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 menyelesaikan suatu sasaran tertentu*. Sedangkan menurut GOR [2] mendefinisikan sistem adalah " *Sistem terdiri dari bagian – bagian yang bersama beroperasi untuk mencapai berberapa 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 system dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini dapat dipandang sebagai satu kesatuan dan menunjukan ruang lingkup (*scope*) dari sistem tersebut.

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 unutk 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.

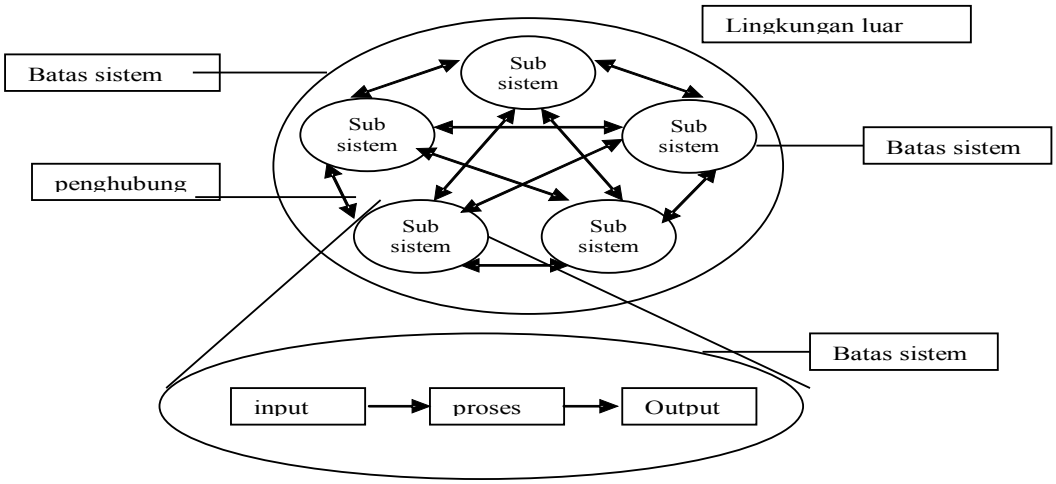
7. Pengolah (*Process*)

   Suatu 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 2.1 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 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 sesuatu 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 dalam format yang mudah dimengerti.
6. Dapat diuji kebenarannya, yaitu memungkinkan dua orang yang kompeten 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 terintegrasi 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* berpegaruh dalam bahasa pemrograman, metodologi rekayasa, manajemen proyek, *hardware*, dan sebagainya.

2.4.2 Object Oriented Analysis (OOA)

O*bject 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 oriented 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)

Oobject Oriented Design (OOD) adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem.Dalam Ob*ject 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 dari 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 tau 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 sebuah merupakan, 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 menentapkan 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 (*Pengkapsulan*)*

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

4. *Association dan Aggregation*

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

2. 5 Pengenalan UML (Unified Modeling Language)

UML (*Unified Modeling Language*) merupakan bahasa pemodelan yang paling sukses dari tiga metode OO (*Object-oriented*) yang telah ada sebelumnya, yaitu Booch, OMT, and OOSE.UML merupakan kesatuan dari ketiga metode pemodelan tersebut 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 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 diagram.

Beberapa jenis *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 bagaiman *actor* dan *use case* beinteraksi.

b) Logical view

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

c) Component view

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

d) *Deployment view*

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

2. Diagram

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

Adapun jenis diagram antara lain : *use case diagram, class diagram, statechart diagram, sequence diagram, collaboration diagram, activity diagram, component diagram, 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 bagaimana sistem berinteraksi dengan dunia luar. *Use case diagram* dapat digunakan selama proses analisis untuk menangkap *requirements* sistem dan untuk memahami bagaimana sistem seharusnya bekerja.

b) *Class diagram*

Membantu dalam visualisasi 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 dinamis 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 satu 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 aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi 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-komponen *software* seperti *source code*, *binary code*, dan komponen tereksekusi (*executabel component)*.

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. Diagram ini memperlihatkan pemetaan *software* kepada *hardware*.

2.5.3 Indikator Multiplicity

Walaupun *multiplicity* ditentukan untuk *class, multiplicity* menentukan banyaknya *object* yang terlibat dalam relasi.*Multiplicity* menentukan banyaknya *objec*t yang terhubung satu dengan yang lainnya.Indikator *Multiplicity* terdapat pada masing-masing akhir garis relasi, baik pada asosiasi maupun 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 regular.Setiap sistem tersebut memiliki satu atau banyak *workflow*.*Business Modelling* menggambarkan semua *workflow* yang terjadi dalam suatu organisasi.

Secara formal, kita definisikan *business modeling* sebagai segala teknik pemodelan yang digunakan untuk menggambarkan model sebuah bisnis.*Business modeling* dapat 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 organisasi.
3) Yakin bahwa para *customer, 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-case 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 yang dirancang untuk menghasilkan keluaran (*output*) tertentu bagi *customer* tertentu.Sebuah proses bisnis menekankan pada bagaimana sebuah pekerjaan dikerjakan dalam sebuah organisasi, dengan berfokus pada produk yang dihasilkan proses tersebut.proses 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 modeling*, 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 ditujukan untuk suatu *business actor* tertentu.Setiap *business use-case* mewakili satu proses bisnis.

3) *Business Use-Case Realization*

Business Use-Case Realization memberikan sudut pandang dari dalam (*internal view*) terhadap *business use-case*, yakni menentukan bagaimana pekerjaan (dalam *business use-case*) tersebut diatur dan dimainkan untuk mencapai hasil yang diinginkan. Sebuah *Business Use-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* aplikasi yang akan dibuat.*Requirement* untuk *software* aplikasi dibagi ke dalam dua kelompok, yaitu *Function requirement dan Non-function requirement*. *Function requirement* menentukan tindakan yang harus dapat dimainkan sebuah *software* aplikasi.*Requirement* ini sering dimodelkan dalam bentuk use-*case-use-case* dalam *use-case model. Function requirement* juga menentukan masukan dan keluaran dari 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 *(property*) *use-case* tersebut.

2.5.6 Model Analisis

Model analisis (Analysis Model) menggambarkan realisasi dari *use-case-use-case* dalam *use-case model*, tujuan akhir sebuah model analisis adalah untuk membuat pemetaan awal mengenai perilaku yang disyaratkan dalam sistem aplikasi ke dalam elemen-elemen pemodelan.

Elemen model yang terdapat dalam model analisis disebut kelas analisis (*analysis class*).Kelas analisis adalah kelas ber-*stereotype "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.*Entit*y(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 mewakili suatu objek seperti manusia, barang, hewan, peristiwa dan sebagainya.

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

Salah satu pemakai sistem informasi manajemen *database* yaitu administratur basis data (*Database Administrator*/*DBA*) fungsi ini bertanggung jawab untuk melakukan koordinasi, pengawasan, dan pengelolaan *database.*

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 pemakaian *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 saling berhubungan antara satu dengan yang lainnya menggunakan *protocol* komunikasi sehingga dapat saling berbagi informasi, program penggunaan bersama perangkat keras seperti *printer, harddisk*. Selain itu jaringan komputer dapat diartikan sebagai kumpulan sejumlah terminal komunikasi yang berada di berbagai lokasi yang terdiri dari lebih dari satu komputer yang saling berhubungan. Dalam sebuah jaringan komputer dapat diartikan sebagai kumpulan sejumlah terminal komunikasi yang berada di berbagai lokasi yang terdiri dari satu komputer yang saling berhubungan. Dalam sebuah jaringan komputer biasanya terhubung sejumlah komputer ke sebuah/ beberapa *serve*r.

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 dan tanpa adanya kesalahan dari sisi pengirim (*transmisi*) menuju ke sisi penerima (*receiver*) melalui media komunikasi.

TCP/IP (*Trasmission Control Protocol/Internet Protocol*) merupakan protokol standard *internet* yang digunakan untuk melakukan koneksi ke *internet* protokol. TCP/IP memiliki beberapa subyek protocol yang berbeda yang beroperasi pada lapisan yang berbeda dan mempunyai tugas masing-masing.

Berkat adanya protokol 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 hanya 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 dihubungkan. Ada tiga topologi jaringan komputer, yaitu:

1. *Topologi Bus*

Ethernet adalah contoh yang menggunakan topologoi *bus* yang menggunakan satu kawat (kabel) yang berfungsi sebagai medium untuk mentranmisikan 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, datapun 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 ke seluruh *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 bersangkutan 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 yang lain.

2.7.4 Manfaat Jaringan Komputer

Manfaat yang didapat dari membangun jaringan komputer adalah sebagai berikut :

1. Sharing Resources

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

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

2. Media komunikasi

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

3. Integrasi data

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

4. Pengembangan dan Pemeliharaan

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

5. Keamanan Data

   Sistem jaringan komputer memberikan perlindungan terhadap data jaminan keamanan data tersebut di berikan melalui pengaturan hak akses para

pemakai dan *password*, serta teknik perlindungan terhadap *harddisk* sehingga data mendapatkan perlindungan yang efektif.

6. Sumber Daya Lebih Efisien dan Informasi Terkini

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

2.8 Pengertian Client / server

Sistem *client-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 *client* 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 mempunyai kemampuan untuk mengubah atau menghapus data itu.

Sistem *client-server* merupakan suatu sistem *client* komputer yang melibatkan proses-proses *client* yang meminta suatu pelayanan data kepada computer *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 *client* menggunakan konfigurasi yang umum bagi sebuah komputer *desktop* yang 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 Operasi (SO) dan paket aplikasi *database*.Adapun spesifikasi Perangkat Lunak tersebut adalah :

2.9.1 Sekilas Tentang Delphi

Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas aplikasi visual.Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrogramannya yang terstruktur.Keunggulan lain dari Delphi adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis *windows*.

Borland Delphi 7.0 merupakan pengembangan dari Borland Delphi 6.0 lebih banyak keunggulan dan fasilitas baru yang diberikan oleh Borland Delphi 7.0 dibandingkan dengan versi-versi sebelumnya.

2.9.2 **Microsoft Access 2000**

Microsoft Access 2000 atau lebih dikenal dengan sebutan Access 2000 merupakan salah satu *software database* yang berjalan di bawah sistem *Windows*, karena di samping Microsoft Access, masih banyak *software-software database* lainnya yang dapat kita temukan.Dengan Microsoft Access 2000, kita dapat merancang, membuat dan mengelola database dengan cara mudah dan cepat.

Access 2000 merupakan penyempurnaan atau pengembangan dari versi sebelumnya.Beberapa hal kelebihan atau penyempurnaan di dalam Access 2000 antara lain :

1) Fasilitas *subdatasheets* baru memungkinkan kita melihat *record* yang berkaitan dengan sebuah datasheet.
2) Dapat berfungsi sebagai *front end* untuk data Microsoft SQL server.
3) Dilengkapi dengan *Tool Web* yang lebih canggih.
4) Memungkinkan kita untuk mempublikasikan data ke *Web,* namun kita tidak dapat melihat hasil halaman di Access.
5) Dapat mengoperasikan secara langsung dari *database* Access untuk membuka dan *browse* halaman *Web* yang menunjukkan data Access.
6) Adanya fasilitas bantuan *Page Wizard* yang baru menciptakan halaman web sederhana dimana anda dapat menyunting data tabel.

Di dalam mengelola *database*, Access 2000 memiliki sarana atau obyek yang dapat mempermudah dalam kerja kita.Obyek-obyek ini satu sama lain saling berhubungan atau saling terkait.Adapun obyek-obyek tersebut adalah :

1. *Tables*, merupakan tempat atau sarana untuk penyimpanan data

2. *Queries*, digunakan untuk menyaring data dengan berbagai criteria dan urutan yang anda inginkan.
3. *Forms*, digunakan untuk memasukkan dan mengedit data atau informasi yang ada dalam suatu database dengan menggunakan bentuk tampilan formulir
4. *Reports*, digunakan untuk menampilkan atau mencetak data atau informasi dalam bentuk laporan.
5. *Pages*, digunakan untuk membuat halaman *web* berupa data access page.
6. *Macros*, digunakan untuk mengotomatisasikan *database* anda dengan perintah atau cara yang lebih sederhana tanpa membutuhkan pemrograman.
7. *Modules,* merupakan program kecil atau prosedur yang kegunaannya adalah untuk perancangan modul aplikasi pengolahan *database* tingkat lanjut.

Ada beberapa istilah dalam Access 2000, yaitu :

1) Tabel, adalah sekumpulan data yang memiliki topik tertentu.Tabel mengatur data di dalam kolom (dikaitkan dengan *field)* dan baris (dikaitkan dengan *record*).Sebagai contoh, sebuah tabel dapat berisi tentang Mahasiswa, Dosen, atau data Pegawai.
2) *Field*, tempat atau dimana data atau informasi dalam kelompok yang sama atau sejenis dimasukkan.*Field* pada umumnya tersimpan dalam bentuk kolom secara vertikal pada tabel, seperti NIM, NAMA, ALAMAT, dan sebagainya.

3) *Record*, merupakan data lengkap dalam jumlah tunggal biasanya tersimpan dalam bentuk baris secara *horizontal* pada tabel.

2.9.3 Rational Rose

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

Rational Rose memiliki keunggulan, diantaranya :

1. Bahasa yang digunakan adalah bahasa pemodelan standar yaitu UML, akan meningkatkan komunikasi intra tim.
2. *Rational Rose* mendukung *round-trip engineering*, sehingga kita dapat men-*generate* 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. Membangun *software* menggunakan *Rational Rose* memudahkan dalam memperbaiki *software* tersebut karena apabila 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 *client/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 dirancang/dianalisis**

Apotek Utama Jaya yang bergerak dalam bidang kesehatan dimana terjadi proses penjualan dan pembelian obat**.Di bawah ini penjelasan tentang penjualan, pembelian, obat.**

2.10.1 Penjualan

Penjualan merupakan salah satu fungsi dari pemasaran atau merupakan bagian dari kegiatan pemasaran.Penjualan sangat pentingdan 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.

Menurut SAL [8] "*Penjualan adalah jumlah seluruh barang-barang yangmerupakan usaha pokok dari perusahaan jika perusahaan itu merupakan*

perusahaan dagang, maka sales perusahaan itu adalah hasil penjualan barang-barang dagangan yang dijual oleh perusahaan tersebut ".

2.10.2 Pembelian

Pembelian merupakan salah 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"*.

2.10.3 Pengertian Obat

Obat adalah senyawa kimia yang dapat dipakai pada atau diberikan pada manusia atau binatang sebagai alat Bantu diagnosis.

Manfaat Obat :

a) Sebagai pengobatan dan pencegahan penyakit atau keadaan abnormal lainnya.
b) Untuk menghilangkan rasa sakit dan penderitaan atau untuk mengendalikan atau memperbaiki setiap keadaan fisiologik atau patologik

Obat terbagi menjadi dua macam :

1) Obat Generik

   Obat yang di produksi oleh perusahaan negeri.Contoh : Parasetamol, Antalgin.

2) Obat Paten

   Obat yang di produksi oleh perusahaan-perusahaan swasta.Contoh : Lanamol, Neuromach, Amoxsan.