BAB II

LANDASAN TEORI

2.1 Pengertian Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerja sama membentuk suatu kesatuan, suatu sistem betapapun kecilnya selalu mengandung komponen-komponen, dimana komponen-komponen dapat berupa subsistem-subsistem atau bagian dari sistem yang mempunyai sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan

Beberapa para ahli mengemukakan pengertian sistem seperti dibawah ini;

Menurut [Jog 2] *:*

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

Menurut [Ray 5] *:*

" Sistem adalah himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk suatu kesatuan yang utuh dan terpadu ".

Dari definisi diatas dapat diambil kesimpulan bahwa suatu sistem merupakan kumpulan dari komponen-komponen yang saling terstruktur dan terpadu serta saling bekerja sama untuk melakukan fungsi dari sistem sehingga adanya ketercapaian tujuan dari sistem.

Menyangkut karateristik sistem ini, penulis [Jog 2] berpendapat bahwa sistem mempunyai karakteristik atau sifat-sifat tertentu yaitu sebagai berikut:

1. Komponen sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen–komponen sistem atau elemen–elemen 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 dengan *supra system*.

2. Batas sistem

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 satu kesatuan. Batas suatu sistem menunjukan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan luar sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan dari sistem

4. Penghubung sistem

Penghubung (*Interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber–sumber daya mengalir dari satu subsistem ke subsistem lainnya. Keluaran (*Output*) dari satu subsitem akan menjadi masukan (*input*) untuk subsitem yang lainnya dengan memlalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

5. Masukan Sistem

Masukan adalah bahan atau energi yang dimasukkan kedalam sistem. Masukan dapat berupa perawatan (*maintance input*) dan masukan sinyal (*signal input*). *Maintance input* adalah energi yang dimasukan supaya sistem tersebut dapat beroperasi. *signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem

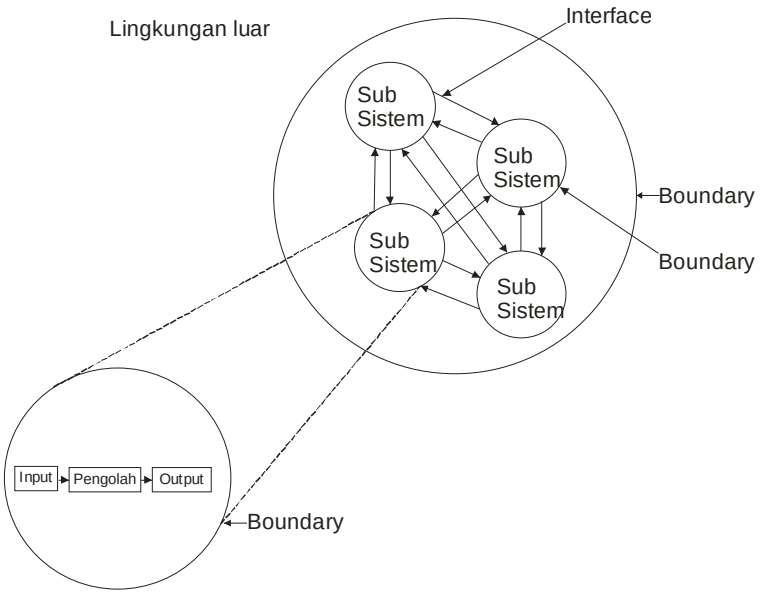
Keluaran (*output*) adalah hasil dari bahan atau energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lainnya atau kepada *supra* sistem.

7. Pengolah Sistem

Pengolah sistem adalah mesin atau mekanisme yang digunakan untuk mengubah masukan menjadi keluaran. Pengolah memiliki peranan yang penting, karena disinilah proses perubahan dan pendayagunaan masukan terjadi sehingga menghasilkan keluaran yang sesuai dengan tujuan sistem.

8. Sasaran dan Tujuan

Suatu sistem pasti mempunyai tujuan *(goal)* atau sasaran (*objective*). Tujuan merupakan hal akhir yang ingin dicapai oleh suatu sistem, sedangkan sasaran merupakan hal-hal yang menjadi objek dan titik fokus untuk meraih tujuan. Suatu sistem bisa dikatakan berhasil menjalankan fungsinya bila berhasil mencapai sasaran dan tujuan dari sistem tersebut.



Gambar 2.1 *Karateristik suatu sistem*
(Sumber :[Jog 2])

2.2 Pengertian Informasi

Menurut [Jog 2] :

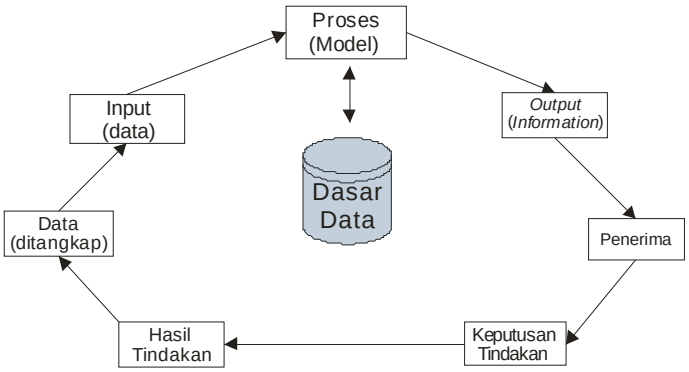
"Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya".

Menurut [Ray 5] :

“ Informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang ”

Jadi pengertian 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 – laporan yang diperlukan.

Data yang diolah melalui suatu model menjadi informasi, penerima kemudian memberi informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus.



Gambar 2.2 *Siklus Informasi*
(Sumber : [Jog 2])

Kualitas dari sistem informasi yang harus dihasilkan harus akurat, tepat waktunya, relevan. Dan yang menentukan nilai dari informasi adalah manfaat dan biaya untuk mendapatkan.

2.3 Pengertian Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen dalam pengambilan keputusan. Informasi itu sendiri dapat diperoleh dari sistem informasi

Menurut Robert A. dan K. Roscue, [Jog 2], Sistem informasi dapat didefinisikan sebagai berikut:

"Sistem Infrormasi 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"

Menurut [Ray 5] :

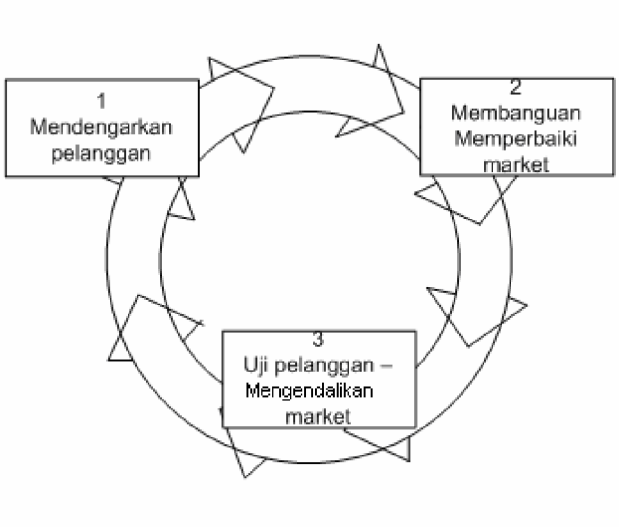
"Sistem Informasi merupakan sistem yang mempunyai kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi "

2.4 Metode pengembangan Sistem

Metode pengembangan sistem adalah menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang telah berjalan. Dalam pengembangan sistem ini metode

pengembangan menggunakan metode *Prototype*, karena metode ini memberikan ide bagi pengembang maupun pelanggan tentang cara sistem akan berfungsi dan yang akan dikembangakan.

Metode *prototype* terdapat 3 *(Tiga)* tahapan untuk dapat mengembangkan suatu perangkat lunak seperti gambar di bawah ini :



Gambar 2.3 *Model Prototipe*
(Sumber : [Rog 1])

Tahapan tersebut antara lain:

1. Pada tahap pertama ini, pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan dari perangkat lunak, mengidentifikasikan segala kebutuhan yang diketahui.
2. Pada tahap kedua, pengembang melakukan perancangan yang berfokus pada penyajian dari apsek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contoh pendekatan input dan format output).
3. Pada tahap ketiga, ini pelanggan atau pemakai mengevaluasi dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak pada tahap ini dimungkinkan perangkat lunak untuk di setting ulang untuk memenuhi

kebutuhan pelanggan, dan pada saat yang sama memungkinkan pengembang untuk secara lebih baik memahami apa yang harus dilakukannya.

Alasan penyusun menggunakan metode prototipe ini, karena metode prototipe ini memiliki kelebihan-kelebihan sebagai berikut:

1. Metode ini cukup efektif sebagai paradigma dalam rekayasa perangkat lunak. Karena mendapatkan kebutuhan dan aturan yang jelas yang disetujui oleh pelanggan, dalam pembuatan perangkat lunak bisa dilakukan secara cepat dan memungkinkan untuk merubah kembali perangkat lunak agar sesuai dengan kebutuhan pelanggan. Walaupun pada umumnya *prototype* akan dihilangkan dan dibuat perangkat yang sebenarnya.
2. Ketika diperlihatkan versi pertama atau tampilan input dan format output dalam media kertas, pelanggan bisa langsung mengetahui apakah perangkat lunak tersebut bisa memenuhi kebutuhannya atau tidak.

2.5 Pemograman Beroreantasi Objek (UML)

Menurut [Fow 4], *Unifieed Modelling Language* (UML) adalah keluarga notasi grafis yang di dukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemograman beroreantasi objek (OO) pemodelan ini merupakan bahasa standar yang digunakan untuk visualisasi, merancang serta mendokumentasikan perangkat lunak, khususnya sistem yang dibangun menggunakan pemograman berorientasi objek (OO).

Adapun tahapan yang terdapat dalam pemodelan yang merupakan berbentuk diagram adalah sebagai berikut :

Tabel 2.1 *Jenis Diagram Resmi UML*
(Sumber : [Fow 4])

Diagram	Kegunaan
Activity	*Behavior* prosedural dan paralel
Class	*Class*, fitur, dan hubungan - hubungan
Communication	Interaksi antar objek; penekanan pada jalur
Component	Struktur dan dan koneksi komponen
Composite structure	Dekomposisi *runtime* sebuah class
Deployment	Pemindahan artifak ke node
Interaction overview	Campuran *sequence* dan *activity* diagram
Object	Contoh konfigurasi dari contoh–contoh
Package	Struktur hirarki *compile-time*
Sequence	Interaksi antar objek; penekanan pada *sequence*
State machine	Bagaimana *even* mengubah objek selama aktif
Timing	Interaksi antar objek; penekanan pada *timing*
Use Case	Bagaimana pengguna berinteraksi dengan sebuah sistem

2.5.1 *Object Management Group (OMG)*

Menurut [Suh 3], *Object Management Group, Inc. (OMG)*. Adalah sebuah organisasi internasional yang dibentuk pada tahun 1989, didukung lebih dari 800 angggota, terdiri dari perusahaan sistem informasi, *software developer* dan para *user* sistem komputer. OMG mempromosikan teori dan praktek-praktek *object oriented technology* dalam rekayasa *software*. Organisasi ini salah satunya bertugas membuat spesifikasi “manajemen objek” untuk menetapkan kerangka bersama dalam rekayasa *software.* Spesifikasi tersebut dibuat dengan tujuan

utama untuk mendapat *reusability, portabillity,* dan *interoperabillity sofware* yang berdasarkan OO, dalam lingkungan yang heterogen, dapat dioperasikan dalam semua *platform hardware* dan sistem operasi.

Sasaran OMG adalah membantu perkembangan *object oriented technology* dan mengalahkan dengan mendirikan *Object Manajement Architecture (OMA)*. OMA menentukan infastruktur konseptual yang didasarkan pada seluruh spesifikasi yang dikeluarkan OMG.

OMG kemudian mengeluarkan UML, dimana dengan adanya UML ini diharapkan dapat mengurangi kekacauan dalam bahasa pemodelan yang selama ini terjadi dalam lingkungan industri. UML diharapkan juga dapat menjawab masalah penotasian dan mekanisme tukar menukar model yang terjadi selama ini.

2.5.2 *Artifact* UML

Menurut [Suh 3], untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut :

1. *use-case diagram*

   Menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diliuar sistem (actor). Diagram ini menunjukkkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar

2. *class diagram*

   Membantu dalam visualisasistruktur kelas-kelas dari suatu sistem dan merupakan tipe diagramyang paling banyak dipakai.

3. behavior diagram :

3.1 statechart diagram

Memperlihatkan urutan keadaan sesaat yang dilalui sebuah objek, kehjadian yang menyebabkan sebuah transisidari suatu state atau aktvitas kepada yang lainnya, dan aksi yang menyebabkan perubahan satu state atau aktivitas.

3.2 activity diagram

Memodelkan alur kerja sebuah proses bisnis dan urutan aktivitas dalam suatu proses.

3.3 interaction diagram

3.3.1 Squence diagram

Menjelaskan interaksi objek yang disusun dalam suatu urutan waktu.

3.3.2 Collaboration diagram

Melihat pada interaksi dan hubungan tersetruktur antar objek

4. implementation diagram :

4.1 component diagram

Menggambarkan alokasi semua kelas dan objek kedalam komponen-komponen dalam desain fisik sistem *software.*

4.2 deployment diagram

Diagram ini memperlihatkan pemetaan *software* kepada *hardware*.

Diagram–diagram tersebut diberi nama berdasarkan sudut pandang yang berbeda–beda terhadap sistem dalam proses analisis atau rekayasa

Dibuatnya berbagai jenis diagram di atas karena :

1. Setiap sistem yang kompleks selalu paling baik jika di dekati melalui himpunan berbagai sudut pandang yang kecil yang satu sama lain hampir saling bebas (*independent*). Sudut pandang tunggal senantiasa tidak mencukupi untuk melihat sistem yang besar dan kompleks.
2. Diagram yang berbeda-beda tersebut dapat menyatakan tingkatan yang berbeda-beda dalam proses dalam proses rekayasa.
3. Diagram-diagram tersebut dibuat agar model yang dibuat semakin mendekati realitas.

Diagram–diagram ini ditambah dengan kemampuan dokumentasi merupakan *artifacs* utama UML.

2.5.3 Semantik dalam UML

Menurut [Suh 3], OMG telah menetapkan semantik (makna istilah) semua notasi UML dalam model struktur dan model *behavioral*. Model *structural* (disebut juga model statis) menekankan struktur objek dalam sebuah sistem, menyangkut kelas–kelas, *interface* dan hubungan antar komponen. Model *behavioral* (juga disebut sebagai model dinamis) menekankan prilaku objek dalam sebuah sistem, termasuk metode, interaksi, kolaborasi, dan *state history*.

2.5.4 Faktor yang mendorong dibuatnya UML

Menurut [Suh 3], Faktor penting yang mendorong dibuatnya UML ini dikarenakan sebagai berikut :

1. Pentingnya model.

a. Membangun model untuk suatu sistem *software* (terlebih *software* untuk suatu organisasi bisnis) sangat bergantung pada konstruksinya atau kemudahan dalam memperbaikinya. Oleh karena itu, membuat model sangat penting sebagaimana pentingnya kita memiliki cetak biru untuk suatu bangunan yang besar.

b. Model yang bagus sangat penting untuk menghasilkan komunikasi yang baik antar anggota tim dan untuk meyakinkan sempurnanya arsitektur sistem yang dibangun.

c. Jika kita membangun suatu model dari suatu sistem yang kompleks, tidak mungkin kita dapat memahaminya secara keseluruhan. Dengan meningkatnya komplektisitas sistem, visualisasi dan pemodelan menjadi sangat penting. UML dibuat untuk merespon kebutuhan tersebut.

2. Kecenderungan dunia industri terhadap *software.*

a Sebagai suatu nilai yang strategis bagi pasar *software* adalah dengan meningkatnya kebutuhan dunia industri untuk memiliki teknik otomatisasi dengan *software.* Oleh karena itu, penting sekali adanya teknik rekayasa software yang dapat meningkatkan kualitas dan mengurangi biaya dan waktu. Termasuk dalam teknik rekayasa *software* adalah teknik *visual programming*, juga pembuatan *framework* dan pola.

b Kompleksitas dunia industri semakin meningkat karena bertambah luasnya ruang lingkup dan tahapan proses. Satu motivasi kunci bagi para pembangunan UML adalah untuk membuat suatu himpunan simantik dan notasi yang mampu menangani kerumitan arsitektural dalam semua ruang lingkup.

3. Terjadinya konvergensi metode dan *tool* pemodelan.

Sebelum adanya UML, terdapat ketidakjelasan bahasa pemodelan apa yang paling unggul. Para *user* harus memilih diantara bahasa pemodelan dan *tool* (*software*) pemodelan yang banyak dan mirip. UML dibuat untuk membuat integrasi baru dalam bahasa pemodelan antar *tool* dan "proses".

2.5.5 Tujuan UML

Menurut [Suh 3], tujuan utama UML diantaranya untuk :

1. memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemograman dan proses rekayasa.
3. Menyatukan praktek–praktek terbaik yang terdapat dalam pemodelan.

2.5.6 Cakupan UML

Menurut [Suh 3] :

Pertama, UML menggabungkan konsep Booch, OMT, dan OOSE, sehingga UML merupakan suatu bahasa pemodelan tunggal yang umum dan digunakan secara luas oleh para *user* ketiga model tersebut dan bahkan para *user* metode lainnya.

Kedua, UML menekankan pada apa yang dapat dikerjakan dengan metode–metode tersebut.

Ketiga, UML berfokus pada suatu bahasa pemodelan standar, bukan pada proses standar. Mekipun UML harus diaplikasikan dalam konteks sebuah proses, dari pengalaman bahwa organisasi dan masalah yang berbeda juga.

UML tidak mencakup :

1. Bahasa pemograman.

UML adalah bahasa pemodelan visual, bukan dimaksudkan untuk menjadi suatu bahasa pemograman visual, tetapi UML memberikan arah untuk bergerak kearah kode.

2. *Tool* (*software* aplikasi).

Membuat standar sebuah bahasa diperlukan oleh *tool–tool* dan proses. UML mendefinisikan semantikdan notasi, bukan sebuah *tool*. Contoh *tool* yang menggunakan UML sebagai bahasanya adalah *National Rose* dan *Enterprise Architect.*

3. Proses rekayasa.

UML digunakan sebagai bahasa dalam proyek dengan proses yang berbeda–beda. UML bebas dari proses dan mendefinisikan sebuah proses standar bukan tujuan UML atau RFP dari OMG.

2.5.7 Perbandingan UML dengan bahasa pemodelan lainnya

Menurut [Suh 3], UML merupakan gabungan dai Booch, OMT, dan OOSE dengan membuang elemen-elemen yang tidak praktis, ditambah elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu, sehingga UML lebih ekspresif dan seragam dari pada metode lainnya.

2.6 Arsitektur Aplikasi

Arsitektur aplikasi merupakan sebuah bentuk aplikasi yang akan dibangun dalam penelitian ini. Di dunia ini banyak sekali sistem jaringan yang dapat di implementasikan secara global. Di bawah ini akan dijelaskan secara umum mengenai jaringan komputer.

2.6.1 Pengertian Jaringan Komputer

Menurut [Onno 7], Jaringan Komputer dapat diartikan sebagai suatu himpunan interkoneksi sejumlah komputer otonom. Dua buah komputer dikatakan membentuk suatu network bila keduanya dapat saling bertukar informasi. Pembatasan istilah otonom disini adalah untuk membedakan dengan sistem master/slave. Bila sebuah komputer dapat membuat komputer lainnya aktif atau

tidak aktif dan mengontrolnya, maka komputer komputer tersebut tidak otonom. Sebuah sistem dengan unit pengendali (*control unit*) dan sejumlah komputer lain yang merupakan slave bukanlah suatu jaringan; komputer besar dengan remote printer dan terminalpun bukanlah suatu jaringan.

2.6.2 Manfaat Jaringan Komputer

1. *Resource sharing*

Dapat menggunakan sumber daya yang ada secara bersama-sama. Misal seorang pengguna yang berada 100 km jauhnya dari suatu data, tidak mendapatkan kesulitan dalam menggunakan data tersebut, seolah-olah data tersebut berada didekatnya. Hal ini sering diartikan bahwa jaringan komputer mengatasi masalah jarak.

2. Reabilitas tinggi

Dengan jaringan komputer kita akan mendapatkan reliabilitas yang tinggi dengan memiliki sumber-sumber alternatif persediaan. Misalnya, semua *file* dapat disimpan atau *dicopy* ke dua, tiga atau lebih komputer yang terkoneksi ke jaringan. Sehingga bila salah satu mesin rusak, maka salinan di mesin yang lain bisa digunakan.

3. Menghemat uang

Komputer berukuran kecil mempunyai *rasio* harga atau kinerja yang lebih baik dibandingkan dengan komputer yang besar. Komputer besar seperti *mainframe* memiliki kecapatan kira-kira sepuluh kali lipat kecepatan komputer kecil atau pribadi. Akan tetap, harga *mainframe* seribu kali lebih mahal dari komputer pribadi. *rasio* harga atau kinerja dan kecepatan inilah

membuat para perancang sistem untuk membangun sistem yang terdiri dari komputer-komputer pribadi.

2.6.3 Jenis-Jenis Jaringan Komputer

Secara umum jaringan komputer dibagi atas lima jenis, yaitu ;

1. *Local Area Network* (*LAN*).

Local Area Network (*LAN*), merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik-pabrik untuk memakai bersama sumber daya (*resouce*, misalnya *printer*) dan saling bertukar informasi.

2. *Metropolitan Area Network* (MAN).

Metropolitan Area Network (MAN), pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum. MAN mampu menunjang data dan suara, bahkan dapat berhubungan dengan jaringan televisi kabel.

3. *Wide Area Network* (WAN).

Wide Area Network (WAN), jangkauannya mencakup daerah geografis yang luas, seringkali mencakup sebuah negara bahkan benua. *WAN* terdiri dari kumpulan mesin- mesin yang bertujuan untuk menjalankan program-program (aplikasi) pemakai.

4. *Internet*.

Sebenarnya terdapat banyak jaringan didunia ini, seringkali menggunakan perangkat keras dan perangkat lunak yang berbeda-beda. Orang yang terhubung ke jaringan sering berharap untuk bisa berkomunikasi dengan orang lain yang terhubung ke jaringan lainnya. Keinginan seperti ini memerlukan hubungan antar jaringan yang seringkali tidak kampatibel dan berbeda. Biasanya untuk melakukan hal ini diperlukan sebuah mesin yang disebut *gateway* guna melakukan hubungan dan melaksanakan terjemahan yang diperlukan, baik perangkat keras maupun perangkat lunaknya. Kumpulan jaringan yang terinterkoneksi inilah yang disebut dengan internet.

5. Jaringan Tanpa Kabel.

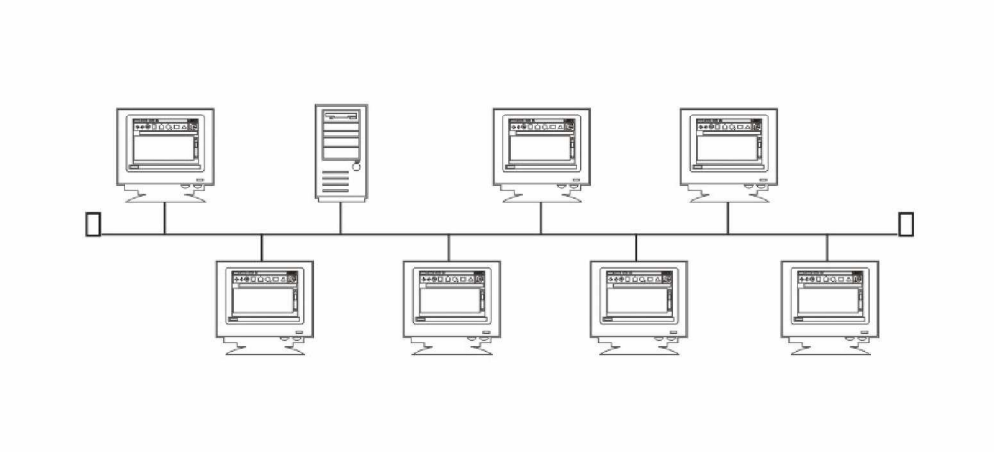
Jaringan tanpa kabel merupakan suatu solusi terhadap komukasi yang tidak bisa dilakukan dengan jaringan yang menggunakan kabel. Misalnya orang yang ingin mendapat informasi atau melakukan komunikasi walaupun sedang berada diatas mobil atau pesawat terbang, maka mutlak jaringan tanpa kabel diperlukan karena koneksi kabel tidaklah mungkin dibuat di dalam mobil atau pesawat. Saat ini jaringan tanpa kabel sudah marak digunakan dengan memanfaatkan jasa satelit dan mampu memberikan kecepatan akses yang lebih cepat dibandingkan dengan jaringan yang menggunakan kabel.

2.6.4 Topologi Jaringan Komputer

Menurut [Bud 6], mengatakan, topologi secara fisik dari suatu jaringan lokal adalah merujuk kepada konfigurasi kabel, komputer, dan perangkat lainnya. Adapun tipe– tipe topologi jaringan komputer adalah sebagai berikut :

1. *Linier Bus* (garis Lurus).

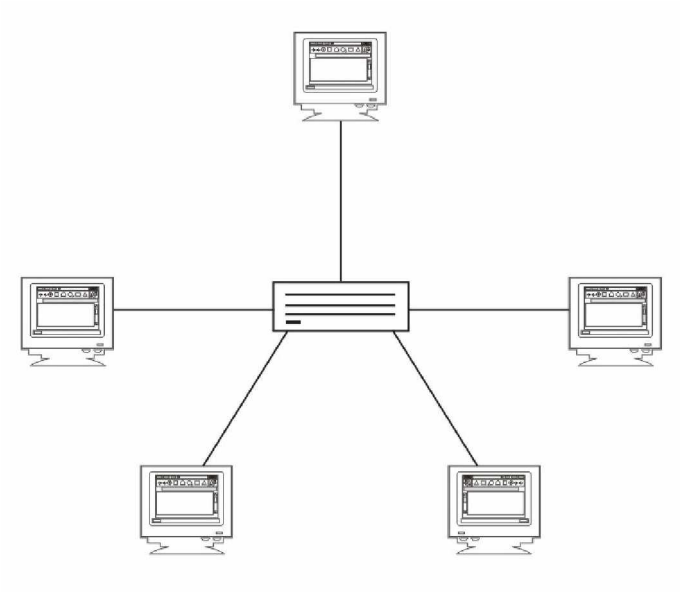
Topologi *Linier* Bus (Garis Lurus) terdiri dari satu jalur kabel utama dimana pada masing–masing ujungnya diberikan sebuah terminator. Semua *nodes* pada jaringan (*file server, workstation*, dan perangkat lainnya) terkoneksi sebuah kabel utama (*backbone*).



Gambar 2.4 *Topologi Linier Bus (Garis Lurus)*
(Sumber : [Bud 6])

2. *Star* (Bintang).

Topologi model ini dirancang, yang mana setiap *nodes* (*file server, workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *connector*. Data yang dikirim ke jaringan lokal akan melewati *connector* sebelum melanjutkan ke tempat tujuannya, *connector* akan mengatur dan mengendalikan keseluruhan fungsi jaringan, dan juga bertindak sebagai *repeater* (penguat aliran data). Konfigurasi pada jaringan model ini menggunakan kabel *Twisted Pair*, dan dapat digunakan pula kabel *Coaxial* atau kabel *fibre optic.*



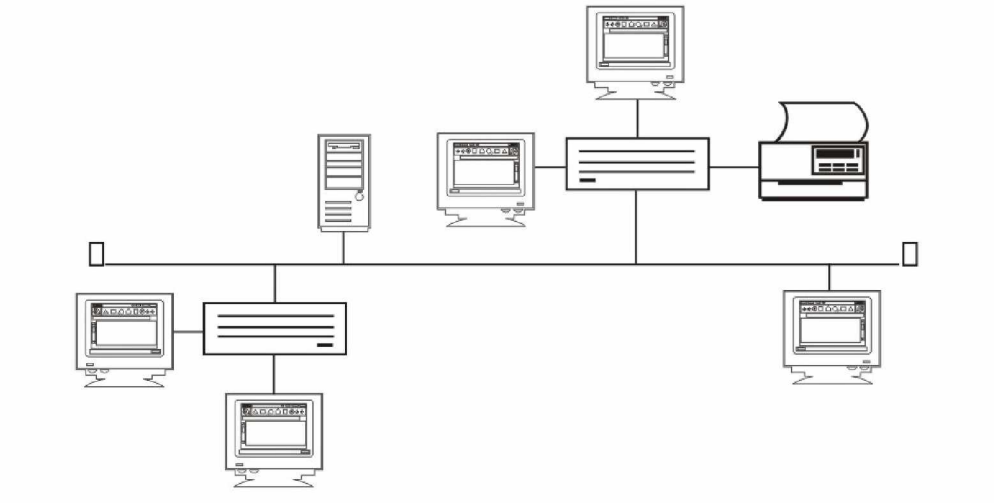
Gambar 2.5 *Topologi Star (Bintang)*
(Sumber : [Bud 6])

3. *Ring* (Cincin).

Topologi *Ring* (Cincin) menggunakan teknik konfigurasi yang sama dengan topologi *star* tetapi pada topologi ini terlihat pada jalur media transmisi menyerupai suatu lingkaran tertutup menyerupai cincin (lingkaran), sehingga diberi nama topologi bintang dalam lingkaran.

4. *Tree* (Pohon).

Topologi model ini merupakan perpaduan antara topologi *linear bus* dan *star*, yang mana terdiri dari kelompok-kelompok dari *workstation* dengan konfigurasi star yang terkoneksi ke kabel utama yang menggunakan topologi *linear bus*. Topologi ini memungkinkan untuk pengembangan jaringan yang telah ada, dan memungkinkan untuk mengkonfigurasi jaringan sesuai dengan kebutuhan.

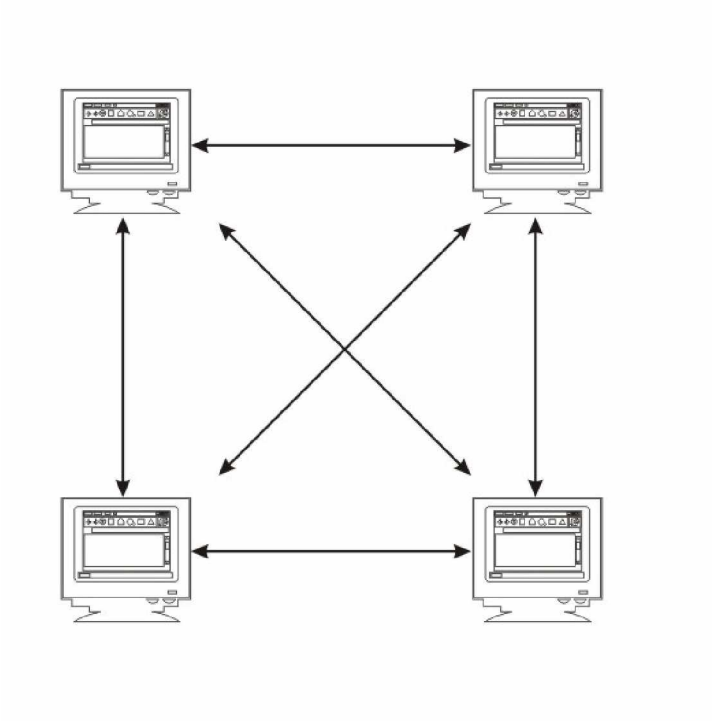


Gambar 2.6 *Tree (Pohon)*
(Sumber : [Bud 6])

2.6.5 Model Hubungan pada LAN

1. *Peer-to-Peer*

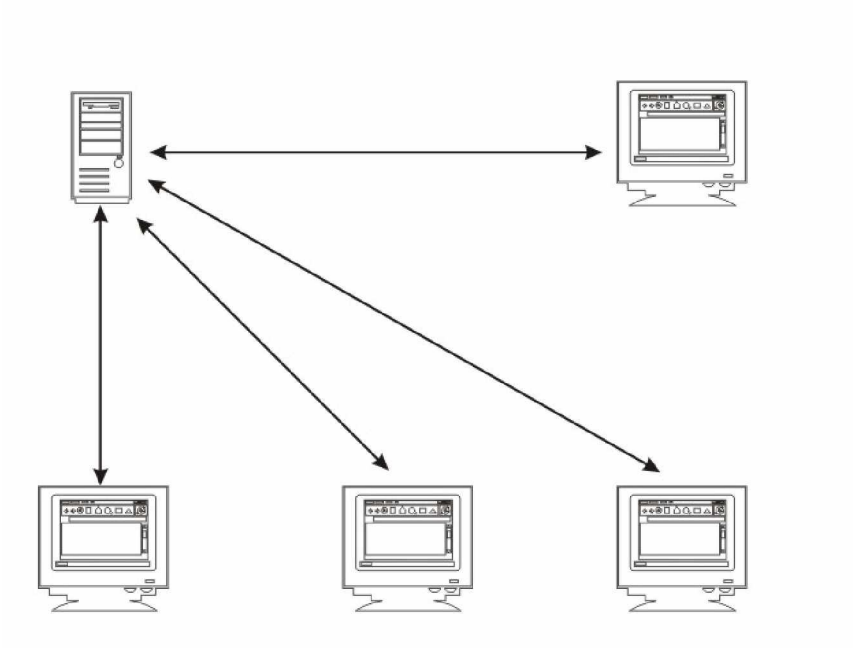
Model hubungan *peer to peer* memungkinkan *user* membagi sumber dayanya yang ada di komputernya baik itu berupa *file*, layanan *printer* dan lain-lain serta mengakses sumber daya yang terdapat pada komputer lain. Namun model ini tidak mempunyai *file server* atau sumber daya yang terpusat. Didalam model hubungan *peer to peer* ini, seluruh komputer adalah sama, yang mana mempunyai kemampuan yang sama untuk memakai sumber daya yang tersedia di dalam jaringan model ini di desain untuk jaringan bersekala kecil dan menengah.



Gambar 2.7 *Model Hubungan Peer-to-peer*
(Sumber : [Bud 6])

2. *Client/Server.*

Model hubungan *Client/Server* memungkinkan jaringan untuk mensentralisasikan fungsi dan aplikasi kepada satu atau dua *dedicated file server*. Sebuah *file server* menjadi jantung dari keseluruhan sistem, memungkinkan untuk mengakses sumber daya, dan menyediakan keamanan. *Workstation* yang berdiri sendiri dapat mengambil sumber daya yang ada pada *file server*. Model hubungan menyediakan mekanisme untuk mengintegrasikan seluruh komponen yang ada di jaringan dan memungkinkan banyak pengguna secara bersama-sama memakai sumber daya pada *file server*.



Gambar 2.8 *Model Hubungan Client Server*
(Sumber : [Bud 6])

2.7 Perangkat Lunak Pendukung

Sistem Informasi Pemeriksaan pada Instalasi Laboratorium dikembangkan dan diimplementasikan menggunakan Bahasa Pemrograman *Borland Delphi* 7.0 dengan Sistem Operasi *Microsoft Windows Server* 2003. Serta menggunakan *Database Microsoft* SQL *Server* dan *Rational Rose* 2002.05.00 sebagai program bantuan untuk membuat UML.

2.7.1 Sekilas Tentang *Borland Delphi* 7.0

Menurut [Ted 8], 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). *Delphi* juga merupakan sebuah adalah kompiler atau penterjemah bahasa *Delphi* (awalnya dari *Pascal*) yang merupakan bahasa tingkat tinggi sekelas dengan *Basic*, C.

Bahasa Pemrograman di *Delphi* disebut bahasa *prosedural* artinya bahasa atau sintaknya mengikuti urutan tertentu atau *prosedur*. Ada jenis pemrograman non-prosedural seperti pemrograman untuk kecerdasan buatan seperti bahasa Prolog. *Delphi* termasuk keluarga *Visual* sekelas *Visual Basic*, *Visual* C, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Pemrogram tinggal memilih objek apa yang ingin dimasukkan kedalam *Form* atau *Window*, lalu tingkah laku objek tersebut saat menerima *event* atau aksi tinggal dibuat programnya.

Delphi merupakan bahasa berorentasi objek, artinya nama objek, properti dan *methode*/ *procedure* dikemas menjadi satu kemasan (*encapsulate*). Semenjak Delphi 6, Delphi telah di lengkapi dengan sejumlah komponen yang tergolong sebagai *dbExpress*, yang memungkinkan koneksi ke *Microsoft* SQL *Server* 2000 ataupun *Oracle* dilakukan dengan mudah, sehingga *Delphi* dapat digunakan sebagai aplikasi *front-end* yang berhubungan dengan *database server*. Pada versi 7.0, komponen yang tergolong *dbExpress* sedikit berubah. Ada beberapa hal yang menjadi daya tarik dari *Delphi* sendiri, *Delphi* langsung membuat *Application* sendiri (.EXE) dan *listing* bisa dibuka dalam *notepad* (.*pas).

2.7.2 Sekilas *Microsoft SQL Server 2000*

Menurut [Ted 9], mengatakan, *Microsoft* SQL *Server* 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 *system 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*, *Microsoft* SQL *Server* 2000 dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh single user, kecepatan *query Microsoft* SQL Server 2000 bisa sepuluh kali lebih cepat *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase.*

Microsoft SQL *Server* 2000 adalah satu dari sekian banyak system *database* merupakan terobosan solusi yang tepat dalam aplikasi *database*. Didukung oleh ribuan bahkan jutaan komunitas pengguna di *internet* yang siap membantu. Selain itu juga tersedia *mailing list* dan *homepage* khusus yang memberikan *tutorial* serta dokumentasi lengkap

Adapun keistimewaan *Microsoft* SQL *Server* 2000 adalah sebagai berikut :

1. *Multi User*.

Microsoft SQL *Server* 2000 dapat digunakan oleh beberapa *user* dalam waktu bersamaan tanpa mengalami masalah, hal ini yang memungkinkan *database Microsoft* SQL *Server* 2000 dapat diakss *client* secara bersamaan.

2. *Performance Tuning*.

Microsoft SQL *Server* 2000 memiliki kecepatan yang menajubkan dalam menangani *query* sederhana, dapat memproses lebih banyak SQL dalam persatuan waktu.

3. *Colom Type*.

Microsoft SQL Server 2000 memiliki tipe kolom yang kompleks, seperti; *integer, float, double, char,* n*varchar, datetime, image, datetime* serta yang lainnya.

4. *Command* dan *Function*

Microsoft SQL *Server* 2000 memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* ,FROM, *WHERE, UPDATE, DELETE* dalam *query*.

5. *Security.*

Microsoft SQL *Server* 2000 memiliki beberapa lapisan sekuriti seperti *subnetmask,* nama *host* dan izin *akses user* dengan sistem perizinan yang mendetail.

6. *Connectivity*.

Microsoft SQL *Server* 2000 dapat melakukan koneksi dengan *client* menggunakan *protocol* TCP/IP, atau *Named Pipes* (NT).

7. *Localisation.*

Microsoft SQL *Server* 2000 dapat mendeteksi pesan kesalahan (*error code*).

8. *Interface.*

Microsoft SQL *Server* 2000 memiliki *interface* (antarmuka) terhadap berbagi aplikasi dan bahasa pemrograman dengan menggunakan API (*Application Progrmming Interface*).

9. *Clients dan Tools*.

Microsoft SQL Server 2000 dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *databse*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

2.8 Pengertian Pemeriksaan, Instalasi dan Laboratorium

2.8.1 Pengertian Pemeriksaan

Menurut [Poer 10] :

"Perbuatan (hasil dsb) memeriksa"

Menurut [Mar 11] :

"*Proses, cara, perbuatan memeriksa"*

2.8.2 Pengertian Instalasi

Menurut [Poer 10] :

"Hal memasang/mengatur"

Menurut [Mar 11] :

"Perangkat peralatan teknik beserta perlengkapannya yang dipasang pada posisinya dan siap dipergunakan"

2.8.3 Pengertian Laboratorium

Menurut [Poer 10] :

"Tempat mengadakan percobaan (penyelidikan) segala sesuatu yang berhubungan dengan ilmu fisika, kimia, biologi, dsb."

Menurut [Mar 11] :

"Tempat atau kamar dan sebagainya tertentu yang dilengkapi dengan peralatan untuk mengadakan percobaan/penelitian