BAB II
LANDASAN TEORI

2.1 Konsep Dasar Sistem

Menurut JOG[4] terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya.

Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai "*suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu"*.

Sedangkan pendekatan sistem yang menekankan pada komponen atau elemen-elemen mendefinisikan bahwa "*Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu*".

2.1.1 Karakteristik Sistem

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama untuk membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem

   Batasan sistem merupakan daerah yang membatasi diantara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batas sistem ini

memungkinkan suatu sistem dipandang sebagai suatu kesatuan yang menunjukan ruang lingkup sistem itu.

3. Lingkungan luar sistem

   Lingkungan luar dari suatu sistem adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga dapat bersifat merugikan sistem tersebut lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus dijaga dan dipelihara sedangkan lingkungan luar yang merugikan harus ditahan dan dikendalikan, apabila tidak akan mengganggu kelangsungan hidup dari sistem.

4. Penghubung sistem

   Penghubung merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem kesubsistem yang lainnya dengan melalui penghubung. Dengan adanya penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk suatu satu kesatuan.

5. Masukan sistem

   Masukan adalah suatu yang dimasukan kedalam sistem. Masukan dapat berupa masukan perawatan,masukan signal. Masukan perawatan adalah sesuatu yang dimasukan supaya sistem tersebut dapat beroprasi sedangkan masukan signal adalah sesuatu yang diproses untuk didapatkannya suatu keluaran.

6. Keluaran sistem

   Keluaran sistem adalah hasil dari masukan yang diolah dan diklasifikasikan menjadi yang berguna. Keluaran dapat berupa masukan untuk subsistem lainnya.

7. Pengolahan sistem

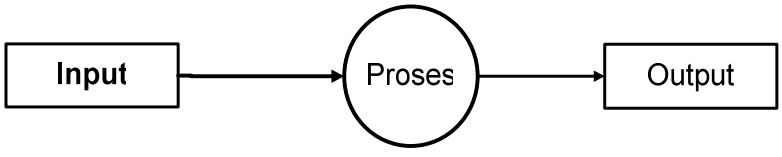
   Sesuatu dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahannya. Pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem

   Suatu sistem pasti memiliki, tujuan/sasaran apabila suatu sistem tidak mempunyai sasaran maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem dapat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan oleh sistem. Suatu sistem dikatakan berhasil bila sasaran yang diharapkan mengenai tujuannya.

2.1.2 Bentuk Umum Sistem

Pada dasarnya bentuk umum suatu Sistem terdiri atas Masukan (*Input*), Proses dan Keluaran (*Output*), adapun bentuk umum dari suatu Sistem adalah sebagai berikut:



Gambar 2.1 Bentuk Umum Sistem
[Sumber : JOG[4]]

2.2 Konsep Dasar Informasi

Informasi adalah faktor terpenting dalam sistem untuk pengambilan suatu keputusan. Definisi informasi sendiri dapat didefinisikan menurut JOG[4] "*Informasi adalah data yang diolah menjadi bentuk yang lebih berguna bagi yang menerimanya*."

Sumber dari informasi adalah data, yang kemudian diolah dengan kriteria tertentu untuk menghasilkan informasi yang dibutuhkan. Data adalah suatu kenyataan yang menggambarkan suatu kejadian-kejadian dan menjadi kasatuan yang nyata. Kejadian adalah sesuatu 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 segala sesuatu yang perlu diolah terlebih dahulu untuk mendapatkan suatu informasi.

Kualitas informasi yang diharapkan tergantung tiga hal pokok :

1. Akurat

   Berarti informasi harus bebas dari kesalahan-kesalahan, tidak biasa, tidak menyesatkan dan mencerminkan maksud serta tujuannya

2. Tepat Waktu

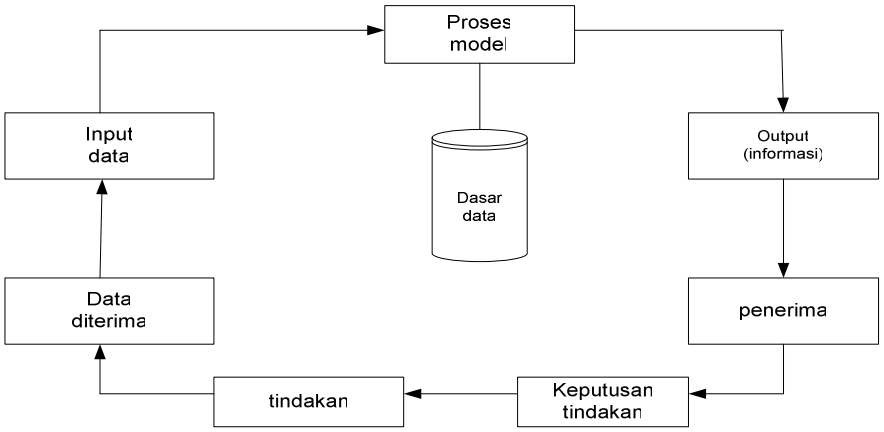
   Berarti informasi yang sampai ke penerima tidak terlambat, karena informasi adalah landasan untuk mengambil suatu keputusan. Untuk itu diperlukan suatu teknologi yang mengolah dan mengirim informasi dengan cepat dan tepat.

3. Relevan

   Berarti informasi mempunyai manfaat dan berguna bagi pemakainya, karena batas relevansi seseorang berbeda, maka informasi bisa dikatakan berguna jika benar-benar berguna dan dibutuhkan pemakainya.

2.2.1 Siklus Informasi

Informasi merupakan suatu proses perubahan dunia menjadi informasi, data yang diolah menjadi suatu model menjadi informasi penerima, kemudian meneriama informasi tersebut membuat suatu kepeutusan dan meklakukan tindakan yang berarti menghasilkan suatu keputusan dan melakukan tindakan yang lain yang akan membuat sejumlah data kembali, untuk lebih jelasnya dapat dilihat pada gambar siklus informasi berikut ini



Gambar 2.2 Siklus Informasi
[Sumber : JOG [4]]

2.3 Konsep Dasar Sistem Informasi

Menurut JOG[4] Sistem informasi adalah "*suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung operasi bersifat managerial dan kegiatan strategi dari suatu*

organisasi dan menyediakan pihak luar tertentu dengan laporan yang diperlukan".

Informasi didefinisikan oleh Robert A.Leitch dan K. Roscoe Davis sebagai berikut :

" *Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi yang mendukung operasi bersifat manajerial dan kegiatan strategi-strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan."*

2.3.1 Sistem Informasi Penjualan

Penjualan berhubungan erat dengan pemasaran, tetapi penjualan hanyalah merupakan segi dari pemasaran artinya dalam suatu penjualan hanya memikirkan dan mengurusi soal pertukaran baik berupa barang maupun jasa.

Sehingga dari sini setiap perusahaan tentunya akan menyusun suatu akuntansi penjualan. Penyusunan akuntansi penjualan menurut sumber MID[6] ini bertujuan untuk mengetahui:

a. Penjualan merupakan sumber pendapatan perusahaan
b. Pendapatan dari penjualan merupakan sumber pendanaan perusahaan
c. Akibat penjualan akan menambah posisi harta pada perusahaan

Adapun klasifikasi transaksi penjualan dibagi menjadi:

1 Penjualan secara tunai
2 Penjualan secara kredit
3 Penjualan secara tandom

4 Penjualan secara ekspor

5 Penjualan secara konsinyasi

6 Penjualan secara grosir

Seperti dikemukakan sebelumnya bahwa tujuan penjualan adalah melakukan transaksi jual beli yang sebanyaknya, adapun jenis-jenis penjualan yang dilakukan diantaranya:

a. Penjualan secara percobaan, ialah penjualan baru terjadi apabila pihak pembeli telah menyatakan persetujuannya terhadap barang yang dijual.
b. Penjualan dengan contoh, ialah penjualan yang terjadi atas dasar contoh barang tersebut baik mengenai barang maupun kualitasnya dan sesuai harganya.
c. Penjualan dengan cicilan atau angsuran, ialah pembeli membayar harga barang dengan beberapa kali bayaran.
d. Penjualan dengan kontan, dalam hal ini harga dibayar tunai barang dibawa langsung oleh pembeli.

2.3.1.1 Fungsi Yang Terkait Dalam Sistem Penjualan

Beberapa fungsi yang terkait dalam sistem penjualan adalah :

1. Fungsi Penjualan

   Fungsi ini bertanggung jawab untuk menerima pesanan dari pembeli, mengedit pesanan dari pelanggan untuk menambahkan informasi yang belum ada pada surat pesanan tersebut, meminta otorisasi kredit,

menentukan tanggal pengiriman dan jumlah pesanan dari gudang barang yang akan dikirim dan mengisi pesanan pengiriman.

2. Fungsi Gudang

   Fungsi ini bertanggung jawab untuk menyiapkan barang yang dipesan oleh pelanggan serta menyerahkan barang ke fungsi pengiriman.

3. Fungsi Pengiriman

   Fungsi ini bertanggung jawab untuk menyerahkan barang atas dasar surat pemesanan pengiriman yang diterimanya dari fungsi penjualan, fungsi ini bertanggung jawab untuk menjamin bahwa tidak ada barang yang keluar dari perusahaan tanpa ada otorisasi dari yang berwenang.

4. Fungsi Administrasi

   Fungsi ini bertanggung jawab untuk membuat dan mengirimkan faktur penjualan kepada pelanggan, serta menyediakan copy faktur bagi kepentingan pencatatan transaksi penjualan oleh fungsi akuntansi.

5. Fungsi Akuntansi

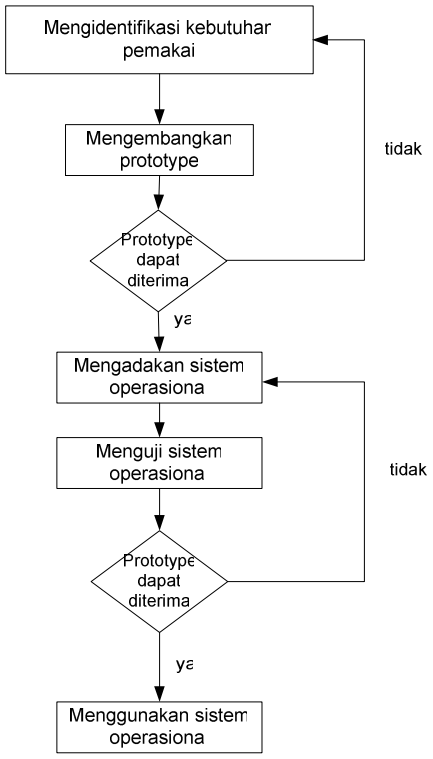
   Fungsi ini bertanggung jawab untuk mencatat transaksi penjualan serta membuat laporan penjualan. Disamping itu fungsi ini juga bertanggung jawab untuk mencatat harga pokok persediaan yang dijual ke dalam kartu persediaan.

2.4 Metode Pengembangan Sistem

Metode yang digunakan untuk perancangan sistem informasi ini adalah metode *prototype*. Adapun tahapan-tahapan dari metode *prototype* menurut RAY[9] adalah sebagai berikut:

a. **Mengidentifikasi kebutuhan pemakai.** Pada tahap ini analisis sistem akan melakukan studi kelayakan dan studi terhadap kebutuhan pemakai, baik yang meliputi model *interface*, teknik prosedural, maupun dalam teknologi yang akan digunakan.
b. **Mengembangkan *prototype*.** Pada tahap kedua ini, analisa sistem bekerja sama dengan pemrogram dalam mengembangkan *prototype* sistem untuk memperlihatkan kepada pemesanan permodelan sistem yang akan dibangunnya.
c. **Menentukan apakah *prototype* dapat diterima oleh pemesan atau pemakai.**
d. **Mengadakan sistem operasional**. Melalui pemrogram berdasarkan permodelan sistem yang telah disepakati oleh pemesan sistem.
e. **Menguji sistem operasional.** Pada tahap ini pemrogram akan melakukan uji coba baik menggunakan data sekunder maupun data primer untuk memastikan bahwa sistem dapat berlangsung dengan baik dan benar, sesuai dengan kebutuhan pemesan.
f. **Menentukan sistem operasional.** Apakah dapat diterima oleh pemesan atau harus dilakukan beberapa perbaikan atau bahkan harus dibongkar semuanya dimulai dari awal lagi.

g. **Jika sistem telah disetujui,** maka tahap terakhir adalah melakukan implementasi sistem.



Gambar 2.3 Metode *Prototype*
[Sumber: RAY[9]]

2.5 Pengenalan UML

UML (*Unified Modeling Language*) merupakan pengganti dari metode analisis berorientasi objek (OOA/D) yang ada sekitar akhir tahun 80-an dan awal 90-an.

UML merupakan gabungan dari metode *Booch*, *Rumbaugh (OMT)* dan *Jacobson*. Tetapi UML ini akan mencakup lebih luas dari pada OOA&D. pada pertengahan pengembangan UML dilakukan standarisasi proses dengan OMG

(Object Management Group) dengan harapan UML akan menjadi bahasa standar pemodelan pada masa yang akan datang.

UML disebut *sebagai bahasa pemodelan. 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.*

Bahasa pemodelan merupakan bagian terpenting dari metode. Ini merupakan bagian kunci tertentu untuk komunikasi. Jika anda ingin berdiskusi tentang desain dengan seseorang, maka anda hanya akan membutuhkan bahasa pemodelan bukan proses yang digunakan untuk mendapatkan desain.

UML merupakan bahasa standar untuk penulisan *blueprint software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

2.5.1 Pengertian UML

Pengertian *Unified Modelling Language* (UML) menurut sumber ILM[11] adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan perangkat lunak

dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram perangkat lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya, yaitu: Grady Booch OOD *(Object-Oriented Design)*, Jim Rumbaugh OMT *(Object Modeling Technique)*, dan Ivar Jacobson OOSE *(Object-Oriented Software Engineering)*.

2.5.2 Bagian-bagian UML

Bagian-bagian utama dari UML adalah *view, diagram, model element, dan general mechanism.*

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 UML antara lain adalah *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 oleh *external actors. Actor* yang berinteraksi dengan

sistem dapat berupa user atau sistem lainnya. *View* ini digambarkan dalam *use case diagrams* dan kadang-kadang dengan *activity diagrams.*

View ini digunakan terutama untuk pelanggan, perancang (*designer*), pengembang (*developer*), dan penguji sistem (*tester*).

b. Logical View

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

View ini digambarkan dalam *class diagrams* untuk struktur statis dan dalam *state, sequence, collaboration,* dan *activity diagram*s untuk model dinamisnya. View ini digunakan untuk perancang (*designer*) dan pengembang (*developer*).

c. Component View

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

View ini digambarkan dalam *component view* dan digunakan untuk pengembang (*developer*).

d. Concurrency View

Membagi sistem kedalam proses dan prosesor. View ini digambarkan dalam diagram dinamis (*state, sequence, collaboration,* dan *activity diagrams*) dan diagram implementasi (*component* dan *deployment*

diagrams) serta digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

e. Deployment View

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (*nodes*) dan bagaimana hubungannya dengan yang lain. View ini digambarkan dalam *deployment diagrams* dan digunakan untuk pengembang (*developer*), pengintegrasi (*integrator*), dan penguji (*tester*).

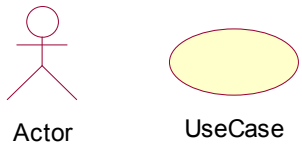
2. Diagram

Diagram berbentuk grafis 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. Adapun jenis diagramnya antara lain:

a. Use Case Diagram

Menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah deskripsi fungsi yang disediakan oleh sistem kedalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*.

Use case digambarkan hanya yang dilihat dari luar oleh *actor* (keadaan lingkungan sistem yang dilihat user) dan bukan bagaimana fungsi yang ada di dalam sistem. Simbol yang digunakan yaitu:

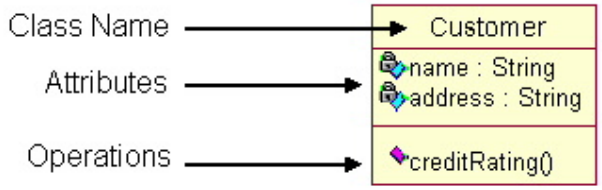


Gambar 2.4 *Use Case* Model
[Sumber: ATL[10]]

b. Class Diagram

Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialed* (satu *class* merupakan spesialisasi dari *class* lainnya) atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram.*

Suatu *class* biasanya terdiri dari 3 bagian, yaitu nama, atribut, dan operasi. Berikut adalah contoh dari suatu *class*:

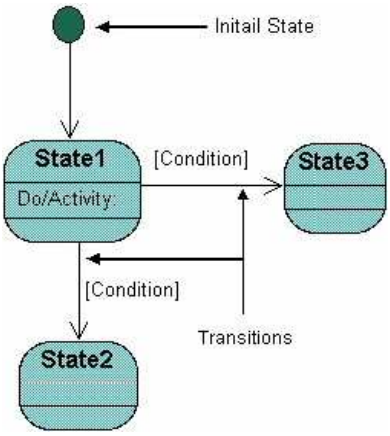


Gambar 2.5 Bagian-bagian dari C*lass*
[Sumber: ATL[10]]

c. *State Diagram*

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan.

State class tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda.

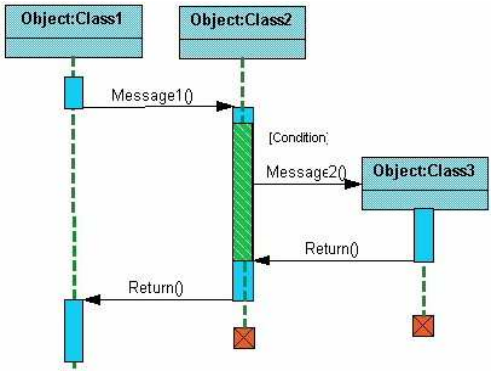


Gambar 2.6 *State Diagram*
[Sumber: ATL[10]]

d. *Sequence Diagram*

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

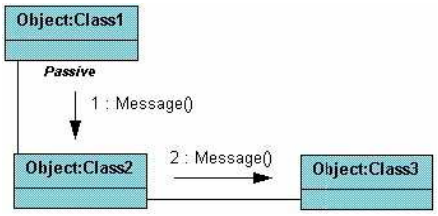
Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan proses vertikal.



Gambar 2.7 *Sequence Diagram*
[Sumber: ATL[10]]

e. Collaboration Diagram

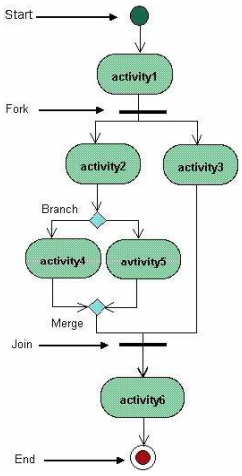
Menggambarkan kolaborasi dinamis seperti *sequence diagrams*. Dalam menunjukkan pertukaran pesan, *collaboration diagrams* menggambarkan object dan hubungannya (mengacu ke konteks). Jika penekanannya pada waktu atau urutan maka gunakan *sequence diagrams*, tapi jika penekanannya pada konteks gunakan *collaboration diagrams*.



Gambar 2.8 *Collaboration Diagram*
[Sumber: ATL[10]]

f. Activity Diagram

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



Gambar 2.9 *Activity Diagram*
[Sumber: ATL[10]]

g. Component Diagram

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

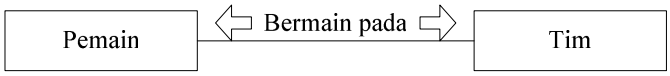
h. Deployment Diagram

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

dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

2.5.3 *Indicator Multiplicity*

Multiplicity pada kasus asosiasi menunjukkan bahwa ada sejumlah objek pada sebuah *class* yang berhubungan dengan sebuah objek pada sebuah asosiasi *class*. Untuk menggambarkan hal tersebut bias dilihat pada fakta bahwa sebuah tim mungkin terdiri dari beberapa pemain. Untuk menunjukkan hal itu angka harus diletakkan di dekat *class* yang dimaksud.



Gambar 2.10 *Multiplicity* pada Asosiasi
[Sumber: MUN[8]]

Ada banyak multiplicity yang mungkin untuk dipakai. Sebuah class dapat berhubungan denga yang lain dalam satu ke satu *(one-to-one)*, satu ke banyak (*one-to-many*), satu ke satu atau lebih (*one-to-one or more*), satu ke nol atau satu, satu ke sebuah interval tertentu (contoh satu ke lima sampai dengan sepuluh), satu ke sejumlah n atau satu ke serangkaian pilihan (contoh satu ke sembilan atau sepuluh).

Pada notasi UML, untuk menampilkan lebih atau banyak digunakan (*). Untuk menunjukkan ‘atau’ digunakan titik dua (..) seperti contohnya 1..* (berarti satu atau lebih). Notasi ‘atau’ bisa juga digunakan tanda koma (,). Contoh dari *multiplicity* diatas dapat dilihat pada gambar berikut:



Gambar 2.11 Macam-macam *multiplicity*
[Sumber: MUN[8]]

2.6 Basis Data

Menurut FAT[3] : *" Basis data adalah Kumpulan atau koleksi dari data-data yang disimpan pada alat-alat penyimpanan tertentu dengan struktur penyimpanan yang khas dan digunakan perangkat lunak untuk memanipulasinya"*.

Basis data merupakan kumpulan *file-file* yang memiliki hubungan antara satu *file* dengan *file* lainnya sehingga membentuk suatu kesatuan data yang memberikan informasi tentang suatu organisasi. Basis data dapat dijalankan dengan mengguakan beberapa konfigurasi, salah satunya cukup terkenal saat ini adalah dengan metode *client/server*.

Prinsip utama suatu basis data adalah pengaturan data atau arsip, sedangkan tujuan utama basis data adalah kemudahan dan kecepatan dalam pencarian data atau arsip yang kita inginkan.

Penyusunan suatu basis data digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu :

a. Redudansi dan inkonsistensi data.
b. Kesulitan pengaksesan data.
c. Isolasi data untuk standarisasi.
d. *Multiple user* (banyak pemakai).
e. Masalah keamanan (*security*).
f. Masalah integrasi (kesatuan).
g. Masalah data *independence* (kebebasan data).

Proses dasar yang dimiliki oleh database ada 4, yaitu :

1. Pembuatan data-data baru (*create database*).
2. Penambahan data (*insert*)
3. Mengubah data (*up date*)
4. Menghapus data (*delete*)

Sistem manajemen database (*Database Management System*) merupakan sistem pengoperasian dan sejumlah data pada komputer. Dengan sistem ini dapat merubah data, memperbaiki data yang salah dan menghapus data yang tidak dapat dipakai.

2.7 Arsitektur Aplikasi

Dalam melengkapi semua kebutuhan sistem informasi ini, maka diperlukan perantara yang berfungsi sebagai jalur data antara beberapa komputer yang biasa disebut jaringan komputer.

2.7.1 Pengertian Jaringan Komputer

Jaringan komputer menurut AND[2]: " *merupakan sekelompok komputer otonom yang saling berhubungan antara satu dengan lainnya menggunakan protocol komunikasi melalui media komunikasi sehingga dapat saling berbagi informasi, program – program, penggunaan bersama perangkat keras seperti printer, harddisk, dan sebagainya*."

Selain itu jaringan komputer bias diartikan sebagai kumpulan sejumlah terminal komunikasi yang berada di berbagi lokasi yang terdiri dari lebih satu computer yang saling berhubungan.

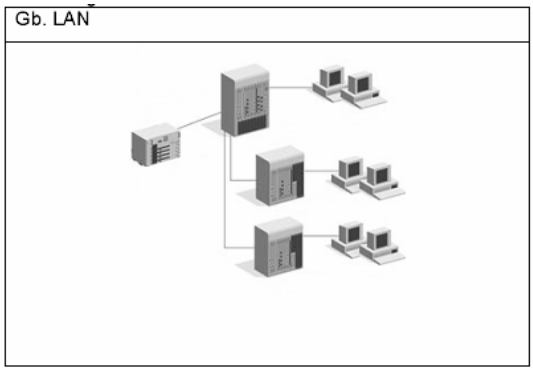
Tujuan di bangunnya suatu jaringan komputer adalah membawa informasi dari sisi pengiriman (*transmitter*) menuju ke sisi penerima (*receiver*) melalui media komunikasi.

2.7.2 Jenis – Jenis Jaringan Komputer

Ada beberapa jenis jaringan komputer, diantaranya yaitu:

1. *Local Area Network* (LAN)

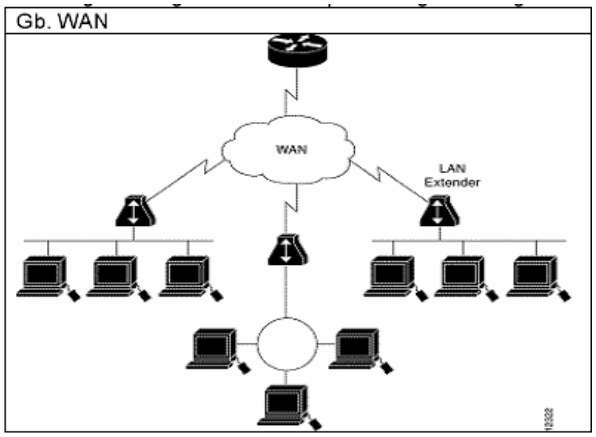
   Jaringan komputer yang secara geografis terletak berdekatan dan saling berhubungan.



Gambar 2.12 Jaringan LAN
[Sumber: ILM [11]]

2. *Wide Area Network* (WAN)

   Jaringan computer yang secara geografis terpisah jarak yang jauh dan dihubungkan dengan saluran telepon atau gelombang radio.



Gambar 2.13 Jaringan WAN
[Sumber: ILM [11]]

3. *Metropolitan Area Network* (MAN)

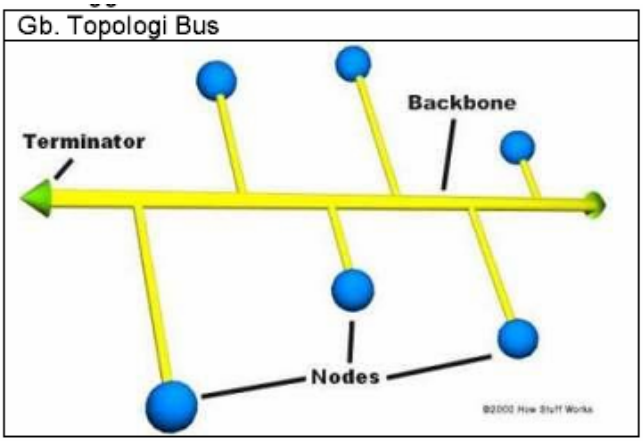
   Jaringan komputer yang menghubungkan komputer-komputer di seluruh dunia. Lebih dikenal sebagai internet.

2.7.3 Topologi Jaringan Komputer

Topologi jaringan adalah cara pengelompokan/konfigurasi elemen-elemen dari jaringan. Dua jaringan dikatakan mempunyai topologi yang sama bila konfigurasi koneksinya sama, meskipun secara fisik, jarak, dan transmisinya beda. Beikut adalah beberapa jenis topologi jaringan, yaitu:

1. ***Bus***

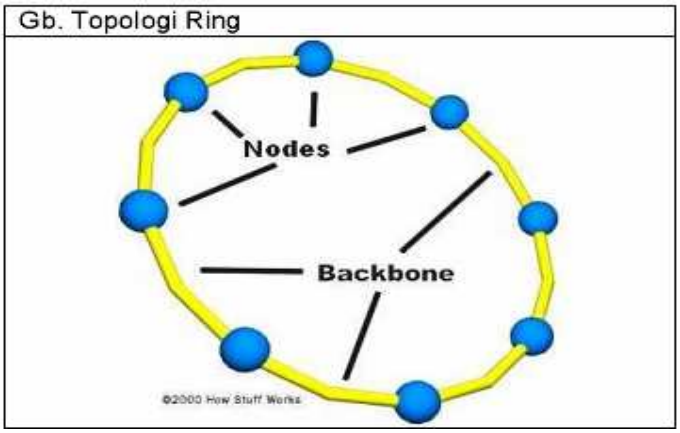
   Topologi jaringan yang konfigurasi koneksi semua nodenya terhubung menggunakan *bus* tunggal.



Gambar 2.14 Topologi Jaringan Bus
[Sumber: ILM [11]]

2. ***Ring***

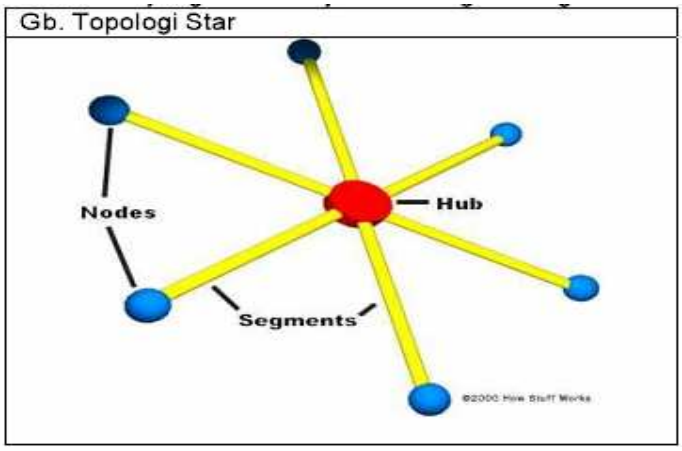
   Topologi ini menggunakan metoda token-passing dimana data yang dikirim akan berputar melalui node ke node sampai node tujuan ditemukan. Bila diurutkan, koneksi ini akan berbentuk melingkar.



Gambar 2.15 Topologi Jaringan Ring
[Sumber: ILM [11]]

3. *Star*

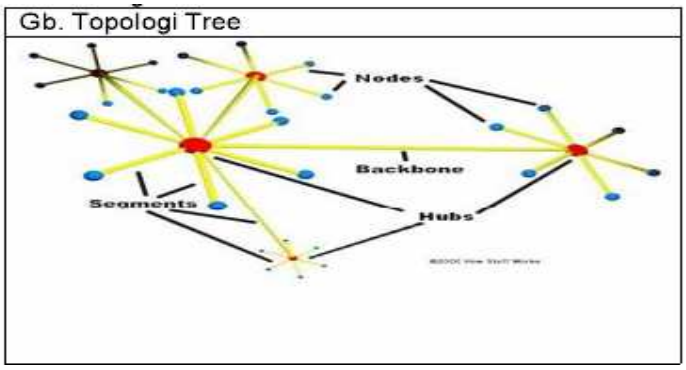
Topologi jaringan yang konfigurasi koneksinya adalah semua node terhubung pada satu node pusat, yang biasanya menggunakan peralatan hub atau switch. Node pusat ini yang melakukan rebroadcast semua transmisi yang diterima dari node-node yang ada ke tujuan masing-masing.



Gambar 2.16 Topologi Jaringan *Star*
[Sumber: ILM [11]]

4. ***Tree***

   Topologi jaringan yang konfigurasinya adalah gabungan dari beberapa topologi *star* sekaligus.



Gambar 2.17 Topologi Jaringan *Tree*
[Sumber: ILM [11]]

2.7.4 Manfaat Jaringan Komputer

Menurut AND[1] secara umum jaringan komputer mempunyai beberapa manfaat di bandingkan dengan komputer yang berdiri sendiri, diantaranya :

1. Sharing Resources

   Bertujuan agar seluruh program , peralatan atau peripheral lainnya dapat di manfaatkan oleh setiap orang yang ada pada jaringan komputer tanpa terpengaruh oleh lokasi maupun pengaruh dari pemakai.

2. Media Komunikasi

   Jaringan komputer memungkinkan terjadinya komunikasi antar pengguna, baik untuk telekonference maupun untuk mengirim pesan informasi yang penting lainnya meskipun jaraknya saling berjauhan.

3. Integrasi Data

   Pembangaunan jaringan komputer dapat mencegah ketergantungan pada komputer pusat, melainkan dapat didistribusikan ke komputer di tempat lainnya.

4. Pengembangan dan Pemeliharaan

   Dengan adanya jaringan komputer ini, maka pengembangan peralatan dapat dilakukan dengan mudah dan menghemat biaya, misalnya dengan kualitas pencetakkan dari dot metrik printer ke laser printer, maka tidak perlu membeli laser printer untuk jumlah komputer yang ada melainkan cukup satu buah karena printer tersebut dapat digunakan secara bersama – sama. Dalam hal pemeliharaan jaringan komputer memudah pemakai dalam merawat harddisk dan peralatan lainnya, misalnya untuk memberikan perlindungan terhadap serangan virus.

5. Keamanan Data

   Sistem jaringan komputer memberikan perlindungan terhadap data melalui pengaturan hak akses para pemakai.

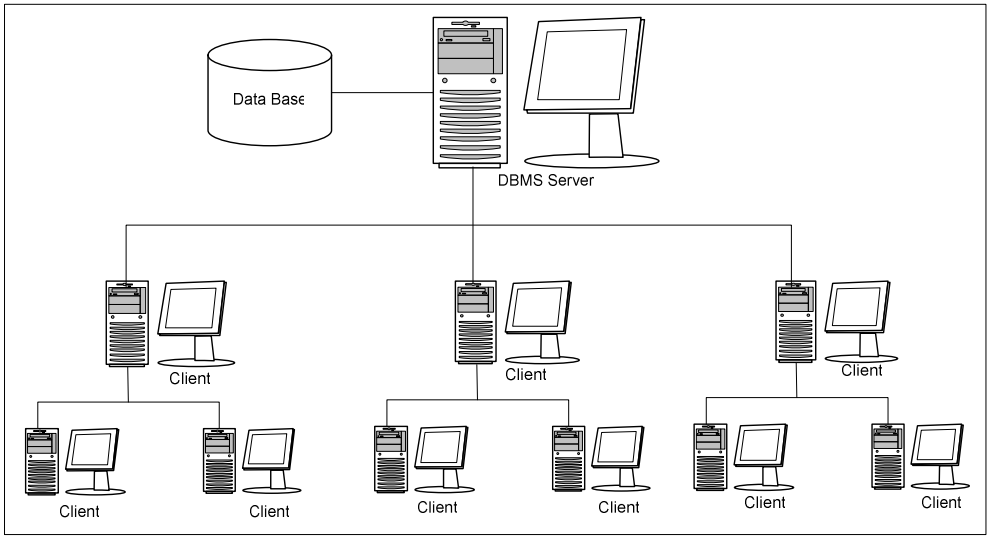
6. Sumber Daya Lebih Efisien dan informasi terkini

   Dengan adanya pemakaian sumber daya secara bersama – sama, maka pemakai bisa mendapatkan hasil dengan maksimal dan kualitas yang tinggi. Selain itu data atau informasi yang diakses selalu terbaru, karena setiap ada perubahan yang terjadi dapat segera langsung di ketahui oleh setiap pemakai

2.8 Pengertian *Client / Server*

Client server menurut Spencerken Miller Ken, MIL[7] merupakan komputer yang memanfaatkan sumber daya dalam jaringan yang disediakan oleh komputer lainnya, yang disebut dengan server. Client juga merupakan sebuah aplikasi atau proses yang meminta pelayanan dari komponen atau proses lainnya. Server merupakan perangkat khusus dalam jaringan komputer yang menjadi tempat bagi semua nodes di dalam jaringan untuk bisa melakukan *resource sharing*. *Client server* merupakan model jaringan yang memisahkan secara jelas mana yang dapat memberikan layanan jaringan (*server*) dan mana yang hanya menerima layanan (*client*).

Arsitektur adalah desain sebuah aplikasi. Arsitektur *client/server* merupakan arsitektur terdistribusi artinya pemrosesan dari suatu aplikasi terjadi pada lebih satu mesin. Dalam arsitektur *client/server* aplikasi dipecah menjadi koponen-komponen yang bekerja sama untuk mencapai tujuan.



Gambar 2.18 Sistem *Client-Server* Kompleks
[Sumber: FAT[3]]

2.9 Perangkat Lunak Pendukung

Suatu Perangkat lunak *(Software)* pendukung sangat berperan penting dalam membangun suatu aplikasi yang baik, karena berguna untuk melengkapi kebutuhan aplikasi yang akan dibangun. Adapun perangkat lunak yang digunakan yaitu : Borland Delphi 6.0 dan SQL Server 2000.

2.9.1 Sekilas Borlan Delphi 6.0

Menurut AGU[1]: *"Delphi adalah aplikasi yang digunakan untuk pengembangan dengan memanfaatkan keistimewaan konsep-konsep antar muka grafis dalam microsoft windows"*. Aplikasi yang dihasilkan Delphi berkaitan dengan windows itu sendiri sehingga dibutuhkan pengetahuan bagaimana cara kerja *windows.*

Delphi adalah software buatan Borland yang sangat terkenal. Berbeda dengan software windows umumnya, Delphi bukanlah software aplikasi seperti MS Office atau Software windows lainnya. Delphi adalah suatu bahasa pemograman, *Developher Language*, aplikasi uantuk membuat aplikasi. Delphi digunakan uantuk membangun aplikasi windows, aplikasi grafis, aplikasi visual, bahkan aplikasi jaringan (client/server) dan lain sebagainya.

Jadi Delphi merupakan program yang sangat cocok dan tepat dalam membangun aplikasi ini. Delphi hampir sama dengan pascal sangat baik untuk pemograman, karena sifatnya yang terstruktural dan prosedural serta penting untuk dikuasai oleh siapa saja yang berkecimpung dalam dunia komputer.

2.9.2 SQL Server 2000

Menurut sumber Martina Inge MAR[5] bahwa Microsoft SQL *server* 2000 adalah *perangkat lunak Relation Database Management Sistem (RDBMS) yang handal*. Di design untuk mendukung proses transaksi yang besar seperti *order entry online*, *inventori*, akuntansi atau manufaktur. Microsoft SQL Server 2000 dapat dijalankan pada NT 4.0 Server atau Microsoft Windows 2000 *Server*, selain itu dapat pula di install pada *personal desktop* di Windows 2000 Profesional, Windows 98 Milenium.

Sistem akan dipergunakan oleh beberapa komputer (*client-server*) maka database yang dibangun adalah merupakan database yang berfungsi untuk menunjang hal tersebut untuk itu dalam penulisan ini dipakai database Microsoft SQL Server 2000, selain itu database ini juga *compatible* dengan bahasa pemrograman yang digunakan. Secara teoritis, program SQL yang sudah terinstal pada komputer yang dapat menampung 32.767 database dan terdapat lebih dari 2 billion objek.

2.10 Kasus yang Dianalisis/Dirancang

Kasus yang dianalisis, teliti dan dirancang penulis menyangkut proses penjualan, karena proses pencatatan transaksinya tersebut banyak yang masih dilakukan secara manual dimana banyak kesalahan-kesalahan yang tidak disengaja sering terjadi. Untuk itu Penulis mengambil kasus penjualan Tunai Kredit pada PT. BAHANA YAMAHA BANDUNG agar semua masalah yang menyangkut penjualan baik tunai maupun kredit dapat dicegah dan diatasi.