BAB 2

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

Pengertian sistem menurut Andri Kristanto (2008:1) adalah sebagai berikut : ” Sistem merupakan kumpulan elemen - elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditujukan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang di inginkan.”

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

Prosedur adalah suatu urut-urutan operasi tulis - menulis dan biasa nya melibatkan beberapa orang didalm satu atau lebih *department* yang diterapkan, untuk menjamin penanganan yang seragam dari transaksi - transaksi bisnis yang terjadi.

2.1.1 Pengertian Data

Data adalah bahan yang akan diolah atau diproses yang bisa berupa angka-angka, huruf-huruf, simbol-simbol yang menunjukan suatu situasi dan lain-lain yang berdiri sendiri atau merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

Menurut Robert N.antony dan John Dearden, Data adalah : ‘ Bentuk jamak dari bentuk tunggal datum atau data-item’.

dan

“ Data Merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.” (Jogiyanto, 2005;8).

2.1.2 Pengertian Sistem

Suatu sistem sangatlah dibutuhkan dalam suatu perusahaan atau instansi pemerintahan, karena sistem sangatlah menunjang terhadap kinerja perusahaan atau instansi pemerintah, baik yang berskala kecil maupun besar. Supaya dapat berjalan dengan baik diperlukan kerjasama diantara unsur-unsur yang terkait dalam sistem tersebut.

Ada berbagai pendapat yang mendefinisikan pengertian sistem, seperti dibawah ini :

“Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”.(Jogiyanto,2005;1).

Masih dalam buku ‘Analisia dan Desain sistem informasi’ karangan Jogiyanto menerangkan:

“Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”.(Jogiyanto,2005;2)

2.1.3 Pengertian informasi

Dalam manajemen, informasi merupakan data yang telah diproses sehingga mempunyai arti tertentu bagi penerimanya. Sumber dari informasi adalah Data,

sedangkan data itu sendiri adalah kenyataan yang menggambarkan suatu kejadian, sedangkan kejadian itu merupakan suatu peristiwa yang terjadi pada waktu tertentu .dalam hal ini informasi dan data saling berkaitan.

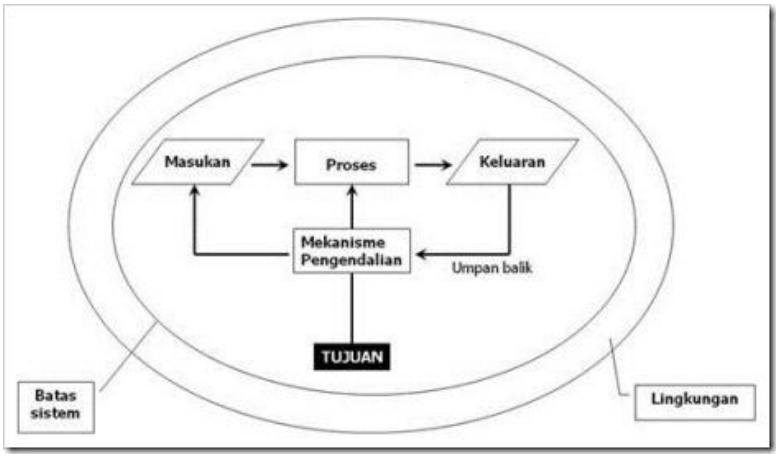
Menurut Jogiyanto dalam buku 'Analisis dan desain sistem informasi' adalah :"Informasi diartikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya" . (Jogiyanto,2005; 8).

Sedangkan definisi dari Robert A.leitch dan K.Roscoe davis 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-laporan yang diperlukan'. (Jogiyanto,2005;11)

Pengertian Informasi selalu dikaitkan dengan data, namun arti dari masing-masing kata dalam pengertian tersebut berbeda. Keberadaan suatu data sangat menunjang terhadap informasi, karena data merupakan bahan mentah yang diperlukan untuk mengambil keputusan.

2.1.4 Karakteristik Sistem



Gambar 2.1 Karakteristik Sistem

[Sumber : http://dimas347.files.wordpress.com]

Adapun sistem mempunyai beberapa karakteristik atau sifat-sifat tertentu antara lain yaitu:

a. Komponen sistem

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 sistem. Setiap sistem tidak perduli 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. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila 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 menunjukkan ruang lingkup (scope) dari sistem tersebut.

c. Lingkungan luar sistem (*environment*)

Adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat menguntungkan dan merugikan.

1. Keuntungan : Merupakan energi dari sistem yang harus tetap dijaga dan dipelihara karena bersifat menguntungkan.
2. Kerugian : Merupakan energi 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 yang lainnya. Adapun kegunaan penghubung lainnya adalah.

1. Memungkinkan sumber-sumber dapat mengalir dari subsistem yang 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 dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). Maintenance input adalah energi yang dimasukkan 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 dari energi yang diolah oleh sistem.

g. Pengolah sistem (*Process*)

Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.

h. Sasaran sistem

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

2.1.5 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya adalah sebagai berikut ini:

1. Sistem Abstrak dan Sistem Fisik

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sedangkan sistem fisik adalah sistem yang tampak secara fisik.

2. Sistem Alamiah dan Sistem Buatan Manusia.

   Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Sistem buatan manusia merupakan sistem yang melibatkan hubungan manusia dengan mesin yang disebut dengan human machine system.

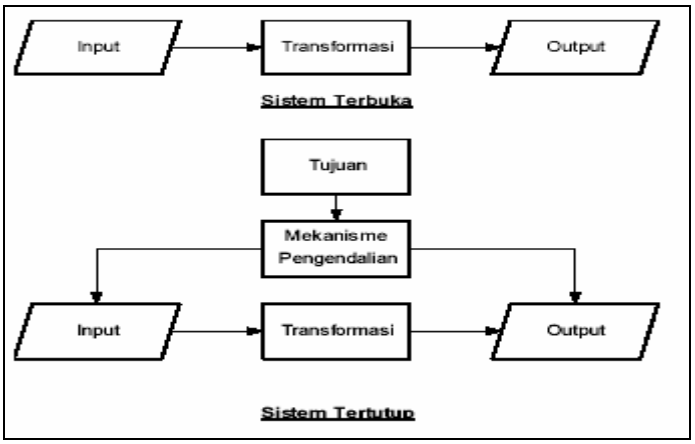
3. Sistem Deterministik dan Sistem Probabilistik.

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem Tertutup dan Sistem Terbuka.

   Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem terbuka adalah sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.

Klasifikasi sistem terbuka dan tertutup dapat digambarkan sebagai berikut :



Gambar 2.2 Klasifikasi sistem terbuka dan tertutup

[sumber : http://www.unhas.ac.id/]

2.2 Konsep Dasar Informasi

Menurut Jogiyanto dalam buku ‘Analisis dan desain sistem informasi’ adalah :“Informasi diartikan sebagai data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya” . (Jogiyanto,2005; 8).

Sumber dari informasi adalah data. Sedangkan data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu. Didalam dunia bisnis, kejadian nyata yang sering terjadi adalah perubahan dari suatu nilai yang disebut dengan transaksi. Misalnya penjualan adalah transaksi perubahan nilai barang menjadi nilai uang atau nilai piutang dagang. Satuan nyata (*fact dan entity*) adalah berupa suatu obyek nyata seperti tempat, benda dan orang yang betul – betul ada dan terjadi.

Adapun pengertian data adalah:

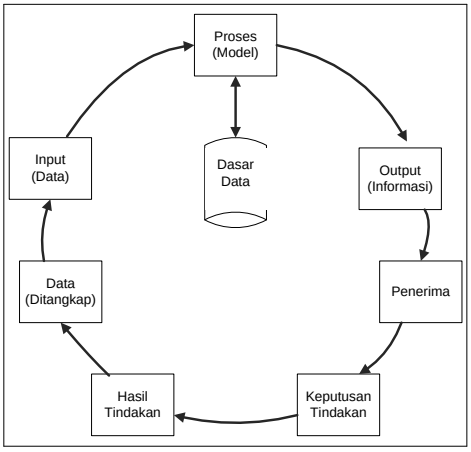
“ Data Merupakan kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.” (Jogyanto, 2005;8).

Pengertian Informasi selalu dikaitkan dengan data, namun arti dari masing-masing kata dalam pengertian tersebut berbeda. Keberadaan suatu data sangat menunjang terhadap informasi, karena data merupakan bahan mentah yang diperlukan untuk mengambil keputusan.

2.2.1 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk dihasilkan informasi.

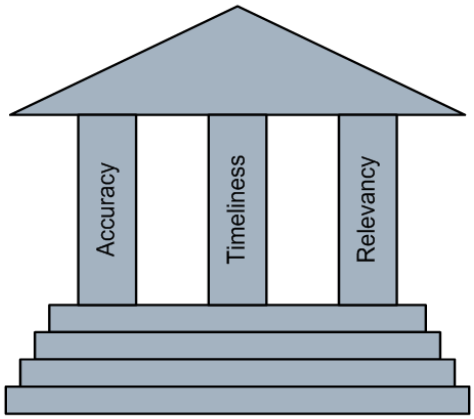
Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, siklus ini oleh John Burch disebut dengan siklus informasi. Siklus ini disebut juga dengan siklus pengolahan data.



Gambar 2.3 Siklus Informasi

[Sumber:Sistem Informasi Manajemen, Azhar Susanto, 2004, Penerbit Linggajaya]

2.2.2 Kualitas Informasi



Gambar 2.4 Kualitas Informasi

[Sumber: http://fairuzelsaid.files.wordpress.com/]

Kualitas dari suatu informasi tergantung dari tiga hal, yaitu informasi harus akurat, tepat pada waktunya dan relevan.

1. Akurat

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan *(noise)* yang dapat merubah atau merusak informasi tersebut.

2. Tepat pada waktunya

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi. Karena informasi merupakan landasan didalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka berakibat fatal untuk organisasi.

3. Relevan

Informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda.

2.2.3 Nilai Informasi

Nilai dari informasi (*value of information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah yang tertentu untuk memperolehnya, karena sebagian besar informasi dinikmati tidak hanya oleh satu pihak tetapi oleh banyak perusahaan.

2.3 Konsep Dasar Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen didalam pengambilan keputusan. Informasi dapat diperoleh dari sistem informasi (*information system*) atau disebut juga dengan *processing system* atau *Information processing system atau information – generating system*

Sistem Informasi menurut (Jogiyanto, 2005; 2) adalah :

"Sistem informasi adalah suatu sistem di dalam 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-lapoaran yang diperlukan".

2.3.1 Komponen Sistem Informasi

John Burch Gary Grudnitski mengemukakan bahwa:

"...sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan, yaitu blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya."

Adapun komponen-komponen dari suatu sistem informasi meliputi:

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” dalam sistem informasi. Teknologi digunakan untuk menerima input, 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*).

5. Blok Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat lunak untuk memanipulasinya. Data perlu disimpan didalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisensi kapasitas penyimpanannya. Basis data di akses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan BBMS (*data base management system*).

6. Blok Kendali.

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

2.3.2 Komponen Sistem Informasi Berbasis Komputer

Pengelompokan komponen-komponen sistem informasi adalah sebagai berikut:

1. Perangkat Keras (*Hardware*)

Perangkat keras (*hardware*) mencakup piranti-piranti fisik seperti komputer dan printer.

2. Perangkat Lunak (*Software*) atau Program

Perangkat lunak (*software*) merupakan sekumpulan instruksi yang memungkinkan perangkat keras untuk memproses data atau merupakan program-program yang digunakan untuk menjalankan komputer.

3. Prosedur (*Procedure*)

Prosedur (*procedure*) merupakan sekumpulan aturan yang dipakai untuk mewujudkan pemrosesan data dan pembangkitan keluaran yang dikehendaki.

4. Orang (*Brainware*)

Orang (*brainware*) merupakan semua pihak yang bertanggungjawab dalam pengembangan sistem informasi, pemrosesan, dan penggunaan keluaran sistem informasi.

5. Basis Data (*Database*)

Basis data (*database*) merupakan sekumpulan tabel, hubungan dan lain-lain yang berkaitan dengan penyimpanan data.

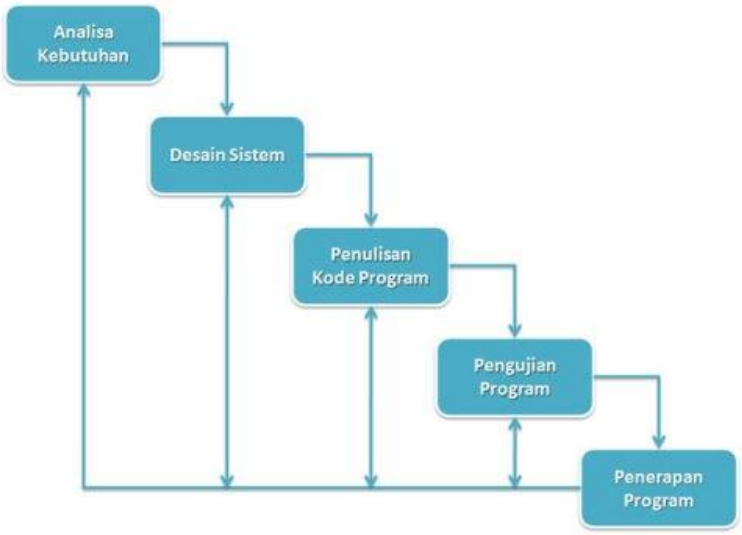
6. Jaringan Komputer dan Komunikasi Data

Jaringan komunikasi merupakan sistem penghubungan yang memungkinan sesumber (*resource*) dipakai secara bersama atau diakses oleh sejumlah pemakai.

2.4 Metode Pengembangan Sistem

Fase pengembangan sistem informasi disebut juga sebagai siklus hidup pengembangan sistem informasi yang garis besarnya terdiri dari enam langkah. Tahapan – tahapan pekerjaan dalam pelaksanaan tidak kaku namun dapat disesuaikan dengan kebutuhan, seperti cara literasi.

Metode yang dilakukan dalam pengembangan sistem pada penelitian ini menggunakan metode *waterfall* dengan langkah – langkah sebagai berikut :



Gambar 2.5 Model Air Terjun Rekaya Perangkat Lunak

Sumber[http://www.budihermawan.net/wp-content/uploads/waterfall.jpg]

Tahapan utama dalam siklus klasik / model air terjun rekaya perangkat lunak adalah sebagai berikut :

1.Analisis dan rekayasa sistem (*Sistem Engineering*)

Software merupakan bagian dari sebuah Sistem informasi. Tahap analisis dan rekayasa sistem dilakukan untuk mendapatkan gambaran yang meluas pada lingkup sistem.

2.Analisis sistem (*Analysis*) Tahap ini lebih terarah ke software, analisis persyaratan berusaha mengetahui aspek what. Tahapan ini banyak melibatkan pemakai dan pengembang.

3.Perencanaan (*Design*)

Tahap ini bertujuan menerjemahkan persyaratan menjadi suatu bentuk representatif yang dapat di evaluasi kualitas sebelum tahap coding dilakukan.

4.Penulisan Program (*Coding*)

Coding merupakan tahap penerjemah rancangan ke dalam bentuk yang dapat di mengerti komputer.

5.Pengujian (*Testing*)

Tahap ini berfokus pada pengujian rincian logika software. Pengujian bertujuan mengungkapkan dan menghilangkan kesalahan – kesalahan yang ada sehingga software bekerja sesuai yang diharapkan.

6.Pemeliharaan (*Maintance*)

Tahap memelihara meliputi kegiatan – kegiatan koreksi kesalahan dan penyesuaian software terhadap perubahan lingkungannya.

2.5 Pengertian UML

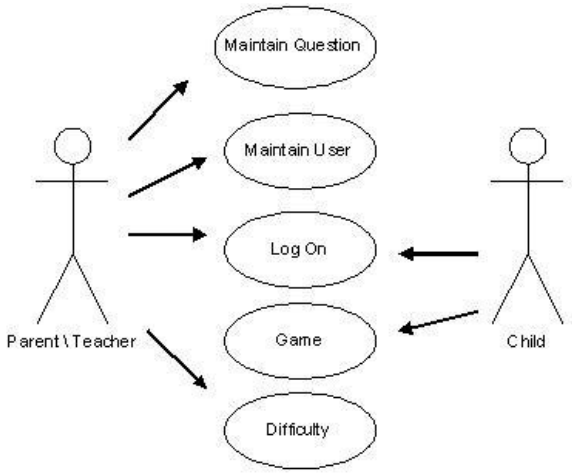
Unified Modeling Language (*UML*) adalah keluarga notasi grafis yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (*OO*). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang *UML* berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif (Martin 2005:1).

Pada tahap analisis, meliputi usaha untuk mengetahui apa kemampuan sebuah sistem yang diinginkan pengguna dan pelanggan dari sebuah perangkat lunak.

2.5.1 Diagram-Diagram Dalam *UML*

Berikut ini merupakan beberapa penjelasan tentang diagram – diagram yang ada dalam UML ini.

1. *Use Case Diagram*



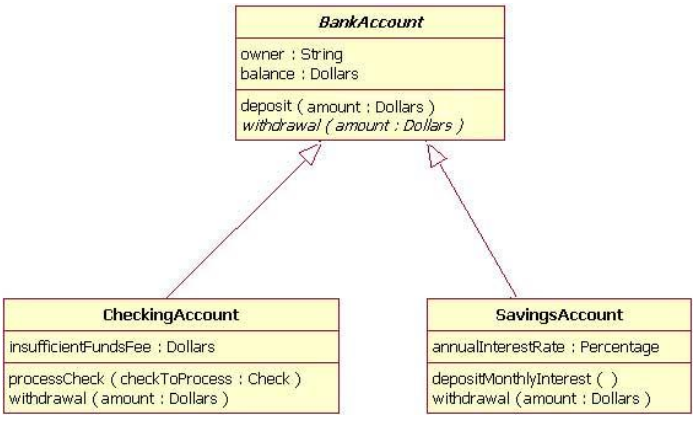
Gambar 2.6 Contoh Diagram Use Case

[Sumber : http://en.wikipedia.org/wiki/File:Use_Case_diagram_V1.JPG/]

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng- *create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat

membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di- *include* akan dipanggil setiap kali *use case* yang meng- *include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common* . Sebuah *use case* juga dapat meng- *extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

2. *Class Diagram*



Gambar 2.7 Contoh Class Diagram

[Sumber:http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/]

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)

2. Atribut

3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metoda. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class.*

Dengan demikian *interface* mendukung resolusi metoda pada saat *run-time*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*. Kita juga dapat membuat diagram yang terdiri atas *package*.

Hubungan Antar *Class* :

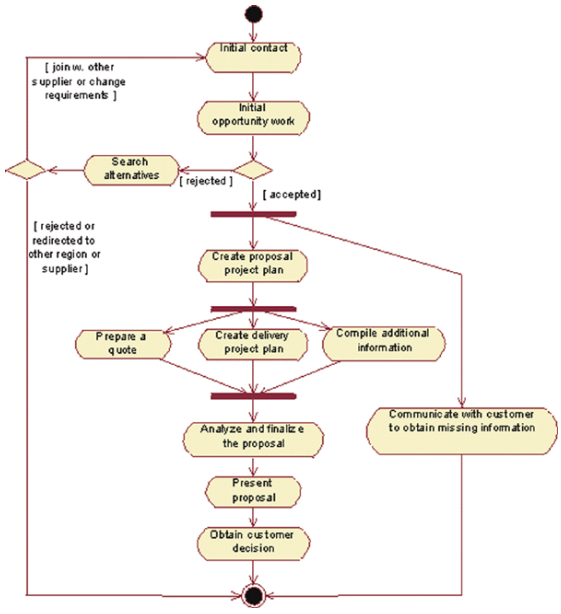
1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.

2. Agregasi, yaitu hubungan yang menyatakan bagian ("terdiri atas..").

3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.

4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

3. *Activity Diagram*



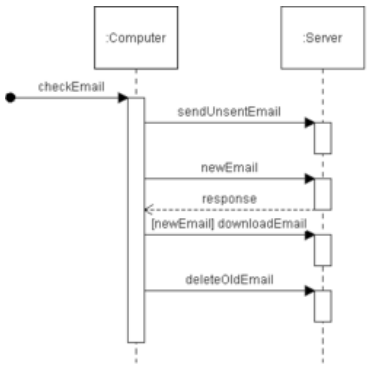
Gambar 2.8 Contoh Activity Diagram

[Sumber:http://www.ibm.com/developerworks/rational/library/2802.html]

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 segiempat 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.

4. *Sequence Diagram*



Gambar 2.9 Contoh Sequence Diagram

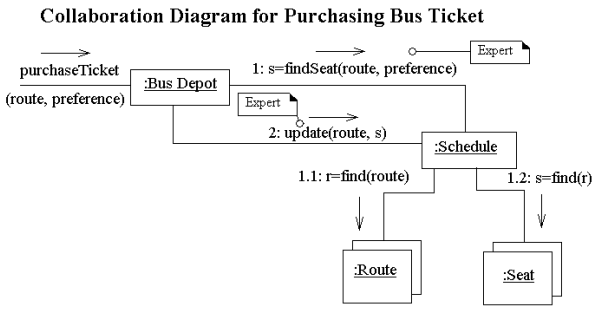
[Sumber : http://en.wikipedia.org/wiki/Sequence_diagram]

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* , dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar 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*

7. Collaboration Diagram

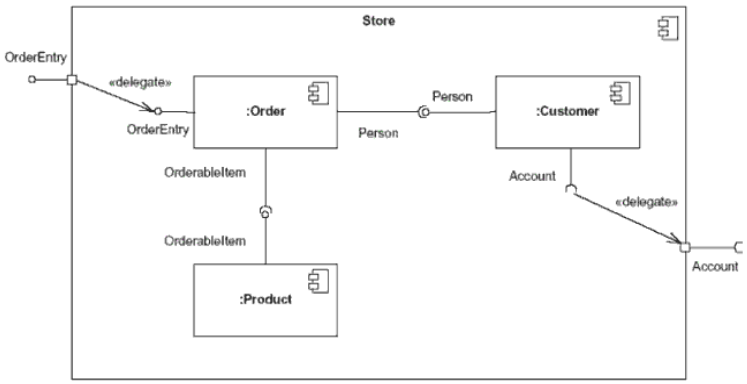


Gambar 2.10 Contoh *Collaboration Diagram*

[Sumber:http://www.comptechdoc.org/independent/uml/begin/umlcoldiagram.html]

Collaboration diagram juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number* , di mana *message* dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.

8. Component Diagram



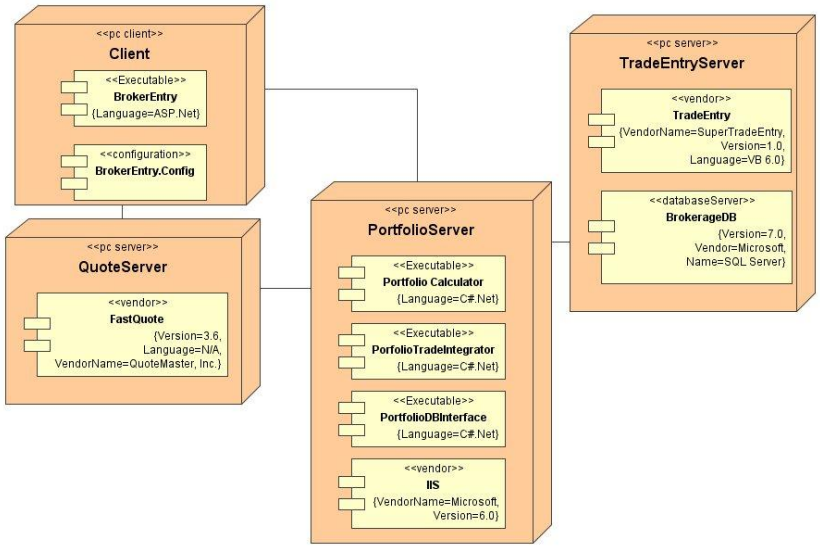
Gambar 2.11 Contoh *Component Diagram*

[Sumber: http://www.ibm.com/developerworks/rational/library/dec04/bell/]

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 lain.

9. Deployment Diagram



Gambar 2.12 Contoh *Deployment Diagram*

[Sumber: http://www.devx.com/enterprise/Article/27899/1954]

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server, *workstation* , atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

2.6 Perangkat Lunak

Adapun model – model proses perangkat lunak sebagai berikut :

1. *Linier Sequential Model* :

Model ini sering disebut model klasik atau *waterfall*. Model ini menyarankan pendekatan pengembangan secara sekuen dan sistematik untuk pengembangan perangkat lunak dimulai di level sistem, berlanjut ke analisis, lalu perancangan, pemograman, pengujian dan pemeliharaan. Model ini merupakan model yang tertua.

Kelemahan model ini adalah :

a. Proyek – proyek nyata jarang mengikuti alur sekuen yang diusulkan model. Meskipun linier model dapat mengakomodasikan iterasi, namun model melakukan secara tidak langsung. Sebagai hasilnya, perubahan – perubahan dapat menyebabkan kebingungan saat tim pengembangan melakukanya.

b. Model ini memerlukan pernyataan eksplisit itu dan sulit mengakomodasi ketidakpastian yang terdapat di awal dari kebanyakan proyek.

c. Versi yang dapat bekerja dari program tidak akan tersedia sampai akhir dari proyek. Kesalahan besar utama, Jika tidak terdeteksi sampai pada program kerja dikaji ulang,

Maka kesalahan itu dapat mengakibatkan program sama sekali tidak dapat digunakan. Menyiapkan banyak sumber daya.

Alasan kelemahan model ini adalah ;

a. Kebutuhan harus telah ditetapkan di awal siklus hidup.

b. Kebutuhan validasi terlalu lambat

Meskipun terdapat kelemahan, Namun model ini memberikan daftar lengkap jumlah aktivitas minimal yang perlu terdapat di model pengembangan perangkat lunak. Umumnya, model – model pengembangan yang lain merupakan superset dari model pengembangan model waterfall, Memuat aktivitas – aktivitas yang lebih banyak dari pada model waterfall ini.

2.10 Analisis dan Perancangan Sistem

Tahap analisis sistem merupakan tahap yang dikerjakan sebelum tahap perancangan. Pada tahap analisis sistem yang dikerjakan adalah menganalisis sistem untuk menentukan kelemahan-kelemahan sehingga dapat diusulkan perbaikannya.

2.7.1 Pengertian Analisis Sistem

Menurut Azhar Susanto (2008:318) analisis sistem adalah studi tentang sistem bisnis yang sedang berjalan dan permasalahannya, menentukan kebutuhan aktifitas bisnis dan permintaan-permintaan pemakai sistem dan melakukan evaluasi terhadap berbagai alternatif solusi.

Berdasarkan penjelasan diatas, penulis dapat menarik kesimpulan, bahwa analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi

yang utuh ke dalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan.

Langkah-langkah Menganalisis Sistem

Di dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem sebagai berikut ini :

1. *Identify*, yaitu memahami kerja dari sistem yang ada.

2. *Understand*, yaitu memahami kerja dari sistem yang ada.

3. *Analyze*, yaitu menganalisis sistem.

4. *Report*, yaitu membuat laporan hasil analisis.

2.7.2 Perancangan Sistem

Perancangan sistem dilakukan setelah tahap analisis sistem dan memperoleh gambaran dengan jelas apa yang harus dikerjakan dan bagaimana membentuk sistem tersebut.

2.7.2.1 Pengertian Perancangan

Menurut Azhar Susanto (2008:318) perancangan sistem adalah spesifikasi umum dan terinci dari pemecahan masalah berbasis komputer yang telah dipilih selama tahap analisis.

Tujuan dari perancangan sistem adalah sebagai berikut :

1. Untuk memenuhi kebutuhan pada pemakai sistem.

2. Untuk memberikan gambaran yang jelas dan rancang bangun yang lengkap kepada pemrogram komputer dan ahli-ahli teknik lainnya yang terlibat.

Untuk mencapai tujuan ini, analisis sistem harus dapat mencapai sasaran sebagai berikut :

1. Perancangan sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.

2. Perancangan sistem harus dapat mendukung tujuan utama perusahaan sesuai dengan yang telah didefinisikan pada tahap analisis sistem.

3. Perancangan sistem harus efisien dan efektif untuk dapat mendukung berbagai kegiatan pengolahan data.

4. Perancangan sistem harus dapat mempersiapkan rancang bangun yang terinci untuk masing-masing komponen dari sistem informasi.

Berdasarkan penjelasan diatas penulis dapat menarik kesimpulan, bahwa tujuan dari perancangan sistem itu sendiri adalah memberikan gambaran secara umum/global kepada pemakai tentang sistem yang akan dikembangkan dan berfungsi sebagai persiapan untuk tahap perancangan sistem secara terinci. Tahapan dari perancangan sistem meliputi desain input, proses, dan output.

2.7.3 Pengolahan Data

Data berasal dari kata latin yaitu ”datum” yang berarti fakta. Jadi data merupakan suatu kumpulan kejadian yang diangkat dari suatu kenyataan yang dapat

berupa angka, huruf atau simbol-simbol. Sebelum diolah, data tidak memilika makna apa-apa, akan tetapi setelah diolah akan menjadi berarti bagi pengelolanya, sehingga oleh karenanya diperlukan sistem pengolahan data.

Dalam perancangan database, data diasumsikan sebagai penjelasan atau gambaran dari fenomena atau ide yang dipertimbankan oleh manusia dalam rangka memformulasikan dan mencatatnya.

Menurut Azhar Susanto (2008:38) data adalah fakta atau apapun yang dapat digunakan sebagai input dalam menghasilkan informasi. Pengolahan data didefinisikan sebagai berikut : ”pengolahan data adalah segala kegiatan, baik dengan menggunakan peralatan elektronik, mekanis, maupun secara manual yang bertujuan mengolah data menjadi informasi”.

Langkah-langkah pengolahan data

Untuk prosedur pengolahan data biasanya menggunakan langkah-langkah dari pengumpulan sejauh operasi pengolahan data yang tediri dari :

a. Pencatatan (*Recording*)

Proses pencatatan data dari suatu kejadian ke dalam bentuk formulir atau dokumen yang nantinya berguna untuk masukan pengolahan data.

b. Pemeriksaan data *(Verifying*)

Proses pemeriksaan data terhadap data masuk, bertujuan agar data yang ada di dokumen dapat dipindahkan ke dalam komputer secara cermat, untuk menjaga kesalahan pencatatan karena dalam proses manual tidak luput dari kesalahan-kesalahan.

c. Pengelompokan data (*Classifying*)

Proses pengelompokkan data dapat diklasifikasikan menjadi beberapa kategori tertentu dan bertujuan untuk memudahkan atau dapat memberi arti tertentu bagi pengolah data.

d. Penyusunan *(Shorting*)

Proses penyusunan data di dasarkan peringkat tertentu dengan menentukan item penyortiran yang disebut *key*.

e. Penggabungan (*Merging*)

Dalam proses penggabungan data dilakukan pencampuran dua atau lebih kumpulan data, dimana kumpulan data diurutkan dan diklasifikasikan menurut *key* yang sama serta meletakkan kumpulan data tersebut bersama-sama menjadi bentuk kumpulan data tersebut yang bersama-sama membentuk data tunggal.

f. Perhitungan (*Calculating*)

Dalam pengolahan data diperlukan perhitungan yang cermat dan akurat, agar dapat diperoleh hasil akhir sebagai laporan yang mendekati kebenaran.

g. Penyimpanan data (*Storing*)

Proses ini dilakukan menginvetarisir data ke dalam suatu media penyimpanan. Dimana penyimpana data ini bertujuan untuk memudahkan dalam pencarian data kembali apabila diperlukan.

2.8 Pengertian jaringan komputer

Jaringan komputer adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung. Informasi dan data bergerak melalui kabel-kabel sehingga

memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencatak pada printer yang sama dan bersama-sama menggunakan hardware atau

software yang terhubung dengan jaringan. Tiap komputer, printer atau *periferal* yang terhubung dengan jaringan disebut node. sebuah jaringan komputer dapat memiliki dua puluhan, ribuan atau bahkan jutaan *node*.

Sebuah jaringan biasanya terdiri dari dua atau lebih komputer yang saling berhubungan diantara satu dengan yang lain, dan saling berbagi sumber daya misalnya *CDROM, printer*, pertukaran *file*, atau memungkinkan untuk saling berkomunikasi secara elektronik. Komputer yang terhubung tersebut, dimungkinkan berhubungan dengan media kabel, saluran telepon, gelombang radio, satelit, atau sinar infra merah.

1. Jenis-jenis jaringan komputer

Menurut Azhar Susanto (2008:308) ada tiga macam jenis jaringan/*network* yaitu :

a. *Local Area Network* (*LAN*)/jaringan area lokal.

Sebuah *LAN* adalah jaringan yang dibatasi oleh area yang relative kecil, umumnya dibatasi oleh area lingkungan seperti sebuah perkantoran di sebuah gedung, atau sebuah sekolah, dan biasanya tidak jauh dari sekitar 1 km2. Beberapa model konfigurasi *LAN*, satu komputer biasanya dijadikan sebuah *file server*, yang mana digunakan untuk menyimpan perangkat lunak (*software*) yang mengatur aktifitas jaringan, ataupun sebagai perangkat lunak yang dapat digunakan oleh komputer-komputer yang terhubung ke dalam *network*. Komputer-komputer yang terhubung ke dalam jaringan (*network*) itu biasanya disebut dengan *workstation*. Biasanya kemampuan *workstation* lebih di bawah dari *file server* dan mempunyai aplikasi lain di dalam harddisknya selain aplikasi untuk jaringan. Kebanyakan *LAN* menggunakan media kabel untuk menghubungkan antara satu komputer dengan komputer lain.

b. *Metropolitan Area Network* (*MAN*)/Jaringan Area Metropolitan

Sebuah *MAN* biasanya meliputi area yang lebih besar dari *LAN* dengan rentang sekitar 10-45 km, misalnya antar wilayah dalam satu propinsi. Dalam hal ini jaringan menghubungkan beberapa buah jaringan-jaringan kecil ke dalam lingkungan area yang lebih besar.

c. *Wide Area Network* (*WAN*)/ Jaringan Area Skala Besar

Wide Area Network (*WAN*) adalah jaringan yang lingkupnya biasanya sudah menggunakan sarana satelit ataupun kabel bawah laut jaringan bisa mencakup antarkota, antar propinsi, antarnegara, dan bahkan antar dunia.

2. Topologi Jaringan Komputer

Menurut Azhar Susanto (2008:306) topologi jaringan adalah bentuk/konfigurasi dari sistem jaringan. Topologi suatu jaringan didasarkan pada cara penghubung sejumlah *node* atau sentral dalam membentuk suatu sistem jaringan. Topologi jaringan yang umum dipakai adalah : *Mesh*, Bintang (*Star)*, *Bus, Tree*, dan Cincin (*Ring*).

1. Topologi Jaringan *Mesh*

Topologi jaringan ini menerapkan hubungan antar sentral secara penuh. Tingkat kerumitan jaringan sebanding dengan meningkatnya jumlah sentral yang terpasang. Dengan demikian disamping kurang ekonomis juga relatif mahal dalam pengoperasiannya.

2. Topologi Jaringan Bintang (*Star*)

Dalam topologi jaringan bintang, salah satu sentral dibuat sebagai sentral pusat. Bila dibandingkan dengan sistem *mesh*, sistem ini mempunyai tingkat kerumitan jaringan yang lebih sederhana sehingga sistem menjadi lebih ekonomis, tetapi beban yang dipikul sentral pusat cukup berat. Dengan demikian kemungkinan tingkat kerusakan atau gangguan dari sentral ini lebih besar.

3. Topologi Jaringan *Bus*

Pada topologi ini semua sentral dihubungkan secara langsung pada medium transmisi dengan konfigurasi yang disebut *Bus*. Transmisi sinyal dari suatu sentral tidak dialirkan secara bersamaan dalam dua arah. Hal ini berbeda sekali dengan yang terjadi pada topologi jaringan *mesh* atau bintang, yang pada kedua sistem tersebut dapat dilakukan komunikasi atau interkoneksi antar sentral secara bersamaan. Topologi jaringan *bus* tidak umum digunakan untuk interkoneksi antar sentral, tetapi biasanya digunakan pada sistem jaringan komputer.

4. Topologi Jaringan Pohon (*Tree*)

Topologi jaringan ini disebut juga sebagai topologi jaringan bertingkat. Topologi ini biasanya digunakan untuk interkoneksi antar sentral dengan hirarki yang berbeda. Untuk hirarki yang lebih rendah digambarkan pada lokasi yang rendah dan semakin keatas mempunyai hirarki semakin tinggi. Topologi jaringan jenis ini cocok digunakan pada sistem jaringan komputer.

5. Topologi Jaringan Cincin (*Ring*)

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.

Yang paling banyak digunakan dalam jaringan komputer adalah jaringan bertipe bus dan pohon *(tree*), hal ini karena alasan kerumitan, kemudahan instalasi dan pemeliharaan serta harga yang harus dikeluarkan. Tapi hanya jaringan bertipe pohon saja yang diakui kehandalannya karena putusnya salah satu kabel pada *client*, tidak akan mempengaruhi hubungan client yang lain.

3. Manfaat Jaringan Komputer

Satu jaringan komputer memiliki banyak manfaat, diantaranya yaitu:

1. *Resource sharing*, yaitu dapat menggunakan data dan informasi yang ada secara bersama-sama.
2. Reliabilitas tinggi, dengan komputer yang terhubung jaringan, akan mendapatkan reliabilitas yang tinggi dengan memiliki sumber-sumber alternatif persediaan.
3. Ekonomis, dapat memanfaatkan periferal yang terbatas secara bersama-sama.

4. Sistem Client Server

Sistem *Client* dan *Serve*r 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.

2.9 Perangkat Lunak Pendukung

Perangkat lunak (*software)* adalah peralatan untuk menunjang untuk kerja dari perangkat keras (*hardware*). Perangkat lunak memberikan instruksi-instruksi yang dapat ditanggapi dan dimengerti oleh perangkat keras komputer. Perangkat lunak komputer (*software*) dapat dikelompokkan ke dalam tiga kelompok yaitu :

a. Sistem Operasi (*Operating Sistem*)

Merupakan program yang berfungsi untuk mengendalikan dan mengkoordinasikan kegiatan sistem komputer.

Contoh : Windows 9x, NT, Me, Xp, Vista, 7, 8, Linux, dan lain-lain.

b. Perangkat lunak aplikasi (*Aplication Software*)

Merupakan program yang ditulis dan dierjemahkan oleh bahasa pemrograman untuk keperluan aplikasi tertentu.

c. Bahasa Pemrograman (*Programming Language*)

Merupakan program yang digunakan untuk menterjemahkan suatu bahasa pemrograman ke dalam bahasa mesin, agar dapat dimengerti oleh komputer.

Contoh : Microsoft Word, Excel, Corel draw, dan lain-lain.

1. Java

Dikutip dari (http://id.wikipedia.org/wiki/java)

Bahasa Java diciptakan oleh James Gosling dan Patrick Naughton dalam suatu proyek dari SUN Microsystem. Pada mulanya ingin diberi nama OAK dari pohon yang terdapat pada kantor James Gosling, tetapi kata OAK telah ada pada sun microsystem, maka diberi nama java (terinspirasi minum kopi). Browser pertama yang dapat membaca java adalah Hot Java. Setelah Browser Netscape dari perusahaan Netscape Navigator dan IE dari perusahaan Microsoft Inc dapat membaca script java, maka bahasa java semakin popular. *Vendor-vendor* lain, seperti IBM, Oracle, Symantec, Inprise (Borland Inc), dan perusahaan-perusahaan mobile, seperti Nokia, Sony Ericsson, Motorola, dan Samsung juga mengadopsi teknologi java.

2. MySQL

Dikutip dari (http://id.wikipedia.org/wiki/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.

3. XAMPP

Dikutip dari (http://id.wikipedia.org/wiki/Xampp)

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost)*, yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License*, merupakan web server yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat men-*download* langsung dari web resminya.

4. NetBeans

Dikutip dari (http://www.hi-techmall.org/geek/blog/sejarah-fungsi-pengertian-netbeans)

NetBeans mengacu pada kedua platform kerangka untuk aplikasi desktop Java, dan sebuah lingkungan pengembangan terpadu (IDE) untuk pengembangan dengan Java , JavaScript , PHP , Python , Ruby , Groovy , C , C + + , Scala , Clojure , dan lain-lain.

NetBeans IDE ditulis dalam Java dan berjalan di mana-mana mana JVM diinstal, termasuk Windows, Mac OS, Linux, dan Solaris.Sebuah JDK diperlukan untuk

pengembangan fungsionalitas Jawa, tetapi tidak diperlukan untuk pembangunan di bahasa pemrograman lain.

Platform NetBeans memungkinkan aplikasi untuk dikembangkan dari satu set modular komponen software yang disebut modul. Aplikasi berbasis platform NetBeans (termasuk IDE NetBeans) dapat diperpanjang oleh pengembang pihak ketiga .

Awal sejarah

NetBeans dimulai pada tahun 1996 sebagai Xelfi (kata bermain pada Delphi), Java IDE proyek mahasiswa di bawah bimbingan Fakultas Matematika dan Fisika di Charles University di Praha . Pada tahun 1997 Staněk Romawi membentuk perusahaan sekitar proyek tersebut dan menghasilkan versi komersial NetBeans IDE hingga kemudian dibeli oleh Sun Microsystems pada tahun 1999. Komunitas NetBeans sejak terus tumbuh, berkat individu dan perusahaan yang menggunakan dan berkontribusi dalam proyek ini.

versi Lancar

NetBeans IDE 6.0 memperkenalkan dukungan untuk mengembangkan modul IDE dan aplikasi klien kaya berdasarkan platform NetBeans, Java Swing GUI builder (sebelumnya dikenal sebagai "Proyek Matisse"), meningkatkan CVS dukungan, WebLogic 9 dan JBoss 4 dukungan, dan perangkat tambahan banyak editor. NetBeans 6 is available in official repositories of major Linux distributions. NetBeans 6 tersedia dalam repositori resmi dari distribusi Linux utama.

Selain itu, NetBeans Enterprise Pack mendukung pengembangan aplikasi Java EE 5 perusahaan, termasuk SOA alat desain visual, skema XML tools, web orkestrasi layanan (untuk BPEL), dan UML modeling. The NetBeans IDE Bundle for C/C++ supports C/C++ development. The NetBeans IDE Bundle untuk C / C + + mendukung C / C + + pembangunan.

Hosting pengembang sumber terbuka proyek di kenai.com tambahan manfaat dari instant messaging dan pelacakan masalah integrasi dan navigasi kanan dalam IDE, dukungan untuk pengembangan aplikasi web dengan PHP 5.3 dan kerangka Symfony, dan kode selesai diperbaiki, layout, petunjuk dan navigasi dalam proyek JavaFX.

dirilis pada bulan Juni 2010, menambahkan dukungan untuk OSGi , Spring Framework 3.0, Java EE injeksi ketergantungan (JSR-299), Zend Framework untuk PHP , dan navigasi kode lebih mudah (seperti "Apakah / ditimpa Penerapan" penjelasan), format , petunjuk, dan refactoring di beberapa bahasa.

2.10 Akademik

Menurut Kamus Besar Bahasa Indonesia edisi kedua, akademi adalah lembaga pendidikan tinggi kurang lebih tiga tahun lamanya yang mendidik tenaga profesi. Sedangkan akademik adalah ”bersifat akademik”.

2.11 Sistem Informasi Akademik

Sistem informasi pengolahan data akademik adalah sistem informasi yang memberikan layanan informasi berupa data dalam hal yang berhubungan dengan

akademik. Dimana dalam hal ini pelayanan yang diberikan yaitu seperti : pengolahan nilai siswa, proses pembayaran iuran wajib sekolah (SPP), dan informasi absensi/kehadiran siswa dengan penyampaian informasi berupa : informasi nilai siswa, informasi jadwal pembayaran iuran wajib sekolah (SPP), dan informasi absensi/kehadiran siswa.