merupakan elemen dari sistem yang bertugas untuk mengolah atau memproses seluruh masukan data menjadi suatu informasi yang lebih berguna. Misalkan sistem produksi akan mengolah bahan baku yang berupa bahan mentah menjadi bahan jadi yang siap digunakan.

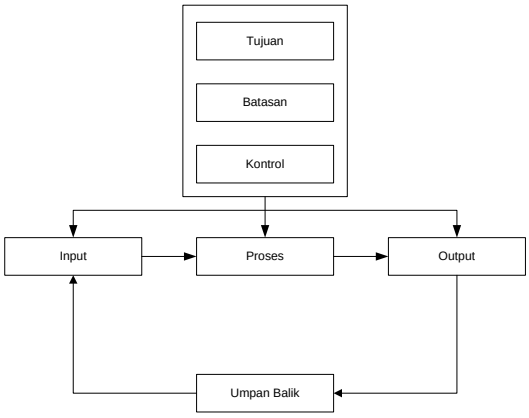
d. *Output*

Output merupakan hasil dari *input* yang telah diproses oleh bagian pengolah dan merupakan tujuan akhir sistem. *Output* ini bisa berupa laporan grafik, diagram batang dan sebagainya.

e. Umpan Balik

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

Hubungan antara elemen-elemen dalam sistem dapat dilihat pada gambar 2.1.



Gambar 2.1

Elemen-elemen sistem (Sumber : Andri Kristanto)

Dari gambar 7.1, bisa dijelaskan sebagai berikut : tujuan, batasan dan kontrol sistem akan berpengaruh pada input, proses dan output. Input yang masuk dalam sistem akan diproses dan diolah sehingga

menghasilkan *output*. *Output* tersebut akan dianalisa dan akan menjadi umpan balik bagi si penerima dan dari umpan balik ini akan muncul segala macam pertimbangan untuk input selanjutnya. Selanjutnya siklus ini akan berlanjut dan berkembang sesuai dengan permasalahan yang ada.

2.2 Konsep Dasar Informasi

"Informasi merupakan kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima. Tanpa suatu informasi, suatu sistem tidak akan berjalan dengan lancar dan akhirnya bisa mati. Suatu organisasi tanpa adanya suatu informasi maka organisasi tersebut tidak bisa berjalan dan tidak bisa beroperasi." (Sumber : Andri Kristanto)

"Informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian – kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan suatu keputusan." (Sumber : Teguh Wahyono)

Beberapa hal yang perlu diperhatikan dalam pembahasan konsep dasar informasi ini diantaranya adalah :

2.2.1 Pengolahan data

"Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian adalah sesuatu yang terjadi pada saat tertentu. Jadi data merupakan gambaran suatu kejadian yang sedang terjadi, dimana data tersebut akan diolah dan diterapkan dalam sistem menjadi input yang berguna dalam suatu sistem."

Salah satu metode untuk mengolah data adalah dengan media pengolah data yang menggunakan komputer. Dengan media ini semua permasalahan yang ada dapat diselesaikan secara cepat baik itu permasalahan yang menggunakan perhitungan matematis atau fungsi-fungsi lainnya. Selain itu dengan komputer, permasalahan yang ada dapat diselesaikan dengan sedikit mungkin kesalahan yang akan terjadi.

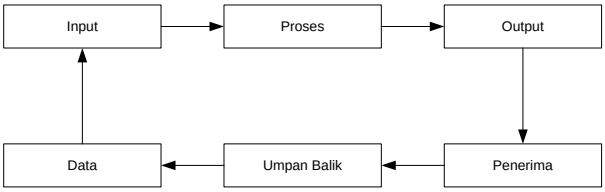
Operasi yang dilakukan dalam pengolahan data (Sumber : Andri Kristanto) antara lain :

1. Input data meliputi
    a. Mencatat transaksi data ke sebuah pengolahan data medium
    b. Melakukan pengkodean transaksi data ke dalam bentuk lain.
    c. Menyimpan data atau informasi untuk pengambilan keputusan.
2. Transformasi data meliputi :
    a. *Calculating*, adalah operasi aritmetika terhadap data *field* yang dimasukkan.
    b. *Summarizing*, adalah proses akumulasi beberapa data.
    c. *Classifying*.
3. *Output* data meliputi :
    a. *Displaying result*, yaitu menampilkan informasi yang dibutuhkan pemakai melalui monitor atau cetakan.
    b. *Reproducing*, penyimpanan data yang digunakan untuk pemakai lain yang membutuhkan.

c. *Telecomunicating*, penyimpanan data secara elektronik melalui saluran komunikasi.

2.2.2 Siklus Informasi

Data yang masih merupakan bahan mentah apabila tidak diolah maka data tersebut tidak akan berguna. Data tersebut akan berguna dan menghasilkan suatu informasi apabila diolah melalui suatu model. Model yang digunakan untuk mengolah data tersebut disebut dengan model pengolahan data atau lebih dikenal dengan nama siklus pengolahan data. Perhatikan gambar 2.2.



Gambar 2.2

Siklus pengolahan data (Sumber : Andri Kristanto)

Dari gambar di atas dapat dijelaskan bahwa data yang merupakan suatu kejadian yang menggambarkan kenyataan yang terjadi dimasukkan melalui elemen input kemudian data tersebut akan diolah dan diproses menjadi suatu *output* dan *output* tersebut adalah informasi yang dibutuhkan. Informasi tersebut akan diterima oleh pemakai/penerima, kemudian penerima akan memberikan umpan balik yang berupa evaluasi terhadap informasi tersebut dan hasil umpan balik tersebut akan menjadi data yang akan dimasukkan menjadi input kembali. Begitu seterusnya.

2.2.3 Kualitas Informasi

Kualitas informasi tergantung dari 4 hal yang sangat dominan (Sumber : Andri Kristanto), yaitu :

1. Akurat

   Informasi yang dihasilkan harus bebas dari kesalahan-kesalahan dan tidak menyesatkan bagi orang-orang yang menerima informasi tersebut. Adapun komponen akurat meliputi :

   a. Completeness

   b. Correctness

   c. Security

2. Tepat waktu

   Informasi yang diterima harus tepat pada waktunya, sebab kalau informasi yang diterima terlambat maka informasi tersebut sudah tidak berguna lagi.

3. Relevan

   Informasi harus mempunyai manfaat bagi si penerima, sebab informasi ini yang akan digunakan untuk pengambilan suatu keputusan dalam pemecahan suatu permasalahan yang ekonomis, efisien dan dapat dipercaya.

2.3 Konsep Dasar Sistem Informasi

"Sistem informasi merupakan suatu sistem yang dibuat oleh manusia yang terdiri dari komponen – komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.(Sumber : Teguh Wahyono)

"Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut. Selain itu dapat juga memegang peranan yang sangat penting dalam sistem informasi." Data yang akan dimasukkan adalah sebuah sistem informasi dapat berupa formulir-formulir, prosedur-prosedur dan bentuk data lainnya.(Sumber : Andri Kristanto)

Selain itu pula, sistem informasi dapat didefinisikan sebagai berikut :

1. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
2. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi.
3. Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Maka dapat kita simpulkan bahwa sistem informasi adalah sebuah sistem yang sengaja diciptakan oleh manusia untuk memenuhi kebutuhan informasi guna mencapai suatu tujuan tertentu.

2.4. Pengertian Absensi

Berdasarkan kamus bahasa Indonesia absensi adalah daftar administrasi ketidakhadiran pegawai.

2.5. Perancangan Proses

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

2.5.1. 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.

2.4.2. Diagram Konteks

Diagram *konteks* menurut Andri Kristanto 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.

2.4.3. Data Flow Diagram

1. 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.
2. 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."

3. 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.

2.4.4. 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.

2.4.5. 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.*

2.4.6. 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.

2.4.7. ERD

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.

2.4.8. 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.

2.5. Perangkat lunak pendukung

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.)