BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Sistem adalah kumpulan dari unsur atau elemen-elemen yang saling berkaitan atau berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

Menurut Jerry Fith Gerald pengertian sistem yang menekankan pada prosedurnya adalah

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

Menurut Ludwig Von Bartalanfy pengertian sistem yang mengaitkan unsure-unsur dengan lingkungan

Sistem merupakan seperangkat unsur yang saling terkait dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan.

Menurut Anatol Raparot, pengertian tentang sistem yaitu

Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lainnya.

Menurut L. Ackof, pengertian tentang sistem

Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

(Sumber : http://mnfuady.wordpress.com/2008/07/29/konsep-dasar-informasi/)

2.1.1. Karakteristik Sistem

Adapun sistem memiliki beberapa karakteristik atau sifat-sifat tertentu antara lain :

a. Memiliki komponen

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

b. Batas Sistem *(Boundary)*

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

c. Lingkungan Luar Sistem *(Environment)*

Merupakan apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat menguntungkan dan juga merugikan.

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

d. Penghubung Sistem *(Interface)*

Merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Adapun penghubung lainnya yaitu :

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

e. Masukan Sistem *(input)*

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

f. Keluaran Sistem *(output)*

Merupakan hasil energi yang diolah oleh sistem.

g. Pengolah sistem *(proses)*

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

h. Sasaran sistem

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

2.2. Konsep dasar Informasi

Informasi adalah data yang dirubah menjadi bentuk yang lebih berguna dan berarti bagi pihak yang menerimanya. Sedangkan data adalah suatu kenyataan yang menggambarkan suatu kejadian-kejadian dari kesatuan nyata.

Menurut Barry E. Cushing, informasi didefinisikan sebagai hasil dari pengolahan data yang diorganisasikan dan berguna bagi orang atau pihak yang menerimanya. Informasi dikatakan berguna apabila mempunyai kualitas yang

baik dalam membantu seorang menejer mengambil keputusan dan dapat menentukan kebijaksanaan-kebijaksanaan guna mencapai tujuan organisasi.

Adapun kualitas dari suatu informasi ditentukan oleh karakteristik–karakteristik sebagai berikut :

1. Akurat *(Acurate)*

Diartikan bahwa suatu informasi harus benar-benar dapat memberikan suatu kebenaran dan bebas dari kesalahan-kesalahan dan harus jelas mencerminkan maksudnya.

2. Tepat Waktu *(Time Liness)*

Diartikan bahwa informasi yang digunakan bukan merupakan informasi yang telah usang dan tidak bernilai lagi, serta harus sampai pada penerima tidak terlambat sehingga dapat membuat keputusan secara tepat waktu.

3. Sejalan *(Relevan)*

Diartikan bahwa Informasi yang disampaikan mempunyai hubungan dengan masalah yang akan digunakan bersama informasi tersebut.

4. Lengkap

Diartikan bahwa informasi yang disampaikan kepada penerima tidak terpotong-potong, Karena hal itu mempengaruhi dalam pengambilan keputusan.

5. Jelas dan lugas

Diartikan bahwa informasi yang disampaikan kepada penerima harus jelas sehingga dapat diterima dan mudah dipahami.

(Sumber : http://mnfuady.wordpress.com/2008/07/29/konsep-dasar-informasi/)

2.3. konsep dasar sistem informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan. Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

2.3.1. Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan *(building blok),* yang terdiri dari komponen *input*, komponen model, komponen *output*, komponen teknologi, komponen *hardware*, komponen *software*, komponen basis data, dan komponen kontrol. Semua komponen tersebut saling berinteraksi satu dengan yang lain membentuk suatu kesatuan untuk mencapai sasaran.

1. Komponen *input*

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

2. Komponen model

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

3. Komponen *output*

Hasil dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua pemakai sistem.

4. Komponen teknologi

Teknologi merupakan "*tool box*" dalam sistem informasi, Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, neghasilkan dan mengirimkan keluaran, dan membantu pengendalian dari sistem secara keseluruhan.

5. Komponen *hardware*

Hardware berperan penting sebagai suatu media penyimpanan vital bagi sistem informasi.Yang berfungsi sebagai tempat untuk menampung database atau lebih mudah dikatakan sebagai sumber data dan informasi untuk memperlancar dan mempermudah kerja dari sistem informasi.

6. Komponen *software*

Software berfungsi sebagai tempat untuk mengolah, menghitung dan memanipulasi data yang diambil dari hardware untuk menciptakan suatu informasi.

7. Komponen basis data

Basis data *(database)* merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan yang lain, tersimpan di perangkat keras komputer dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan 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 menggunakan perangkat lunak paket yang disebut DBMS *(Database Management System).*

8. Komponen kontrol

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, teperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, 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.

(Sumber : http://apr1l-si.comuf.com/komponen.php/)

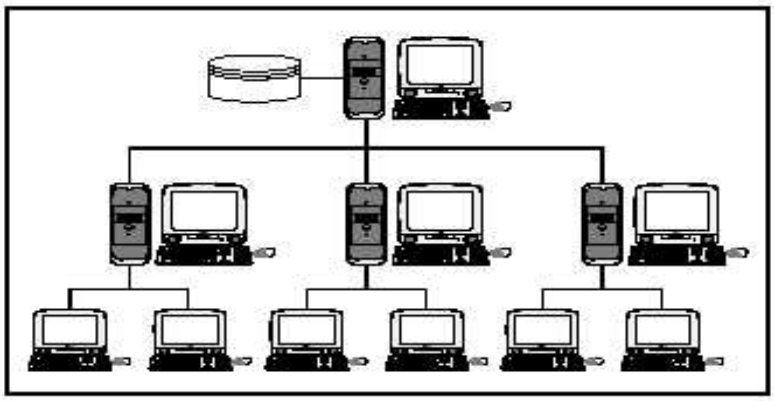
2.4. Client server

Arsitektur jaringan *client server* merupakan pengembangan dari arsitektur *file server.* Arsitektur ini adalah model konektifitas pada jaringan yang mengenal adanya *server* dan *client,* dimana masing-masing memiliki fungsi yang berbeda satu sama lainnya. Server dapat berbagi pakai data, aplikasi dan peripheral seperti

harddisk, printer, modem dan lain-lain. Oleh karena itu, tidak jarang juga tercipta sebutan print server, communication server dan lain sebagainnya. Prinsip kerjanya sangat sederhana, dimana server akan menunggu permintaan dari client, memproses dan memberikan hasilnya kepada client. Sedangkan client akan mengirimkan permintaan ke server, menunggu proses dan melihat visualisasi hasil prosesnya.

2.4.1. Sistem Client Server

Sistem Client dan Server terdiri atas dua komponen (mesin) utama, yaitu Client dan Server. Client berisi aplikasi basis data dan server berisi DBMS dan basis data. Setiap aktifitas yang dikehendaki para pemakai akan lebih dahulu ditangani oleh client. Client menangani proses yang menjadi tanggung jawabnya. Jika ada proses yang harus melibatkan data yang tersimpan pada basis data yang terletak di server, barulah client mengadakan hubungan dengan server. Pada bentuk sistem client server untuk memenuhi kebutuhan client akan megirimkan pesan atau perintah Query pengambilan data. Selanjutnya server yang menerima pesan tersebut akan menjalankan Query tersebut dan hasilnya akan dikirimkan kembali ke client. Dengan begitu, transfer datanya jauh lebih efisien. Untuk lebih jelasnya, dapat dilihat pada gambar sistem client server berikut ini :



Gambar 2.1. Sistem *Client-Server* Kompleks

(Sumber gambar : http://cutegalzjelek.blogspot.com/2012/04/model-arsitektur-clientserver.html)

2.4.2. Komponen dasar Client Server

1. Client

Client merupakan terminal yang digunakan oleh pengguna untuk meminta layanan tertentu yang dibutuhkan. Terminal *client* dapat berupa PC, ponsel, komunikator, robot, televisi dan peralatan lain yang membutuhkan informasi.

2. *Middleware*

Middleware merupakan komponen perantara yang memungkinkan *client* dan *server* untuk saling terhubung dan saling berkomunikasi satu sama lainnya. *Middleware* ini dapat berupa *transaction monitor* / TP. *Remote procedure call* atau *object Request Broker / ORB.*

3. Server

Server merupakan komputer khusus yang bertugas melayani aplikasi-aplikasi jaringan / pihak yang menyediakan layanan. Server ini

akan dapat berupa basis data SQL, monitor TP, *server groupware ,* server object dan web. Secara umum server berperan menerima pesan penerima layanan dari client, memproses permintaan tersebut dan mengirimkan hasil permintaan kepada client.

2.4.3. karakteristik *Server* dan *Client*

1. karakteristik Server
   a. Pasif
   b. Menunggu request
   c. Menerima request, memproses mereka dan mengirimkan balasan berupa service.
2. karakteristik client
   a. Aktif
   b. Mengirim request
   c. Menunggu dan menerima balasan dari server

http://cutegalzjelek.blogspot.com/2012/04/model-arsitekturclientserver.html

2.5. Pemodelan

Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pemodelan dapat menggunakan bentuk yang sama dengan realitas misalnya jika seorang arsitek ingin memodelkan sebuah gedung yang akan dibangun maka dia akan memodelkannya dengan menggunakan maket (tiruan).

Pada dunia pembangunan perangkat lunak sistem informasi juga diperlukan pemodelan. Pemodelan perangkat lunak digunakan untuk mempermudah langkah

langkah berikutnya dari pengembangan sebuah sistem informasi sehingga lebih terencana.

Pemodelan perangkat lunak memiliki beberapa abstraksi, misalnya sebagai berikut :

1. Petunjuk yang berfokus pada proses yang dimiliki oleh sistem
2. Spesifikasi Struktur secara abstrak dari sebuah sistem (belum detail)
3. Spesifikasi lengkap dari sebuah sistem yang sudah final
4. Spesifikasi umum atau khusus sistem
5. Bagian penuh atau parsial dari sebuah sistem

Perangkat pemodelan adalah suatu model yang digunakan untuk menguraikan sistem menjadi bagian-bagian yang dapat diatur dan mengkomunikasikan ciri konseptual dan fungsional kepada pengamat. Peran perangkat pemodelan :

1. Komunikasi

   Perangkat pemodelan dapat digunakan sebagai alat komunikasi antara pemakai dengan analisis sistem maupun *developer* dalam pengembangan sistem.

2. Eksperimentasi

   Pengembangan sistem yang bersifat *"trials and error"*.

3. Prediksi

   Model meramalkan bagaimana suatu sistem akan bekerja

Salah satu perangkat pemodelan adalah *Unified Modeling Language (UML).*

(Sumber : *http://cutegalzjelek.blogspot.com*)

2.5.1. Pengertian UML

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah system. Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan syntax/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT *(Object Modeling Technique),* dan Ivar Jacobson OOSE *(Object-Oriented Software Engineering).*

Sejarah UML sendiri cukup panjang. Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah : metodologi booch , metodologi coad , metodologi OOSE , metodologi OMT , metodologi shlaer-mellor , metodologi wirfs-brock , dsb.

Masa itu terkenal dengan masa perang metodologi (method war) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan group/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikata metodologinya banyak digunakan mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh **Object Management.**

2.5.2. Diagram UML

a. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang yang akan dibuat. Secara kasar *use case* digunakan untuk fungsi mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut actor dan *use case.*

1. Aktor merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi actor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

b. *Class Diagram*

Class diagrams menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimilki oleh suatu kelas

kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

   Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan

2. Kelas yang menangani tampilan sistem

   Kelas yang mendefinisikan dan mengatur tampilan pemakai

3. Kelas yang diambil dari pendefinisian *use case*

   Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*

4. Kelas yang diambil dari pendefinisian data

   Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Jenis-jenis kelas diatas juga dapat digabungkan satu sama lainnya sesuai dengan pertimbangan yang dianggap baik asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada. Susunan kelas juga dapat ditambahkan kelas utilitas seperti koneksi ke basis data, membaca *file* teks, dan lain sebagainya sesuai kebutuhan.

c. Activity diagrams

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya *(internal processing).* Oleh karena itu activity diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu use case atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Sama seperti state, standar UML menggunakan segi empat dengan sudut membulat untuk menggambarkan aktivitas. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan

proses-proses paralel *(fork dan join)* digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

d. *Sequence Diagrams*

Sequence diagrams menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagrams* terdiri atas dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek-objek yang terkait). *Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline vertikal. Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class.

Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan icon khusus untuk objek *boundary*, *controller* dan *persistent entity*.

e. Component Diagrams

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency)* di antaranya. Komponen piranti lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun *executable*, baik yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari beberapa *class* dan atau *package,* tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lainnya.

2.6. Pengenalan Jaringan

Berdasarkan kriterianya, jaringan komputer dibedakan menjadi 4 yaitu:

1. Berdasarkan distribusi sumber informasi/data

a. Jaringan terpusat

Jaringan ini terdiri dari komputer client dan server yang mana computer client yang berfungsi sebagai perantara untuk mengakses sumber informasi/data yang berasal dari satu komputer server

b. Jaringan terdistribusi

Merupakan perpaduan beberapa jaringan terpusat sehingga terdapat beberapa komputer server yang saling berhubungan dengan client membentuk sistem jaringan tertentu.

2. Berdasarkan jangkauan geografis dibedakan menjadi:

a. Jaringan LAN

Merupakan jaringan yang menghubungkan 2 komputer atau lebih dalam cakupan seperti laboratorium, kantor, serta dalam 1 warnet.

b. Jaringan MAN

Merupakan jaringan yang mencakup satu kota besar beserta daerah setempat. Contohnya jaringan telepon lokal, sistem telepon seluler, serta jaringan relay beberapa ISP internet.

c. Jaringan WAN

Merupakan jaringan dengan cakupan seluruh dunia. Contohnya jaringan PT. Telkom, PT. Indosat, serta jaringan GSM Seluler seperti Satelindo, Telkomsel, dan masih banyak lagi.

3. Berdasarkan peranan dan hubungan tiap komputer dalam memproses data.

a. Jaringan *Client-Server*

Pada jaringan ini terdapat 1 atau beberapa komputer server dan komputer client. Komputer yang akan menjadi komputer server maupun menjadi komputer client dan diubah-ubah melalui software jaringan pada protokolnya. Komputer client sebagai perantara untuk dapat mengakses data pada komputer server sedangkan komputer server menyediakan informasi yang diperlukan oleh komputer client.

b. Jaringan Peer-to-peer

Pada jaringan ini tidak ada komputer client maupun komputer server karena semua komputer dapat melakukan pengiriman maupun penerimaan informasi sehingga semua komputer berfungsi sebagai client sekaligus sebagai server.

4. Berdasarkan Media Transmisi Data

a. Jaringan Berkabel *(Wired Network)*

Pada jaringan ini, untuk menghubungkan satu komputer dengan komputer lain diperlukan penghubung berupa kabel jaringan. Kabel jaringan berfungsi dalam mengirim informasi dalam bentuk sinyal listrik antar komputer jaringan.

b. Jaringan Nirkabel *(Wireless Network)*

Merupakan jaringan dengan medium berupa gelombang elektromagnetik. Pada jaringan ini tidak diperlukan kabel untuk menghubungkan antar komputer karena menggunakan gelombang elektromagnetik yang akan mengirimkan sinyal informasi antar komputer jaringan.

2.7. Perangkat Lunak Pendukung

Dalam pembuatan aplikasi berbentuk software ini, kita membutuhkan perangkat lunak sebagai penunjang pembuatannya baik dalam text editor maupun bahasa pemrogramannya.

2.7.1. Pengenalan Java

a. Sejarah singkat Java

Pada 1991, sekelompok insinyur Sun dipimpin oleh Patrick Naughton dan James Gosling ingin merancang bahasa komputer untuk perangkat konsumer seperti cable TV Box. Karena perangkat tersebut tidak memiliki banyak memori, bahasa harus berukuran kecil dan mengandung kode yang liat. Juga karena manufaktur–manufaktur berbeda memilih processor yang berbeda pula, maka bahasa harus bebas dari manufaktur manapun. Proyek diberi nama kode *"Green"*. Kebutuhan untuk fleksibilitas, kecil, liat dan kode yang netral terhadap platform mengantar tim mempelajari implementasi Pascal yang pernah dicoba. Niklaus Wirth, pencipta bahasa Pascal telah merancang bahasa portabel yang menghasilkan intermediate code untuk mesin hipotesis. Mesin ini sering disebut dengan mesin maya *(virtual machine).* Kode ini kemudian dapat digunakan di sembarang mesin yang memiliki interpreter. Proyek Green menggunakan mesin maya untuk mengatasi isu utama tentang netral terhadap arsitektur mesin. Karena orang – orang di proyek Green berbasis C++ dan bukan Pascal maka kebanyakan sintaks diambil dari C++, serta mengadopsi orientasi objek dan bukan prosedural. Mulanya bahasa yang diciptakan diberi nama "Oak" oleh James Gosling yang mendapat inspirasi dari sebuah pohon yang berada pada seberang kantornya, namun dikarenakan nama Oak sendiri merupakan nama bahasa pemrograman yang telah ada sebelumnya, kemudian SUN menggantinya dengan JAVA. Nama JAVA sendiri

terinspirasi pada saat mereka sedang menikmati secangkir kopi di sebuah kedai kopi yang kemudian dengan tidak sengaja salah satu dari mereka menyebutkan kata JAVA yang mengandung arti asal bijih kopi. Akhirnya mereka sepakat untuk memberikan nama bahasa pemrograman tersebut dengan nama Java. Produk pertama proyek Green adalah Star 7 (*7), sebuah kendali jarak jauh yang sangat cerdas. Dikarenakan pasar masih belum tertarik dengan produk konsumer cerdas maka proyek Green harus menemukan pasar lain dari teknologi yang diciptakan. Pada saat yang sama, implementasi WWW dan Internet sedang mengalami perkembangan pesat. Di lain pihak, anggota dari proyek Green juga

menyadari bahwa Java dapat digunakan pada pemrograman internet, sehingga penerapan selanjutnya mengarah menjadi teknologi yang berperan di web.

(sumber : http://julio.staff.ipb.ac.id/files/2011/12/JENI-Intro1-Bab02-Pengenalan-Bahasa-JAVA.pdf)

2.7.2. Pengenalan MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. 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 *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL *(General Public License).* Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language)*. SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain :

1. *Portability*

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

2. *Open Source*

   MySQL didistribusikan secara open source (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*

   MySQL dapat digunakan oleh beberapa user 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. *Column types*

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

6. *Command dan functions*

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

7. *Security*

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

8. *Scalability* dan *limits*

   MySQL mampu menangani database dalam skala besar, dengan jumlah 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. *Connectivity*

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

10. *Localisation*

    MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface*

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

12. Clients dan tools

MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi database, dan pada setiap tool yang ada disertakan petunjuk online.

13. Struktur table

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan database lainnya semacam PostgreSQL ataupun Oracle.

(Sumber:http://rpl10rizkyapryanti.wordpress.com/2009/04/01/pengenalan-mysql/)

2.7.3. Pengenalan Java Netbeans

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra. Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Saat ini terdapat dua produk : NetBeans IDE dan NetBeans Platform.

The NetBeans IDE adalah sebuah lingkungan pengembangan – sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java – namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan.

2.8. Sistem Informasi Pelayanan Tamu Hotel

Berdasarkan penjelasan sistem, informasi dan sistem informasi maka beberapa pengertian yang berkaitan dengan judul skripsi yaitu antara lain.

2.8.1. Pengertian Pelayanan

Pelayanan adalah suatu kegiatan atau urutan kegiatan yang terjadi dalam interaksi langsung antara seseorang dengan orang lain atau mesin secara fisik dan menyediakan kepuasan pelanggan. Dalam kamus besar bahasa Indonesia dijelaskan pelayanan sebagai usaha melayani kebutuhan orang lain.

[http://www.damandiri.or.id/file/nurhasyimadunairbab2.pdf/ 15 april 2012]

2.8.2. Pengertian Tamu Hotel

Tamu Hotel adalah Setiap orang yang datang dengan tujuan untuk menginap di hotel dan atau mempergunakan fasilitas atau jasa- jasa hotel yang telah disediakan."

2.8.3. Pengertian Hotel secara Umum

Hotel dapat diartikan sebagai tempat / bangunan yang diorganisasikan secara komersial, dengan sistem yang baik dan fasilitas yang memuaskan kepada setiap tamu yang ada.(TriznaTarmoezi dan Heldin Manurung, 2004 : 1)

Dari pengertian diatas dapat disimpulkan bahwa hotel adalah tempat / bangunan yang diorganisasikan secara komersial, dengan sistem yang baik dan fasilitas yang memuaskan kepada setiap tamu yang ada.