BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

2.1.1. Pengertian Sistem

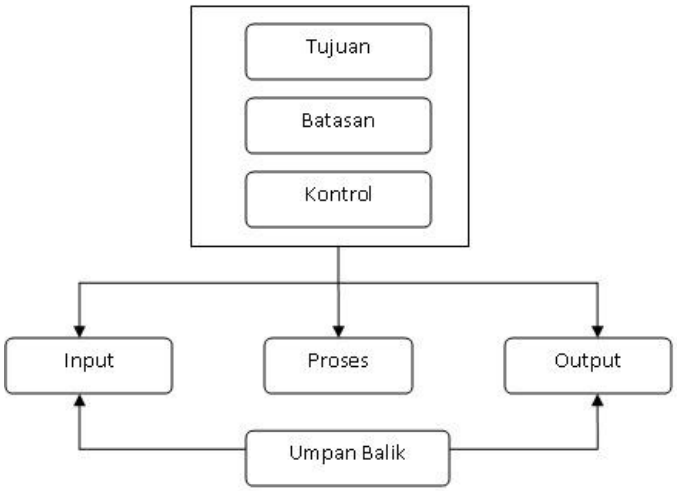
Menurut Jogiyanto (2005:1) terdapat dua kelompok pendekatan dalam mendefinisikan sistem, yaitu pendekatan yang menekankan pada prosedurnya dan pendekatan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang menekankan pada prosedurnya mendefinisikan sistem sebagai berikut:

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

Adapun pendekatan sistem yang kedua menurut Jogianto (2005:2) yaitu pendekatan yang menekankan pada komponen atau elemennya mendefinisikan sistem sebagai berikut :

"Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu." Dari dua pendekatan definisi sistem diatas, maka penulis mengambil kesimpulan bahwa sistem adalah "sistem adalah kumpulan dari komponen-komponen atau elemen-elemen yang saling berhubungan untuk mencapai suatu tujuan atau sasaran tertentu".

2.1.2. Elemen Sistem



Gambar 2.1. Elemen Sistem

[Sumber:http://oktavianibangun.blogspot.com/2009/05/pengertian-sistem-informasi.html tanggal 21/5/2013]

1. Tujuan Sistem

Tujuan sistem merupakan tujuan dari sistem tersebut dibuat. Tujuan sistem dapat berupa tujuan organisasi, kebutuhan organisasi, permasalahan yang ada dalam suatu organisasi mapun urutan prosedur untuk mencapai tujuan organisasi.

2. Batasan Sistem

Batasan sistem merupakan sesuatu yang membatasi sistem dalam mencapai tujuan sistem. Batasan sistem dapat berupa peraturan – peraturan yang ada dalam suatu organisasi, biaya – biaya yang dikeluarkan, orang – orang yang ada dalam organisasi, fasilitas baik itu sarana dan prasarana maupun batasan.

3. Kontrol Sistem

Kontrol atau pengawasan sistem merupakan pengawasan terhadap pelaksanaan pencapaian tujuan dari sistem tersebut. Kontrol sistem dapat berupa

kontrol terhadap pemasukan data (input), kontrol terhadap keluaran data (output), kontrol terhadap pengolahan data, kontrol terhadap umpan balik dan sebagainya.

4. Input

Input merupakan elemen dari sistem yang bertugas untuk mengolah atau memproses seluruh masukan data, dimana masukan tersebut dapat berupa jenis data, frekuensi pemasukan data dan sebagainya.

5. Proses

Proses merupakan elemen dari sistem yang bertugas untuk mengolah atau memproses seluruh masukan menjadi suatu informasi yang lebih berguna.

6. Output

Output merupakan hasil dari input yang telah diproses oleh bagian pengolah dan merupakan tujuan akhir sistem. Output ini bisa berupa laporan grafik, diagram batang dan sebagainya

7. Umpan Balik

Umpan balik merupakan elemen dalam sistem yang bertugas mengevaluasi bagian dari output yang dikeluarkan, dimana elemen ini sangat penting demi kemajuan sebuah sistem. Umpan balik ini dapat merupakan perbaikan sistem, pemeliharaan sistem dan sebagainya.

2.1.3. Karakteristik Sistem

Menurut Jogiyanto (2005:3) bahwa suatu sistem mempunyai karakteristik atau sifat – sifat tertentu, yaitu memiliki komponen – komponen (*components*), batas sistem (*boundary*), lingkungan sistem (*Environment*), penghubung

(*Interface*), masukan (*Input*), keluaran (*Output*), pengolah (*Proses*), dan sasaran (*Objective*), dan tujuan (*Goal*).

1. Komponen sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagain-bagian dari sistem. Setiap subsitem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengarui suatu sistem secara keseluruhan.

2. Batas Sistem (*boundary*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem di pandang sebagai satu kesatuan. Batas suatu sistem menunjukan ruang lingkup dari sistem tersebut.

3. Lingkungan Sistem (*environment*)

Linkungan luar dari sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan (harus dijaga dan merupakan energi dari sistem) dan dapat bersifat merugikan (harus ditahan dan dikendalikan).

4. Penghubung Sistem (*interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya

melalui penghubung, Dengan penghubung satu subsistem dapat berinteraksi yang lainnya membentuk satu kesatuan.

5. Masukan Sistem (*input*)

Masukan (*input*) adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal* input). Maintenance input adalah energi yang masukan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (*output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan subsistem yang lain atau kepada supersistem.

7. Pengolah Sistem

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

8. Sasaran Sistem

Suatu sistem pasti mempunya tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya Suatu sistem dikatakan berasil bila mengenai sasaran atau tujuan.

2.1.4. Klasifikasi Sistem

Menurut Jogiyanto (2005:6) sistem dapat diklasifikasikan dari beberapa sudut pandangan, diantaranya adalah sebagai berikut ini.

1. Sistem diklasifikasikan berdasarkan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*).

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak nampak, misalnya sistem teologi. Sistem fisik adalah sistem yang ada secara fisik misalnya sistem komputer.

2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia.

   Sistem alamiah adalah sistem yang terjadi melalui proses alam dan tidak di buat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang di rancang oleh manusia yang melibatkan interaksi manusia dengan mesin yang disebut dengan human-machine system atau man-machine system.

3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*).

   Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya didekteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Misalnya sistem pada komputer. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat dipredisikan karena mengandung unsur probabilitas.

4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*).

   Sistem tertutup adalah sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tertutup ini

ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar atau subsistem yang lainnya. Karena sistem sifatnya terbuka dan terpengaruh oleh lingkungan luarnya, maka suatu sistem harus mempunyai suatu sistem pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan secara otomatis dan terbuka hanya untuk pengaruh yang baik.

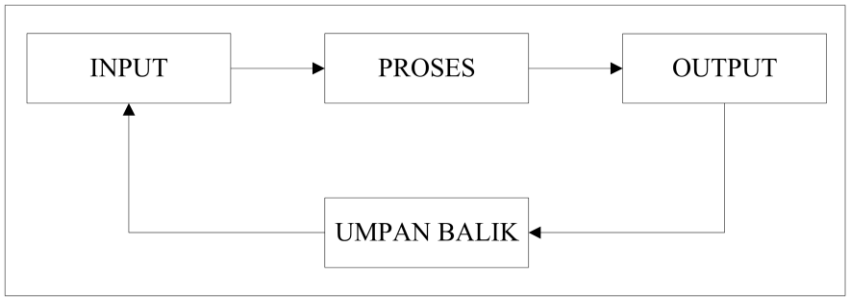
2.2. Konsep Dasar Informasi

2.2.1. Pengertian Informasi

Menurut pakar dalam bidang penelitian, **Donald Cooper,** data merupakan "kumpulan sejumlah fakta atau kenyataan yang dapat dipercaya kebenarannya sehingga dapat digunakan untuk menarik suatu kesimpulan".

Menurut Andri Kristanto (2008 : 10). Data yang masih merupakan bahan mentah apabila tidak diolah maka data tersebut tidak berguna. Data tersebut akan berguna dan menghasilkan suatu informasi apabila diolah melalui suatu model. Model yang digunakan untuk mengolah data tersebut disebut dengan model pengolahan data atau lebih dikenal dengan nama siklus pengolahan data.

Berikut adalah gambar ilustrasi Proses Informasi yang berjalan dan di hasilkan :



Gambar 2.2. Siklus Informasi Berjalan

[Sumber : Kristanto, Andi. Perancangan Sistem Informasi, 2008:10]

Dari gambar diatas dapat dijelaskan bahwa data yang merupakan suatu kejadian yang menggambarkan kenyataan yang terjadi dimasukan melalui elemen input kemudian data tersebut akan diolah dan diproses menjadi suatu output, dan output tersebut adalah informasi yang dibutuhkan. Informasi tersebut akan diterima oleh pemakai atau penerima, kemudian penerima akan memberikan umpan balik yang berupa evaluasi terhadap informasi tersebut dan hasil umpan balik tersebut akan menjadi data yang akan dimasukan menjadi input kembali. Begitu seterusnya alur pengolahan data. Kualitas suatu informasi tergantung dari tiga hal, yaitu :

a. Akurat

Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksud dari informasi tersebut.

b. Tepat Waktu

Tepat pada waktunya berarti sebuah informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai

lagi. Karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi.

c. Relevan

Berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan yang lainnya berbeda. Menurut (McFadden, dkk 1999) mendefinisikan informasi sebagai data yang telah diproses sehingga mempunyai arti dan dapat meningkatkan pengetahuan seseorang yang menggunakan data tersebut.

Kesimpulan : Suatu sistem tanpa informasi akan tidak berguna, karena suatu sistem yang kurang mendapatkan informasi akan mengalami kemacetan dan akhirnya berhenti. Dengan demikian informasi sangat penting bagi suatu sistem. Informasi sendiri berasal dari data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.2.2. Nilai Informasi

Menurut Jogiyanto (2005:11) 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. Informasi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sebagian besar informasi tidak dapat ditaksir dari keuntungannya dengan satuan nilai uang, tetapi dapat ditaksir dengan nilai efektivitasnya.

2.3. Konsep Dasar Sistem Informasi

2.3.1. Pengertian Sistem Informasi

Sistem informasi dalam sebuah system meliputi pemasukan data (*input*) kemudian diolah melalui suatu model dalam pemrosesan data, dan hasil informasi akan ditangkap kembali sebagai suatu input dan seterusnya sehingga membentuk siklus informasi yang dapat diperoleh dari system informasi sebagai sistem khusus dalam organisasi untuk mengolah informasi tersebut. Sistem informasi didefinisikan sebagai Jogiyanto.

Menurut Jogiyanto (2005:11) yang dimaksud dengan sistem informasi adalah:

"Suatu sistem dalam sebuah suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial dan kegiatan strategis dari suatu organisasi dan menyediakan pihakluar tertentu dengan laporan-laporan yang diperlukan".

2.3.2. Komponen Sistem Informasi

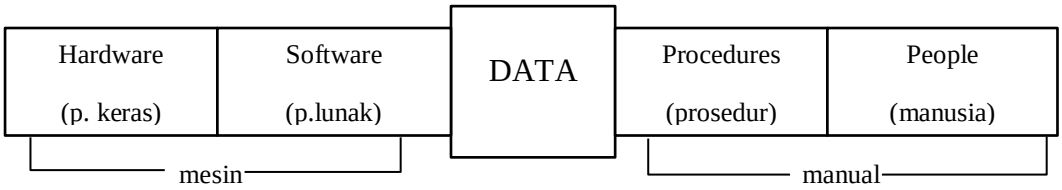
Adapun beberapa komponen sistem informasi dalam buku Al-bahra (2005:14) dapat diklasifikasikan sebagai berikut:

Hardware dan software yang berfungsi sebagai mesin. Hardware merupakan adalah semua bagian fisik komputer. Sedangkan software merupakan kumpulan dari perintah / fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer dalam melaksanakan tugas tertentu.

People dan *procedures* yang merupakan manusia dan tatacara menggunakan mesin. *People* (manusia) adalah mereka yang terlibat dalam

kegiatan sistem informasi seperti operator, pemimpin sistem informasi dan sebagainya. Sedangkan *procedures* (prosedur) merupakan tatacara yang menghubungkan berbagai perintah, dan aturan yang akan menentukan rancangan dan penggunaan sistem informasi.

Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi dan merupakan penghubung antara manusia dan mesin agar terjadi suatu proses pengolahan data.



Gambar 2.3. Lima komponen Sistem Informasi

[Sumber: Al-bahra bin Ladjamudin. 2005. Analisis dan Desain Sistem Informasi. Graha Ilmu. Yogyakarta.]

2.3.3. Kegiatan Sistem Informasi

Adapun kegiatan sistem informasi, diantaranya Al-Bahra (2005:22) :

1. *Input*; Menggambarkan suatu kegiatan untuk menyediakan data untuk diproses.
2. Proses; Menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.
3. *Output*; Suatu kegiatan untuk menghasilkan dari kegiatan Proses.
4. Penyimpanan; Suatu kegiatan untuk memelihara dan menyimpan data.

5. *Control*; Suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan seperti yang diharapkan.

2.4. Pengertian Sistem Informasi Pengelolaan Data Pasien

Sistem informasi pengelolaan data pasien merupakan proses pengolahan data pasien yang melakukan pengobatan ke klinik mulai dari proses pendaftaran, pemeriksaan dan pembayaran. Dari sistem informasi ini dapat dihasilkan berupa laporan kunjungan pasien, laporan rekam medis pasien dan laporan pembayaran pasien di klinik.

2.4.1. Definisi Klinik

Sebuah klinik (atau rawat jalan klinik atau klinik perawatan rawat jalan) adalah fasilitas kesehatan yang dikhususkan untuk perawatan pasien rawat jalan. Klinik dapat dioperasikan, dikelola dan didanai secara pribadi atau publik, dan biasanya meliputi perawatan kesehatan primerkebutuhan populasi di masyarakat lokal, berbeda dengan rumah sakit yang lebih besar yang menawarkan perawatan khusus dan mengakui pasien rawat inap untuk menginap semalam. Beberapa klinik bisa tumbuh menjadi lembaga besar seperti rumah sakit besar, atau menjadilembaga yang berhubungan dengan rumah sakit atau sekolah kedokteran, sementara beberapa tetap mempertahankan nama "klinik".

Klinik juga sering dikaitkan dengan praktek medis umum, dijalankan oleh satu atau beberapa dokter umum atau manajer praktek.

[Sumber:http://www.richamorku.com/artikel/artikel_pengertian_arti_klinik.htm 1/Juni/2013]

2.4.2. Definisi Pasien

Pengertian pasien menurut “http://www.wikipedia.org/wki/pasien” adalah seseorang yang menerima perawatan medis, seringkali pasien menderita penyakit atau cedera dan memerlukan bantuan dokter untuk memulihkannya.

Kata pasien dari bahasa Indonesia analog dengan kata patient dari bahasa Inggris. Patient diturunkan dari bahasa latin yaitu patiens yang memiliki kesamaan arti dengan kata pati yang artinya menderita.

[Sumber:http://nanangbudianas.blogspot.com/2013/02/pengertian-pasien-dan pasien-rawat-inap.html 1/Juni/2013]

2.5. Alat Bantu yang Digunakan

Adapun alat-alat bantu yang digunakan dalam penelitian ini akan dijelaskan seperti di bawah ini :

2.5.1. Konsep Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah mempermudah programmer didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Suatu perusahaan software yaitu Rational Software, telah membentuk konsarium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysist Design* (OOAD).

2.5.1.1.Pengertian UML

Menurut Adi Nugroho (2010:6) UML (*Unified Modelling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’. Pemodelan (*modelling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahn yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. UML yang sesungguhnya merupakan metodologi kolaborasi antara metode-metode Booch yang dikembangkan oleh Graddy Booch, OMT (*Object Modeling Technique*) yang dikembagkan oleh DR. James Rumbaugh, serta OOSE (*Object Orientid Software Engineering*) yang dikembangkan oleh Ivar Jacobson.

2.5.1.2.Tujuan UML

Tujuan UML yaitu diantaranya:

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 pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
4. Untuk membuat suatu model, UML memiliki diagram grafis sebagai berikut:
    a) Business Use Case model
    b) Activity Diagram
    c) Use Case model
    d) Behavior diagram : Sequence diagram

e) Implementation diagram : Component diagram, Deployment diagram

f) Generate Code

Diagram diagram tersebut diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisis atau rekayasa. Dibuatnya berbagai jenis diagram diatas karena :

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

Diagram-diagram ini ditambah dengan kemampuan dokumentasi perupakan artifacts utama UML. *Data-Flow Diagram* dan tipe diagram lain yang tidak terdapat dalam UML tidak termasuk dalam paradigma *object-oriented*. *Activity diagram* dan *collaboration diagram* yang terdapat dalam UML menggantikan data-flow diagram. *Activity diagram* juga sangat bermanfaat untuk membuat *workflow*.

2.5.1.3.Diagram-Diagram Dalam UML

Adapun diagram – diagram dalam UML, menurut Adi Nugroho (2010:34) terdapat beberapa diagram antara lain yaitu:

a. Use Case Diagram

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.

b. Class Diagram

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*.

Adapun hubungan antar *class*, sebagai berikut :

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.

c. Activity Diagram

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 (*forkdan 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 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*.

e. Collaboration Diagram

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.

f. Component Diagram

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.

g. Deployment Diagram

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.

2.5.2. Pengertian StarUML

StarUML adalah sebuah proyek open source untuk mengembangkan cepat, fleksibel, extensible, featureful, dan bebas-tersedia UML / platform MDA berjalan pada platform Win32. Tujuan dari proyek StarUML adalah untuk membangun sebuah alat pemodelan perangkat lunak dan juga platform yang menarik adalah pengganti alat UML komersial seperti Rational Rose.

2.6. Perangkat Lunak Pendukung

Perangkat lunak pada penelitian ini adalah Java Netbeans 7.2.1, Xampp, MySQL. Berikut ini adalah penjelasan dari kedua perangkat pendukung tersebut.

2.6.1. Java Netbeans IDE 7.2.1

NetBeans IDE **7.2.1** adalah sebuah lingkungan pengembangan sebuah tools untuk programmer menulis, mengkompilasi, 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. (sumber : www.netbeans.org)

NetBeans mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop java, dan sebuah Integrated Development Environment (IDE) yang dibangun menggunakan platform NetBeans. Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut “modul”. Sebuah modul adalah suatu arsip Java (Java archive) yang memuat kelas-kelas Java untuk berinteraksi dengan NetBeans Open API dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang

dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform NetBeans dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan powerful.

Lingkungan Pengembangan yang terintegrasi pada netbean IDE, memudahkan pengguna untuk membuat beragam aplikasi dengan mudah. Adapun komponen – komponen Neatbeans adalah sebagai berikut:

1. *Menu Bar*

Letak *Menu Bar* pada *Palette* yaitu pada *Swing Menus*, untuk menggunakan button ini klik pada button tersebut. Fungsi daripada menu bar ini untuk menciptakan tab-tab menu yang nantinya dipergunakan untuk perintah menu yang diinginkan.

2. *Toolbar*

ToolBar berfungsi sebagai tempat untuk menampilkan tool-tool yang bertujuan mempermudah user dalam mengoperasikan beberapa tool-tool yang mungkin sering digunakan.

3. *Project Explorer*

Project Explorer akan digunakan untuk melihat file – file yang berupa class java yang terbentuk melalui Neatbeans.

4. *Panel Inspector*

Panel *inspector* akan tampil apabila mengaktifkan dokumen yang mengandung *container* atau pemograman grafis (GUI). Pada panel ini

menampilkan komponen yang digunakan oleh file yang bersangkutan, seperti container, komponen kontrol, komponen menu, komponen border dan lainnya.

5. Jendela Utama

Jendela utama adalah sebuah objek yang digunakan untuk menempatkan objek-objek yang dipilih dari kontrol komponen.

6. Kontrol Komponen

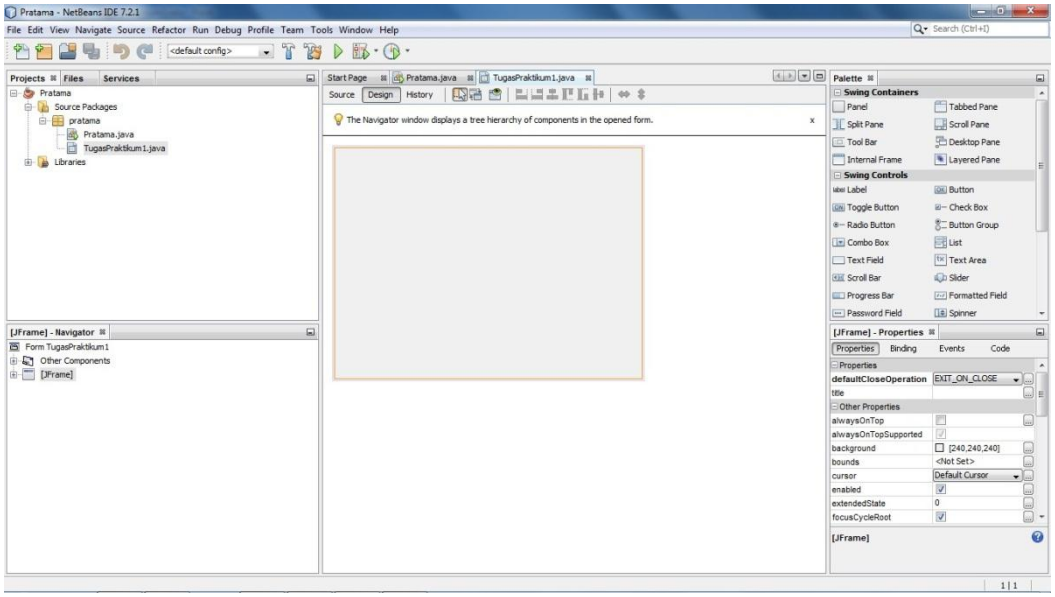
Kontrol komponen merupakan jendela yang menyediakan tool-tool untuk mendesign form berbasis grafis (GUI). Tool ini dibagi menjadi beberapa kategori, dimana setiap kategori menyediakan tool-tool GUI Builder sesuai dengan kategorinya. untuk menggunakannya, tinggal menyeret tool-tool kedalam area desain.

7. Jendela Properties

Jendela Properties berfungsi menampilkan properties komponen yang aktif untuk mengatur properties yang dimiliki oleh suatu komponen.

8. Jendela Keluaran (*debugging*)

Terdiri dari jendela Debugger dan Output. Jendela Debugger mempunyai tab panel untuk menset breakpoint, memonitor thread, dan mengawasi nilai variabel-variabel. Jendela Output menampilkan pesan dari debugger. Jika ada file yang terbuka, *Debugging workspace* juga mengandung *Source Editor*.



Gambar 2.4. Tampilan Netbeans IDE 7.2.1

[Sumber : http://probase-tama.blogspot.com/2013/05/ 1/Juni/2013]

2.6.2. Xampp

XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan project non-profit yang di kembangkan oleh Apache Friends yang didirikan Kai 'Oswalad' Seidler dan Kay Vogelgesang pada tahun 2002, project mereka ini bertujuan mempromosikan pengunaan Apache web server.

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 dan bebas, merupakan web server yang

mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. Untuk mendapatkanya dapat mendownload langsung dari web resminya. Mengenal bagian XAMPP yang biasa digunakan pada umumnya:

9. htdoc adalah folder tempat meletakkan berkas-berkas yang akan dijalankan, seperti berkas PHP, HTML dan skrip lain.
10. PhpMyAdmin merupakan bagian untuk mengelola basis data MySQL yang ada dikomputer. Untuk membukanya, buka browser lalu ketikkan alamat http://localhost/phpMyAdmin, maka akan muncul halaman phpMyAdmin.
11. Kontrol Panel yang berfungsi untuk mengelola layanan (*service*) XAMPP. Seperti menghentikan (*stop*) layanan, ataupun memulai (*start*).

2.6.3. MySQL

MySQL merupakan aplikasi database server. Perkembangannya disebut SQL yang merupakan kepanjangan dari Structured Query Language. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database. SQL pertama kali didefinisikan oleh American National Standards Institute (ANSI) pada tahun 1986. MySQL adalah sebuah sistem manajemen database yang bersifat open source. MySQL adalah pasangan serasi dari PHP. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia. MySQL dapat digunakan untuk membuat dan mengola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam database. MySQL merupakan sisitem manajemen database yang bersifat relational, artinya data-data yang dikelola dalam database akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan

menjadi jauh lebih cepat. MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar. MySQL juga dapat menjalankan perintah-perintah Structured Query Language (SQL) untuk mengelola database-database yang ada di dalamnya. Hingga kini, MySQL sudah berkembang hingga versi 5.

MySQL 5 sudah mendukung trigger untuk memudahkan pengelolaan tabel dalam database.

2.7. Arsitektur Aplikasi

Arsitektur aplikasi meliputi pengertian jaringan komputer, tipe-tipe jaringan, jenis topologi jaringan, dan pengertian client server.

2.7.2. Pengertian Jaringan Komputer

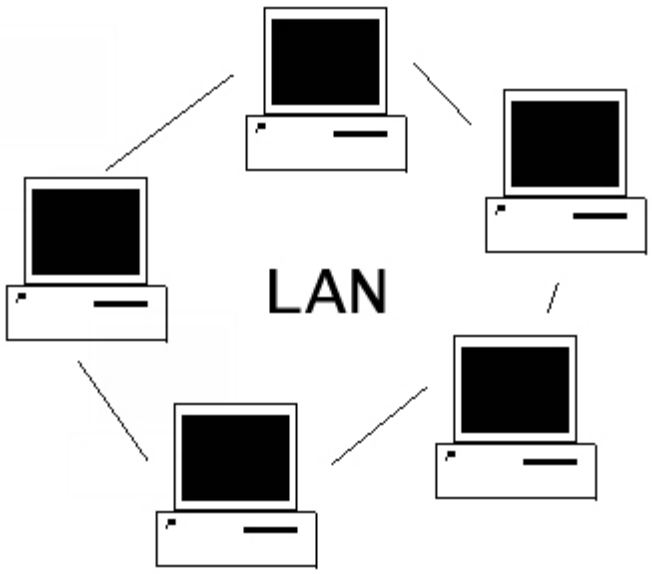
Menurut Abdul Kadir (2003:346) yang dimaksud dengan jaringan komputer (*computer network*) adalah hubungan dua buah simpul (umumnya berupa komputer) atau lebih yang bertujuan utamanya adalah untuk melakukan pertukaran data.

2.7.3. Tipe-tipe Jaringan Komputer

1. LAN (*Local Area Network)*

LAN adalah suatu jaringan yang digunakan untuk menghubungkan komputer yang berada dalam satu area yang kecil seperti dalam satu ruang, satu gedung, atau beberapa gedung yang berdekatan. LAN umumnya menggunakan media transmisi berupa kabel tetapi ada juga yang tidak mengguanakan kabel dan disebut dengan wireless LAN atau LAN tanpa kabel. Jarak komputer yang

dihubungkannya bisa mencapai 5 sampai 10 kilometer. Kecepatan LAN berkisar dari 10 sampai 100 Mbps.



Gambar 2.5. Arsitektur LAN

[sumber : http://wally.cs.iupui.edu/n241-new/webMag/index.html 1/Juni/2013]

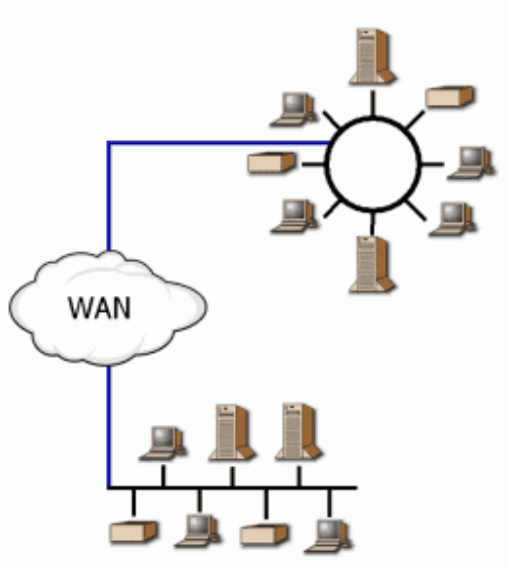
2. MAN (Metropolitan Area Network)

MAN adalah suatu jaringan yang cakupannya meliputi suatu kota atau dengan rentang sekitar 10 sampai 45 kilometer. MAN menghubungkan LAN-LAN yang lokasinya berjauhan. MAN umumnya menggunakan media transmisi dengan mikro gelombang atau gelombang radio, tetapi ada juga yang menggunakan jalur sewa *(leased line)*. Kecepatan MAN berkisar dari 1,5 sampai 150 Mbps.

3. WAN (Wide Area Network)

WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada cakupan geografis yang luas, seperti hubungan dari kota ke kota lain dalam

satu negara. Cakupan WAN meliputi 100 sampai 1000 kilometer, dan kecepatan antar kota bisa bervariasi antara 1,5 Mbps sampai 2,5 Gbps.



Gambar 2.6. Arsitektur WAN

[sumber : http://3.bp.blogspot.com/ 9/Juni/2013]

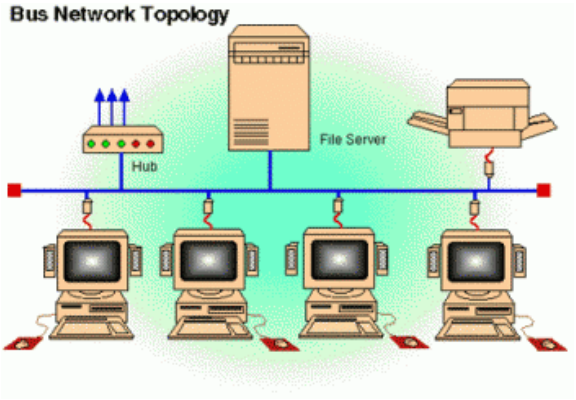
4. GAN *(Global Area Network)*

GAN adalah jaringan yang menghubungkan negara-negara diseluruh dunia. Kecepatan GAN bervariasi mulai dari 1,5 Mbps sampai 100 Gbps, cakupan WAN mencapai ribuan kilometer.

2.7.4. Topologi Jaringan

Menurut Dede Sopandi (2008:27) topologi menggambarkan metode yang digunakan untuk melakukan pengabelan secara fisik dari suatu jaringan. Topologi Jaringan adalah susunan atau pemetaan interkoneksi anatara *node*, dari suatu jaringan, baik secara fisik (*riil*) dan logis (*virtual*).

1. Topolgoi Bus



Gambar 2.7. Topologi Bus

[Sumber : http://id.wikipedia.org/wiki/Topologi_bus 1/Juni/2013]

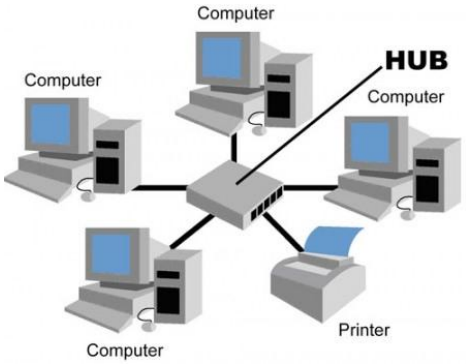
Pada topologi Bus, kedua unjung jaringan harus diakhiri dengan sebuah terminator. Barel connector dapat digunakan untuk memperluasnya. Jaringan hanya terdiri dari satu saluran kabel yang menggunakan kabel BNC. Komputer yang ingin terhubung ke jaringan dapat mengkaitkan dirinya dengan mentap Ethernetnya sepanjang kabel. Linear Bus: Layout ini termasuk layout yang umum. Satu kabel utama menghubungkan tiap simpul, ke saluran tunggal komputer yang mengaksesnya ujung dengan ujung. Masing-masing simpul dihubungkan ke dua simpul lainnya, kecuali mesin di salah satu ujung kabel, yang masing-masing

hanya terhubung ke satu simpul lainnya. Topologi ini seringkali dijumpai pada sistem client/server, dimana salah satu mesin pada jaringan tersebut difungsikan sebagai File Server, yang berarti bahwa mesin tersebut dikhususkan hanya untuk pendistribusian data dan biasanya tidak digunakan untuk pemrosesan informasi. Instalasi jaringan Bus sangat sederhana, murah dan maksimal terdiri atas 5-7 komputer. Kesulitan yang sering dihadapi adalah kemungkinan terjadinya tabrakan data karena mekanisme jaringan relatif sederhana dan jika salah satu node putus maka akan mengganggu kinerja dan trafik seluruh jaringan.

Keunggulan topologi Bus adalah pengembangan jaringan atau penambahan workstation baru dapat dilakukan dengan mudah tanpa mengganggu workstation lain. Kelemahan dari topologi ini adalah bila terdapat gangguan di sepanjang kabel pusat maka keseluruhan jaringan akan mengalami gangguan.

[Sumber : http://id.wikipedia.org/wiki/Topologi_bus 1/Juni/2013]

2. Topologi Star/Bintang



Gambar 2.8. Topologi Star

[Sumber :http://id.wikipedia.org/wiki/Topologi_bintang 1/Juni/2013]

Topologi bintang merupakan bentuk topologi jaringan yang berupa konvergensi dari node tengah ke setiap node atau pengguna. Topologi jaringan bintang termasuk topologi jaringan dengan biaya menengah.

Kelebihan

Kerusakan pada satu saluran hanya akan mempengaruhi jaringan pada saluran tersebut dan station yang terpaut.

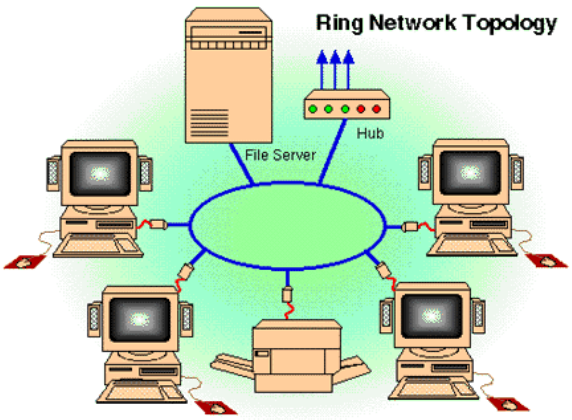
a. Tingkat keamanan termasuk tinggi.
b. Tahan terhadap lalu lintas jaringan yang sibuk.
c. Penambahan dan pengurangan station dapat dilakukan dengan mudah.

Kekurangan

Jika node tengah mengalami kerusakan, maka seluruh jaringan akan terhenti.

[Sumber :http://id.wikipedia.org/wiki/Topologi_bintang 1/Juni/2013]

3. Topologi Ring/Cincin



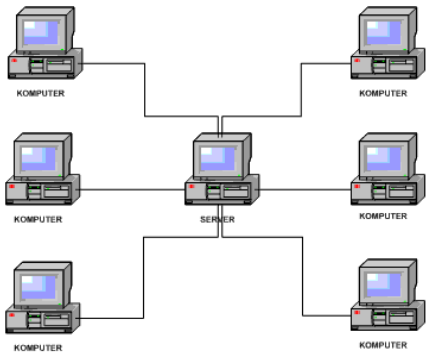
Gambar 2.8. Topologi Ring

[Sumber :http://id.wikipedia.org/wiki/Topologi_cincin 1/Juni/2013]

Topologi cincin adalah topologi jaringan dimana setiap titik terkoneksi ke dua titik lainnya, membentuk jalur melingkar membentuk cincin. Pada topologi cincin, komunikasi data dapat terganggu jika satu titik mengalami gangguan. Jaringan FDDI mengantisipasi kelemahan ini dengan mengirim data searah jarum jam dan berlawanan dengan arah jarum jam secara bersamaan.

[Sumber :http://id.wikipedia.org/wiki/Topologi_cincin 1/Juni/2013]

4. Topologi Mesh



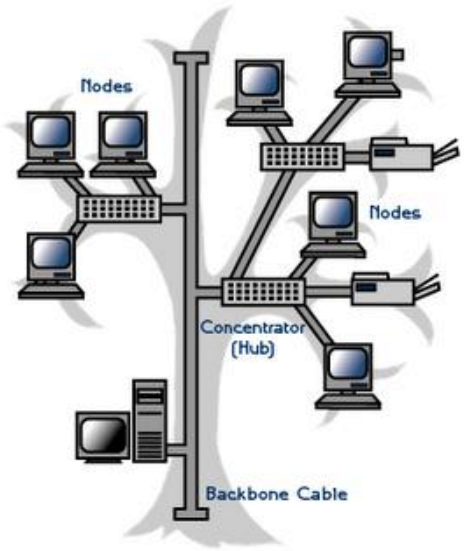
Gambar 2.9. Topologi Mesh

[Sumber : http://id.wikipedia.org/wiki/Topologi_mesh 1/Juni/2013]

Topologi jaringan ini menerapkan hubungan antar sentral secara penuh. Jumlah saluran harus disediakan untuk membentuk jaringan Mesh adalah jumlah sentral dikurangi 1 (n-1, n = jumlah sentral). Tingkat kerumitan jaringan sebanding dengan meningkatnya jumlah sentral yang terpasang. Dengan demikian disamping kurang ekonomis juga relatif mahal dalam pengoperasiannya.

[Sumber : http://id.wikipedia.org/wiki/Topologi_mesh 1/Juni/2013]

5. Topologi Tree



Gambar 2.10. Topologi Tree

[Sumber : http://id.wikipedia.org/wiki/Topologi_pohon 1/Juni/2013]

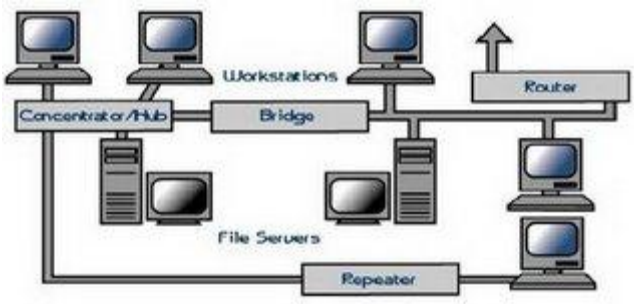
Topologi Jaringan Pohon (Tree) Topologi jaringan ini disebut juga sebagai topologi jaringan bertingkat. Topologi ini biasanya digunakan untuk interkoneksi antar sentral denganhirarki yang berbeda. Untuk hirarki yang lebih rendah digambarkan pada lokasi yang rendah dan semakin keatas mempunyai hirarki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

Pada jaringan pohon, terdapat beberapa tingkatan simpul (node). Pusat atau simpul yang lebih tinggi tingkatannya, dapat mengatur simpul lain yang lebih rendah tingkatannya. Data yang dikirim perlu melalui simpul pusat terlebih dahulu. Misalnya untuk bergerak dari komputer dengan node-3 kekomputer node-7 seperti halnya pada gambar, data yang ada harus melewati node-3, 5 dan node-6

sebelum berakhir pada node-7. Keungguluan jaringan model pohon seperti ini adalah, dapat terbentuknya suatu kelompok yang dibutuhkan pada setiap saat. Sebagai contoh, perusahaan dapat membentuk kelompok yang terdiri atas terminal pembukuan, serta pada kelompok lain dibentuk untuk terminal penjualan. Adapun kelemahannya adalah, apabila simpul yang lebih tinggi kemudian tidak berfungsi, maka kelompok lainnya yang berada dibawahnya akhirnya juga menjadi tidak efektif. Cara kerja jaringan pohon ini relatif menjadi lambat.

[Sumber : http://id.wikipedia.org/wiki/Topologi_pohon 1/Juni/2013]

6. Topologi Linier



Gambar 2.11. Topologi Linier

[Sumber : http://id.wikipedia.org/wiki/Topologi_linier 1/Juni/2013]

Jaringan komputer dengan topologi linier biasa disebut dengan topologi linier bus, layout ini termasuk layout umum. Satu kabel utama menghubungkan tiap titik koneksi (komputer) yang dihubungkan dengan konektor yang disebut dengan T Connector dan pada ujungnya harus diakhiri dengan sebuah terminator. Konektor yang digunakan bertipe BNC (British Naval Connector), sebenarnya BNC adalah nama konektor bukan nama kabelnya, kabel yang digunakan adalah RG 58 (Kabel Coaxial Thinnet). Installasi dari topologi linier bus ini sangat sederhana dan murah tetapi maksimal terdiri dari 5-7 Komputer.

Keuntungan, hemat kabel, layout kabel sederhana, mudah dikembangkan, tidak butuh kendali pusat, dan penambahan maupun pengurangan terminal dapat dilakukan tanpa mengganggu operasi yang berjalan.

Kerugian, deteksi dan isolasi kesalahan sangat kecil, kepadatan lalu lintas tinggi, keamanan data kurang terjamin, kecepatan akan menurun bila jumlah pemakai bertambah, dan diperlukan Repeater untuk jarak jauh.

[Sumber : http://id.wikipedia.org/wiki/Topologi_linier 1/Juni/2013]