BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

2.1.1 Definisi Sistem Informasi

Sistem informasi adalah suatu sistem yang dapat memberikan informasi kepada orang lain atau orang tertentu mengenai hasil pengolahan data-data yang telah dilakukan di suatu perusahaan atau instansi dan dibutuhkan oleh manajer sebagai bahan acuan untuk memutuskan suatu hal.[7]

Telah diketahui sebelumnya bahwa informasi suatu hal yang sangat penting bagi manajemen didalam menggambil suatu informasi tersebut dapat diperoleh dari sistem informasi (*Information system*) atau disebut juga dengan *processing system* atau *information processing system* atau *Information generating system*. Menurut Robert A. Leitch dan K. Roscoe Davis Sistem Informasi didefinisikan sebagai berikut : “Sistem Informasi adalah suatu sistem didalam 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 yang diperlukan”. [8]

Menurut Robert dan Donald Symazky Sistem Informasi sendiri mempunyai sejumlah komponen yang berbeda yaitu : manusia, data, hardware, dan software.

Setiap suatu sistem, setiap komponen tersebut berinteraksi untuk mencapai sasarannya.[8]

2.1.1.1 Komponen Sistem Informasi

John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technologi block*), blok basis data (*database block*), dan blok kendali (c*omputers block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.[9]

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” (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima *output*, 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*). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beroperasi. Misalnya teknisi adalah operator komputer, pemrogram, operator pengolah kata, spesialis telekomunikasi, analis sistem, penyimpanan data dan lain sebagainya.

5) Blok Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik

juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management Systems*).

6) Blok Kendali

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

2.1.1.2 Arsitektur Informasi

Arsitektur Informasi adalah suatu pemetaan atau rencana kebutuhan-kebutuhan informasi di dalam organisasi. Arsitektur ini berguna sebagai penuntun bagi operasi sekarang atau menjadi *blue print* untuk arahan dimasa mendatang.

Tujuan dari arsitektur ini adalah agar bagian teknologi informasi memenuhi kebutuhan-kebutuhan bisnis strategi organisasi oleh karena itu arsitektur informasi memadukan kebutuhan informasi, komponen sistem informasi, dan teknologi pendukung.

2.1.1.3 Tools Yang Digunakan untuk Analisis dan Perancangan Sistem

Untuk menjelaskan konsep yang telah di sebutkan diatas maka diperlukan suatu alat bantu untuk memodelkan sistem pada tahap analisis maupun pada tahap perancangan. Oleh karena itu, dibagi dalam dua kategori yaitu melakukan pemodelan terhadap proses dan pemodelan terhadap data.

1) Model Proses

Proses pemodelan ini akan menggunakan tool seperti diagram kontek diagram aliran data event list, kamus data. Spesifikasinya sebagai berikut : [4]

a. DFD (Data Flow Diagram)

Data flow diagram adalah alat pembuat model yang memungkinkan *profesional system* untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan aliran data baik secara manual maupun komputerisasi. Untuk memudahkan pembacaan DFD, maka penggambaran DFD dibantu dengan beberapa simbol diantaranya :

i. Kesatuan Luar (*External Entity*)

Setiap sistem pasti mempunyai batas sistem yang memisahkan suatu sistem dengan lingkungan luarnya. Namun sistem akan menerima *input* dan menghasilkan *output* kepada lingkungan luarnya.

ii. Arus data (*Data Flow*)

Suatu data flow atau alur data dapat dipresentasikan dengan anak panah yang menunjukan arah menuju kedalam dan keluar dari suatu

proses. Aliran daya ini digunakan untuk menerangkan perpindahan data atau paket data atau informasi dari suatu bagian sistem ke bagian lainnya aliran data dapat disimbolkan dengan panah.

iii. Proses (*Process*)

Proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komponen dari hasil suatu atau data yang masuk ke dalam proses untuk dihasilkan atau data yang akan keluar dari proses.

iv. Penyimpanan data (Data Store)

Penyimpanan data (*data store*) merupakan simpanan dari atau yang dapat berupa sebagai berikut ini :

- suatu file / database di dalam sistem computer
- suatu arsip atau catatan manual
- suatu agenda atau buku

b. Daftar kejadian (Event List)

"*Event list* adalah daftar naratif dari kejadian yang terjadi di luar system yang menyebabkan sistem melakukan respon terhadap kejadian – kejadian tersebut .[10]

c. Kamus data (Data Dictionary)

Kamus data atau *data dictionary* disebut juga dengan istilah sistem dari *dictionary* adalah katalog fakta tentang data dan kebutuhan–kebutuhan dari suatu sistem informasi dengan menggunakan kamus data

analis sistem cepat mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data dibuat pada sikap analis sistem dan digunakan pada setiap analis maupun pada tahap perencanaan sistem.

"Kamus data adalah dokumentasi yang mendukung data flow diagram terdiri dari definisi dari setiap data flow dan *data store* yang berada pada *data flow diagram* tersebut. Kamus data terdiri dari dua macam yaitu kamus data komposit dan kamus data elementer".[4]

"Kamus data komposit digunakan untuk menjelaskan komposisi dari paket data yang komplek yang dapat dipecah menjadi beberapa item – item elementer". [11]

"Kamus data elementer digunakan untuk menjelaskan data – data elementer, yang tidak dapat dipecah – pecah lagi".[10]

Kamus data mendefinisikan elemen data dengan fungsi sebagai berikut :

1. Menjelaskan arti aliran data dan penyimpanan dalam DFD.
2. Mendefinisikan komponen paket data yang bergerak melalui aliran.
3. Mendeskripsikan komposisi penyimpanan data.

d. Spesifikasi Proses

Spesifikasi porses adalah proses yang menjelaskan tentang kejadian–kejadian yang terjadi pada buble yang terdapat di level terbawah pada data flow diagram.

"Spesifikasi proses menggambarkan kejadian di dalam setiap buble pada level terbawah pada *data flow diagram*. Spesifikasi proses mendefinisikan kegiatan yang harus dilakukan untuk mengubah input menjadi *output*.[12]

Contoh Spesifikasi Proses :

Tabel Error! No text of specified style in document**..1 Spesifikasi Proses**

Nama Komponen	**Keterangan**
No.proses	No .proses berdasarkan DFD pada level terbawah
Nama .proses	Nama proses berdasarkan DFD pada level terbawah
Deskripsi	Penjelasan singkat tentang proses diatas
Dipanggil oleh proses	Proses diatas dipanggil oleh no proses berapa
Memanggil .proses	Proses diatas memanganggil no proses berapa
Flow Input	Masukan aliran data ke proses
Flow Output	Keluaran aliran data ke proses
Store Input	Pengambilan data store oleh suatu proses
Store Output	Penyimpanan data store oleh suatu proses
Logika proses	Berisi narasi/logika setiap proses pada level terbawah

2) Model Data

Model data yang digunakan dalam pembuatan laporan ini adalah *Model Entity Relationship* dan skema relasi.[4]

a. Model ERD (*Entity-Relationship Diagram*)

Entity Relationship diagram adalah suatu model jaringan (*network*) yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Tujuan utama penggambaran ERD adalah untuk menunjukan struktur objek data *(entity)* dan hubungan *(Relationship)* yang menunjukan struktur objek data (*entity*) dan hubungan yang ada pada objek tersebut.

Komponen – komponen pembentuk model ERD ini adalah entitas *(entity)* dan relasi *(relation).* Dan berikut ini adalah penjelasan dari komponen pembentuk ERD :

i. *Entity*

Entity adalah sesuatu yang dapat diberikan dalam dunia nyata dengan keberadaan yang bebas baik secara fisik maupun secara abstrak (konsep), mempunyai karakteristik tertentu, dimana informasi yang berkaitan dengannya dikumpulkan. Tipe entitas adalah sekumpulan entitas yang menggunakan sifat dan karakteristik yang sama, sedangkan instance entitas adalah empat persegi panjang serta pemberian nama biasanya dengan menggunakan kata benda.

ii. Relasi

"Relasi adalah hubungan yang terjadi antara *instance* dari satu atau lebih tipe entitas. *Relationship* tidak mempunyai keberadaan fisik kecuali yang mewarisi dari hubungan entitas tersebut. *Relationship set* adalah kumpulan *relatioanship* yang sejenis. Simbol yang menggunakan kata kerja.

iii. Kardinalitas Pemetaan (*mapping constraint*)

Kardinalitas pemetaan atau rasio kardinalitas menunjukkan jumlah *entity* yang dapat dihubungkan ke satu *entity* lain dengan suatu *relationship sets*. Kardinalitas pemetaan meliputi :

- Hubungan satu ke satu (*one to one*)

  Yaitu satu *entity* dalam A dihubungkan dengan maksimum satu *entity* dalam B.

- Hubungan satu ke banyak (*one to many*)

  Yaitu satu *entity* dalam A dihubungkan dengan sejumlah *entity* dalam B. Satu *entity* dalam B dihubungkan dengan maksimum satu *entity* dalam A.

- Hubungan banyak ke satu (*many to one*)

  Yaitu satu *entity* dalam A dihubungkan dengan maksimum satu *entity* dalam B. Satu *entity* dalam B dapat dihubungkan dengan sejumlah *entity* dalam A.

- Hubungan banyak ke banyak (*many to many*)

  Satu *entity* dalam A dihubungkan dengan sejumlah *entity* dalam B.

b. Relationship

Sebuah relationship mempresentasikan suatu kumpulan yang mungkin terjadi diantara kejadian *entity*. Kardinaliti mengidentifikasikan jumlah keberadaan dari sebuah *entity* dalam relasinya dengan *entity* yang lain.

c. Field / Atribut Key

Atribut key terdiri dari *Candidat key* (kunci kandidat), *Primary key* (Kunci Primer), *Foreign key* (Kunci Tamu).

1. Kunci Kandidat *(Candidate Key*)

   Kunci kandidat adalah satu atribut atau satu set minimal atribut yang mengidentifikasikan secara unik suatu kejadian spesifik dari entity. satu minimal set dari atribut menyatakan secara tak langsung dimana anda tidak dapat membuang beberapa attribute dalam set tanpa merusak kepemilikan yang unik. Jika satu kunci kandidat berisi lebih dari satu *attribute*, maka biasanya disebut sebagai *composite key* (kunci campuran/gabungan).

2. Kunci Primer (*Primary Key*)

   Primary key adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasikan secara unik suatu kejadian spesifik, tapi juga dapat mewakili setiap kejadian dari suatu *entity*.

   Setiap kunci kandidat punya peluang menjadi *primer key*, tetapi sebaliknya dipilih satu saja yang dapat mewakili secara menyeluruh terhadap *entity* yang ada.

3. Kunci Tamu (*Foreign Key*)

   Foreign key adalah satu atribut (atau satu set atribut) yang melengkapi satu *relationship* yang menunjukan ke induknya. Kunci tamu ditempatkan pada *entity* anak dan sama dengan kunci primer induk direlasikan. Hubungan antara *entity* induk dengan anak adalah hubungan satu lawan banyak (*one to many relationship*).

3) **Konsep Dasar Basis Data**

Basis data terdiri dari dua kata, yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang tempat berkumpul. Sedangkan data merupakan suatu kumpulan kejadian yang diambil dari suatu kenyataan yang dapat berupa angka, huruf, simbol khusus yang saling berhubungan satu dengan yang lainnya dan diorganisasikan sedemikian rupa sehingga suatu saat dapat dimanfaatkan kembali dengan mudah dan cepat.[3]

i. Sistem Basis Data

Sistem basis data merupakan sekumpulan komponen-komponen yang terdiri dari kumpulan file atau tabel yang mempunyai fungsi masing-masing yang saling berhubungan dan sekumpulan program (DBMS) yang memungkinkan pemakai dan program lain untuk mengakses dan memanipulasi file-file atau tabel-tabel tersebut. Adapun komponen-komponen basis data adalah sebagai berikut :

a) Perangkat keras (*Hadware*)

b) Perangkat lunak (*Software*)

c) Sistem operasi (*Operating System*)

d) Basis data (*Database*)

e) Pengelola basis data (DBMS)

f) Pemakai (*User*)

ii. Konsep DBMS (*Database Management System*)

Semua operasi yang berhubungan dengan basis data harus menggunakan *Database Management System* (DBMS) yang menentukan bagaimana data diorganisasikan atau dikelola, misalnya untuk menambah data, menghapus data, dan membaca data. Bila pemakai mengakses basis data maka DBMS menyediakan antara pemakai dengan basis data.

2.1.2 Rekayasa Perangkat Lunak

2.1.2.1 Evolusi Perangkat Lunak

Sebuah perangkat lunak yang disimpan atau digunakan dalam jangka waktu yang cukup lama akan menimbulkan masalah apabila sering diabaikan atau tidak diperhatikan perkembangannya. Permasalahan yang akan timbul dari penggunaan perangkat lunak yang cukup lama adalah : [4]

1. Adanya kebutuhan baru. Sebagai contoh pada saat ini berkembangnya perangkat lunak berbasis *Web*.
2. Untuk mengurangi kompleksitas, biaya , waktu pemasaran. Sebagai contoh pemanfaatan dari bahasa perkembangan bahasa pemrograman tingkat tinggi.
3. Mengurangi cacat. Sebagai contoh dengan cara membuat standarisasi dari pengkodean yang tidak sesuai.

Untuk dapat menyelesaikan permasalahan yang timbul diatas, maka diperlukannya melakukan evolusi dari perangkat lunak tersebut.

1. Pembangunan Perangkat Lunak

   Yang termasuk dalam pembangunan perangkat lunak itu sendiri meliputi dari daur hidup perangkat lunak, yaitu : permintaan (*requirement*), spesifikasi, perancangan (*design*), testing dan sebagainya.

2. Perawatan Perangkat Lunak

   Merupakan proses - proses untuk memperpanjang waktu penggunaan sistem perangkat lunak yang ada, sehingga tetap dapat dipergunakan sebagaimana mestinya sesuai dengan baik.

3. Migrasi Perangkat Lunak

   Merupakan proses - proses yang memindahkan sistem yang ada ke sistem yang baru dikarenakan perkembangan dari kebutuhan perangkat lunak tersebut.

Untuk dapat menjaga kualitas dari perangkat lunak tersebut dalam melakukan evolusi perangkat lunak, maka dibuatlah *"Laws"* dari perangkat lunak tersebut. Adapun *"Laws"* dari perangkat lunak tersebut menurut Lehman adalah :

1. Perubahan yang terus menerus secara kontinyu.

   Program yang berada pada lingkungan dunia nyata harus mengalami perubahan atau program tersebut menjadi tidak berguna pada lingkungan tersebut.

2. Meningkatnya kompleksitas.

   Seiring dengan berkembangnya sebuah program maka program tersebut menjadi lebih kompleks. Kemudian penambahan sumber daya juga dibutuhkan untuk memelihara dan menyederhanakan struktur programnya.

   Ada sebuah pernyataan yang mengatakan :

   "Most often overlooked risk in software engineering: As the system grows over time, it will become too complex or disjointed to understand or make work reliably." Deutsch (1998).

3. Aturan yang fundamental dari sebuah program evolusi.

   Pada saat membuat sebuah program evolusi telah ditentukan ukuran, metriks serta indikatornya.

4. Tetap dijaga stabilitas dari organisasinya.

   Sebagai contoh dengan menambahkan sumber daya (misal manusia) tetapi tidak mengubah produktivitas.

5. Tetap *familiarity*.

   Dimana pada saat membuat fungsi yang baru perbedaannya tidak jauh dari fungsi-fungsi yang sebelumnya.

2.1.2.2 Strategi Evolusi

Dalam melakukan evolusi perangkat lunak ada beberapa strategi yang dapat digunakan, diantaranya :

1. *Forward Engineering*

   Forward Engineering adalah sebuah proses pengubahan dari abstraksi level yang paling tinggi(*Requirement*) dan logik ke level design sampai ke level fisik (C*ode*)dari sistem.

2. *Restructuring*

   Restructuring adalah merupakan proses perubahan perangkat lunak yang terjadi pada level phisik(*Code*).

3. *Redocumenting*

   Redocumenting adalah proses revisi terhadap dokumentasi system yang telah ada pada setiap level abstraksi.

4. *Reverse Engineering*

   Reverse Engineering adalah proses untuk mengindentifikasi sistem yang bermula dari level abstraksi yang paling rendah (misal *object code*), untuk menghasilkan spesifikasi formal.

5. *Reengineering*

   Reengineering adalah proses untuk mengindentifikasi sistem yang bermula dari level abstraksi yang paling rendah (misal *object code*), untuk menghasilkan spesifikasi formal sehingga terbentuk *source code* baru.

6. *Roundtrip Engineering*

   Roundtrip Engineering merupakan proses untuk menjaga sinkronisasi antara *requirements, designs,* dan *code*.

7. *Retirement*

   Retirement adalah proses dimana sebuah perangkat lunak secara keseluruhan sudah tidak dipergunakan kembali (dipensiunkan).

Setiap strategi evolusi dapat dilakukan otomatisasi, dimana dapat dilakukan pada level fisik (*Code*). Proses tersebut terjadi pada bagian source code dengan mekanisme tertentu.

2.1.2.3 Model Proses Rekayasa Perangkat Lunak

Model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan alat-alat bantu yang akan dipakai, dan kontrol serta penyampaian yang dibutuhkan. Perkembangan perangkat lunak bisa dianggap sebagai lingkaran pemecahan masalah dimana terdapat empat keadaan berbeda, yaitu status quo, definisi masalah, perkembangan teknis memecahkan masalah di keseluruhan aplikasi dari banyak aplikasi, dan integrasi pemecahan menyampaikan hasil kepada siapa yang membutuhkan pertama kali. Bermacam-macam model proses yang berbeda pada perangkat lunak sebagai berikut:

1. Model Sekuensial Linier atau *Waterfall*

   Sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan.

2. Model Prototipe

   Prototyping paradigm dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut,yang akan nampak bagi pelanggan / pemakai (contohnya pendekatan *input* dan format *output*).

3. Model RAD

   Rapid Application Development (RAD) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier di mana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.

4. Model Proses Perangkat Lunak Evolusioner

Model evolusioner adalah model iteratif. Model itu ditandai dengan tingkah laku yang memungkinkan perekayasa perangkat lunak mengembangkan versi perangkat lunak yang lebih lengkap sedikit demi sedikit.

a. Model Pertambahan

Model inkeremental menggabungkan elemen-elemen model sekuensial linier (diaplikasikan secara berulang) dengan filosofi prototipe iteratif. Model pertambahan memakai urutan-urutan linier di dalam model yang membingungkan, seiring dengan laju waktu kalender. Model pertambahan berfokus pada penyampaian produk operasional dalam setiap pertambahannya.

b. Model Spiral

Model spiral (*spiral model*) adalah model proses perangkat lunak yang evolusioner yang merangkai sifat iteratif dari prototipe dengan cara komputer dan aspek sistematis dari model sekuensial linier.

c. Model Rakitan Komponen

Model ratikan komponen menggabungkan beberapa karakteristik model spiral. Model ini bersifat evolusioner, sehingga membutuhkan pendekatan iteratif untuk mencapai perangkat lunak. Tetapi model rakitan komponen merangkai aplikasi dari komponen perangkat lunak sebelum dipaketkan (kadang-kadang disebut “kelas”).

d. Model Perkembangan Konkuren

Model proses yang konkuren dapat disajikan secara skematis sebagai sederetan aktivitas teknik mayor, tugas-tugas, dan keadaannya yang lain. Contohnya aktivitas rekayasa yang dibatasi untuk model spiral dipenuhi dengan melakukan tugas-tugas sebagai berikut: *prototyping* dan atau pemodelan analisis, spesifikasi kebutuhan, dan rancangan.

5. Model Formal

Model metode formal mencakup sekumpulan aktivitas yang membawa kepada spesifikasi matematis perangkat lunak komputer. Metode formal memungkinkan perekayasa perangkat lunak untuk mengkhususkan, mengembangkan, dan memverifikasi sistem berbasis komputer dengan menggunakan notasi matematis yang tepat.

2.1.3 Borland Delphi 7.0.

Delphi adalah suatu program berbasis bahasa *Pascal* yang berjalan dalam lingkungan *Windows*. Delphi telah memanfaatkan suatu teknik pemrograman yang disebut RAD yang telah membuat pemrograman menjadi lebih mudah. Delphi adalah suatu bahasa pemrograman yang telah memanfaatkan metode pemrograman *Object Oriented Programming* (OOP).

2.1.4 MySQL

Mysql merupakan sofware sistem manajemen *database* (*Database Management System - DBMS*) yang sangat populer di kalangan pemrograman. MySQL merupakan *database* yang paling populer digunakan untuk membangun aplikasi web yang menggunakan *database* sebagai sumber dan pengelola data.

Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan , cepat secara kinerja query, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan skala menengah kecil.

Keistimewaan MySQL yaitu :

1. *Portability*

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *Windows, Unix, Free BSD, Mac OS X Server* dan lain-lain.

2. *Open Source*

   MySQL di distribusikan secara *Open Source* sehingga dapat digunakan secara bebas.

3. *Multi User*

   MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah.

4. *Performance Tuning*

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

5. *Column Types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *Signed/Unsigned integer, float, double, char, varchar, text, blob, date, time, times stamp, year, set* dan *enum.*

6. *Command dan Function*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *Query*

7. *Security*

   MySQL memiliki lapisan-lapisan sekuritas seperti level *Subnetmask,* nama *host,* dan ijin akses *user* dengan *system,* perijinan yang mendetail seperti password *terenkripsi.*

8. *Scalability dan Limits*

   MySQL mampu menangani *database* dengan skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 juta tabel serta 5 miliar baris. Selain itu, batas *indeks* yang dapat ditampung mencapai 32 *indeks* pada setiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan *client* menggunakan *protocol* TCP/IP, *Unix Socet (Unix),* atau *Named Pipe (NT).*

10. *Localitation*

    Deteksi pesan kesalahan pada *Client* dengan menggunakan lebih dari 20 bahasa

11. *Interface*

Interface terhadap berbagai palikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Client dan Tools*

Dilengkapi dengan berbagai *tools* yang dapat digunakan sebagai administrasi *database*, dan pada setiap *tool* yang ada di sertakan petunjuk *online*.

13. *Struktur Table*

Struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, di banding *database* lainnya semacam *postgre SQL* ataupun *oracle.*

2.1.5 Konsep Dasar 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 permintan-permintaan dari komputer *client* dan bertindak sebagai *sever database* yang menyimpan data. *Client* yaitu komputer / w*orkstation* yang melakukan pengiriman permintaan-permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya.

Sistem *client* / *server* merupakan suatu sistem komputer yang melibatkan proses-proses *client* yang meminta suatu pelayanan data kepada komputer *server* yang menyediakan layanan data tersebut, sehingga baik *client* maupun *server* sama-sama melakukan suatu pekerjaan. Dengan adanya kombinasi *client* dan

server ini maka kumpulan dari modul-modul program tidak dieksekusi dalam memori yang sama namun terbagi dalam komputer *client* dan *server*. Dengan arsitektur seperti ini, modul yang memanggil menjadi *client* yang meminta suatu layanan data dan modul yang dipanggil mencari *server* yang menyediakan layanan tersebut.

Dari jenis-jenis jaringan yang ada seperti (LAN, WAN, MAN, GAN) yang akan dijelaskan dalam oleh penulis adalah jenis LAN (*Local Area Network*), karena LAN merupakan jaringan yang akan penulis pakai.

2.1.5.1 Pengertian LAN (Local Area Network)

Local Area Network merupakan salah satu arsitektur jaringan yang paling sederhana dan dapat dikembangkan menjadi arsitektur jaringan yang lebih luas cakupannya. Luas cakupan LAN itu sendiri tidak melebihi dari satu area yang terdiri dari beberapa terminal yang saling dihubungkan sehingga menambahkan fungsi dari terminal itu sendiri layanan-layanan yang dapat diberikan LAN adalah penggunaan file bersama (*file SHAring)* atau penggunaan printer bersama *(printer SHAring*).

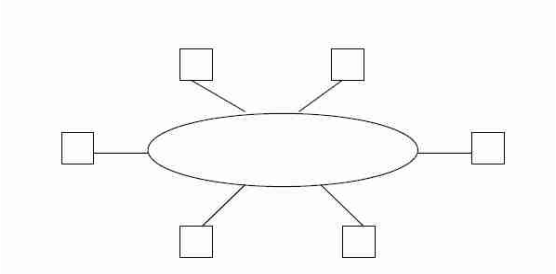
Biasanya LAN menggunakan satu *server* untuk melayani kebutuhan *client*-nya, tetapi tidak menutup kemungkinan untuk menggunakan lebih dari satu *server,* tergantung kebutuhan dari *client* itu sendiri. Biasanya yang menjadi pertimbangan adalah jenis layanan yang dibutuhkan dan performansi jaringan itu sendiri.

2.1.5.2 Topologi Jaringan

Apabila dilihat dari jenis hubungannya, maka topologi jaringan dapat dibagi menjadi tiga, yaitu:

a. Topologi cincin (ring topology)

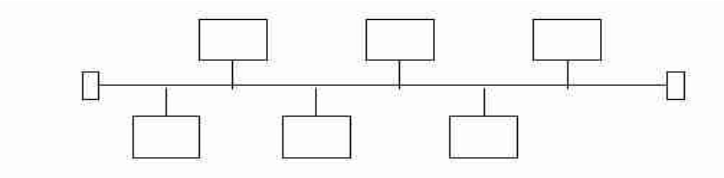
Untuk membentuk jaringan cincin, setiap sentral harus dihubungkan seri satu dengan yang lain dan hubungan ini akan membentuk loop tertutup. Dalam sistem ini setiap sentral harus dirancang agar dapat berinteraksi dengan sentral yang berdekatan maupun berjauhan. Dengan demikian kemampuan melakukan switching ke berbagai arah sentral. Keuntungan dari topologi jaringan ini antara lain : tingkat kerumitan jaringan rendah (sederhana), juga bila ada gangguan atau kerusakan pada suatu sentral maka aliran trafik dapat dilewatkan pada arah lain dalam sistem.



Gambar 2.1 Topologi Cincin

b. Topologi bus (*bus topology*)

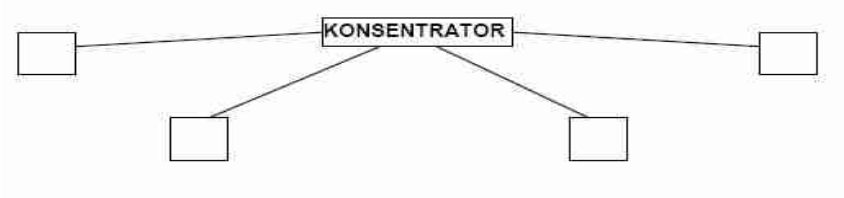
Topologi bus berupa komputer-komputer yang dihubungkan melalui satu jalur kabel. Kelemahan topologi bus adalah bila pada sutau titik jaringan mengalami kerusakan, seluruh jaringan akan mati. Implementasi topologi ini menggunakan kabel koaksial dan membutuhkan sedikitnya duah buah alat yang disebut terminator pada ujung-ujung kabel. Terminator berfungsi untuk memberikan hambatan sebesar 50 *ohm* antara kabel inti dengan serabut kawat.



Gambar 2.2 Topologi Bus

c. Topologi bintang (star topology)

Implementasi topologi star memerlukan hardware tambahan, yaitu konsentrator berupa switch atau hub yang berfungsi untuk mengatur lalu lintas data. Topologi ini lebih tahan terhadap gangguan dibandingkan topologi bus karena kerusakan di salah satu titik tidak akan mematikan seluruh jaringan, tetapi membutuhkan biaya lebih karena membutuhkan konsentrator. Dalam topologi *star* digunakan kabel *twisted pairs*.



Gambar 2.3 Topologi Bintang (Star)

2.1.5.3 Deskripsi Istilah-istilah Perbankan

Berikut ini adalah data yang diperlukan dan yang disusun dalam penyusunan laporan keuangan :

1. Neraca

   Pada umumnya neraca adalah laporan yang menggambarkan jumlah aktiva dan pasiva dalam suatu pengolahan keuangan

   Laporan neraca yang dibuat oleh perusahaan per periode. Neraca yang digunakan pada penyusunan laporan keuangan ini adalah neraca pada akhir tahun tutup buku.

2. Laba Rugi

   Laba rugi adalah laporan tentang selisih antara total pendapatan dan beban.

   Laporan laba rugi yang dibuat oleh perusahaan per periode. Laba rugi yang digunakan pada penyusunan laporan keuangan ini adalah laba rugi pada akhir tahun tutup buku.

3. Jurnal Umum

   Jurnal umum adalah catatan transaksi yang terjadi dalam suatu periode.

   Laporan jurnal umum yang dibuat oleh perusahaan per periode. Jurnal Umum yang digunakan pada penyusunan laporan keuangan ini adalah laba rugi pada akhir tahun tutup buku.

4. Inkubasi

   Memberdayakan dan membina bidang usaha yang dibawahinya.