BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem

Menurut Schmidt & Taylor menyatakan sistem sebagai berikut : **“Sistem adalah suatu kumpulan komponen (Entitas) yang berinteraksi dan bereaksi antar atribut komponennya untuk mencapai suatu tujuan akhir yang logis”DIK[3].**Misalnya sistem teologi adalah susunan yang teratur dari gagasan-gagasan tentang Tuhan, manusia dan sebagainya. Sistem bersifat fisis adalah serangkaian unsur yang bekerja sama untuk mencapai suatu tujuan”.

Hick berpendapat bahwa “**sistem adalah kumpulan komponen yang saling berinteraksi dan beroperasi dalam suatu pembatasan. Pembatas akan menyaring setiap jenis dan tingkat aliran input dan output antara sistem dan lingkungannya.” DIK[3].].**

Murdick & Ross, menyatakan bahwa “**Sistem adalah sekumpulan elemen-elemen yang berinteraksi dan membentuk suatu kesatuan terintegrasi.” DIK[3].**

Dari sekian banyak arti dari kata sistem maka penulis mengambil kesimpulan bahwa sistem adalah suatu kesatuan utuh yang terdiri dari beberapa bagian yang saling berhubungan dan berinteraksi untuk mencapai tujuan tertentu. Juga dapat disimpulkan bahwa:

1. Sistem terdiri dari beberapa komponen/elemen/unsure/entitas.
2. Sistem disatukan oleh hubungan antara komponen tersebut.
3. Sistem memiliki suatu tujuan.
4. Sistem memiliki pembatas.

2.1.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, antara lain sebagai berikut**(DIK[3])** :

1. Komponen Sistem *(Component*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan.

2. Batas Sistem *(Boundary)*

   Batas sistem merupakan daerah yang membatasi antara satu sistem dengan sistem lain atau dengan lingkungan luar.

3. Lingkungan Luar Sistem *(Environment)*

   Lingkungan luar sistem dari satu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem *(Intervace)*

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lain.

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 dimasukan supaya sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapatkan keluaran.

6. Keluaran Sistem *(Output)*

   Keluaran *(output)* adalah hasil dari energi yang diolah dan diklarifikasi menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem *(Process)*

   Suatu sistem biasa mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

   Suatu sistem pasti mempunyai tujuan *(goal)* dan sasaran *(objective)*. Suatu sistem dikatakan berhasil apabila mengenai sasaran atau tujuan dari sistem tersebut.

2.1.2 Klasifikasi Sistem

Sistem dapat dikelompokan atau diklasifikasikan menjadi beberapa kelompok karena sistem bersifat umum maka ada baiknya untuk memahami berbagai konsep kategori sistem melalui identifikasi terhadap sistem yang dimaksud untuk menyajikan perilaku dan karakteristiknya**(DIK[3])**.

1. Sistem diklasifikasikan sebagai sistem Abstrak *(abstract system)* dan sistem Fisik *(physical system)*

   Sistem abstrak adalah suatu susunan yang teratur dari gagasan atau konsepsi yang saling tergantung satu sama lain. Sistem fisik adalah kumpulan elemen-elemen yang beroperasi secara bersama-sama untuk mencapai tujuannya.

2. Sistem diklasifikasikan sebagai sistem Alam dan Buatan

   Sistem alamiah adalah sistem yang terbentuk dengan sendirinya yang dapat ditemui di alam bebas. Sedangkan sistem buatan adalah sistem yang diciptakan untuk mencapai tujuan tertentu.

3. Sistem diklasifikasikan sebagai sistem Terbuka dan Tertutup

   Sistem terbuka adalah sistem yang mampu berinteraksi dengan lingkungannya dimana dimungkinkan adanya pertukaran materi, energi, maupun informasi

dengan lingkungannya. Sistem tertutup adalah sistem yang tidak mempunyai relasi atau interaksi terhadap lingkungannya.

4. Sistem diklasifikasikan sebagai sistem Permanen dan Sementara

   Semua sistem yang berlaku untuk rentang waktu yang cukup panjang, dibandingkan dengan kegiatan manusia dalam sistem tersebut, dapat digolongkan sebagai sistem permanent. Sedangkan sistem yang bersifat sementara diadakan untuk jangka waktu tertentu saja dan sesudahnya dihapuskan atau dimodifikasi kembali dapat digolongkan sebagai sistem sementara.

2.2 Konsep Dasar Informasi

Menurut Gordon B.Davis dalam bukunya *Management Information System : Conceptual Foundation, structures, and Development* menyebutkan informasi sebagai data yang telah diolah menjadi bentuk yang lebih berguna bagi penerimanya dan nyata, berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan. **WAH[9].**

Menurut Barry E. Cushing dalam bukunya *Accounting Information System and Business Organization,* dikatakan bahwa informasi merupakan sesuatu yang menunjukan hasil pengolahan data yang diorganisasi dan berguna kepada orang yang menerimanya.

Menurut Robert N. Anthony dan John Dearden dalam bukunya *Management Control System,* menyebutkan informasi sebagai suatu kenyataan, data, item yang menambah pengetahuan bagi penggunanya. **WAH[9].**

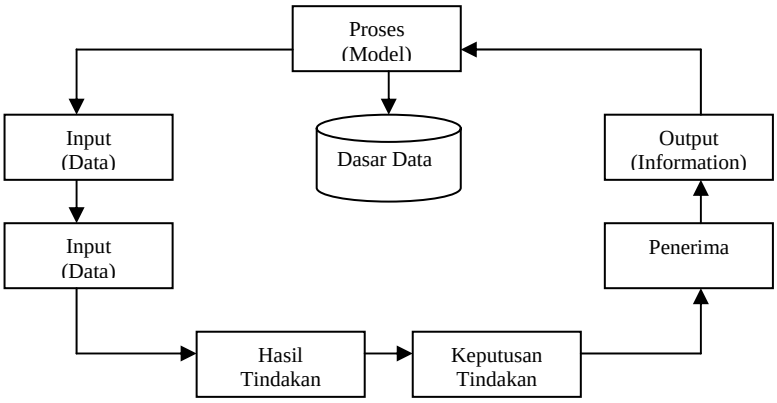
Menurut Stephen A. Moscove dan Mark G. Simkin dalam bukunya *Accounting Information System : Concepts and Practice* mengatakan informasi

sebagai kenyataan atau bentuk-bentuk yang berguna yang dapat digunakan untuk pengambilan keputusan bisnis. **WAH[9].**

Dari keempat pengertian seperti tersebut di atas dapat disimpulkan bahwa informasi merupakan hasil dari pengolahan data mentah menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambar suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan keputusan.

2.2.1 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data diolah melalui suatu model untuk menghasilkan informasi, informasi sendiri merupakan suatu proses perubahan data menjadi informasi. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut membuat suatu keputusan dan melakukan tindakan yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya sehingga membentuk suatu siklus.



Gambar 2.1. Siklus Informasi

(Sumber : JOG[5])

2.2.2 Kualitas Informasi

Kualitas informasi merupakan salah satu yang benar-benar diperhatikan karena tingkat nilai suatu informasi ditentukan oleh kualitas informasi itu sendiri, kualitas informasi tergantung dari 3 (tiga) hal, yaitu**(WAH[9])** :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya.
2. Tepat waktu, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi merupakan landasan dalam pengambilan keputusan sehingga apabila keputusan terlambat dapat berakibat fatal pada perusahaan.

Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakai. Relevan informasi untuk tiap-tiap orang antara satu dengan yang lain bisa berbeda.

2.2.3 Pengertian Sistem Informasi

Sistem informasi adalah kumpulan dari berbagai atau komponen atau subsistem baik fisik ataupun nonfisik yang saling berhubungan satu sama lain dan bekerjasama secara harmonis untuk mencapai suatu tujuan tertentu, yaitu mengolah data menjadi informasi**.(WAH[9])**.

Pengertian Sistem Informasi adalah : “Komponen yang saling berhubungan dan bekerjasama untuk mengumpulkan, memproses, menyimpan dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengendalian dan untuk memberikan gambaran aktivitas di dalam perusahaan**”.(WAH[9])**

Pendapat lain mengatakan bahwa Sistem informasi adalah sebagai berikut“Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan

kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan**.(WAH[9])**.

2.3 Pengelolaan Perpustakaan

1. Pengertian Pengelolaan

Pengelolaan (Manajemen), merupakan proses merencanakan, memimpin, dan mengendalikan pekerjaan anggota organisasi dan menggunakan semua sumber daya organisasi untuk mencapai sasaran organisasi yang sudah di tetapkan.**(ILM[4])**

2. Pengertian Perpustakaan

Perpustakaan adalah sebuah koleksi buku dan majalah. Dia dapat menunjuk ke koleksi pribadi perseorangan, tetapi lebih umum sebuah koleksi besar yang dibiayai dan dirawat oleh sebuah kota atau institusi, dan dibagi oleh banyak orang yang tidak mampu membeli banyak buku untuk diri mereka sendiri**.(ILM[4])**

Tetapi, dengan koleksi dan penemuan media baru selain buku untuk menyimpan informasi, banyak perpustakaan kini juga merupakan tempat penyimpanan dan/atau akses ke map, cetak atau hasil seni lainnya, mikrofilm, mikrochips, tape audio, CD, LP, tape video dan DVD, dan menyediakan fasilitas umum untuk mengakses gudang data CD-ROM dan internet.

3. Pengertian Buku

Buku adalah sebuah media penyambung ilmu yang efektif bagi pembacanya yang dapat memberikan ilmu dan pengetahuan**.(ILM[4])**, banyak sekali manfaat yang terkandung jika membaca buku, selain menambah pengetahuan juga

memberikan kesenangan tersendiri Namun entah mengapa minat membaca buku di negara ini masih sangat kurang, padahal dengan membaca buku maka satu ilmu didapatkan.

4. Pengertian Anggota

Anggota adalah setiap orang yang telah terdaftar secara resmi sebagai suatu kesatuan dalam suatu organisasi, yang mempunyai hak serta kewajiban yang telah diatur berdasarkan kebijakan dari managemen suatu organisasi. **.(ILM[4])**

5. Pengertian Peminjaman

Peminjaman adalah suatu proses dimana setiap orang yang terdaftar secara resmi sebagai anggota suatu organisasi mempunyai hak untuk melakukan peminjaman sesuai dengan bentuk dari organisasi. **.(ILM[4])**

6. Pengertian Pengembalian

Pengembalian adalah suatu proses dimana setiap orang yang melakukan peminjaman sejumlah tertentu mempunyai kewajiban untuk mengembalikan apa yang sudah dipinjam sesuai dengan peraturan yang berlaku diorganisasi tersebut.**(ILM[4])**

7. Pengertian Denda

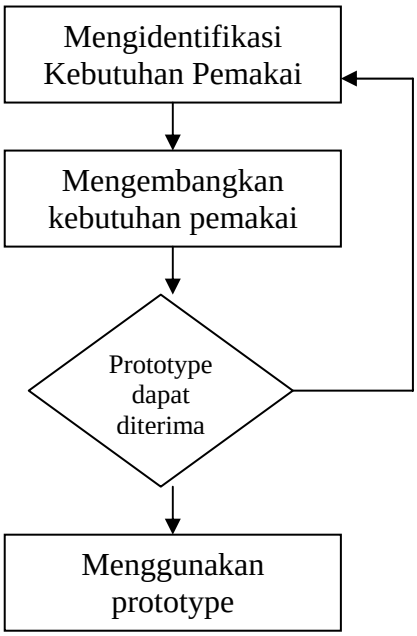
Denda adalah suatu proses dimana setiap orang/anggota yang melakukan pelanggaran terhadap ketentuan yang telah dibuat atau diatur oleh suatu organisasi akan dikenakan sangsi sesuai dengan aturan yang telah ditetapkan oleh organisasi tersebut. **.(ILM[4])**

2.4 Metode Pengembangan Sistem

2.4.1 Prototype

Metode yang digunakan untuk Sistem Informasi Pengelolaan Perpustakaan adalah *prototype*. *Prototipe* merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai.

Adapun teknik pengembangan sistem yang terlihat pada gambar 1.1 langkah-langkah metode penelitian :



Gambar 2.2 Metode Prototype 1

(sumber : NUG[8])

2.4.2 Waterfall

Model ini telah diperoleh dari proses engineering lainnya. Model ini menawarkan cara pembuatan perangkat lunak secara lebih nyata. Langkah-langkah yang penting dalam model ini adalah :

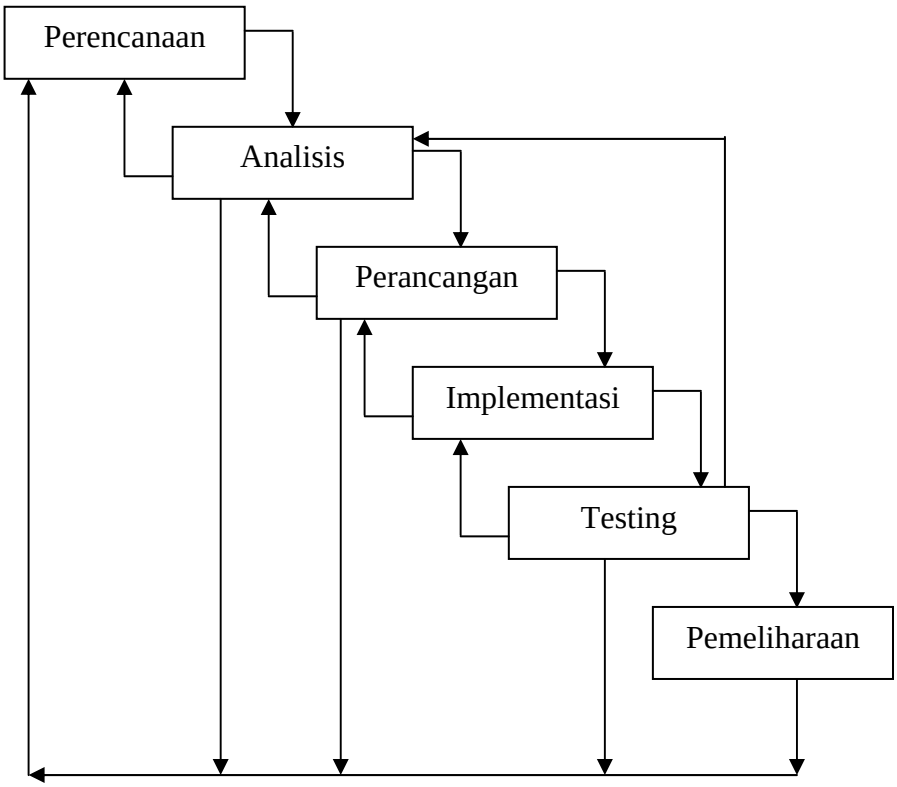
1. Penentuan dan analisis spesifikasi Jasa, kendala dan tujuan dihasilkan dari konsultasi dengan pengguna sistem. Kemudian semuanya itu dibuat dalam bentuk yang dapat dimengerti oleh user dan staf pengembang.

2. Desain sistem dan perangkat lunak, Proses desain sistem membagi kebutuhan-kebutuhan menjadi sistem perangkat lunak atau perangkat keras. Proses tersebut menghasilkan sebuah arsitektur sistem keseluhan. Desain perangkat lunak termasuk menghasilkan fungsi sistem perangkat lunak dalam bentuk yang mungkin ditransformasi ke dalam satu atau lebih program yang dapat dijalankan.

3. Implementasi dan ujicoba unit, Selama tahap ini desain perangkat lunak disadari sebagai sebuah program lengkap atau unit program. Uji unit termasuk pengujian bahwa setiap unit sesuai spesifikasi.

4. Integrasi dan ujicoba system, Unit program diintegrasikan dan diuji menjadi sistem yang lengkap untuk menyakinkan bahwa persyaratan perangkat lunak telah dipenuhi. Setelah ujicoba, sistem disampaikan ke kastamer

5. Operasi dan pemeliharaan Normalnya, ini adalah phase yang terpanjang. Sistem dipasang dan digunakan.



Gambar 2.3 Metode Waterfall

(sumber : NUG[8])

Berdasarkan penjelasan mengenai metode prototype dan waterfall di atas akhirnya penulis memutuskan menggunakan metode prototype untuk pengembangan dalam tugas akhir ini dengan alasan metode prototype ini lebih mudah di implementasikan di bandingkan dengan waterfall yang lebih panjang dalam phase penerapannya dan ketidak luwesan pembagian proyek kedalam langkah yang nyata/jelas. Sedangkan system yang di sampaikan kadang tidak sesuai dengan keinginan pelanggan.

2.5 Analisis dan Perancangan Object Oriented

2.5.1 Teknologi Object Oriented

Teknologi *object-oriented* merupakan paradigma baru dalam rekayasa software yang didasarkan pada objek dan kelas. *Object oriented* memandang

software bagian perbagian, dan menggambarkan satu bagian tersebut dalam satu objek. Satu objek dalam sebuah model merupakan suatu fokus selama dalam proses analisis, desain, dan implementasi dengan menekankan pada *state*, perilaku (*behavior*), dan interaksi objek–objek dalam model tersebut.

Orientasi *Object Oriented* menekankan konsep berikut**(ILM[4]):**

1. ***Class*** adalah kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh *'class of dog'* adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.
2. ***Object*** adalah membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
3. ***Abstraksi*** adalah Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" *abstrak* yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi

dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat *abstrak*, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

4. ***Enkapsulasi*** adalah Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi ijin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

5. ***Polimorfisme*** melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

6. ***Inheritansi*** Mengatur polimorfisme dan enskapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada

objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas.)

2.5.2 Metodologi Object Oriented Analysis and Desain(OOAD)

Metodologi adalah cara sistematis untuk mengerjakan pekerjaan analisis dan desain. Dengan motodologi, pihak yang membangun sistem software dapat merencanakan dan mengulangi pekerjaannya di lain waktu. Metodologi yang paling banyak digunkanan dalan *Object oriented analysis and desain*, diantaranya**:(ASU & HAR[1])**

1. Rumbaugh OMT berdasarkan pada analisis terstruktur dan pemodelan *entity-relationship*. Tahapan utama dalam metode ini adalah analisis, desain sistem dan desain objek, dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mengandung semua konsep *Object Oriented*.
2. Metode Shlaer-Mellor menggunakan teknik pemodeln tradisional untuk menjelaskan entitas dalam sistem menggunakan *state diagram* untuk memodelkan keadaan (*state*) entitas, mengunakan *data flow diagram* untuk memodelkan alur data dalam sebuah sistem.
3. Metode Booch sangat terkenal dengan nama metode desain o*bject oriented*. Metode ini menjadikan proses analisis dan desain dalam empat tahapan iteratif (dapat berulang), yaitu identifikai kelas-kelas dan objek-objek, identifikasi *semantic* dan hubungan objek dan kelas tersebut, perincian *interface* dan implementasi.

4. RDD/CRC dari Wirfs-Brock diarahkan pada desain, tetapi sangat berguna untuk memunculkan ide dalam tahap analisis. Keunggulannya adalah mudah untuk digunakan.
5. Metode Coad/Yourdon didasarkan pada pemodelan o*bject oriented dan entry-relationship*. Metode ini menyelesaikan masalah dengan membagi kedalam lima lapisan (*layer*). Lapisan tersebut ditentukan oleh aktifitas yang berbeda.
6. Metode Object Oriented Software Engineering (OOSE) dari Jacobson mengandung elemen-elemen dari metode o*bject oriented* lainnya. Metode ini memberi penekanan lebih pada *use-case*. OOSE memiliki tiga tahapan, yaitu membuat model *requirements* dan analisis, desain dan implementasi, dan model pengujian (*test model*).

2.5.3 Object Oriented Programming (OOP)

Object Oriented Programming (OOP) atau Pemrograman berorientasi objek adalah suatu cara baru dalam berfikir serta berlogika dalam menghadapi masalah-masalah yang akan coba diatasi dengan bantuan komputer**(ASU& HAR[1]).** *Object Oriented Programming (OOP)*, tidak seperti pendahulunya (pemrograman terstruktur) yang mencoba melihat permasalahan lewat pengamatan dunia nayata dimana setiap objek adalah entitas tunggal yang memiliki kombinasi struktur data dan fungsi tertentu. Ini kontras dengan pemrograman terstruktur dimana struktur data dan fungsi didefinisikan secara terpisah dan tidak berhubungan secara erat.

Fisolofi *Object Oriented Programming (OOP)* menciptakan sinergi yang luar biasa sepanjang siklus pengembangan perangkat lunak (perencanaan, analisis, perancangan, serta implementasi) sehingga dapat diterapkan pada perancangan.

2.5.4 Konsep Dasar UML

Unified Modeling Language (UML) merupakan sistem arsitektur yang bekerja dalam *Object Oriented Analysis and Design (OOAD).* Unified Modeling Language (UML) merupakan gabungan dari metode Booch, OMT, dan Jacobson. Pada pengembangan Unified Modeling Language (UML) dilakukan standarisasi proses dengan OMG (*Object Management Group*) dengan harapan Unified Modeling Language (UML) akan menjadi bahasa pemodelan pada masa yang akan datang. Unified Modeling Language (UML) disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat**:(ASU & HAR[1])**.

2.5.5 Pengertian Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa untuk menspesifikasikan, menvisualisasi, membangun, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak, *artifact* tersebut dapat berupa model deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya. Selain itu Unified Modeling Language (UML) adalah bahasa pemodelan yang menggunakan konsep orientasi *object***:(ASU & HAR[1])**.

2.5.6 Bagian-bagian dari UML

Bagian-bagian utama dari Unified Modeling Language (UML) adalah *view*, *diagram, model element,* dan *general mechanism* **(CHO[2])**.

1. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis view dalam Unified Modeling Language (UML) antara lain : *use case view, logical view, component view, concurrency view, dan deployment view*.

a. Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan *external actors*. *Actor* yang berinteraksi dengan sistem dapat berupa *user* atau sistem lainnya.

b. Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (*class, object* dan *relationalship*) dan kolaborasi dinamis yang terjadi ketika *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu.

c. Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari *code module* diperlihatkan dengan struktur dan ketergantungan juga alokasi sumber daya komponen dan informasi administrative lainnya.

d. Concurrency view

Membagi sistem ke dalam proses dan prosesor. *View* ini digambarkan dalam diagram dinamis (*state, sequence, collaboration* dan *activity diagram*).

e. Deployment view

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan yang lain.

2. Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Jenis-jenis diagram antara lain :

a. Use Case Diagram

Menggambarkan sejumlah *external actor* dan hubungannya ke *use case* yang diberikan suatu sistem.

1. Actor

Actor menggambarkan pengguna software aplikasi *(user).* Pada dasarnya *actor* bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa *actor* dimana *actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem, *actor* hanya berinteraksi dengan *use case* tetapi tidak memiliki kontrol atas *use case*.

Adapun beberapa kemungkinan yang menyebabkan actor tersebut terkait dengan sistem, antara lain :

a) Yang berkepentingan terhadap sistem dimana adanya arus informasi baik yang diterimanya maupun yang diinputkan ke sistem.
b) Orang ataupun pihak yang akan mengelola sistem tersebut.
c) External resorce yang digunakan oleh sistem.
d) Sistem lain yang berinteraksi dengansistem yang akan dibuat.

Actor digambarkan dengan *stick man*.



Gambar 2.4. Actor

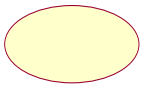
(Sumber : SUH dan HAR[1])

2. Use Case

Menggambarkan fungsional dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun.

Cara menentukan *use case* dalam suatu sistem:

a. Pola perilaku perangkat lunak aplikasi
b. Gambaran tugas dari sebuah *actor*
c. Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada *actor*
d. Apa yang dikerjakan oleh sebuah perangkat lunak (* bukan bagaimana cara mengerjakannya).



Gambar 2.5 Use Case

(Sumber : SUH dan HAR[1])

b. Class Diagram

Menggambarkan struktur statis class di dalam sistem, *class diagram* terdiri atas *object* dan *class*

1. Object

Object adalah gambaran dari entity, baik dunia nyata atau konsep dengan batasan-batasan dan pengertian yang tepat. Object bisa mewakili sesuatu yang nyata seperti komputer, mobil atau dapat berupa konsep seperti proses kimia, transaksi bank, dan lain-lain. Setiap object dalam sistem memiliki 3 (tiga) karakteristik yaitu : *State* (status), *Behaviour* (sifat) dan *Indentity* (identitas).

Cara mendidentifikasi object :

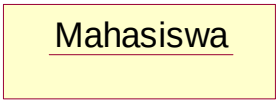
1. Pengelompokan berdasarkan kata/ frase benda pada skenario.
2. Berdasarkan dasar kategori object, antara lain :
   a) Obeject fisik, contoh : pesat telepon
   b) Spesifikasi/ rancangan/deskripsi, contoh : deskripsi pesawat
   c) Tempat, contoh : gudang
   d) Transaksi, contoh : penjualan
   e) Butir yang terlibat pada transaksi, contoh : barang jualan
   f) Peran, contoh : pelanggan
   g) Wadah, contoh : pesawat terbang

h) Kata benda abstrak, contoh : kecanduan

i) Kejadian, contoh : pendaratan

j) Aturan atau kebijakan, contoh : aturan diskon

k) Catalog atau rujukan, contoh : daftar pelanggan

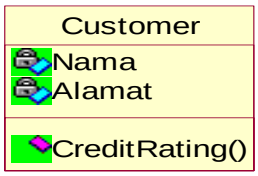


Gambar 2.6 Object

(Sumber : CHO[2])

2. Class

Class adalah deskripsi sekelompok *object* dari property (atribut), sifat (operasi), relasi antar *object* dan semantik yang umum. *Class* merupakan template untuk membentuk *object*. Setiap *object* merupakan contoh dari beberapa *class* dan *object* tidak dapat menjadi contoh dari satu *class*.



Gambar 2.7 Class

(Sumber : CHO[2])

c. State Transition Diagram

State transition diagram menunjukan *state-state* dari *object* tunggal, *event-event* atau pesan yang menyebabkan transisi dari satu *state* ke *state* lain, dan *action* yang merupakan hasil dari perubahan sebuah *state*.

State transition diagram meliputi seluruh pesan dari object yang dapat mengirim dan menerima. Skenario merepresentasikan satu jalur yang

melewati sebuah *State transition diagram.* Oleh karena itu, *sequence diagram* ditentukan untuk menemukan *state-state* sebuah *object.*

1. State

   State adalah sebuah kondisi selama kehidupan sebuah *object* ketika *object* memenuhi beberapa kondisi, melakukan beberapa *action*, atau menunggu sebuah *event*. *State* dari sebuah *object* dapat dikarakteristikkan oleh nilai satu atau lebih atribut-atribut kelas dari *class*.



Gambar 2.8. Notasi UML untuk State

(Sumber : CHO[2])

2. State Transition

   State transition mempresentasikan sebuah perubahan dari *state* awal ke sebuah *state* berikutnya (yang mungkin sama dengan *state* awal). Sebuah *action* dapat menyertai sebuah *State transition*

3. Special State

   Ada dua *state* khusus yang ditambahkan di *State transition diagram.* Pertama *start state*. Masing-masing diagram harus mempunyai dan hanya satu *start state* ketika *object* mulai dibuat. Berikutnya adalah *stop state*, sebuah *stop state* boleh mempunyai banyak *stop state*.



Start state Stop state

Gambar 2.9. Notasi UML untuk start dan stop state

(Sumber : CHO[2])

d. Sequence Diagram

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

e. Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan *object* dan hubungannya (mengacu ke konteks).

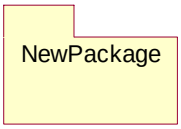
f. Activity Diagram

Menggambarkan rangkaian aliran dari aktifitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

g. Component Diagram

Menggambarkan struktur fisik kode dari komponent. Komponent dapat berupa *source code*, *component biner*, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.

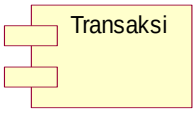
1. Package, adalah mekanisme pengelompokan yang digunakan untuk menandakan pengelompokan elemen-elemen model. Sebuah package dapat mengandung beberapa paket lain didalamnya. Package digunakan untuk memudahkan mengorganisasi elemen-elemen model.



Gambar 2. 10. Package

(Sumber : SUH and HAR[1])

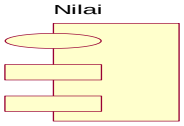
2. Component, merupakan komponen-komponen dalam desain fisik sistem software.



Gambar 2. 11. Component

(Sumber : SUH and HAR[1])

3. Package Spesification Component, menggambarkan komponen spasifikasi dari Package



Gambar 2. 12. Package Spesification Component

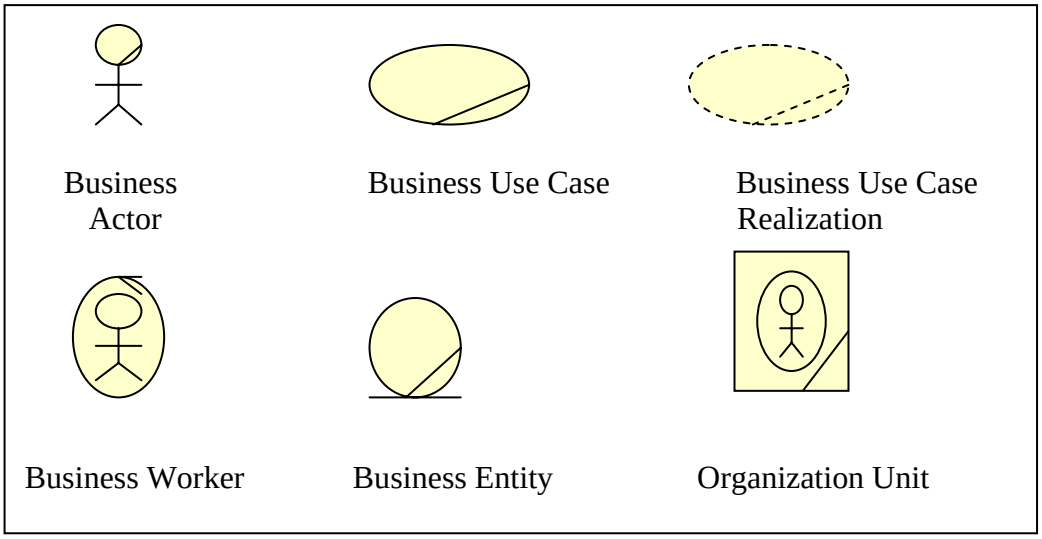
(Sumber : SUH and HAR[1])

h. Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya.

2.5.7 Model Proses Bisnis

Proses bisnis *(Business Process)* adalah sekumpulan altifitas yang dirancang untuk menghasilkan keluaran *(output)* tertentu bagi *customer* tertentu. Sebuah proses bisnis menekankan pada bagaimana sebuah pekerjaan dikerjakan dalam sebuah organisasi, dengan berfokus pada produk yang dihasilkan proses tersebut. Proses ini adalah urutan aktivitas tertentu terhadap waktu dan tempat, dengan titik awal, sebuah titik akhir, dan berbagai *input* dan *output* yang didefinisikan dengan baik. Notasi untuk elemen-elemen model bisnis yang muncul dalam diagram.



Gambar 2.13. Elemen-elemen dalam Model Bisnis

(Sumber : SUH and HAR[1])

Keterangan :

a. *Business Actor* : menggambarkan peran yang dimainkan oleh seseorang atau sesuatu yang dengannya bisnis berinteraksi.
b. *Business Use Case* : merupakan urutan tindakan yang dimainkan suatu bisnis yang menghasilkan sebuah nilai yang dapat dilihat dan ditunjukan untuk suatu *business actor* tertentu.

c. *Business Use Case Realization* : menggambarkan bagaimana realisasi dari setiap use case yang ada pada *business use case*.
d. *Business Worker* : menujukkan sebuah kelas mewakili seorang manusia yang bertindak dalam bisnis.
e. *Business Entity* : menggambarkan entitas yang ada pada suatu bisnis.
f. *Organization Unit* : menggambarkan unit-unit yang ada dalam organisasi bersangkutan.

2.6 Basis Data

Basis data atau database adalah sekumpulan data yang berhubungan. Suatu basis data terdiri dari berbagai object database, meliputi table, view, domain, stored procedure dan trigger**(MAR[7])**. Object-object database menyimpan semua informasi mengenai struktur database dan data. Karena object-object tersebut meng-enkapsulasi informasi mengenai data, object-object database disebut sebagai metadata. Pada waktu lalu, database merupakan sebuah file, misalnya mahasiswa.dbf, pegawai.dbf, yang berisi sebuah table. Di dalam file mahasiswa.dbf terdapat kolom-kolom yang berhubungan, yaitu NPM, Nama, Alamat, TglLahir, dan seterusnya. Seorang mahasiswa mempunyai sebuah baris data. Indek dipakai untuk mempercepat pengaksesan data dan merupakan sebuah file terpisah. Dalam SQL Server, database bukanlah sebuah file tetapi merupakan sebuah konsep logis yang berisi sekumpulan object-object yang berhubungan. Misalnya sebuah database berisi data, struktur database, index, sekuritas, view, dan strored procedure.

2.7 Arsitektur Aplikasi

2.7.1 Pengertian Jaringan Komputer

Jaringan komputer Menurut sumber **ILM [4]** adalah sebuah kumpulan komputer, printer dan peralatan lainnya yang terhubung dalam satu kesatuan. Informasi dan data bergerak melalui kabel-kabel atau tanpa kabel sehingga memungkinkan pengguna jaringan komputer dapat saling bertukar dokumen dan data, mencetak pada printer yang sama dan bersama-sama menggunakan hardware/software yang terhubung dengan jaringan. Setiap komputer, printer atau periferal yang terhubung dengan jaringan disebut **node**. Sebuah jaringan komputer dapat memiliki dua, puluhan, ribuan atau bahkan jutaan node.

2.7.2 Jenis-Jenis Jaringan Komputer

Secara umum jenis jaringan komputer dibagi atas lima jenis, yaitu**:(NUG[8])**

1. *Local Area Network* (LAN)

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

2. *Metropolitan Area Network* (MAN)

Metropolitan Area Network (MAN), pada dasarnya merupakan versi LAN yang berukuran lebih besar dan biasanya menggunakan teknologi yang sama dengan LAN. MAN dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau juga sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi

(swasta) atau umum. MAN mampu menunjang data dan suara, bahkan dapat berhubungan dengan jaringan televisi kabel.

3. *Wide Area Network* (WAN)

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

4. Internet

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

5. Jaringan Tanpa Kabel

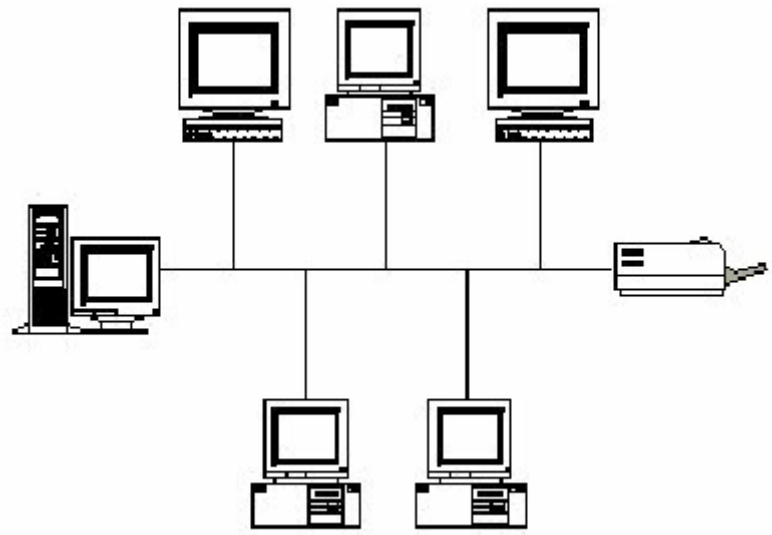
Jaringan tanpa kabel merupakan suatu solusi terhadap komukasi yang tidak bisa dilakukan dengan jaringan yang menggunakan kabel. Misalnya orang yang ingin mendapat informasi atau melakukan komunikasi walaupun sedang berada diatas mobil atau pesawat terbang, maka mutlak jaringan tanpa kabel diperlukan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

memanfaatkan jasa satelit dan mampu memberikan kecepatan akses yang lebih cepat dibandingkan dengan jaringan yang menggunakan kabel.

2.7.3 Topologi Jaringan

Topologi adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Cara yang saat ini banyak digunakan adalah *bus, token-ring, star dan peer-to-peer network***(NUG[5])**. Masing-masing topologi ini mempunyai ciri khas, dengan kelebihan dan kekurangannya sendiri.

a. Topologi BUS



Gambar 2.14 : Contoh gambar topologi jaringan bus.

(Sumber : NUG[8])

Keuntungan

• Hemat kabel

• Layout kabel sederhana

• Mudah dikembangkan

Kerugian

• Deteksi dan isolasi kesalahan sangat kecil

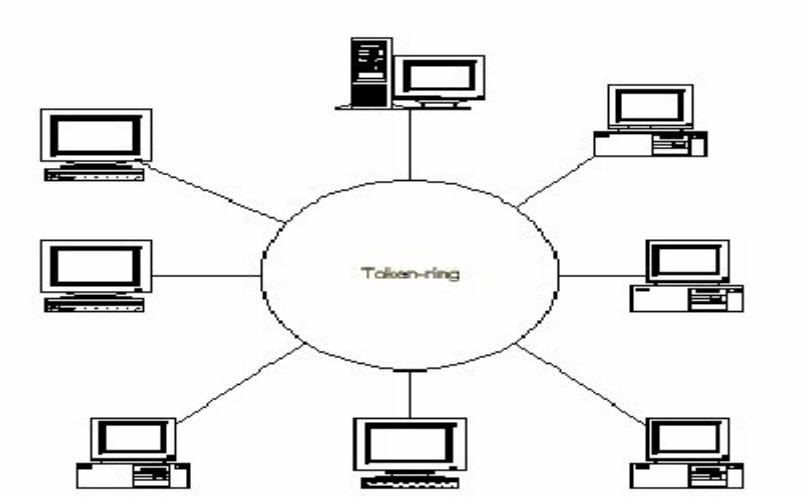
• Kepadatan lalu lintas

• Bila salah satu client rusak, maka jaringan tidak bisa berfungsi.

• Diperlukan repeater untuk jarak jauh

b. Topologi TokenRING

Metode token-ring (sering disebut ring saja) adalah cara menghubungkan computer sehingga berbentuk ring (lingkaran). Setiap simpul mempunyai tingkatan yang sama. Jaringan akan disebut sebagai loop, data dikirimkan kesetiap simpul dan setiap informasi yang diterima simpul diperiksa.



Gambar 2.15 : Contoh gambar topologi jaringan ring.

(Sumber : NUG[8])

Keuntungan

• Hemat Kabel

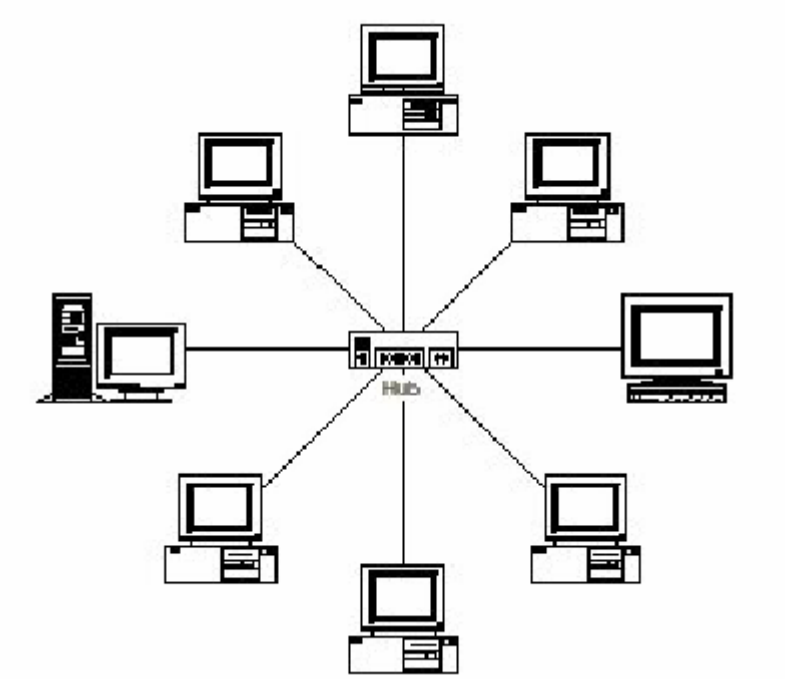
Kerugian

• Peka kesalahan

• Pengembangan jaringan lebih kaku.

c. Topologi STAR

Kontrol terpusat, semua link harus melewati pusat yang menyalurkan data tersebut kesemua simpul atau client yang dipilihnya. Simpul pusat dinamakan stasium primer atau server dan lainnya dinamakan stasiun sekunder atau client server. Setelah hubungan jaringan dimulai oleh server maka setiap client server sewaktu-waktu dapat menggunakan hubungan jaringan tersebut tanpa menunggu perintah dari server.



Gambar 2.16 : Contoh gambar topologi jaringan star.

(Sumber : NUG[8])

Keuntungan

• Paling fleksibel

• Pemasangan/perubahan stasiun sangat mudah dan tidak mengganggu bagian jaringan lain

• Kontrol terpusat

• Kemudahan deteksi dan isolasi kesalahan/kerusakan

Kerugian

• Boros kabel

• Perlu penanganan khusus

• Kontrol terpusat (HUB) jadi elemen kritis

d. Topologi Peer-to-peer Network

Peer artinya rekan sekerja. *Peer-to-peer network* adalah jaringan komputer yang terdiri dari beberapa komputer (biasanya tidak lebih dari 10 komputer dengan 1-2 printer). Dalam sistem jaringan ini yang diutamakan adalah penggunaan program, data dan printer secara bersama-sama. Pemakai komputer bernama Dona dapat memakai program yang dipasang di komputer Dino, dan mereka berdua dapat mencetak ke printer yang sama pada saat yang bersamaan. Sistem jaringan ini juga dapat dipakai di rumah. Pemakai komputer yang memiliki komputer '*kuno*', misalnya AT, dan ingin memberli komputer baru, katakanlah Pentium II, tidak perlu membuang komputer lamanya. Ia cukup memasang netword card di kedua komputernya kemudian dihubungkan dengan kabel yang khusus digunakan untuk sistem jaringan. Dibandingkan dengan ketiga cara diatas, sistem jaringan ini lebih sederhana sehingga lebih mudah dipejari dan dipakai.

2.7.4 Manfaat Jaringan Komputer

a. ***Resource Sharing***, dapat menggunakan sumberdaya yang ada secara bersamasama. Misal seorang pengguna yang berada 100 km jauhnya dari suatu data, tidak mendapatkan kesulitan dalam menggunakan data ersebut,

seolah-olah data tersebut berada didekatnya. Hal ini sering diartikan bahwa jaringan komputer mangatasi masalah jarak.

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

c **Menghemat uang**, Komputer berukutan kecil mempunyai rasio harga/kinerja yang lebih baik dibandingkan dengan komputer yang besar. Komputer besar seperti mainframe memiliki kecapatan kira-kira sepuluh kali lipat kecepatan computer kecil/pribadi. Akan tetap, harga mainframe seribu kali lebih mahal dari computer pribadi. Ketidakseimbangan rasio harga/kinerja dan kecepatan inilah membuat para perancang sistem untuk membangun sistem yang terdiri dari komputerkomputer pribadi.

2.8 Pengertian Client / Server

Dewasa ini konektivitas antara berbagai macam komputer sangatlah tinggi. Beragam komputer dari vendor yang bermacam-macam bisa saling berinteraksi secara client server.

Client adalah suatu proses yang melakukan sesuatu permintaan data atau layanan ke server. Sedangkan server adalah sistem atau proses yang menyediakan data atau layanan yang diminta oleh client.

Client server adalah model jaringan yang memisahkan secara jelas mana yang dapat memberikan layanan jaringan (server) dan mana yang hanya menerima layanan (Client).

2.9 Perangkat Lunak Pendukung

2.9.1 Sekilas Tentang Borland Delphi

Delphi merupakan generasi lanjutan dari turbo pascal (diluncurkan oleh Borland International Incorporation tahun 1983). Sesuai dengan sistem oparsi yang popular pada saat itu, turbo pascal memang dirancang untuk bisa dijalankan pada sistem oprasi DOS kemudian seiring perkembangan zaman dan ilmu pengetahuan yang ada Borland International MC merintis Turbo Pascal for Windows yang bisa dipakai pada windows.

Delphi 7.0 adalah cara yang paling sederhana untuk membangun aplikasi berbasis Windows dan produktifitas dari perangkat lunak yang terdiri dari lima atribut penting, yaitu :

1. Kualitas dan lingkungan pengembangan visual
2. Kecepatan kompiler dibandingkan dengan kompleksitasnya
3. Kekuatan dari bahasa pemrograman dibandingkan dengan kompleksitasnya
4. Fleksibilitas dari arsitektur basis data
5. Pola desain dari pemakai yang dijadikan oleh frame worknya.

2.9.2 Mengenal Database dan Microsoft SQL Server 2000

Database adalah sekumpulan data yang berhubungan. Pada waktu lalu, database merupakan sebuah file, misalnya mahasiswa.dbf, pegawai.dbf, yang berisi sebuah tabel. Didalam file mahasiswa.dbf terdapat kolom-kolom yang berhubungan, yaitu NPM, Nama, Alamat, TglLahir, dan seterusnya. Seorang mahasiswa mempunyai sebuah baris data. Index dipakai untuk mempercepat pengaksesan data dan merupakan sebuah file terpisah**(MAR[7]).**.

SQL server memakai sebuah tipe database yang dinamakan database relational. Database relational adalah database yang mengorganisasikan data dalm bentuk table. Table dibentuk dengan mengelompokan data yang mempunyai subjek yang sama. Table berisi baris-baris dan kolom-kolom informasi. Tabel-tabel dapat saling berhubungan jika diinginkan.

Microsoft SQL server diperkenalkan pada tahun 1990 untuk platform Microsoft OS/2 dalam kerjasamanya dengan Sybase. Produk ini berasal dari Sybase SQL server 4.x untuk Platoform UNIX. Dengan adanya windows NT sehingga dihasilkan Microsoft SQL server versi 4.2 untuk platform Windows NT. Kerjasama dengan Sybase masih berlanjut dan diluncurkan SQL server versi 6.0 dan versi 6.5.

SQL server 6.5 memperbaharui performansi transaksi dan menjadi produk database client/server yang banyak dipakai pada platform Windows NT. Untuk memenuhi kebutuhan SQL server, perlu disain ulang dan kerjasama dengan Sybase diberhentikan dan Micosoft mengembangkan SQL server 7.0. SQL server 2000 dibangun berdasarkan SQL server 7.0**(MAR[7]).**

.