BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Informasi

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling bekerjasama membentuk suatu kesatuan, suatu sistem betapa pun kecilnya selalu mengandung komponen-komponen, dimana komponen-komponen dapat berupa subsistem-subsistem atau bagian dari suatu sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

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

2.1.1 Definisi Sistem

Inti dari sebuah sistem dapat dikelompokkan menjadi dua kelompok dalam mendenifisikan sistem yaitu penekanan pada prosedurnya dan menekankan pada komponen atau elemennya. Denifisi sistem itu sendiri yang menekankan pada komponen atau elemen memiliki pengertian yang lebih luas dari pada penekanan pada prosedur karena pengertian tersebut lebih diterima dikarenakan suatu sistem terdiri dari beberapa subsistem – subsistem.

http://santiw.staff.gunadarma.ac.id/Downloads/files/7691/Pengantar_Sistem_Informasi.doc 23 maret 2009 Dikutip dari Jerry FithGerald, sistem adalah suatu

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

Menurut Fathansyah (2001 : 1) yang dimaksud dengan sistem adalah sebuah tatanan (keterpaduan) yang terdiri atas jumlah sejumlah komponen fungsional (tugas khusus) yang saling berhubungan dan secara bersama-sama bertujuan untuk memenuhi suatu proses.

2.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen, batas sistem, lingkungan luar sistem, penghubung system, masukan sistem, keluaran sistem, pengolah sistem, sasaran sistem. Menurut Jogiyanto (2005 : 3) menyebutkan bahwa karakteristik sistem adalah 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 betapa pun kecilnya selalu mengandung komponen-komponen atau subsistem-subsistem.

2. Batas Sistem

Batas sistem *(boundary)* merupakan daerah yang membatasi antara sustu 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.

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.

4. Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.

5. Masukan Sistem

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

6. Keluaran Sistem

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

7. Pengolah Sistem

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

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan atau sasaran. Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

2.1.3 Klasifikasi Sistem

http://wsilfi.staff.gunadarma.ac.id/*Downloads/files/1004/Konsep+SI.pdf*

23 maret 2009. Klasifikasi Sistem menurut sumber dari internet diatas sistem dapat diklasifikasikan dari beberapa sudut pandangan, yaitu sebagai berikut :

1. Sistem abstrak : sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teologia)

   Sistem fisik : merupakan sistem yang ada secara fisik (sistem komputer, sistem akuntansi, sistem produksi dll.)
2. Sistem alamiah : sistem yang terjadi melalui proses alam. (sistem matahari, sistem luar angkasa, sistem reproduksi dll.

   Sistem buatan manusia : sistem yang dirancang oleh manusia.

   Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut human-machine system (contoh : sistem informasi)
3. Sistem Tertentu (deterministic system) : beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (contoh : sistem komputer)

Sistem tak tentu (probabilistic system) : sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem tertutup (close system) : sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut 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 (open system) : sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

5. Sistem sederhana dan Sistem kompleks

2.1.4 Definisi Informasi

Keterangan, penerangan. Data yang telah diproses ke dalam suatu bentuk yang mempunyai arti bagi si penerima dan mempunyai nilai nyata, sehingga dapat dipakai sebagai dasar untuk mengambil keputusan, dan terasa bagi keputusan saat itu atau keputusan mendatang.

Susunan hirarki informasi mulai dari data atau fakta, kemudian diseleksi dan diolah menjadi sesuatu yang berguna Konsep dasar informasi menurut Jogiyanto (2005 : 8) Informasi adalah :

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

Konsep dasar informasi menurut Menurut Davis [Abd03: 9]

"Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang."

2.1.5 Kualitas Informasi

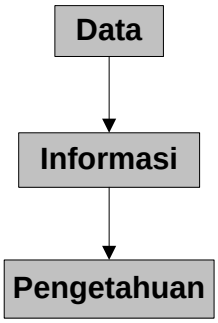
Baik buruknya kualitas suatu informasi dipengaruhi oleh beberapa hal, antara lain :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. 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, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi harus sesuai dengan dengan keadaan saat itu, karena informasi yang sudah usang tidak akan mempunyai nilai lagi. Informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi. Dewasa ini mahalnya informasi disebabkan harus cepatnya informasi tersebut didapat, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.
3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan lainnya berbeda. Misalnya

informasi mengenai sebab-musabab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya informasi mengenai harga pokok produksi untuk ahli teknik merupakan informasi yang kurang relevan, tetapi relevan untuk akuntan.

2.1.6 Hirark Informasi

Informasi memiliki urutan atau tingkatan yang disebut sebagai hirarki makna informasi, yaitu tingkatan informasi dilihat dari kegunaan atau asas manfaatnya.



(Sumber: Jogiyanto, 2005: 9)

Gambar 2.1. Hirarki Informasi

Dari gambar diatas, menurut Jogiyanto (2005: 9) terjadi 3 (tiga) bagian utama, yaitu :

1. Data, yaitu gambaran dari suatu kejadian nyata.
2. Informasi, yaitu suatu hasil dari pengolahan data yang memiliki nilai guna bagi penerima yang membutuhkan.
3. Pengetahuan, yaitu nilai lebih yang didapat dari sebuat informasi.

2.1.7 Sistem Informasi

Sistem yang ada berfungsi sebagai penghasil suatu keluaran, baik berupa informasi maupun berupa objek / benda. Untuk keluaran berupa informasi, maka sistem tersebut dikatakan sebagai sistem informasi

2.1.8 Pengertian Sistem Informasi

Definisi menurut Jogiyanto (2005 : 11) Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, pendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan.

Menurut Jogiyanto (2005: 12) faktor – faktor yang menentukan kehandalan dari suatu sistem informasi atau informasi dapat dikatakan baik jika memenuhi kriteria-kriteria sebagai berikut :

1. Keunggulan (*usefulness*)

   Yaitu suatu sistem yang harus dapat menghasilkan informasi yang tepat dan relevan untuk mengambil keputusan manajemen dan personil operasi dalam organisasi.

2. Ekonomis

   Kemampuan sistem yang mempengaruhi sistem harus bernilai manfaat minimal, sebesar biayanya.

3. Kehandalan (*Reliability*)

   Keluaran dari sistem harus mempunyai tingkat ketelitian tinggi dan sistem tersebut harus beroperasi secara efektif.

4. Pelayanan (*Customer Service*)

   Yakni suatu sistem memberikan pelayanan yang baik dan efisien kepada para pengguna sistem pada saat berhubungan dengan organisasi.

5. Kapasitas (*Capacity*)

   Setiap sistem harus mempunyai kapasitas yang memadai untuk menangani setiap periode sesuai yang dibutuhkan.

6. Sederhana dalam kemudahan (*Simplicity*)

   Sistem tersebut lebih sederhana (umum) sehingga struktur dan operasinya dapat dengan mudah dimengerti dan prosedure mudah diikutin.

7. Fleksibel (*Fleksibility*)

   Sistem informasi ini harus dapat digunakan dalam kondisi yang bagaimana yang diinginkan oleh organisasi tersebut atau pengguna tertentu.

2.1.9 Komponen Sistem Informasi

http://wsilfi.staff.gunadarma.ac.id/*Downloads/files/1004/Konsep+SI.pdf*

23 maret 2009, Istilah dalam komponen sistem informasi adalah blok bangunan (*building block*) yang dapat di bagi menjadi enam blok, yaitu :

1. Blok masukan (*input block*)

   Blok input merupakan data–data yang masuk ke dalam sistem informasi, yang dapat berupa document-document dasar yang dapat diolah menjadi suatu informasi tertentu.

2. Blok model (*model block*)

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan mengolah data input untuk menghasilkan suatu informasi yang dibutuhkan.

3. Blok keluaran (*output block*)

   Merupakan informasi yang menghasilkan sekumpulan data yang nantinya akan disimpan berupa data ceta laporan.

4. Blok teknologi (*technologi block*)

   Blok teknologi merupakan penunjang utama dalam berlangsunganya sistem informasi. Yang memiliki beberapa komponen yaitu diantaranya alat memasukan data (*input device*), alat untuk menyimpan dan mengapses data (*storege device*), alat untuk menghasilakan dan mengirimkan keluaran (*output divice*) dan alat untuk membantuk pengendalian sistem secara keseluruan (*control device*). Teknologi informasi terdiri dari 3 (tiga) bagian utama, yaitu teknisi (*humanware* atau *brainare*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok basis data (*database block*)

   Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkap keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu di simpan dan perlu di organisasi sedemikian rupa, supaya informasi yang dihasilkan berkualitas.

6. Blok kendali (*control block*)

   Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal–hal yang dapat merusak sistem dapat di cegah bila terlanjur terjadi kesalahan-kesalahan kerusakan dalam penggunaan sistem.

2.2 Definisi dari Kasus yang Di Analisis

2.2.1 Pengertian Forex

Forex Trading adalah transaksi perdagangan nilai tukar mata uang asing di pasar uang internasional. Pasar *forex* merupakan pasar uang terbesar di dunia, lebih besar dari saham , options , futures. Siapa yang melakukan transaksi forex ? yang melakukan transaksi di pasar *forex* adalah: pemerintah-pemerintah di dunia, bank-bank utama dunia, perusahaan bertaraf internasional, hedge fund, spekulan valas maupun individu. Sehingga dengan banyaknya pemain di pasar forex ini menyebabkan perputaran uang menjadi sangat cepat. Transaksi yang terjadi lebih dari 1,9 triliun US dollar setiap hari sehingga membuat uang dapat berpindah tangan dari satu tempat ke tempat lain hanya dalam beberapa detik.

Dengan berkembangnya teknologi internet, sekarang ini siapapun bisa melakukannya dari rumah. Ditambah lagi dengan sistem margin trading dengan menggunakan leverage dan nilai kontrak yang sangat kecil, memungkinkan investor dengan dana yang relatif kecil. Di Fx-Indo sendiri, Anda bahkan bisa mulai dengan $1 saja!

Para pemain forex dapat melakukan trading dengan menggunakan jasa perusahaan pialang (commision house) atau melakukannya sendiri secara online melalui internet. Perdagangan forex memiliki beberapa kelebihan dibandingkan dengan perdagangan produk-produk keuangan lain seperti perdagangan saham, yaitu :

1. *24 Hours Trading*

Dapat dilakukan 24 jam sehari, 5 hari seminggu, kapan dan dimanapun kita berada. 24 jam terus berjalan non stop : Dimulai dari waktu new zealand & australia jam 05.00 - 14.00 wib , terus ke pasar asia yaitu Jepang, Hongkong & Singapura jam 07.00 - 16.00 wib , ke pasar eropa yaitu Jerman & Inggris jam 13.00 - 22.00 wib , sampai ke pasar Amerika jam 20.30 - 10.30 wib.

Semua mata uang dunia yang umum dan memiliki daya jual tinggi. Contoh : USDollar, Yen, Euro, Franc, Poundsterling. (EUR/USD, GBP/USD, USD/JPY, USD/CHF). dan lain-lain. Kemudian bagaimana proses transaksinya? dan apa itu Beli/Jual dalam bursa valuta asing ini ? di bursa valas (forex) ini anda dapat membeli ataupun menjual mata uang yang diperdagangkan. Secara obyektif adalah untuk mendapatkan profit atau keuntungan dari posisi transaksi yang anda lakukan.

Segala transaksi FOREX anda dilakukan oleh anda sendiri secara online via internet (bisa di rumah, warnet, hotel, cafe, mobil, via PDA, dll)

2. ***Likuiditas***

Sangat likuid dengan banyaknya broker/dealer yang bermain dalam pasar forex.

3. ***Rendahnya biaya transaksi***

Komisi broker relatif kecil, bahkan untuk trading online melalui internet tidak ada biaya transaksi namun hanya dikenakan biaya yang jumlahnya beragam. Selain itu spreadnya juga kecil.

4. ***Potensi keuntungan 2 arah (naik maupun turun)***

Memiliki potensi keuntungan baik pada mata uang yang menguat maupun pada mata uang yang melemah.

5. ***Margin Trading***

Perdagangan dengan margin membuat daya beli investor melebihi jumlah modal yang dimiliki. Setelah anda tahu penjelasan singkat ini , langkah selanjutnya adalah memilik BROKER Forex ada di pasaran yang jumlahnya ratusan. Pilih broker yang sesuai dengan pilihan anda , anda nyaman trading forex di broker tersebut , anda tidak banyak mengalamai masalah dengan broker tersebut , anda untung trading dgn broker tersebut. Strategy memilih broker forex akan saya jelaskan di artikel selanjutnya.

2.3 Pengertian *Internet* dan *Web Site*

Dalam penelitian ini dibutuhkan beberapa teori yang mendukung dalam penyajian informasi, hal ini dianggap perlu untuk memudahkan dalam pembuatan sistem informasi khususnya yang berbasis *web.*

2.3.1 *Internet*

Jaringan komputer di dunia telah berkembang selama lebih dari 40 tahun, dan dalam kurun waktu tersebut, fungsinya telah berubah dari sekedar alat percobaan dilaboratorium menjadi suatu sistem yang digunakan oleh berjuta-juta orang setiap harinya. Jaringan yang pertama kali adalah *Arpanet*, hingga sekarang telah memberikan manfaat yang luar biasa bukan karena telah digunakan oleh ribuan pakar komputer untuk mengakses dan berbagi serta menerima mail tetapi lebih dari pada itu. *Arpanet* akan tercatat dalam sejarah sebagai jaringan perintis terciptanya suatu sistem informasi global yang menjadikan dunia ini semakin kecil dan meniadakan apa yang disebut batas-batas suartu Negara.

Hal ini dapat dimaklumi, karena dengan adanya *Internet*, suatu sistem informasi yang dulu sulit diperoleh, kini semuanya bisa diperoleh hanya dalam beberapa saat saja. Bahkan bukan tidak mungkin, pada masa yang akan datang dengan semakin berkembangnya *Internet*, semua orang diseluruh dunia akan saling bertukar informasi hanya dalam sekejap. Untuk itu patut diingat, bahwa dunia informasi kelak akan sangat bergantung kepada *Internet* dan berarti internet tidak

akan pernah mundur apalagi punah, sebalikya *Internet* akan semakin terus berkembang dan berkembang tanpa dibatasi.

Jadi *Internet* adalah jaringan komputer global yang terdiri dari beberapa komputer yang berhubungan yang dapat menjangkau seluruh dunia, hanya saja ruang lingkup terbatas hanya pada suatu wilayah tertentu atau hanya bersifat *Local Network* seperti halnya pada organisasi perusahaan dan lembaga-lembaga atau dalam suatu gedung.

2.3.2 *World Wide Web* (www)

Sistem pengaksesan informasi dalam internet yang paling terkenal adalah *World Wide Web* (WWW) atau biasa dikenal dengan istilah web. Pertama kali diciptakan pada tahun 1991 di CERN, laboratorium fisika partikel Eropa, Jenewa, Swiss. Tujuan awalnya adalah untuk menciptakan media yang mudah untuk berbagi informasi diantara para fisikawan dan ilmuwan.

Web menggunakan protocol yang disebut HTTP *(HyperText Transfer Protocol)* yang berjalan pada TCP/IP. Adapun dokumen web ditulis dalam format HTML *(Hyper Text Markup Language)*. Dokumen ini diletakkan dalam Web server (server yang melayani permintaan halaman web) dan diakses oleh klien (pengakses informasi) melalui perangkat lunak yang disebut web browser atau sering disebut browser saja.

Konsep *hypertext* yang digunakan pada web sebenarnya telah dikembangkan jauh hari sebelum kehadiran web. Dengan menggunakan *hypertext*, pemakai dapat

melompat dari suatu dokumen ke dokumen lain dengan mudah, dengan cukup mengklik teks-teks khusus yang pada awalnya ditandai dengan garis bawah.

2.3.3 Arti *Web Site*

Menurut penulis pengertian *Web Site* adalah "Sebuah tempat di internet dimana orang bisa menempatkan informasi , sehingga siapa saja didunia ini dapat menunjunginya, kapan saja dapat mengetahui tentang dirinya, memberikan pertanyaan kepada orang yang menggunakan *web site* ini, memberikan masukan atau bahkan mengetahui informasi yang dimasukan oleh pemiliknya".

2.4 Arsitektur Aplikasi

Arsitektur aplikasi terdiri dari pengertian jaringan komputer, Jenis-jenis jaringan komputer, topologi jaringan komputer, dan manfaat jaringan komputer.

2.4.1 Pengertian Jaringan Komputer

Jaringan komputer atau yang sering disebut dengan jaringan saja adalah hubungan dua buah simpul (umumnya berupa komputer) atau lebih yang tujuan utamanya adalah untuk melakukan pertukaran data. Dalam prakteknya, jaringan komputer memungkinkan untuk melakukan berbagi perangkat lunak, perangkat keras, dan bahkan kekuatan pemprosesan.

2.4.2 Jenis-Jenis Jaringan Komputer

1. LAN

   LAN (*Local Area Network*) merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer.

2. MAN

   MAN (*Metropolitan Area Network*) merupakan versi LAN yang berukuran lebih besar dan biasanyamemakai teknologi yang sama dengan LAN.

3. WAN

   WAN (*Widel Area Network*) merupakan sebuah jaringan yang memiliki jarak yang sangat luas, karena radiusnya mencakup sebuah negara dan benua.

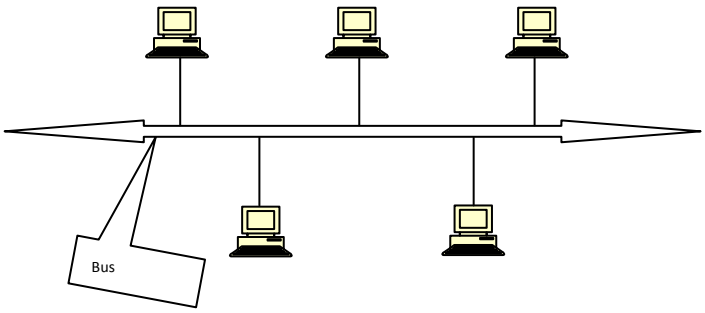
4. Jaringan tanpa kabel.

2.4.3 Topologi Jaringan Komputer

Topologi Jaringan adalah gambaran secara fisik dari pola hubungan antara komponen-komponen jaringan, yang meliputi *server, workstation, hub* dan pengkabelannnya. Terdapat tiga macam topologi jaringan umum digunakan, yaitu *Bus* (Garis Lurus), *Star* (Bintang), *Ring* (Cincin) dan *Tree* (Pohon).

1. *Linear Bus* (Garis Lurus)

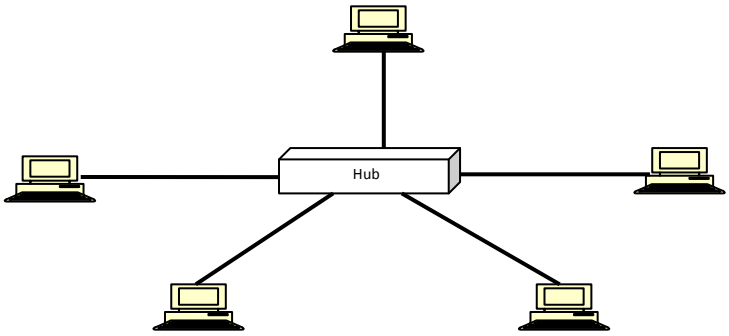
   Topologi *Linear Bus* (Garis Lurus) terdiri dari satu jalur kabel utama dimana pada masing-masing ujungnya diberikan sebuah terminator. Semua *nodes* pada jaringan (file server, worstation, dan perangkat lainnya) terkoneksi sebuah kabel utama (*backbone*). Jaringan-jaringan *Ethernet* dan *Local Talk* menggunakan topologi linear ini.



(Sumber: Abdul Kadir, 2003: 353)
Gambar 2.2. Topologi Jaringan komputer Linear *Bus*

2. *Star* (Bintang)

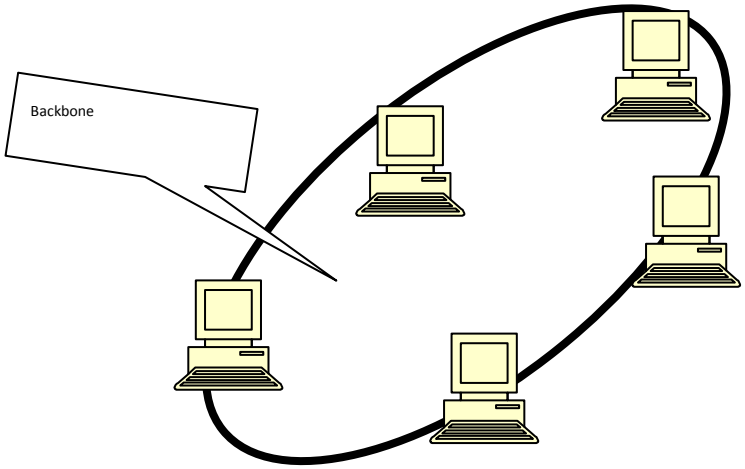
   Topologi model ini dirancang, yang mana setiap *nodes* (*file server*, *workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator*. Data yang dikirim ke jaringan lokal akan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. *Condentrator* akan mengatur dan 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*.



(Sumber: Abdul Kadir, 2003: 355)
Gambar 2.3. Topologi Jaringan komputer *Star*

3. *Ring* (Cincin)

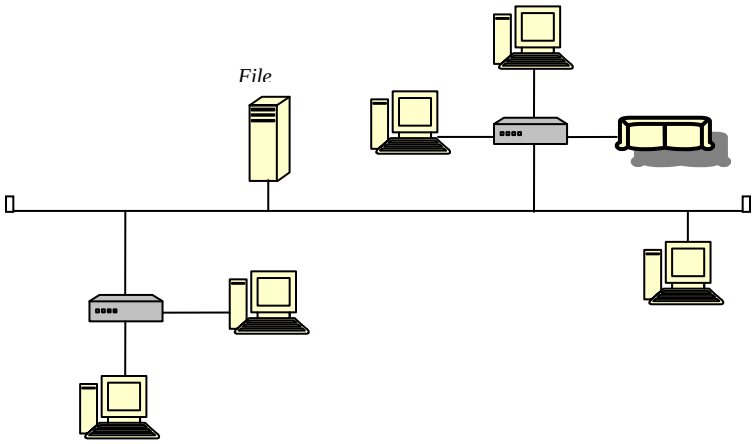
   Topologi *Ring* (Cincin) menggunakan teknik konfigurasi yang sama dengan topologi star tetapi pada topologi ini terlihat bahwa jalur media transmisi menyerupai suatu lingkaran tertutup menyerupai cincin (lingkaran), sehingga diberi nama topologi bintang dalam lingkaran atau *star-wired ring.*



(Sumber: Abdul Kadir, 2003: 354)
Gambar 2.4. Topologi Jaringan komputer *Ring*

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.



(Sumber: Abdul Kadir, 2003: 356)
Gambar 2.5. Topologi Jaringan komputer *Tree*

2.4.4 Manfaat Jaringan Komputer

Manfaat yang didapat dari membangun jaringan komputer adalah sebagai berikut:

1. *Sharing Resources*

   Sharing resources bertujuan agar seluruh program, peralatan/ periperal lainnya dapat dimanfaatkan oleh setiap orang yang ada pada jaringan komputer tanpa terpengaruh pada jaringan komputer tanpa terpengaruh oleh

lokasi maupun pengaruh dari pemakai. Dengan kata lain, seorang pemakai yang letaknya jauh sekalipun dapat memanfaatkan data maupun informasi yang lainnya tanpa mengalami kesulitan.

2. Media Komunikasi

   Jaringan komputer memungkinkan terjadinya komunikasi antara pengguna, baik untuk *teleconference* maupun untuk mengirim pesan/ informasi yang penting lainnya.

3. Integrasi Data

   Pembangunan jaringan komputer dapat mencegah ketergantungan pada komputer pusat. Setiap proses data tidak harus dilakukan satu komputer saja, melainkan dapat didistribusikan ke tempat lainnya oleh sebab inilah maka dapat terbentuk data yang terintegrasi sehingga dengan demikian memudahkan pemakai untuk memperoleh dan mengolah informasi setiap saat.

4. Pengembangan dan Pemeliharaan

   Dengan adanya jaringan komputer ini, maka perkembangan peralatan dapat dilakukan dengan mudah dan menghemat biaya. Jaringan komputer bisa memudahkan pemakai dalam merawat *harddisk* dan peralatan lainnya. Misalnya untuk memberikan perlindungan serangan *virus* maka pemakai cukup memusatkan perhatian pada *harddisk* yang ada di komputer pusat.

5. Keamanan Data

   Sistem jaringan komputer memberikan perlindungan terhadap data jaminan keamanan data tersebut di berikan melalui pengaturan hak akses para pemakai dan password, serta teknik perlindungan terhadap harddisk sehingga data mendapatkan perlindungan yang efektif.

6. Sumber Daya Lebih Efisien dan Informasi Terkini

   Dengan adanya pemakaian sumber daya secara bersama-sama maka pemakai bisa mendapatkan hasil dengan maksimal informasi yang di akses selalu terbaru, karena setiap ada perubahan yang terjadi dapat secara langsung di ketahui oleh setiap pemakai.

2.5 Perangkat Lunak Pendukung

Komputer tidak berguna tanpa keberadaan perangkat lunak (*software*). Komputer bekerja atas dasara instruksi. Sekumpulan instruksi diberikan untuk mengendalikan perangkat keras komputer. Sekumpulan instruksi inilah yang dikenal dengan sebutan program atau program komputer. Secara lebih umum, program komputer inilah yang disebut perangkat lunak.

Perangkat lunak biasa dikelompokkan menjadi program aplikasi (*application program*) dan program sistem (*system program*)

Program sistem adalah program yang digunakan untuk mengontrol sumber daya komputer, seperti CPU dan piranti masukan/keluaran. Kedudukan program ini

adalah sebagai perantara antara program aplikasi dan perangkat keras komputer. Itulah sebabnya, peran program sistem sering kali tidak terlihat secara langsung.

Program sistem dapat dikelompokkan menjadi tiga golongan, yaitu program pengendalian sistem, program pendukung sistem, dan program pengembangan sistem.

Program aplikasi adalah program yang dibuat oleh pemakai yang ditujukan untuk melakukan suatu tugas khusus. Program seperti ini biasa dikelompokan menjadi dua, yaitu program aplikasi serbaguna dan program aplikasi spesifik.

2.5.1 PHP

Bahasa pemograman yang digunakan penulis adalah PHP (*Personal Home Page tool*), PHP ini bahasa pemograman untuk membuat *Website* yang dinamis dan mudah untuk di *Update* setiap saat dari *Browser,* dibutuhkan sebuah program yang mampu mengolah data dari komputer *Client* atau dari komputer *Server* itu sendiri sehingga mudah dan nyaman untuk disajikan di *Browser*. Salah satu program yang dapat dijalankan di *Server* dan cukup handal adalah PHP. PHP adalah salah satu bahasa pemograman yang berjalan dalam sebuah *Web Server* dan berfungsi sebagai pengolahan data pada sebuah *Server*. Dengan menggunakan program PHP, sebuah *Website* akan lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung *Website/*komputer *Client* akan diolah dan disimpan pada *Database Web Server* dan dapat ditampilkan kembali apabila diakses. Untuk menjalankan kode-kode program PHP ini, file data atau file dari komputer *Client* ke dalam *Web Server*.

Banyak sekali pemrograman yang berjalan pada *Server*, masing-masing memililki kelebihan dan kekurangan. Pada saat ini banyak *Website* yang menggunakan program PHP sebagai dasar pengolahan data. Beberapa keunggulan yang dimiliki program PHP adalah :

1. Tingkat akses PHP yang lebih cepat dan memiliki tingkat keamanan yang tinggi.
2. PHP memiliki tingkat *Lifecycle* yang cepat, sehingga selalu mengikuti perkembangan terknologi *Internet*.
3. PHP mampu berjalan dibeberapa *Server* yang ada, misalnya *Apache, Microsoft IIS, PWS, AOLserver, phttpd, fhttpd* dan *Xitami*.
4. PHP mampu berjalan di *Linux* sebagai *Platform* system operasi utama bagi PHP, tetapi juga dapat berjalan di *FreeBSD, Unix, Solaris, Windows*, dan yang lain.
5. PHP juga mendukung akses kebeberapa *Database* yang sudah ada, baik yang bersifat *free*/gratis ataupun komersial. *Database* itu antara lain *MySQL, PosgreSQL, mSQL, Informix, Microsoft SQL Server*.
6. PHP bersifat *free*/gratis.

2.5.2 MySQL

MySQL merupakan *software* sistem manajemen *database* (Database *Management System* – DBMS) yang sangat populer dikalangan *web*, terutama dilingkunan Linux dengan menggunakan script PHP dan Perl. *Software* database ini kini telah tersedia juga pada platform sistem operasi Windows (98 / ME atau pun NT / 2000 / XP).

MySQL merupakan database yang paling popular digunakan untuk membangun aplikasi *web* yang menggunakan database sebagai sumber dan pengelola datanya.

Kepopuleran MySQL dimungkinkan karena kemudahannya untuk digunakan, cepat secara kinerja *query*, dan mencukupi untuk kebutuhan database perusahaan – perusahaan skala menengah – kecil. MySQL merupakan database yang digunakan oleh situs – situs terkemuka di *Internet* untuk menyimpan datanya.

Software database MySQL kini dilepas sebagai *software* manajemen database yang open source, sebelumnya merupakan software database yang shareware. Shareware adalah suatu *software* yang dapat didistribusikan secara bebas untuk keperluan penggunaan secara pribadi, tetapi jika digunakan secara komersial maka pemakai harus mempunyai lisensi dari pembuatnya. *Software* open source menjadikan *software* dapat didistribusikan secara bebas dan dapat dipergunakan untuk keperluan pribadi atau pun komersial, termasuk didalamnya source code dari *software* tersebut.

Database MySQL tersedia secara bebas – Cuma-cuma dan boleh digunakan oleh setiap orang, dengan lisensi open source GNU General Public License (GPL) atau pun lisensi komersial non GPL. Saat ini diperkirakan lebih dari 3 juta pemakai di seluruh dunia, dengan lebih dari setengah juta server yang memasangnya, termasuk didalamnya Yahoo!, MP3.com, Motorola, NASA, Silicon Graphics, HP, Xerox, Cisco, dan Texas Instruments.

Database MySQL merupakan database yang menjanjikan sebagai alternatif pilihan database yang dapat digunakan untuk sistem database personal atau organisasi kita. Oracle sebagai database besar telah membuat kit (modul) untuk memudahkan proses migrasi dari MySQL ke dalam Oracle, hal ini dapat menunjukkan bahwa Oracle telah memperhitungkan database MySQL sebagai database alternatif masa depan. Demikian juga dengan pengguna dari database MySQL, menunjukkan makin banyaknya perusahaan besar menggunakannya.

MySQL dikenal sebagi database yang pertama kali didukung oleh bahasa pemrograman script untuk Internet – PHP dan Perl. MySQL dan PHP dianggap sebagai pasangan *software* pengembangan aplikasi berbasis *web* yang ideal.

Antar muka (front end) untuk aplikasi database MySQL dapat menggunakan bahasa pemrograman umum seperti bahasa pemrograman Java, C/C++, MS Visual BASIC, atau pun Borland Delphi, hasil akhir dari model aplikasi yang dihasilkan adalah aplikasi *Client / Server*. Umumnya akses kepada database MySQL dari bahasa pemrograman yang disebutkan jika di lingkungan Windows menggunakan MyODBC, driver koneksi database dengan menggunakan standar ODBC.

2.5.3 Apache

Apache adalah web browser *open source* yang tersedia di berbagai platform, termasuk Linux dan Windows. Apache dikembangkan berdasarkan kode-kode dan ide dari NCSA httpd 1.3. apache merupakan server HTTP yang paling terkenal pada awal tahun 1995.

Nama apache merupakan singkatan dari "*A PATCHY SERVER*" yang kira-kira berarti server yang dibangun dari kode-kode yang telah ada dan file patch. Menurut *Neteraft* survey, Apache saat ini adalah server yang paling popular dipakai di internet.

2.5.4 HTML (Hyper Text Markup Language)

HTML adalah suatu bahasa pemrograman yang dipergunakan untuk menyusun dan membentuk suatu dokumen agar dapat ditampilkan pada program browser world wide web dalam bentuk yang dikehendaki oleh pembuatnya.

HTML juga dipergunakan untuk menciptakan *hypertext link* atau hubungan antara teks dan dokumen lain atau juga bagian dari dokumen lain, HTML hanya berupa file teks yang bisa dibaca dengan suatu editor teks biasa, sesuai dengan namanya bahasa ini menggunakan tanda (*markup*) untuk menandai perintah-perintahnya.

2.5.5 Javascript

Merupakan sosok pemrograman *modern*, yaitu bahasa pemrograman berorientasi script (*object-oriented scripting language*), yang berbasis kepada kosep *prototype*. Bahasa ini banyak dikenal di dalam lingkungan *website*. Bahasa ini diperkenalkan pertama kali oleh Brendan Eich dari perusahaan *Netscape Corporation* dengan nama awal Mocha kemudian berubah menjadi *LiveScript* namun diganti akhirnya menjadi *Javascript.*

2.5.6 Rational Rose 2002 Enterprise Edition

Dalam hal pembuatan CASE terpopuler software yang salalu digunakan adalaha Rational Rose 2002 Enterprise Edition , software ini membantu dalam pengembangan sistem atau perangkat lunak yang bermetodologi berorientasi objek.

2.6 Pengertian Pemrogramman Berorientasi Objek

Dalam perkembangan Rekayasa Perangkat Lunak Pemograman Berorientasi Objek Merupakan hal perkembangan yang baru dikarenakan sistemnya di pandang sebagai kumpulan-kumpulan objek diskrit yang saling berinteraksi.

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi objek. Berikut empat prinsip dasar dari pemograman berorientasi objek yang terdiri dari :

1. Abstraksi : memfokuskan pada karakteristik objek
2. Enkapsulasi : Menyembunyikan banyak hal yang terdapat dalam objek yang tidak perlu diketahui objek lain
3. Modularitas : Membagi sistem yang rumit menjadi bagian-bagian yang lebih kecil
4. Hirarki : berhubungan dengan abstraksi dan modularitas yaitu pembagian berdasarkan urutan dan pengelompokan sesuatu

2.6.1 Pengertian UML

Unified Modeling Language (UML) merupakan sistem arsitektur yang bekerja dalam OOAD (*Object-Oriented Analysis/Design*) dengan satu bahasa yang konsisten

untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasikan *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa software, dapat berupa model, deskripsi, atau software) yang terdapat dalam sistem software. UML merupakan bahasa pemodelan yang paling sukses dari tiga metode OO yang telah ada sebelumnya, yaitu Booch, OMT (*Object Modeling Technique*), dan OOSE (*Object-Oriented Software Engineering*).

UML merupakan kesatuan dari dari ketiga pemodelan tersebut dan ditambah kemampuan lebih karena mengandung metode tambahan untuk mengatasi masalah pemodelan yang tidak dapat ditangani ketiga metode tersebut. UML dikeluarkan oleh OMG (Object Management Group, Inc) yaitu organisasi internasional yang dibentuk pada 1989, terdiri dari perusahaan sistem informasi, *software developer*, dan para *user* sistem komputer.

Beberapa konsep dasar dalam OOAD :

1. *Object*

   Objek (*object*) adalah "benda", secara fisik atau konseptual, yang dapat kita temui disekeliling kita. Hardware, software, dokumen, manusia, dan bahkan konsep semuanya adalah contoh objek. Untuk kepentingan memodelkan perusahaannya, seorang kepala eksekutif akan melihat karyawan, gedung, divisi, dokumen, dan keuntungan sebagai objek. Seorang teknisi mesin akan melihat ban, pintu, mesin, laju tertinggi, dan banyaknya bahan bakar sebagai objek. Dan, seorang *software engineer* akan memandang tumpukan, antrian,

jendela (window), dan check box sebagai objek. Sebuah objek memiliki keadaan sesaat (state) dan perilaku (behavior). State dari sebuah objek adalah kondisi objek tersebut atau himpunan dari keadaan yang menggambarkan objek tersebut. Sebagai contoh, bola lampu adalah objek, dan salah satu keadaan nyala atau tidak adalah state dari objek bola lampu tersebut. Salah satu state dari objek jam adalah waktu saat ini. State dinyatakan dengan nilai dari atribut (attribute) objeknya. Atribut adalah nilai internal suatu objek yang mencerminkan antara lain karakteristik objek, kondisi sesaat, koneksi dengan objek lain, dan identitasnya. Perubahan state dicerminkan oleh perilaku (behavior) objek tersebut.

Behavior suatu objek mendefinisikan bagaimana sebuah objek bertindak (beraksi) dan memberi reaksi. Behavior ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan dalam objek itu sendiri. Behavior dari sebuah objek dicerminkan oleh interface, service, dan method dari objek tersebut. Interface adalah pintu untuk mengakses *service* objek. Service adalah fungsi yang bisa diemban objek. Method adalah mekanisme internal objek yang mencerminkan perilaku (behavior) objek tersebut. Sebagai contoh, jika printer sebuah merupakan perilaku (behavior) atau service-nya mencetak apapun yang dia terima.

2. *Class*

   Kelas (class) adalah definisi umum (pola, template atau cetak biru) untuk himpunan objek sejenis. Kelas menetapkan spesifikasi perilaku (behaviors) dan atribut objek-objek tersebut. Class adalah keniskalan (abstraksi) dari entitas dalam dunia nyata. Objek adalah "contoh" (instance) dari sebuah kelas. Sebagai contoh, atribut untuk kelas binatang adalah berkaki empat dan memiliki ekor. Perilakunya adalah tidur dan makan. Contoh (instance) yang mungkin dari kelas binatang ini adalah kucing, gajah, dan kuda.

3. *Inheritance*

   Seperti yang sudah diuraikan di atas, objek adalah contoh/ instance dari sebuah class. Hal ini mempunyai konsekuensi yang penting yaitu sebagai instande sebuah class, sebuah objek mempunyai semua karakteristik dari classnya. Inilah yang disebut dengan inheritance (pewarisan sifat). Dengan demikian apaun attribute dan operation dari class akan dimiliki pula oleh semua objek yang diinherit/ diturunkan dari class tersebut. Sifat ini tidak hanya berlaku untuk objek terhadap class, akan tetapi juga berlaku untuk class terhadap class lainnya.

4. *Polimorphisme*

   Polimorphisme adalah konsep yang handal bagi pengembangan perangkat lunak untuk pemisahan secara jelas diantara sub sistem yang berbeda. Dengan

demikian sebuah sistem akan bisa dimodifikasi secara mudah karena hanya dibutuhkan interface antar class.

5. *Encapsulation*

   Encapsulation adalah proses menyembunyikan detil implementasi sebuah objek. Satu-satunya jalan untuk mengakses data objek tersebut adalah melalui interface. Interface melindungi internal state sebuah objek dari "campur tangan" pihak luar. Oleh karena itu objek sering digambarkan sebagai kotak hitam (black box) yang menerima dan mengirim pesan-pesan (messages). Dalam object-oriented programming kotak hitam tersebut berisi kode (himpunan intruksi dengan bahasa yang dipahami komputer) dan data (informasi dimana intruksi tersebut beroperasi dengannya). Dalam object-oriented programming, kode dan data disatukan dalam sebuah "benda" yang tersembunyi isinya, yaitu objek. Pengguna objek tidak perlu mengetahui isi dalam kotak tersebut. Untuk dapat berkomunikasi dengan objek, diperlukan pesan (messages). Secara formal kita definisikan message sebagai permintaan untuk objek penerima (receiver object) untuk membawa metode yang ditunjukkan atau perilaku dan mengembalikan result dari aksi tersebut kepada objek pengirim (sender object). Sebagai contoh, satu object orang mengirim object bola lampu sebuah pesan (message) untuk menyala (melalui saklar). Objek bola lampu menyalakan dirinya dan menunjukkan kepada objek orang tersebut bahwa state barunya adalah menyala.

6. *Association dan Aggregation*

   Association (asosiasi) adalah hubungan antar objek yang saling membutuhkan. Sedangkan, *aggregation* (agregasi) adalah bentuk khusus dari asosiasi yang menggambarkan seluruh bagian suatu objek merupakan bagian dari objek lainnya. Sebagai contoh, objek tanggal dapat disusun dari objek hari, objek bulan, dan objek tahun. Satu alasan mengapa object-oriented menguntungkan bagi programmer adalah karena programmer dapat mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata. Keuntungan yang lain adalah proses pembuatan software dapat dilakukan dengan lebih cepat karena software dibangun dari object-object standar, dapat menggunakan model yang ada, dan dapat membuat model dengan cepat melalui metodologi. Kualitas yang tinggi dari software dapat dicapai karena adanya tested components. Lebih mudah dalam maintenance karena perbaikan kode hanya diperlukan pada satu tempat (bukan diurut dari awal). Mudah dalam membangun sistem yang besar karena subsistem dapat dibuat dan diuji secara terpisah. Mengubah sistem yang sudah ada tidak memerlukan membangun ulang keseluruhan sistem.

2.6.2 Alat Bantu UML

Sebagaimana yang telah dijelaskan diatas tentang UML, bahwa untuk mendapatkan banyak pandangan terhadap sistem informasi yang akan dibangun,

UML menyediakan beberapa alat bantu diagram visual yang menunjukan berbagai aspek dalam sistem. Ada beberapa alat bantu diagram yang disediakan dalam UML antara lain :

1. *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.

2. *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*. Hubungan Antar *Class* adalah 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.

3. *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, dimana 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 vertical. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

4. *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.

5. *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.

6. *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.

7. *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.