# BAB II

# LANDASAN TEORI

## 2.1 Konsep Dasar Sistem

Terdapat 2 kelompok pendekatan didalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau bagiannya. Menurut JOG [5] *sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk mnyelesaikan suatu sasaran tertentu*. Sedangkan menurut GOR [2] mendefinisikan sistem adalah *"sistem terdiri dari bagian-bagian yang bersama beroperasi untuk mencapai beberapa tujuan."*

Menurut JOG [5] suatu sistem mempunyai karakteristik (sifat-sifat tertentu), yaitu :

1. Komponen – komponen (*Components*)

   Suatu sistem terdiri dari jumlah komponen yang saling berinteraksi, bekerja sama untuk membentuk suatu kesatuan komponen – komponen sistem berupa sub-sistem atau bagian – bagian dari sistem.

2. Batasan Sistem (*Boundary*)

   Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini dapat dipandang sebagai satu kesatuan dan menunjukan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*Environment*)

   Segala sesuatu diluar batas dari sistem yang mempengaruhi operasi sistem yang bersifat menguntungkan atau merugikan.

4. Penghubung (*Interface*)

   Merupakan media penghubung antara satu sub-sistem dengan sub-sistem yang lainnya. Melalui penghubung ini memungkinkan sumber daya – sumber daya mengalir dari sub-sistem yang lainnya.

5. Masukan (*Input*)

   Energi yang dimasukan ke dalam sistem yang berupa masukan perawatan (*Maintenace Input*) agar sistem tersebut dapat beroperasi dan masukan sinyal (*Signal Input*) yang diproses untuk menghasilkan keluaran.

6. Keluaran (*Output*)

   Merupakan hasil dari energi yang telah diproses dan dapat diklarifikasikan antara lain menjadi *output* yang langsung digunakan, *output* yang bisa dijadikan *input* untuk sub-sistem yang lain atau kepada supra sistem serta *output* bagi sistem yang tidak digunakan lagi (sisa pembuangan) tapi bagi sistem lain sangat berguna.

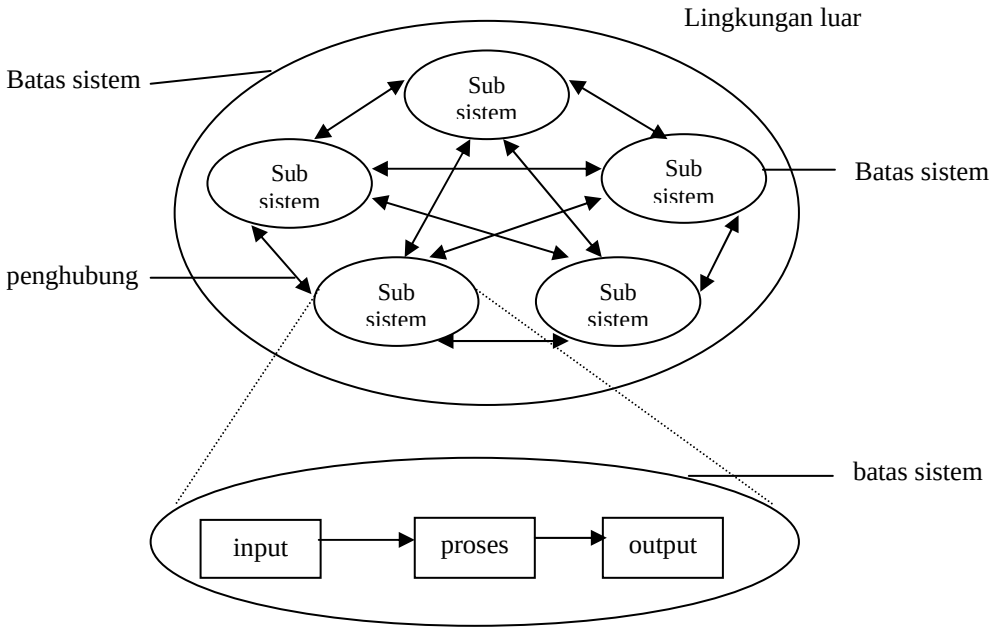
7. Pengolah (*Process*)

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

8. Sasaran (*Objectives*) dan Tujuan (*Goal*)

   Suatu sistem dikatakan berhasil jika mengenai sasaran atau tujuannya.

Karakteristik atau sifat-sifat suatu sistem dapat kita lihat seperti pada gambar berikut ini :




*gambar 2.1 : Karakteristik Suatu Sistem*

*(Sumber : Jogiyanto HM, 2001, Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur, Andi Offset, Yogyakarta*)

## 2.2 Konsep Dasar Informasi

Informasi adalah faktor terpenting dalam sistem untuk pengambilan keputusan. Definisi informasi sendiri 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

suatu kesatuan yang nyata. Kejadian adalah suatu yang terjadi pada saat tertentu yang menyangkut perubahan nilai yang disebut transaksi. Sedangkan kesatuan nyata adalah berupa objek nyata yang terjadi pada saat kejadian berlangsung. Data juga dapat diartikan segala sesutau yang perlu diolah terlebih dahulu untuk mendapatkan suatu informasi.

Informasi adalah "*Data yang telah diproses kedalam suatu bentuk tertentu yang mempunyai nilai nyata dan terasa bagi keputusan saat ini dan kebutuhan mendatang.*" GOR [2]. Sedangkan menurut JOG [5] " *Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya*."

Agar bermanfaat, informasi harus memiliki kualitas sebagai berikut :

1. Relevan, yaitu menambah pengetahuan atau nilai bagi para pembuat keputusan, dengan cara mengurangi ketidakpastian, menaikkan kemampuan untuk memprediksi, atau menegaskan ekspektasi semula.
2. Dapat dipercaya, yaitu bebas dari kesalahan atau bisa secara akurat menggambarkan kejadian atau aktivitas organisasi.
3. Lengkap, yaitu tidak menghilangkan data penting yang dibutuhkan oleh para pemakai.
4. Tepat waktu, yaitu disajikan pada saat yang tepat untuk mempengaruhi proses pembuatan keputusan.
5. Mudah dipahami, yaitu disajikan format yang mudah dimengerti.
6. Dapat diuji kebenarannya, yaitu memungkinkan dua orang yang kompoten untuk menghasilkan informasi yang sama secara *independent.*

## 2.3 Konsep Dasar Sistem Informasi

Sistem informasi dapat diartikan sebagai suatu cara mengelola pekerjaan informasi dengan menggunakan pendekatan sistem. Sistem informasi merupakan suatu proses sistem manusia dan mesin yang terintregasi secara harmonis untuk menyediakan informasi yang dapat digunakan oleh *user* untuk mendukung fungsi operasi, manajemen dan pengambilan keputusan di dalam organisasi baik secara manual maupun dengan kemampuan teknologi informatika (komputer).

"*Sistem informasi adalah suatu sistem dari suatu sistem organisasi yang mempertemukan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dan laporan yang diperlukan*." GOR [2].

Sistem informasi merupakan proses yang diselenggarakan untuk mendapatkan informasi melalui suatu prosedur yang memanfaatkan sumber daya manajemen kedalam suatu keperluan tertentu.

Komponen sistem informasi adalah sebagai berikut :

1. *Hardware* (perangkat keras), terdiri dari komputer, printer, jaringan.
2. *Software* (perangkat lunak).
3. Data, merupakan komponen dasar informasi.
4. Manusia (*user*).
5. Prosedur, seperti buku penyusun operasional dan elektronik.

Kegiatan – kegiatan di dalam suatu informasi meliputi :

1. *Input*, kegiatan menyediakan data untuk diproses.
2. Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan informasi yang bernilai lebih.

3. *Output*, kegiatan yang menghasilkan keluaran berupa informasi hasil dari proses.
4. Penyimpanan, kegiatan untuk memelihara dan menyimpan data.
5. Kontrol (pengendali), suatu aktifitas untuk menjamin bahwa sistem informasi berjalan sesuai dengan jalan yang diharapkan.

## 2.4 Metode Pendekatan Sistem

Metode pendekatan sistem yang digunakan adalah *Object Oriented* yang menekankan pada solusi *logic* berbasis objek. Teknologi *object oriented* merupakan paradigma baru dalam rekayasa perangkat lunak yang didasarkan pada objek dan kelas.

### 2.4.1 Metode Berorientasi Objek

*Object-oriented* mencakup bidang aplikasi yang sangat luas. Para pengguna sistem komputer dan sistem lain yang didasarkan atas teknologi komputer merasakan efek *object-oriented* dalam bentuk meningkatnya aplikasi *software* yang mudah digunakan dan servis yang lebih fleksibel, yang muncul dalam berbagai bidang industri, seperti dalam perbankan, telekomunikasi, dan sebagainya. Sedangkan bagi *software engineer*, *object-oriented* berpengaruh dalam bahasa pemrograman, metodologi rekayasa, manajemen proyek, *hardware*, dan sebagainya.

### 2.4.2 Object Oriented Analysis

*Object Oriented Analysis (OOA)* adalah metode analisis yang memeriksa *requirements* (syarat/keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan.

Dalam *object orinted analysis* (OOA), terdapat lima lapisan (tidak bersifat *sequential*), yaitu:

1) Menentukan lapisan subjek
2) Menentukan lapisan kelas dan objek
3) Menentukan lapisan struktur
4) Menentukan lapisan atribut
5) Menentukan lapisan servis

### 2.4.3 Object Oriented Design (OOD)

*Object Oriented Design (OOD)* adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem. Dalam *Object Oriented Design (OOD)*, terdapat lima lapisan (tidak bersifat *sequential*), yaitu:

1) Menentukan lapisan subjek
2) Menentukan lapisan kelas dan objek
3) Menentukan lapisan struktur
4) Menentukan lapisan atribut
5) Menentukan lapisan servis

### 2.4.4 Konsep Dasar Dalam OOAD

Konsep dasar dalam OOAD terdiri dari objek, kelas, *Encapsulation*, *Association* dan *Aggregation*.

**1. Objek (*object*)**

Adalah "benda" secara fisik atau konseptual, yang dapat kita temui disekeliling kita. *Hardware*, *Software*, dokumen, manusia, dan bahkan konsep semuanya adalah contoh objek.

***a) State***

Adalah kondisi objek tersebut atau himpunan dari keadaan yang menggambarkan objek tersebut. *State* dinyatakan dengan nilai atribut (*attribute*) objeknya.

***b) Atribut***

Adalah nilai internal suatu objek yang mencerminkan antara lain karakteristik objek, kondisi sesaat koneksi dengan objek lain, dan identitas. Perubahan *state* dicerminkan oleh perilaku (*behavior*) objek tersebut.

***c) Behavior***

Suatu objek mendefinisikan bagaimana sebuah objek bertindak (beraksi) dan memberi reaksi. *Behavior* ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan dalam objek itu sendiri. *Behavior* dari sebuah objek dicerminkan oleh *interface*, *service*, dan *method* dari objek tersebut.

***d) Interface***

Adalah pintu untuk mengakses *service* objek.

***e) Service***

Adalah fungsi yang bisa diemban objek.

***f) Method***

Adalah mekanisme internal objek yang mencerminkan perilaku objek tersebut. Sebagai contoh, jika printer merupakan sebuah method, perilaku atau servicenya mencetak apapun yang dia terima.

**2. Kelas (class)**

Kelas *(class)* adalah definisi umum (pola, *template* atau cetak biru) untuk himpunan objek sejenis. Kelas menetapkan spesifikasi perilaku *(behaviors)* dan atribut objek-objek tersebut. *Class* adalah keniskalan (abstraksi) dari entitas dalam dunia nyata. Objek adalah "contoh" *(instance)* dari sebuah kelas.

**3. *Encapsulation* (Pangkapsulan)**

Adalah proses menyembunyikan detil implementasi sebuah objek. Satu-satunya jalan untuk mengakses data objek tersebut adalah *interface. Interface* melindungi *internal state* sebuah objek dari sebuah objek dari pihak luar.

Dalam *object-oriented programming*, kode dan data disatukan dalam sebuah "benda" yang tersembunyi isinya, yaitu objek. Pengguna objek tidak perlu mengetahui isi dalam kotak tersebut. Untuk dapat berkomunikasi dengan objek, diperlukan pesan (*message*).

*Message* dapat didefenisikan sebagai permintaan untuk objek penerima (*receiver object*) untuk membawa metode yang ditunjukan dan mengembalikan *result* (hasil) dari aksi tersebut kepada objek pengirim (*sender objek*).

***4. Association dan Aggregation***

*Association* (asosiasi) hubungan antar objek yang saling membutuhkan. *Aggregation* (agregasi) adalah bentuk khusus dari asosiasi yang menggambarkan seluruh bagian suatu objek yang merupkan bagian dari objek lain.

## 2.5 Pengenalan UML (Unified Moseling Language)

UML *(Unified Modeling Language*) merupakan bahasa pemodelan yang paling sukses dari tiga metode OO *(Object-oriented*) yang telah ada sebelumnya, yaitu Booch, OMT, dan OOS. UML merupakan kesatuan dari ketiga metode pemodelan terebut dan ditambah kemampuan lebih karena mengandung metode tambahan untuk mengatasi masalah pemodelan yang tidak dapat ditangani ketiga metode tersebut.

### 2.5.1 Pengertian UML

UML (*Unified Modeling Language)* adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan *artifacts* (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 UML adalah bahasa pemodelan yang yang menggunakan konsep orientasi objek.

UML menyediakan notasi –notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

Tujuan utama UML diantaranya untuk:

1. Memberikan model yang siap pakai, bahasa pemodelan yang visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

### 2.5.2 Bagian-bagian UML

Bagian-bagian utama dari UML (*Unified Modeling Language)* yaitu:

*1. View*

*View* digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. *View* bukan melihat grafik, tapi merupakan suatu abtraksi yang berisi sejumlah diagaram.

Beberapa janis *View* dalam UML antara lain: *use case view*, *logical view*, *component view*, dan *deployment view*.

*a) Use case view*

*Use case view* digunakan untuk memahami dan menggunakan sistem yang kita modelkan. *View* ini melihat pada bagian *actor* dan *use case* berinteraksi.

*b) Logical view*

*Logical view* mengarah pada persyaratan (*requirements)* fungsional sistem. *View* ini melihat pada kelas-kelas dan hubungan antar kelas-kelas tersebut.

*b) Component view*

*Component view* mengarah pada pengaturan *software*. *View* ini mengandung informasi mengenai komponen-komponen *software*, komponen tereksekusi (*executabel)* dan *library* untuk sistem yang kita modelkan.

*c) Deployment view*

*Deployment view* memperlihatkan pemetaan setiap proses kedalam *hadware*. *View* ini paling bermanfaat ketika kita membuat model suatu sistem yang diterapkan dalam lingkungan arsiktetur yang terdistribusi dimana kita menerapkan aplikasi dan *server* pada lokasi yang berbeda.

*2. Diagram*

*Diagram* berbentuk grafik yang menunjukan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah digram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu.

Adapun jenis diagram antara lain : *use case digaram, class diagram, statechart diagram, sequence diagram, collaboration diagram, activity diagaram, component diagaram, deployment diagram.*

a) *Use case diagram*

Menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor).* Diagram ini menunjukan fungsionalitas suatu sistem atau kelas dan bagaiman sistem berinteraksi dengan dunia luar. *Use case diagram* dapat digunakan selama proses analisis untuk menangkap *requirement*s sistem dan untuk memahami bagaimana sistem seharusnya bekerja.

b) *Class diagarm*

Membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem.

c) *Statechart diagram*

Digunakan untuk memodelkan perilaku diamis satu kelas atau objek. *Statechart diagram* memperlihatkan urutan keadaan sesaat (*state*) yang dilalui sebuah objek, kejadian yang menyebabkan sebuah transisi dari satu *state* atau aktivitas kepada yang lainnya, dan aksi yang menyebabkan perubahan atau *state* dan aktivitas.

d) *Sequence diagram*

Menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. *Sequence diagram* memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case.*

e) *Collaboration diagram*

Melihat pada interaksi dan hubungan terstruktur antar objek. Tipe diagram ini menekankan pada hubungan *(relationship*) antar objek. *Collaboration diagram* digunakan sebagai alat untuk menggambarkan interaksi yang mengungkapkan keputusan mengenai perilaku sistem.

f) *Activity diagram*

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

g) *Component diagram*

Menggambarkan alokasi semua kelas dan objek kedalam komponen-komponen dalam desain fisik sistem *software*. Diagram ini memperlihatkan pengaturan dan kebergantungan antara komponen-kompone *software* seperti *source code, binary code*, dan komponen tereksekusi *(executabel component*)

h) *Deploment diagram*

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Diagram ini memperlihatkan pemetaan *software* kepada *hardware*.

### *2.5.3 Indikator Multiplicity*

Walaupun *multiplicity* ditentukan untuk *class*, *multiplicity* menentukan banyak *object* yang terlibat dalam relasi. *Multiplicity* menentukan banyaknya *object* yang terhubung satu dengan yang lainnya. *Indikator multiplicity* terdapat pada masing-masing akhir garis relasi, baik pada asosiasi agregasi. Beberapa contoh *multiplicity* adalah:

| | |
|---|---|
| 1 | Tepat satu |
| 0..* | Nol atau lebih |
| 1..* | Satu atau lebih |
| 0..1 | Nol atau Satu |
| 5…8 | Range 5 s.d 8 |
| 4..6,9 | range 4 s.d 6 dan 9 |

### 2.5.4 Business Modelling

Dalam dunia bisnis dan industri, terdapat banyak sistem manual dan otomatis yang muncul secara reguler. Setiap sistem tersebut memiliki satu atau banyak *workflow*. *Business Modelling* menggambarkan semua *workflow* yang terjadi dalam suatu organisasi.

Secara formal, kita defenisikan *business modeling* sebagai segala teknik pemodelan yang digunakan untuk meninjau, meningkatkan dan membuat sebuah bisnis.

Tujuan *Business modelling* adalah sebagai berikut:

1. Memahami struktur dan dinamika organisasi.
2. Memahami masalah-masalah dalam mencapai target organisasi dan menemukan potensi untuk kemajuan organiasi.
3. Yakin bahwa para *cutomer, end user, dan developer* mempunyai sebuah pemahaman yang benar mengenai organisasi.
4. Dapat menurunkan/mendapatkan *requirements software* aplikasi yang akan kita buat yang diperlukan untuk mendukung pencapaian target organisasi.

Untuk mencapai tujuan tersebut, harus membuat dua jenis model bisnis, yaitu model use-case bisnis (*business-use-cae model*) dan model objek bisnis *(business object model*) yang berisi proses-proses, peranan-peranan, dan tanggung jawab organisasi untuk mencapai targetnya.

**Proses Bisnis**

Dalam suatu organisasi mungkin terdapat banyak proses bisnis. Proses bisnis *(business process*) adalah sekumpulan aktivitas yag 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 disini adalah urutan aktivitas tertentu terhadap waktu dan tempat, dengan sebuah titik awal, sebuah titik akhir, dan sebagai *input* dan *output* yang didefinisikan dengan baik.

**Stereotype Dalam Model Bisnis**

*Rational Rose* memiliki beberapa jenis elemen yang digunakan dalam *business modelling*, yaitu:

1. *Business Actor*

   *Business Actor* (aktor bisnis)menggambarkan peran yang dimainkan oleh seseorang atau sesuatu yang dengannya bisnis berinteraksi. Sebuah *business actor* menggambarkan seorang *customer* atau partner bisnis, tetapi sebuah sistem informasi yang berhubungan dengan bisnis kita dapat berperan sebagai sebuah *business actor* juga.

2. *Business Use-Case*

   *Business use-case* merupakan urutan tindakan yang dimainkan suatu bisnis yang menghasilkan sebuah nilai yang dapat dilihat dan diwujudkan untuk

suatu *bussiness actor* tertentu. Setiap *business use-case* mewakili satu proses bisnis.

3. *Business Use-Case Realization*

*Business use-case realization* memberika sudut pandang dari dalam (*internal view*) terhadap *business use-case*, yakni menentukan bagaimana pekerjaan (dalam *business-case*) tersebut diatur dan dimainkan untuk mencapai hasil yang diinginkan. *Sebuah business-case* realization menentukan *business worker* dan *business entity* yang terlibat ketika menjalankan sebuah *business use-case.*

4. *Business Worker*

*Business worker* (pekerja bisnis) adalah sebuah kelas yang mewakili seorang manusia yang bertindak dalam bisnis. Sebuah *business worker* berinteraksi dengan para *worker* lainnya dan memainkan entitas bisnis saat berpartisipasi dalam realisasi *suatu business use-case.*

5. *Business Entity*

*Business entity* (entitas bisnis)menggambarkan “benda” yang ditangani atau digunakan oleh *business worker* selama melaksanakan suatu *business use-case*

### 2.5.5 Requirement

*Requirement* adalah kondisi atau kemampuan yang harus dipenuhi oleh *software* apliaksi yang akan dibuat. *Requirement* untuk *software* dibagi dalam dua

kelompok, yaitu *function requirement* dan *non-function requirement. Function requirement* menentukan tindakan yang harus dapat dimainkan sebuah *software* aplikasi. *Function requirement* mencakup fitur, kemampuan *(capabilities*), dan faktor *sequrity software* aplikasi .*Non-function requirement* menggambarkan atribut dari *software* aplikasi dan lingkungannya. *Requirement* ini biasanya masuk di dalam *use-case* dan tercakup dalam sifat *(prooerty) use-case* tersebut.

### 2.5.6 Model Analisis

Model analisis *(analysis model*) menggambarkan realisasi dari *use-case* dalam *use case* model, tujuan akhir sebuah model analisis adalah untuk membuat pemetaan awal mengenai perilku yang diisyaratkan dalam sistem aplikasi kedalam elemen-elemen pemodelan.

Elemen model yang terdapat dalam model anlisis disebut kelas analisis (*analysis class*). Kelas analisis adalah kelas ber-*stereottype* "*boundary*","*control*","*entity*" yang menggambarkan sebuah konsep awal mengenai "benda" dalam sistem aplikasi yang memiliki tanggung jawab dan perilaku. *Boundary* adalah kelas yang memodelkan interaksi antara satu atau lebih *actor* dengan sistem. *Control* digunakan untuk memodelkan "perilaku mengatur", khusus untuk satu atau beberapa *use-case* saja. *Entity* (entitas) memodelkan informasi yang harus disimpan oleh sistem.

## 2.6 Basis Data

Basis data terdiri dari dua kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa dan lain sebagainya.

Basis data merupakan kumpukan dari data-data yang saling terkait dan saling berhubungan satu dengan yang lainnya. Basis data adalah kumpulan-kumpulan *file* yang saling berkaitan.

Salah satu pemakai sistem informasi manajemen *data base* yaitu administratur basis data (*Data Base Administrator/DBA*) fungsi ini bertangung jawab untuk melakukan koordinasi, pengawasan, dan pengelolaan *data base*.

Basis data memiliki tujuan yang diantaranya adalah sebagai berikut:

1. Kemampuan dalam menangani data dalam jumlah besar.
2. Efisiensi pengolahan data yang meliputi *speed, space dan accuracy*.
3. Kemampuan pemakai *resource* basis data secara bersama-sama *(sharebility*).
4. Meniadakan duplikasi data yang tidak perlu dan menghindari inkonsistensi data.

## 2.7 Arsitektur Aplikasi

Arsitektur aplikasi dapat pula dikatakan sebagai struktur desain dari suatu sistem komputer dengan semua kelengkapannya yang telah siap untuk digunakan oleh *user.*

### 2.7.1 Pengertian Jaringan Komputer

Jaringan komputer adalah sekumpulan komputer pribadi yang saling berhubungan antara satu dengan yang lainnya dengan menggunakan *protocol* komunikasi sehingga dapat saling berbagi informasi, program penggunaan bersama perangkat keras seperti printer, hardisk. Selain itu jaringan komuputer dapat diartikan sebagai kumpulan sejumlah terminal komunikasi yang berada diberbagai lokasi yang terdiri dari lebih dari satu komputer yang saling berhubungan. Dalam sebuah jaringan komputer dapat diartikan sebagai kumpulan sejumlah terminal komunikasi yang ada diberbagai lokasi yang terdiri dari berbagai lokasi yang terdiri dari satu komputer yang salig berhubungan . Dalam sebuah jaringan komputer biasanya terhubung sejumlah komputer kesebuah/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 untuk membawa informasi secara tepat dan tanpa adanya kesalahan dari sisi pengirim (transmisi) menuju kesisi penerima (receiver) melalui media komunikasi.

TCP (*Transmission Control Protocol / Internet protocol*) merupakan *protoco*l standar internet yang digunakan untuk melakukan koneksi ke internet *protocol*. TCP/IP memiliki beberapa *subjek protocol* yang berbeda yang beroperasi pada lapisan yang berbeda dan mempunyai tugas masing-masing.

Berkat adanya *protocol* ini setiap komputer dapat berhubungan secara fleksibel dengan *host-host* yang terkoneksi.

### 2.7.2 Jenis-Jenis Jaringan Komputer

Jenis-jenis jaringan ada dua yaitu:

1. *Model peer to peer*

   Menurut model ini, setiap *host* dapat menawarkan layanan *ke peer* lain dan juga mengambil layanan dari *peer* lain. Model ini cocok untuk jaringan kecil.

2. *Model Client / Server*

   Model ini memisahkan secara jelas, mana yang dapat memberikan layanan jaringan **(server**) dan mana yang memberikan layanan (*client).* Beberapa komputer *diset-up* sebagai *server* yang memberikan sumber daya (*resource)* dari jaringan : printer, modem, dan saluran lainnya kepada komputer lain yang dikoneksi kejaringan yang berfungsi sebagai *client.*

### 2.7.3 Topologi Jaringan Komputer

Yang dimaksud topologi jaringan itu adalah susunan fisik bagaimana node-node saling dihubungan. Ada tiga topologi jaringan komputer,yaitu:

1.*Topologi bus*

*Ethernet* adalah contoh yang menggunakan *topologi bus* yang menggunakan satu kawat (kabel) yang berfungsi sebagai medium untuk mentransmisikan data. Node yang merupakan bagian dari jaringan dihubungkan seluruhnya ke kabel tersebut. Node-node yang terhubungkan mengirim dan menerima data jaringan melalui kabel sebagai pembawa sinyal dan melihat apakah data tersebut ditujukan buat dirinya.

2. Topologi Ring

Salah satu node dihubungkan dengan node yang ada didepan dan dibelakangnya sehingga membentuk *ring*. Setiap node mendapatkan giliran menggunakan jaringan dengan mengirim “token”. Node yang mendapat giliran dapat mengirim data dan node lain menerima data, serta melihat apakah data ditujukan kepadanya. Bila ditujukan buat dirinya, data pun disimpan, bila tidak data diteruskan ke node didepannya.

3.*Topologi Star atau hub*

Susunan atau skema dari topologi ini mirip sebuah bintang. Topologi ini memiliki satu *hub* pusat dari mana data ditransmisikan keseluruh node dalam jaringan. Skema ini mempunyai kelebihan dibandingkan dua skema sebelumnya,

yaitu bila terjadi kerusakan pada kabel tidak membawa dampak bagi seluruh node, tapi hanya node yang berangkutan saja sehingga aktivitas jaringan tidak terganggu secara total. Ini berbeda dengan skema bus atau ring, dimana bila terjadi kerusakan pada kabel berakibat pada seluruh jaringan.

Topologi yang digunakan dalam merancang sistem ini adalah topologi *star* atau *hub* karena topologi ini mempunyai banyak kelebihan dibandingkan dengan topologi lain.

### 2.7.4 Manfaat Jaringan Komputer

Manfaat yang di dapat dari membangun jaringan komputer adalah sebagai berikut:

a. *Sharing Resources*

   *Sharing resources* bertujuan agar seluruh program, peralatan/periperal lainnya dapat dimanfaatkan oleh setiap orang yang ada pada jaringan komputer tanpa terpengaruh pada jaringan komputer, tanpa terpengaruh oleh lokasi maupun pengaruh dari pemakai. Dengan kata lain, seorang pemakai yang letaknya jauh sekalipun dapat memanfaatkan data maupun informasi yang lainnya tanpa mengalami kesulitan.

b. Media Komunikasi Jaringan komputer memungkinkan terjadinya komunikasi antara pengguna, baik untuk t*eleconference* maupun untuk mengirim pesan /informasi yang penting lainnya.

c. Integrasi Data

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

d. Pengembangan dan Pemeliharaan

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

e. Keamanan Data

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

f. 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 secara langsung diketahui oleh setiap pemakai.

## 2.8 Pengertian Client / Server

Sistem *clien-server* mempunyai dua komponen utama yaitu komputer *client* dan komputer *server. Server* merupakan komputer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari komputer *clien* dan bertindak sebagai *server database* yang menyimpan data. *Client* yaitu komputer atau *workstation* yang melakukan pengiriman permintaan-permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya. Selain itu *client* juga mempuyai kemampuan untuk mengubah aatu menghapus data itu.

Sistem *clien-server* suatu sistem *clien* 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 memori yang sama namun terbagi dalam komputer *client-server.*

Hal ini menjadikan konfigurasi bagi komputer *client* dan komputer *server* bisa berbeda seperti kapasitas memori, kecepatan *prosesor* 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 *windows NT server*, *windows 2000 server*, sedangkan komputer *clien* menggunakan konfigurasiyang umum bagi sebuah komputer dekstop yag terhubung ke jaringan dengan sistem operasi seperti *windows 98, windows ME, windows XP* dan lain-lain.

## 2.9 Perangkat Lunak pendukung

Perangkat lunak ini terdiri dari Sistem Opersi (SO) dan paket apliaksi *data base*. Adapun spesifikasi perangkat lunak tersebut adalah:

### 2.9.1 Sekilas Tentang Visual Basic 6.0

Microsoft Visual Basic 6.0 adalah bahasa pemrograman yang bekerja dengan lingkup Microsoft Windows. Microsoft Visual Basic 6.0 dapat memanfaatkan kemampuan Microsoft Windows secara optimal. Kemampuannya dapat dipakai untuk merancang program aplikasi yang berpenampilan seperti program aplikasi lainnya berbasis Microsoft Windows. Visual Basic merupakan event-driven programming (programan terkendali kejadian) artinya program menunggu sampai adanya respon dari pemakai berupa event/ kejadian tertentu (tombol, diklik, menu dipilih, dan lain-lain). Ketika event terditeksi, kode yang berhubungan dengan event akan dijalankan.

Visual Basic berawal dari sebuah ide untuk membuat simple *scripting language* untuk GUI Windows. Saat itu masih windows 3.1.1 sebagai bahasa pemrograman yang tergolong sebagai RAD (Rapid Aplication Development) tool, perkembangan visual basic mendapat sambutan yang sangat baik dikalangan *programmer*.

Ledakan jumlah developer atau programmer yang sangat berarti terjadi saat munculnya visual Basic .5 , dimana Visual Basic mampu melakukan dobrakan yang sangat besar, yaitu kemampuan membuat *active control* besderta komponennya (active DDL dan avtive EXE). Hal ini menyedot perhatian developer dan mempengaruhi perubahan teknik pemrograman Visual Basic, terutama pada konsep OOP (*Object Oriented Programming*).

Pada oktober 1997 muncul Visual Basic 6.0 yang tidak jauh berbeda dengan Visual Basic 5 henya saja Visual Basic 6.0 sudah ditambah dengan kemampuan internet, programming, dan juga penyempurnaan pada tingkat keamanan saat menjalankan *thread*, *COM-based application*, yang kemudian membuat trend pemrograman Visul Basic 6.0 mulai bergeser dari aplikasi *desktop* kearah aplikasi *network* dan desain komponen. Hal ini terlihat dengan hadirnya *template IIS application project* dan *DHTML project* pada pilihan menu utama jenis *project.*

Aplikasi (project) terdiri dari :

1. Form – windows/ jendela dimana anda akan membuat *user interface/*tampilan.

2. Kontrol/Contol – tampilan derbasis grafis yang dimasukan pada form untuk memmbuat interaksi dengan pemakai (text box, label, scroll bar, tombol command).

3. Properti/*properties* – nilai /karakteristik yang dimiliki oleh sebuah object Visual Basic. Contoh : *name, caption, size, color.*

4. Metode/ method – serangkaian perintah yang sudah tersedia pada suatu object yang dapat diminta untuk mengrjakan tugas khusus.

5. Prosedur kejadian/ *event procedure* – kode yang berhubungan dengan satu object dimana kode ini akan di eksekusi ketika ada respon dari pemakai berupa event tertentu.

6. Prosedur Umum/ General Procedure – kode yang tak berhubungan dengan suatu object, kode ini harus diminta oleh aplikasi.

7. Modul/ Module – kumpulan dari prosedur umum, deklarasi variable dan definisi konstanta yang digunakan oleh aplikasi.

### 2.9.2 Sekilas tentang My SQL

My Sql adalah multiuser data base yang menggunakan bahasa *Structured Query language* (SQL). My SQL dalam operasi client – server melibatkan server *daemon* My SQL di isi server dan berbagai macam program serta *library* yang berjalan di isi client.

SQL singkatan dari *Structured Query language* sering dibaca sebagai “Sequel” adalah bahasa tersetruktur yang digunakan untuk query meng update dan

mengolah relasi antar data base. User dapat menggunakan bahasa ini untuk memuat, mengurutkan, dan menyaring suatu data sehingga dihasilkan suatu data yang spesifik dari sutu data base.

### 2.9.3 Rational Rose

*Rational Rose* adalah *software* yang memiliki perangkat-perankat pemodelan secara visual untuk membangan suatu solusi dalam rekayasa *software* dan pemodelan bisnis. *Rational Rose* dikeluarkan oleh perusahaan *software* bernama *Rational Software*, perusahaan yang mencetuskan ide pembentukan konsersium bagi perusahaan-perusahaan yang memakai standar UML sebagai bahasa pemodelan di perusahaanya, *Rational Rose* memakai UML sebagai bahasa pemodelannya, ditambah beberapa fitur lain yang membuat *Rational Rose* menjadi *software* pemodelan visual yang termuka. Beberapa fitur yang termuka diantaranya *Rational Rose* memiliki *Rational Unified Process* (RUP) selain itu, *Rational Rose* memiliki kemampuan membuat solusi *client/server*, yang kemudian diterapkan dan di ditribusikan dalam lingkungan perusahaan.

*Rational Rose* memiliki keungulan, diantaranya:

1. Bahasa yang digunakan adalah bahasa pemodelan standar yaitu UML, akan meningkatkan komunikasi intra tim.

2. *Rational rose* mendukung mengandung *round-trip enginering*, sehingga kita dapat mengenerate model kedalam kode (java, C++, Visual Basic, dan sebagainya) dan melakukan *reverse engineering* untuk menampilkan

arsitektur *software* dari kode yang ada. Hal ini dapat dilakukan secara bolak-balik sebagai proses *interative* selama proses rekayasa *software.*

3. Model dan kode senantiasa singkron selama dalam *development cycle*.

4. Membantu *software* menggunakan *Rational Rose* memudahkan dalam memperbaiki *software* tersebut karena apbila suatu saat ditemukan *requirement* baru dapat digambarkan kembali *software* tersebut dalam UML.

5. Para *User Rational Rose* dapat berkomunikasi walaupun bekerja dalam sistem operasi yang berbeda (*windows atau UNIX).*

6. Dengan menggunakan *Rose Web Publisher* suatu tim dapat mengkomunikasikan model dan spesifikasinya dalam *Web Browser*

7. Mendukung rekayasa *software* untuk sistem *clien/server* sehingga *Rational Rose* merupakan *software* pemodelan visual yang tangguh dalam lingkungan *client/server, e-business,* dan lingkungan perusahaan terdistribusi (kantor-kantornya terletak dalam tempat yang terpisah-pisah).

## 2.10 Kasus yang sedang di rancang/dianalisis

PT. Riau Permai bergerak dalam bidang bisnis penjualan dan pembelian dimana terjadi proses penjualan dan pembelian ala-alat tulis kantor.

### 2.10.1 Penjualan

Penjualan adalah peristiwa dimana terjadinya pertukaran barang atau jasa dan dicatat untuk satu periode akuntansi tertentu, baik berdasarkan kas atau berdasarkan aktual. Penjualan merupakan salah satu fungsi dari pemasaran atau merupakan bagian dari kegiatan pemasaran. Penjualan sangat penting dan menentukan, karena suatu perusahaan untuk dapat memasarkan produknya, yaitu dengan cara menjual produk tersebut, dan apabila penjualan tidak dapat dilaksanakan maka fungsi-fungsi lain dari pemasaran tidak akan berjalan.

Pengertian penjualan itu sendiri menurut " *American Marketing Asociation* " diterapkan sebagai berikut :

*" Proses pemberian bantuan atau persuasi secara pribadi atau non pribadi agar membeli suatu komoditi atau jasa agar bertindak yang menguntungkan suatu gagasan atau ide yang mengandung arti komersial bagi penjual*."Berdasarkan pemikiran tersebut diatas maka dapat di ambil kesimpulan bahwa yang dimaksud dengan *system* adalah :

*" Pola hubungan antar bagian-bagian yang saling berkaitan untuk melakukan kegiatan yaitu memproses data penjualan sehingga dihasilkan data yang cepat-cepat dan akurat yang memuaskan keinginan pembeli sehingga tujuan perusahaan dapat tercepat."*

### 2.10.2 Pembelian

Pembelian merupakan salah satu satu fungsi dari pembelanjaan atau merupakan kegiatan dari pembelanjaan. Pembelian sama pentingnya dengan penjualan, yaitu untuk memenuhi kebutuhan tiap perusahaan, seperti kebutuhan peralatan kantor, gedung, peralatan produksi, dan lain sebagainya.

Pengertian pembelian dalam suatu perusahaan menurut AZW [1] yaitu:

"*Pembelian adalah proses terjadinya pengadaan barang yang tergolong kurang memenuhi kebutuhan yang diperlukan"*.

Pembelian terbagi menjadi dua cara yaitu :

1. Pembelian tunai

   Yaitu pembelian barang secara langsung kepada supplier atau pemasok barang dengan pembayaran secara langsung saat itu juga.

2. Pembelian Kredit

   Yaitu pembelian barang secara langsung kepada supplier atau pemasok barang dengan pembayaran secara tidak langsung, melainkan berangsur sesuai kesepakatan.