BAB II

LANDASAN TEORI

Pada landasan teori akan dijelaskan beberapa teori yang berkaitan dengan permasalahan yang akan dibahas sebagai bentuk sarana untuk mendukung dan memperkuat dalam penyusunan suatu sistem informasi yang akan dikembangakan.

2.1. Pengertian Sistem

Menurut *Robert G Murdick* dalam kutipan buku karya *Al Bahra bin Ladjamudin (2005:3)* mendefinisikan:

"Sistem adalah seperangkat elemen-elemen yang terintegrasi dengan maksud yang sama untuk mencapai tujuan bersama".

Menurut *Gerald J* dalam kutipan buku karya Al Bahra bin Ladjamudin (2005:3) mendefinisikan sistem sebagai:

"Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan berkumpul bersama-sama untuk melakukan suatu kegiatan atau fungsi utama dari perusahaan atau organisasi untuk mencapai sasaran tertentu".

Dari pengertian yang dikemukakan oleh *Robert G Murdick* dan *Gerald J* maka dapat disimpulkan bahwa *Sistem* adalah suatu himpunan "benda" yang nyata atau abstrak. dimana sistem terdiri dari bagian–bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, saling mendukung, yang secara keseluruhan bersatu menjadi satu kesatuan untuk mencapai tujuan.

2.1.1. Karakteristik Sistem

Karakteristik sistem adalah sistem yang mempunyai komponen-komponen, batas sistem, lingkungan sistem, penghubung, masukan, keluaran, pengolah dan sasaran. Sistem memiliki karakteristik atau sifat-sifat yang tertentu, yaitu:

1) Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen system atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2) Batas Sistem

   Batasan sistem (boundary) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau denagn lingkungan luarnya. Lingkungan Luar Sistem Lingkungan luar suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut.

3) Penghubung Sistem

   Merupakan media penghubung antara suatu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

4) Masukan (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.

5) Keluaran (Output)

Output merupakan elemen daru sistem yang bertugas untuk mengolah atau memproses seluruh masukan data menjadi suatu informasi yang lebih berguna.

6) Pengolah Sistem

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

7) Sasaran Sistem

Suatu sistem pasti mempunyai tujuan atau sasaran. Sasarn dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem keluaran yang akan dihasilkan sistem.

2.2. Pengertian Informasi

Kata informasi berasal dari kata Perancis kuno informacion (tahun 1387) yang diambil dari bahasa Latin informationem yang berarti “garis besar, konsep, ide”.

Informasi merupakan kata benda dari “informare” yang berarti aktivitas dalam “pengetahuan yang dikomunikasikan”.

Pengertian Informasi Menurut Jogiyanto HM., (1999: 692), “Informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih

berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian – kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan".

Sumber dari informasi adalah data. Data merupakan keterangan tertulis mengenai suatu fakta yang masih berdiri sendiri-sendiri, belum mempunyai pengertian sebagai kelompok, belum terkoordinasi satu sama lain dan belum diolah sesuai dengan keperluan.

Kualitas dari suatu informasi tergantung dari 3 hal, yaitu informasi harus akurat, tepat waktu dan relevan.

1) Akurat, berarti informasi harus bebas dari kesalahan dan tidak bisa atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Ketidakakuratan informasi dapat terjadi karena sumber informasi (data) mengalami gangguan atau kesengajaan sehingga merusak atau merubah data-data asli tersebut.
2) Tepat Waktu, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak mempunyai nilai lagi. sehingga kalau digunakan sebagai dasar dalam pengambilan keputusan akan berakibat fatal atau kesalahan dalam keputusan dan tindakan. Kondisi demikian menyebabkan mahalnya nilai suatu informasi, sehingga kecepatan untuk mendapatkan, mengolah dan mengirimkannya memerlukan teknologi-teknologi terbaru.

3) Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya.

2.3. Pengertian Sistem Informasi

Menurut Dr. Ir. Herry Purnomo, M.com dan Theo Zacharias, ST (2005:252) mendefinisikan sistem informasi:

"Sistem informasi adalah sebuah organisasi informasi yang terdiri atas bagian-bagian yang terkoneksi untuk mencapai tujuan tertentu"

Adapun terdapat beberapa pengertian sistem informasi :

a) Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
b) Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan organisasi.

Dari beberapa definisi yang dikemukakan diatas maka dapat disimpulkan bahwa sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luat tertentu dengan laporan-laporan yang diperlukan.

Komponen-komponen sistem informasi terdiri dari:

1) Hardware

Hardware merupakan kumpulan dari perangkat keras yang terlihat memungkinkan dapat membentuk sistem seperti komputer, printer dan jaringan.

2) Software

Kumpulan dari perintah-perintah atau fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer agar melaksanakan fungsi tertentu.

3) Data

Bahan dasar dari suatu informasi berupa fakta yang mengangkat kejadian-kejadian nyata dan dituangkan kedalam suatu simbol.

4) Prosedur

Procedur merupakan suatu tahapan yang berupa urutan kegiatan yang saling berhubungan untuk mencapai tujuan yang berupa suatu dokumen prosedur seperti buku, petunjuk operasional dan teknis.

5) Manusia

Manusia merupakan pelaksana dari suatu sistem informasi seperti: Operator, Programmer, Analyst, Designer dan sebagainya.

2.4. Bahasa Pemograman Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995.

Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code *(bytecode)* dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM).

Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik *(general purpose)*, dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa *platform* sistem operasi yang berbeda. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

2.4.1. Kelebihan Bahasa Pemograman Java

a) *Multiplatform*. Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas *operating system* Linux tetapi dijalankan dengan baik di atas Microsoft Windows. *Platform* yang didukung sampai saat ini adalah Microsoft Windows,

Linux, Mac OS dan Sun Solaris. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut.

b) OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek)

c) Perpustakaan Kelas Yang Lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

d) Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.

e) Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

2.4.2. Kekurangan Bahasa Pemograman Java

a) Tulis sekali, jalankan di mana saja - Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.

b) Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode* yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada *Microsoft .NET Platform*. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/*direverse-engineer*.

c) Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan *Object Pascal*). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkutat dengan mesin komputer berumur lebih dari 4 tahun.

2.5. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang *multithread*, *multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis

dibawah lisensi GNU *General Public License (GPL)*, tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL *(General Public License)*. Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL *(Structured Query Language)*. SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya.

Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya.

Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web *(wordpress)*, CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

2.5.1. Keunggulan dari MySQL:

1) Portabilitas.

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2) Perangkat lunak sumber terbuka

   MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

3) Multi-user.

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4) Performance Tuning

   MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5) Ragam tipe data

   MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6) Perintah dan Fungsi

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah *(query)*.

7) Keamanan

   MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama *host*, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8) Skalabilitas dan Pembatasan

   MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman *(records)* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9) Konektivitas

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket(UNIX), atau Named Pipes(NT).

10) Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11) Antar Muka

MySQL memiliki antar muka *(interface)* terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API *(Application Programming Interface)*.

12) Klien dan Peralatan

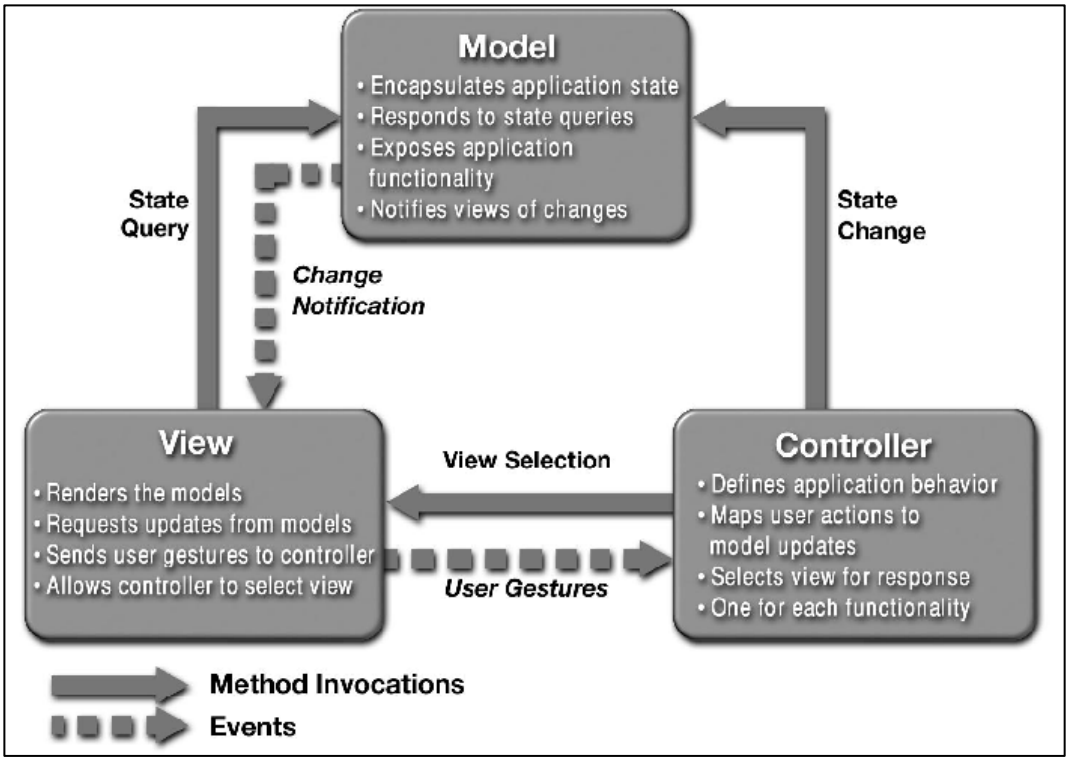
MySQL dilengkapi dengan berbagai peralatan *(tool)* yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

13) Struktur table

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam PostgreSQL atau Oracle.

2.6. Pengenalan Arsitektur MVC

Arsitektur *Model-View-Controller* adalah sebuah pola yang terbukti membangun proyek secara lebih efektif. Hal itu dilakukan dengan memilah komponen antara *Model, View* dan *Controller* pada bagian – bagian dalam proyek.



Gambar 2.1 MVC

2.6.1. Model

Pola MVC memiliki layer yang disebut dengan Model yang merepresentasikan data yang digunakan oleh aplikasi sebagaimana proses bisnis yang diasosiasikan terhadapnya. Dengan memilahnya sebagai bagian terpisah, seperti penampungan data, *persistence*, serta proses manipulasi, terpisah dari bagian lain aplikasi.

Terdapat beberapa kelebihan dalam pendekatan ini. Pertama, membuat detail dari data dan operasinya dapat ditempatkan pada area yang ditentukan *(Model)* dibanding tersebar dalam keseluruhan lingkup aplikasi. Hal ini memberikan keuntungan dalam proses *maintenance* aplikasi.

Kedua, dengan pemisahan total antara data dengan implementasi *interface*, komponen model dapat digunakan kembali oleh aplikasi lain yang memiliki kegunaan yang hampir sama.

2.6.2. View

Layer ini mengandung keseluruhan detail dari implementasi user interface. Disini, komponen grafis menyediakan representasi proses internal aplikasi dan menuntun alur interaksi *user* terhadap aplikasi. Tidak ada *layer* lain yang berinteraksi dengan user, hanya *View*.

Penggunaan *layer View* memiliki beberapa kelebihan : Pertama, memudahkan pengabungan divisi desain dalam development team. Divisi desain dapat berkonsentrasi pada *style, look & feel*, dan sebagainya, dalam aplikasi tanpa harus memperhatikan lebih pada detail yang lain.

Dan juga, memiliki *layer View* yang terpisah memungkinkan ketersediaan multiple interface dalam aplikasi. Jika inti dari aplikasi terletak pada bagian lain (dalam Model), multiple interfaces dapat dibuat *(Swing, Web, Console)*, secara keseluruhan memiliki tampilan yang berbeda namun mengeksekusi komponen *Model* sesuai fungsionalitas yang diharapkan.

2.6.3. Controller

Terakhir, arsitektur MVC memiliki *layer Controller*. *Layer* ini menyediakan detail alur program dan transisi layer, dan juga bertanggungjawab akan penampungan *events* yang dibuat oleh *user* dari *View* dan melakukan *update* terhadap komponen *Model* menggunakan data yang dimasukkan oleh *user*.

Kelebihan dalam penggunaan *layer Controller* secara terpisah : Pertama, dengan menggunakan komponen terpisah untuk menampung detail dari transisi *layer*, komponen *view* dapat didesain tanpa harus memperhatikan bagian lain secara berlebih. Hal ini memudahkan team pengembang *multiple interface* bekerja secara terpisah dari yang lain secara simultan. Interaksi antar komponen *View* terabstraksi dalam *Controller*.

Kedua, dengan menggunakan *layer* terpisah yang melakukan update terhadap komponen *Model*, detail tersebut dihapus dari *layer* presentasi. *Layer* presentasi kembali pada fungsi utamanya untuk menampilkan data kepada user. Detail tentang bagaimana data dari user mengubah ketetapan aplikasi disembunyikan oleh Controller. Hal ini memisahkan dengan jelas antara *presentation logic* dengan *business logic*.