BAB II

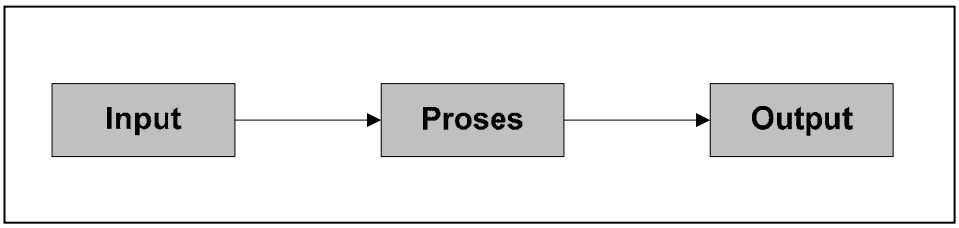
LANDASAN TEORI

2.1 Pengertian Sistem

Menurut Bambang Haryanto (2004:59) pengertian sistem adalah kumpulan objek atau elemen yang saling berinteraksi untuk mencapai satu tujuan tertentu.

2.1.1 Bentuk Dasar Sistem

Bentuk umum dari suatu sistem terdiri atas masukan (*input*), proses dan keluaran (*output*). Dalam bentuk umum sistem ini terdapat satu atau lebih masukan yang akan di proses dan akan menghasilkan suatu keluaran.



Gambar 2.1 Bentuk Dasar Sistem

(Sumber: Jogiyanto, 2005:4)

2.1.2 Karakteristik Sistem

Menurut Jogiyanto (2005: 3) Bahwa suatu sistem mempunyai karakteristik atau sifat – sifat tertentu, yaitu memiliki komponen – komponen (*components*), batas sistem (*boundary*), lingkungan sistem (*Environment*), penghubung (*Interface*), masukan (*Input*), keluaran (*Output*), pengolah (*Proses*), dan sasaran (*Objective*), dan tujuan (Goal).

1. Komponen sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem atau elemen –elemen sistem dapat berupa suatu subsistem atau bagain – bagian dari

sistem. Setiap subsitem mempunyai sifat – sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengarui suatu sistem secara keseluruhan.

2. Batas Sistem (*boundary*)

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

3. Lingkungan Sistem (*environment*)

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

4. Penghubung Sistem (*interface*)

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

5. Masukan Sistem (*input*)

   Masukan (*input*) adalah energi yang dimasukan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (*signal input*). Maintenance input adalah energi yang masukan supaya sistem

tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem (*output*)

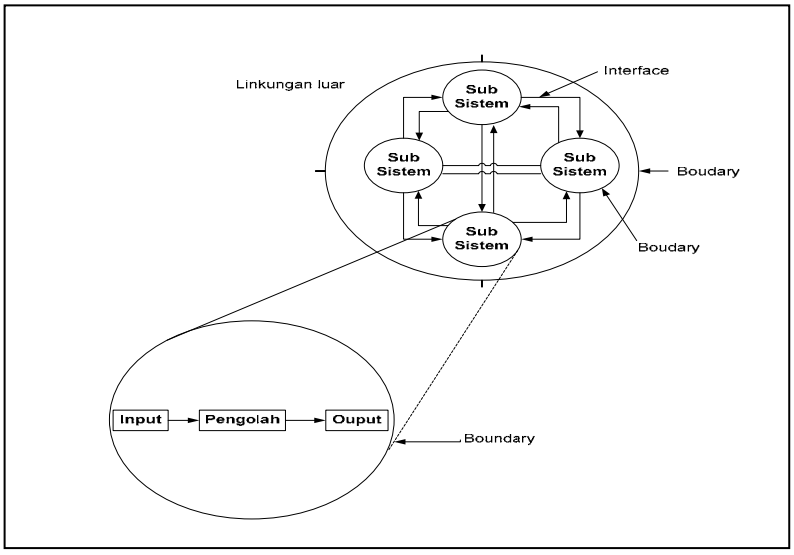
   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan subsistem yang lain atau kepada supersistem.

7. Pengolah Sistem

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

8. Sasaran Sistem

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



Gambar 2.2 Karakteristik Sistem

(Sumber: Jogiyanto, 2005:6)

2.1.3 Klasifikasi Sistem

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

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

   Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak nampak, misalnya sistem teologi. Sistem fisik adalah sistem yang ada secara fisik misalnya sistem komputer.
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia.

   Sistem alamiah adalah sistem yang terjadi melalui proses alam dan tidak di buat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang di rancang oleh manusia yang melibatkan interaksi manusia dengan mesin yang disebut dengan *human-machine system* atau *man-machine system*.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*).

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

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

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

2.1.4 Pengertian Informasi

Informasi merupakan hasil dari data yang di proses yang berasal dari inputan suatu kejadian yang nyata yang berguna bagi pemakainya. Pengertian informasi di atas merupakan pengembangan teori yang di kemukakan oleh Jogiyanto (2005:8) yang menyatakan bawah informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

2.1.5 Konsep dasar informasi

Informasi merupakan hasil dari pengolahan yang di sajikan secara tepat dan akurat. Sumber dari informasi adalah data, data merupakan bentuk jamak dari bentuk tunggal atau data item. Data adalah kenyataan yang menggambarkan suatu

kejadian-kejadian dan kesatuan nyata kejadian-kejadian (*event*). Kejadian adalah suatu yang terjadi pada saat tertentu yang menyangkut perubahan nilai yang disebut transaksi. Sedangkan kesatuan nyata adalah berupa suatu objek yang nyata dan terjadi pada saat kejadian berlangsung. Data juga dapat diartikan suatu yang perlu diolah terlebih dahulu untuk mendapatkan suatu informasi. Menurut Jogiyanto (2005:10) kualitas informasi yang di harapkan tergantung 4 (empat) hal pokok yaitu:

1. Akurat

   Akurat mempunyai arti informasi yang dihasilkan harus bebas dari kesalahan–kesalahan, yang tidak biasa, tidak menyesatkan dan menceminkan maksudnya.

2. Tepat waktu

   Tepat waktu berarti informasi yang disampaikan ke penerima tidak terlambat, karena informasi adalah landasan untuk mengambil suatu keputusan. Untuk itu diperlukan suatu teknologi untuk dan mengirim dengan cepat dan tepat.

3. Relevan

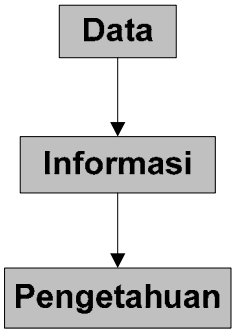
   Berarti informasi mempunyai manfaat dan berguna bagi pemakainya. Karena batas relevensi seseorang berbeda, maka informasi bisa dikatakan berguna jika benar – benar berguna dan dibutuhkan pemakainya.

4. Aman

   Aman berarti informasi harus terbebas dari penyadapan oleh pihak orang yang tidak berwenang dalam penggunaan informasi tersebut.

2.1.6 Hirarki Informasi

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



Gambar 2.3 Hirarki Informasi

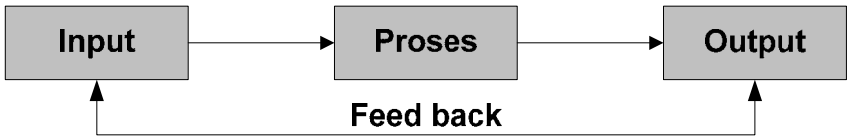
(Sumber: Jogiyanto, 2005:9)

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 Siklus Hidup Informasi

Dari penjelasan sebelumnya telah di kemukakan bahwa informasi adalah merupakan hasil dari pengolahan data. Supaya informasi yang didapat sesuai dengan yang diinginkan dan dapat melangsungkan hidupanya, maka diperlukan *feedback* atau umpan balik dari sistem yang berjalan.



Gambar 2.4 Siklus Hidup Informasi

(Sumber: Jogiyanto, 2005:9)

2.1.8 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.

2.1.9 Pengertian Sistem Informasi

Sistem informasi adalah sekumpulan komponen-komponen sistem yang berada didalam suatu ruang lingkup organisasi, saling berinteraksi untuk menghasilkan sebuah informasi yang bertujuan untuk pihak manajemen tertentu dan untuk mencapai tujuan tertentu.

Menurut Jogiyanto (2005:11) 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 diikuti.

7. Fleksibel (*Fleksibility*)

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

2.1.10 Komponen Sistem Informasi

Menurut Jogiyanto (2005:12) 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 dokumen-dokumen 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 cetak laporan.

4. Blok teknologi (*technologi block*)

   Blok teknologi merupakan penunjang utama dalam berlangsunganya sistem informasi. Yang memiliki beberapa komponen yaitu alat memasukan data (*input device*), alat untuk menyimpan dan mengakses data (*storege device*), alat untuk menghasilkan 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 perangkat keras komputer dan digunakan oleh 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.

2.1.11 Definisi Tenaga kerja

Tenaga kerja adalah tiap orang yang mampu melakukan pekerjaan baik didalam maupun diluar hubungan kerja guna menghasilkan barang atau jasa untuk memenuhi kebutuhan masyarakat.

2.1.12 Definisi Pencari Kerja

Pencari kerja adalah angkatan kerja yang sedang menganggur dan mencari pekerjaan maupun yang yang sudah bekerja tetapi ingin pindah atau alih pekerjaan dengan mendaftarkan diri kepada pelaksana penempatan tenaga kerja atau secara langsung melamar pekerjaan kepada pemberi kerja.

2.1.13 Definisi Pendaftaran Pencari Kerja

Pendaftaran pencari kerja adalah kegiatan untuk mendaftarkan dan mencatat jati diri sebagai pencari kerja.

2.1.14 Pelayanan Penempatan Tenaga Kerja

Pelayanan penempatan tenaga kerja adalah kegiatan untuk mempertemukan pencari kerja dengan pemberi kerja, pengguna tenaga kerja supaya pencari kerja dapat memperoleh pekerjaan yang sesuai dengan bakat dan minat dan kemampuannya serta pemberi kerja memperoleh tenaga kerja yang sesuai dengan kebutuhan.

2.2 Pendekatan Sistem Berorientasi Objek

Pendekatan berorientasi objek adalah cara memandang persoalan menggunakan model-model yang diorganisasikan seputar konsep objek yang mengkombinasikan struktur data dan perilaku suatu entitas.

2.2.1 Unified Modelling Language (UML)

Ada banyak definisi tentang *Unified Modelling Language* (UML) diantaranya adalah sebagai berikut:

Menurut Martin Fowler (2005:1) *Unified Modelling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrogramman berorientasi objek (OO).

Menurut Munawar (2005:17) *Unified Modelling Language* (UML) adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi objek.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software*

Engineering (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design kedalam empat tahapan iteratif, yaitu: identifikasi kelas-kelas dan objek-objek, indetifikasi semantik dari hubungan obyek dan kelas tersebut, perincian interface dan implementasi. Keunggulan metode Booch adalah pada detail dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh Rumbaugh didasarkan pada analisis terstruktur dan pemodelan entity-relationship. Tahapan utama dalam metodologi ini adalah analisis, design sistem, design obyek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO.metode OOSE dari jacobson lebih memberi penekanan pada use case. OOSE memiliki tiga tahapan yaitu membuat model requitment dan analisis, design dan implementasi, dan model pengujian (test model). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

Dengan UML, Metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya.

2.2.2 Model - Model Diagram UML

UML mempunyai sejumlah elemen grafis yang bisa dikombinasikan menjadi diagram. Diagram diklasifikasikan menjadi tiga jenis diagram sebagai berikut:

Tabel 2.1
Klasifikasi Jenis Diagram UML

Structural Diagram	*1. Class Diagram* *2. Composite Structure Diagram* *3. Object Diagram* *4. Component Diagram* *5. Deployment Diagram*
Behavioral Diagram	*1. Use Case Diagram* *2. Activity Diagram* *3. State Machine Diagram*
Interaction Diagram	*1. Sequence Diagram* *2. Interaction Overview Diagram*

(Sumber: Martin Fowler, 2005: 17)

2.2.2.1 Diagram *Use Case*

Diagram *Use case* adalah deskripsi fungsi dari sebuah system dari pandangan pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara user/pengguna sebuah system dengan sistemnya sendiri melalu sebuah cerita bagaimana sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan system disebut scenario. *Use case* dibuat berdasarkan keperluan aktor. *Use case* harus merupakan “Apa” yang dikerjakan software aplikasi bukan “bagaimana” software aplikasi mengerjakannya. Diagram *Use case* memiliki sebuah model khusus yang terbatas untuk kondisi tertentu yang disebut *Stereotype*. Untuk menunjukan *Stereotype* digunakan simbol “<<” diawalnya dan ditutup “>>” diakhirnya. <<extend>> digunakan untuk menunjukan bahwa satu *Use Case* merupakan tambahan fungsional dari *Use Case*

yang lain jika kondisi atau syarat tertentu dipenuhi. Sedankan <<include>> digunakan untuk menggambarkan bahwa suatu *Use Case* seluruhnya merupakan fungsionalitas dari *Use Case* lainnya. Biasanya <<include>> digunakan untuk menghindari peng*copy*an suatu *use case* karena sering dipakai.

2.2.2.2 Diagam *Activity*

Diagram *Activity* adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. *Activity Diagram* mempunyai peran seperti halnya flowchar, akan tetapi perbedaannya dengan flowchart adalah *Activity Diagram* bisa mendukung perilaku paralel sedangkan flowchart tidak bisa

2.2.2.3 Diagram *Sequence*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara obyek-obyek ini dalam usecase.

2.2.2.4 Diagram *Class*

Class Diagram adalah sesuatu yang merupakan cetak biru (*blue-print*) untuk suatu objek. Dengan kata lain, kelas mendefinisikan informasi apa yang dimiliki suatu objek serta mendefinisikan perilaku yang dimilikinya.

2.2.2.5 Diagram *Component*

Component Software adalah bagian fisik dari sebuah sistem, karena menetap di komputer. Komponen bisa berupa *table*, *file data*, *file exe*, *file dynamic link library(DLL)*, *document* dan lain-lain. *Component Diagram* mengandung *component, interface* dan *relationship.*

2.2.2.6 Diagram *Deployment*

Deployment Diagram menunjukan tata letak sebuah sistem secara fisik, menampakan bagian-bagian perangkat lunak yang berjalan pada bagian-bagian perangkat keras. *Deployment Diagram* menyediakan gambaran bagaimana sistem secara fisik akan terlihat. Sistem terdiri dari dari node-node dimana setiap node diwakili untuk sebuah kubus. Garis yang menghubungan antara dua kubus menunjukan hubungan diantara kedua node tersebut. Tipe node bisa berupa device yang berwujud hardware dan bisa juga processor (yang mengeksekusi komponen)atau execution environment (software yang menjadi host atau mengandung software yang lain.

2.2.2.7 Diagram *State (State Machine Diagram)*

State Diagram menelusuri individu-individu objek melalui keseluruhan daur hidupnya, menspesifikasikan semua urutan yang mungkin dari pesan-pesan yang diterima objek tersebut.

2.2.2.8 Diagram *Composite (Composite Struktur Diagram)*

Composite Diagram untuk mendekomposisikan secara hirarkis sebuah class ke sebuah struktur internal. Hal ini memungkinkan untuk memecahkan objek yang kompleks menjadi bagian-bagian kecil.

2.2.2.9 Diagram *Objek*

Objek Diagram adalah gambaran objek-objek secara ringkas disebuah sistem pada suatu waktu. Objek diagram sering disebut sebagai instance diagram karena menunjkan instance-instance dari class. Objek diagram bisa digunakan untuk menunjukan contoh konfigurasi dari objek-objek. Hal ini sangat berguna

untuk menunjukan hubungan yang mungkin ada diantara objek-objek yang sangat kompleks.

2.2.2.10 Diagram *Package*

Package Diagram adalah pengelompokan konstruksi yang memungkinkan untuk mengambil konstruksi tersebut di UML dan mengelompokan elemen-elemen tersebut secara bersama-sama menjadi level yang lebih tinggi.

2.2.2.11 Diagram *Communication*

Communication Diagram adalah sejenis dengan diagram interaksi, yang lebih menekankan pada link data diantara bermacam-macam participant pada interaksi tersebut.

2.2.2.12 Diagram *Interaction Overview*

Interaction Overview Diaram adalahpencakokan secara bersama antara *Activity Diagram* dengan *Sequence Diagram.* Interaction over view diagram bisa dianggap sebagai aktivity diagram dimana semua aktivitas diganti menjadi sedikit sequence diagram, atau bisa juga dianggap sebagai sequence diagram yang dirincikan dengan notasi activity diagram yang digunakan untuk menunjukan aliran pengawasan.

2.2.2.13 Diagram *Timing*

Timing Diagram merupakan bentuk lain dari interaction diagram, dimana terfokus kepada batasan waktu untuk sebuah objek tunggal ataupun, menjadi lebih berguna, untuk sekelompok objek.

2.3 Konfigurasi Jaringan

Menurut Abdul Kadir (2003:415) jaringan komputer adalah hubungan dua buah simpul (umumnya berupa komuputer) atau lebih yang tujuan utamanya adalah untuk melakukan pertukaran data.

Jaringan komputer juga dapat diartikan sebagai kumpulan sebuah terminal komunikasi yang berada diberbagai lokasi yang terdiri dari lebih satu komputer yang saling berhubungan. Dalam sebuah jaringan komputer biasanya terhubung sejumlah komputer ke sebuah/beberapa *server.*

Server adalah komputer yang difungsikan sebagai "pelayanan" pengiriman data dan penerimaan data diantara komputer-komputer yang tersambung.

Tujuan dibangunnya suatu jaringan komputer adalah membawa informasi secara tepat tanpa adanya kesalahan dari sisi pengirim (*transmisi*) menuju ke sisi penerima (*receiver*) melalui media komunikasi.

2.3.1 Jenis-jenis Jaringan Komputer

Menurut Budhi Irawan (2005:29) jenis-jenis jaringan ada dua, yaitu:

1. Model *Peer To Peer*

   Model hubungan *peer to peer* memungkinkan user membagi sumberdaya yang ada dikomputernya baik berupa file, layanan printer dan lain-lain serta mengakses sumber daya yang terdapat pada komputer lain.

2. Model *Client/Server*

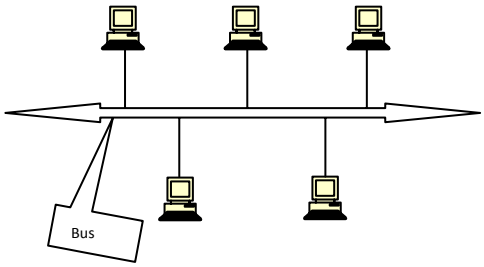
   Model hubungan *client/Server* memungkinkan jaringan untuk mensentralisasi fungsi dan aplikasi kepada satu atau dua *dedicated file* server. Sebuah *file server* menjadi jantung dari keseluruhan sistem, memungkinkan untuk menakses sumber daya, dan menyediakan keamanan.

2.3.2 Topologi Jaringan Komputer

Menurut Abdul kadir (2003:252) yang dimaksud topologi jaringan adalah susunan fisik bagaimana node-node saling dihubungkan. Ada tiga topologi jaringan komputer yaitu :

2.3.2.1 Topologi Bus

Pada topologi ini semua simpul (umumnya komputer) dihubungkan melalui kabel yang disebut bus. Kabel yang digunakan adalah kabel koaksial. Jika seorang pemakai mengirimkan pesan ke seorang pemakai lain maka pesan tersebut akan melalui bus. Setiap komputer perlu membaca alamat dalam pesan. Sekiranya alamat pesan cocok dengan alamat komputer pembaca, komputer tersebut segera mengambil pesan tersebut.

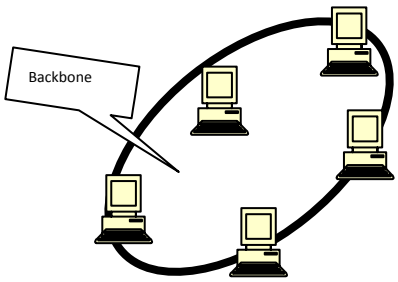


Gambar 2.5 Topologi Bus

(Sumber: Abdul Kadir, 2003: 353)

2.3.2.2 Topologi Cincin (Ring)

Topologi cincin mirip dengan topologi bus. Informasi dikirim oleh sebuah komputer akan dilewatkan ke media transmisi, melewati satu komputer ke komputer berikutnya.

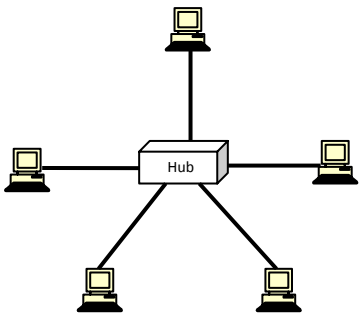


Gambar 2.6 Topologi Cincin

(Sumber: Abdul Kadir, 2003: 354)

2.3.2.3 Topologi Bintang (Star)

Pada topologi ini terdapat komponen yang bertindak sebagai pusat pengontrol. Semua simpul yang hendak berkomunikasi selalu melalui pusat pengontrol tersebut. Dalam hal ini, pusat pengontrol berupa *hub* atau *switch.*



Gambar 2.7 Topologi Bintang

(Sumber: Abdul Kadir, 2003: 355)

2.3.3 Manfaat Jaringan Komputer

Menurut Abdul Kadir (2003:415) 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 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 komunikasi dapat mencegah ketergatungan pada komputer pusat. Setiap proses data tidak harus dilakukan satu komputer saja, melainkan dapat didistribusikan ketempat 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 Pemeliharan

   Menerapkan jaringan komunikasi 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 perindungan serangan virus maka pemakai cukup memusatkan perhatian pada *harddisk* yang ada pada komputer pusat.

5. Keamanan Data

   Sistem jaringan komputer memberikan perlindungan terhadap data jaminan keamanan, data tersebut diberikan 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

   Menerapkan pemakaian sumberdaya secara bersama-sama, maka pemakai bisa mendapatkan hasil dengan maksimal dan kualitas yang tinggi. Selain itu data atau informasi yang di akses selalu terbaru. Karena ada perubahan yang terjadi dapat secara langsung diketahui oleh setiap pemakai.

2.3.4 Sistem *Client/Server*

Menurut Adi Nugroho (2005:211) Setiap Komputer yang menyediakan berkas (file) bagi yang membutuhkan secara umum dinamakan *Server* sementara kompter yang menerima data dapat disebut *Client*.

Sistem *client/server* merupakan suatu sistem client komputer yang melibatkan proses-proses *client* yang meminta suatu pelayanan data kepada komputer *server* yang menyediakan layanan data tersebut. Sehingga *client* maupun *server* sama-sama melakukan pekerjaan. Dengan adanya kombinasi *client (front end*) dan *server (back end)* ini, maka kumpulan dari modul-modul program

tidak dieksekusi dalam memory yang sama namun terbagi dalam komputer *client server.*

Hal ini menjadikan konfigurasi bagi komputer *client* dan komputer *server* bisa berbeda seperti kapasitas memory, kecepatan processor atau alat masukan dan keluaran yang disesuaikan dengan fungsi kerja dari elemen-elemen tersebut. Bagi *server* yang menjalankan tugas pengelolaan suatu *database* digunakan suatu konfigurasi yang khusus menangani tugasnya tersebut dengan sistem operasi yang dikhususkan bagi server seperti milik *Microsoft yaitu Windows NT Server, 2000 Server,* 2003 *Server,* 2008 *Server* atau bisa juga dari varian *Unix* (*Linux, FreeBSD,* dll) sedangkan komputer *client* menggunakan konfigurasi yang umum bagi sebuah komputer *dekstop* yang terhubung ke jaringan dengan sistem operasi milik *Microsoft yaitu Windows 98, ME, XP, Vista* dan lain-lain.

2.3.5 Cara Kerja *Client/Server*

Sistem *client/server* berjalan seutuhnya pada dua sistem komputer yang berbeda. Biasanya sebuah *server* melayani beberapa komputer *client,* walaupun mungkin ada yang hanya melayani satu *client* saja. Apabila pemakai ingin mengakses informasi bagian aplikasi, *client* mengeluarkan permintaan yang dikirimkan melalui jaringan kepada *server* kemudian menjalankan permintaan dan mengirimkan informasi kembali kepada *client.*

Proses *server* berperan sebagai aplikasi yang mengelola sumber daya nilai bersama *(shared resource)* seperti *database,* printer atau jalur komunikasi selain menjalankan tugasnya sebagai back end. Sistem *client server* yaitu pusat

pemrosesan data, sedangkan proses *client* meliputi program-program untuk mengirimkan pesan permintaan pada *server* serta melakukan pengaksesan pada data seperti mengedit, menghapus atau menambah data. Karena itu program *client* adalah aplikasi *front end* yang digunakan sebagai antar muka *(interface)* bagi pemakai untuk berinteraksi dengan *server* selain itu *cient* menangani pemakaian sumberdaya lokal seperti monitor, keyboard dan perangkat lokal lainnya.

2.4 Perangkat Lunak Pendukung

Perangkat lunak pendukung yang digunakan penulis dalam membangun sistem ini da beberapa diantaranya *Rational Rose 2002* digunakan untuk pemodelan atau penotasian diagram berorientasi Objek, *microsoft Visual Basic 6.0* digunakan untuk membangun program, *Seagate Crystal report 8.5* digunakan untuk pembuatan laporan, *Microsoft SQL 2000* sebagai basis data yang digunakan

2.4.1 Sekilas Tentang *Rational Rose 2002*

Rational rose merupakan perangkat lunak CASE (*Computer Aided Software Engineering*) untuk pemrograman berorientasi objek. *Rational Rose* adalah alat atau tools pemoelan visual untuk pengembangan sistem berbasis objek yang sangat handal untuk digunakan sebagai bantuan bagi para pengembang dalam analisis dan perancangan sistem. Rational rose digunakan untuk melakukan pemodelan sistem sebelum pengembang menuliskan kode-kode dalam bahasa pemrogramman tertentu. Rational rose mendukung pemodelan bisnis yang membantu para pengembang untuk memahami sistem secara komprehensif. Rational rose juga membantu analisis sistem dengan cara pengembang membuat

diagram use case untuk melihat fungsionalitas sistem secara keseluruhan sesuai dengan harapan dan keinginan pengguna.

2.4.2 Sekilas Tentang *Microsoft Visual Basic 6.0*

Pemrograman *Microsoft Visual Basic 6.0* adalah pemrograman yang bekerja dalam lingkup *Microsoft Windows*. *Microsoft Visual Basic 6.0* dapat memanfaatkan kemampuan *Microsoft Windows* secara optimal. *Microsoft Visual Basic* menunjukan cara yang digunakan untuk membuat *graphical user interface* dan lingkungan pemrograman *Microsoft Visual Basic* mengandung beberapa *tool* yang bermanfaat untuk menjalankan dan mengelola program yang di buat. *Microsoft Visual Basic* juga mengandung segala sesutau yang dibutuhkan untuk membangun aplikasi untuk *Microsoft Windows* dari awal.

Tool yang terdapat dalam *Microsoft Visual Basic*.

1. *Form* adalah jendela yang bisa di ubah-ubah untuk membuat antar muka program. Pada program set up, *Form* bisa mengandung menu, tombol, kotak daftar, baris penggulung dan item-item lan yang bisa di lihat pada program berbasis Windows lainnya.
2. *Toolbox* untuk menambahkan elemen antarmuka program ke dalam form digunakan *Tool* atau kontrol yang terdapat pada *Toolbox*. *Toolbox* juga mengandung objek yang bisa melakukan operasi “ dibalik layar” pada program *Microsoft Visual Basic*. Objek ini melakukan pekerjaan yang bermanfaat tetapi tidak nampak oleh pemakai apabila program dijalankan. Objek ini diantaranya adalah objek untuk memanipulasi informasi pada *database*.

3. Jendela *Properties* digunakan untuk mengubah karakteristik atau *setting property* dari elemen yang terdapat pada *form*. *Setting property* adalah kualitas objek-objek yang terdapat pada antar muka.

2.4.3 Sekilas Tentang *Seagate Crystal Report 8.5*

Seagate Crystal Report merupakan program yang dapat digunakan untuk membuat, menganalisis dan menerjemahkan informasi yang terkandung dalam database atau program ke dalam berbagai jenis laporan yang sangat fleksibel. Beberapa kelebihan dari *Seagate Crystal Report* adalah:

1. Pembuatan laporannya tidak terlalu rumit sehingga memungkinkan pemrogram pemula sekalipun untuk membuat laporan tanpa harus melibatkan banyak kode pemrograman.
2. Terintegrasi dengan berbagai bahasa pemrograman lain sehingga memungkinkan pemrogram memanfaatkannya dengan keahliannya.
3. Fasilitas impor hasil laporan yang mendukung format yang populer seperti *Microsoft Word*, *Excel*, *Access*, *Adobe Portable Document Format* (PDF), HTML dan sebagainya.

Elemen layar *Seagate Crystal Report* tidak jauh dengan elemen layar *Data Report* (salah satu fasilitas *default* yang disediaakan *Microsoft Visual Basic* untuk membuat laporan). Hanya saja *Seagate Crystal Report* dilengkapi dengan fasilitas yang lebih banyak untuk mengembangkan berbagai jenis laporan.

2.4.4 Sekilas Tentang *Microsoft SQL Server 2000*

Microsoft SQL 2000 diperkenalkan pada tahun 1990 untuk *platform Microsoft OS/2* dalam kerjasamanya dengan *Sybase*. *SQL Server 2000*

dikomersilkan pada tahun 2000 dan mempunyai desain modern. *SQL Server 2000* adalah mesin *database client/server* yang berbeda dengan *database* komputer tunggal tradisional yang memakai sistem pemakaian file secara bersama-sama (misalnya *dBASE, Microsoft Jet, Microsoft Visual Foxpro*). *Database* sistem pemakai file secara bersama-sama bergantung pada sebuah proses *client* tunggal per user untuk memanipulasi data pada *file* yang dipakai pada *server* jaringan. *SQL Server 2000* memberikan bahasa dan antarmuka yang baik untuk pemograman dan komunikasi pada *server.*

Transact-SQL adalah bahasa pemograman *server* yang merupakan superset dari *ANSI-SQL. ANSI-SQL* mendefinisikan empat perintah dasar untuk memanipulasi data yaitu *SELECT, INSERT, UPDATE, DELETE* dan sejumlah perintah untuk mendefinisikan struktur *database.*

SQL Server 2000 memakai konsep transaksi dan *logging* transaksi untuk menjamin konsistensi dan kemampuan *recover* informasi yang disimpan didalam *database SQL Server* mendukung *backup* secara *online.* File-file data yang dipakai pada aplikasi *database* dibuka secara ekslusif oleh proses *server*. Oleh sebab itu melakukan *backup* pada file secara individu tidak memungkinkan, kecuali *server* diberhentikan dulu. *Backup database* dapat dilakukan meskipun ada transaksi yang aktif.

SQL Server mempunyai fungsi utama untuk memproses *query* dan transaksi *database* yang diberikan oleh aplikasi *client* dengan kinerja yang baik. *SQL Server* akan menentukan cara yang paling baik dan efisien untuk memproses

perintah-perintah tersebut dengan sebuah teknik yang dinamakan *Cost Based Query Optimization.*