BAB II

TINJAUAN PUSTAKA

2.1 Profil PT. TELEKOMUNIKASI INDONESIA

2.1.1 Sejarah PT. TELEKOMUNIKASI INDONESIA

Tahun 1975 merupakan awal perjalanan usaha PT Infomedia Nusantara menjadi perusahaan pertama penyedia layanan informasi telepon di Indonesia. Di bawah subdivisi Elnusa GTDI dari anak perusahaan Pertamina, Infomedia telah menerbitkan Buku Petunjuk Telepon Telkom Yellow Pages.

Perkembangan yang tercatat selanjutnya adalah berdirinya PT Elnusa Yellow Pages di tahun 1984 yang berubah nama di tahun 1995 menjadi PT Infomedia Nusantara pada saat PT Telkom Tbk menanamkan investasi. Untuk mendukung implementasi Good Coorporate Governance dalam setiap aspek kegiatan perusahaan, Infomedia telah mengeluarkan kebijakan pedoman tata kelola perusahaan di tahun 2008.

Pada tanggal 30 Juni 2009 PT Telekomunikasi Indonesia (Telkom) melalui PT Multimedia Nusantara (Metra), anak perusahaan yang 99,99% milik Telkom (selanjutnya disebut Telkom Group) telah menandatangani Shares Sales & Purchase Agreement (SPA) untuk membeli 49% saham PT Infomedia Nusantara (Infomedia) milik PT Elnusa Tbk (Elnusa), sehingga 100% saham PT Infomedia Nusantara telah dimiliki oleh Telkom Group.

Saat ini, Infomedia, sesuai dengan visinya menjadi penyedia jasa layanan informasi yang utama dikawasan regional telah melakukan berbagai upaya untuk

mewujudkan visi tersebut dengan mengoptimalkan kompetensi untuk mengambil opportunity dalam pengembangan bisnis kedepan melalui transformasi bisnis dari 3 Pilar Bisnis (Layanan Direktori, Layanan Contact Center dan Layanan Konten) menuju Layanan Outsourcing atau Business Process Outsourcing (BPO) dan Layanan Konten Digital atau Digital Rich Content (DRC).

Layanan Outsourcing atau Business Process Outsourcing (BPO) didefinisikan sebagai bisnis penyediaan jasa alih-daya (outsourcing) oleh pihak ketiga bagi perusahaan untuk satu atau beberapa fungsi bisnis dalam jangka panjang (multi year contract).Bisnis Layanan Outsourcing (BPO) yang telah dijalani Infomedia saat ini berbasis layanan voice yaitu Layanan Contact Center baik untuk inbound maupun outbound dannon voice seperti direct mail dan web development. Namun saat iniInfomedia telah membagi bisnis Layanan Outsourcing (BPO) kedepannya dalam empat kelompok berdasarkan basis layanan yaitu: Contact Center Services, HR Services, IT Services dan Direct Mail.

Sedangkan pengembangan bisnis Layanan Konten Digital (DRC) didasarkan oleh semakin berkembangnya kebutuhan informasi yang semakin cepat danmobile, perubahan gaya hidup dan perkembangan teknologi yang sangat pesat. Infomedia membagi bisnis DRC dalam 3 bagian, yaitu; printed (Yellow Pages, White Pages & Special Directory) , mobile (mobile application, SMS)dan online (online ad, e-commerce, membership).

Keseluruhan produk dan layanan Infomedia merupakan komitmen perusahaan dalam memberikan solusi layanan informasi dan komunikasi yang prima bagi customer dan masyarakat di Indonesia

2.1.2 Logo PT. Telekomunikasi Indonesia



Gambar 2.1 Logo TELKOM

1. Makna dari lingkaran melambangkan Expertise : kelengkapan produk dan layanan dalam portofolio bisnis baru TELKOM yaitu TIME (Telecommunication, Information, Media & Edutainment).
2. Makna dari tangan yang meraih ke luar (Empowering) : mencerminkan pertumbuhan dan ekspansi ke luar.
3. Makna dari jemari tangan (Assured) : memaknai sebuah kecermatan, perhatian serta kepercayaan dan hubungan yang erat.
4. Makna dari kombinasi tangan dan lingkaran (Progressive) : simbol dari matahari terbit yang maknanya adalah perubahan dan awal yang baru.
5. Simbol dari telapak tangan yang mencerminkan kehidupan untuk menggapai masa depan (Heart).
6. Warna Expert Blue pada text Telkom melambangkan keahlian dan pengalaman yang tinggi.
7. Warna Vital Yellow pada telapak tangan mencerminkan sesuatu yang

aktraktif, hangat dan dinamis.

8. Infinite Sky Blue pada text Indonesia dan lingkaran bawah mencerminkan inovasi dan peluang yang tak berhingga untuk masa depan.

2.1.3 Badan Hukum Instansi.

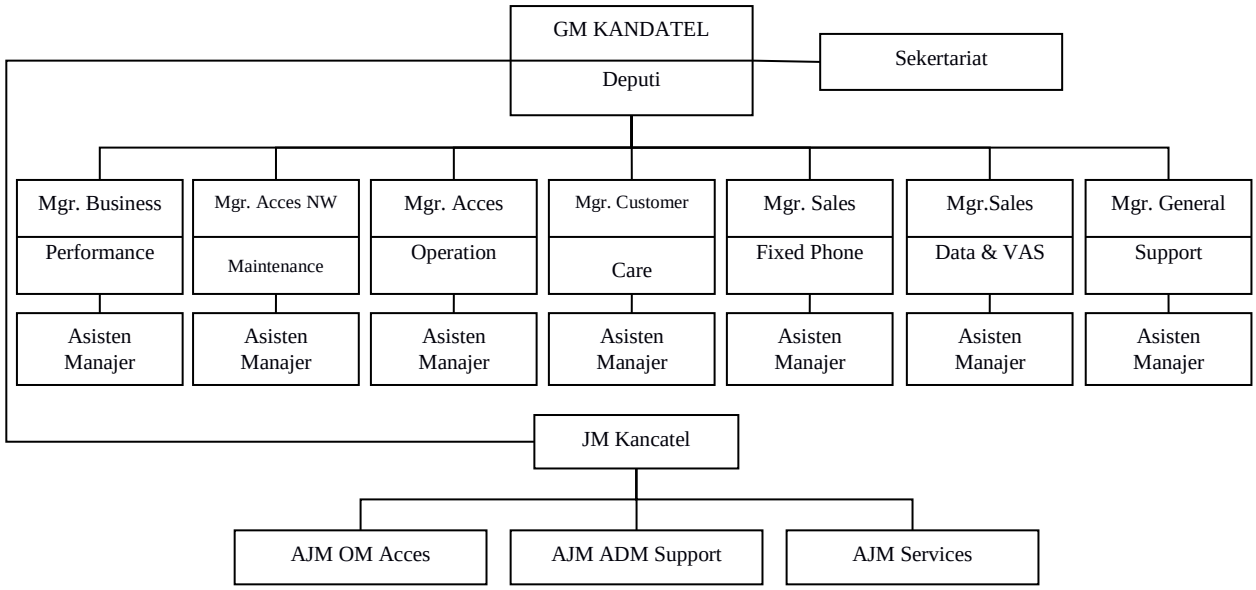
PT. Telekomunikasi Indonesia merupakan perusahaan terkemuka di bidang layanan telekomunikasi di Indonesia. Perusahaan ini terbentuk dan telah berbadan hukum. Berikut penjelasannya :

1. Peraturan pemerintah pengganti Undang-Undang no. 19 tahun 1960, tentang persyaratan sebuah Perusahaan Negara (PN) dengan PERPU no.240 tahun 1961 berubah menjadi PN POS dan TELEKOMUNIKASI. Lapangan usaha PN POS dan TELEKOMUNIKASI ternyata berkembang dengan pesat, maka pada tahun 1965 pemerintah mengadakan peninjuan kembali.
2. Peraturan pemerintah (PP) no.29 dan 30 tahun 1965 terjadi pemecahan menjadi : P.N.POS dan TELEKOMUNIKASI. Selanjutnya mulai tanggal 28 april 1970 berdasarkan S.K Mentri Perhubungan no. 129/U/1970 PN TELEKOMUNIKASI berubah menjadi PERUSAHAAN UMUM TELEKOMUNIKASI yang singkat dengan PERUMTEL.
3. Peraturan pemerintah no.52 tahun 1980 pemerintah mengambil kebijaksanaan dengan membeli seluruh saham PT. INDOSAT sebuah perusahaan swasta yang didirikan dalam rangka penanaman modal asing yang kemudian diubah statusnya menjadi BUMN berbentuk PERSERO.

4. Peraturan pemerintah no.53 tahun 1980, diadakan perubahan atas peraturan pemerintah no.22 tahun 1974, yakni dengan menetapkan PERUMTEL sebagai badan usaha yang diberikan wewenang untuk menyelenggarakan INDOSAT telekomunikasi sebagai badan dalam negeri dan yang diberi wewenang usaha PT. menyelenggaraan telekomunikasi untuk umum internasional.
5. Peraturan pemerintah no.39 tahun 1974 tentang PERUMTEL juga diubah yang dituangkan dalam peraturan pemerintah no.54 tahun 1980.
6. Peraturan pemeritah no.3 tahun 1983 tentang cara pembinaan dan pengawasan PERJAN, PERUM dan PERSERO.
7. Peraturan pemerintah no.21 tahun 1984 tentang PERUMTEL sebagai pengganti dari peraturan pemerintah no.36 tahun 1970 jo. Peraturan pemerintah no.54 tahun 1980. Satu hal yang sangat mengembirakan dalam sejarah perundang-undangan ini adalah ditetapkannya telekomunikasi pengembangan Undang-Undang yang dan no.3 memberikan pembangunan tahun 1989 tentang angin segar dalam pertelekomunikasian di Indonsesia.
8. Peraturan pemerintah no.25 tahun 1991 tanggal 1 Mei 1991 menetapkan pengalihan bentuk perusahaan umum (PERUM) telekomunikasi menjadi perusahaan perseroan (Persero). Peralihan bentuk perusahaan tersebut ditandai dengan penanda tanganan akte pendirian perusahaan perseroan (Persero) PT. Telekomunikasi Indonesia oleh Notaris Imam Fatimah, SH bersama-sama dengan Menparpostel Soesilo Sudarman yang bertindak selaku kuasa dari Mentri Keuangan sebagai pemegang saham, hari selasa tanggal 24 September 1991 jam 09.30 WIB di Depparpostel, Jl. Kebon Sirih 36 Jakarta Pusat.

2.1.4 Struktur Organisasi.

Dalam rangka melaksanakan tugas operasional perusahaan adanya organisasi area cukup di perlukan. Organisasi arean merupakan kerangka dan susunan perwujudan pola hubungan diantara fungsi-fungsi, bagian-bagian dan orang-orang yang menunjukan delagasi tugas , kedudukan dan wewenang serta tanggung jawab yang berbeda-beda dalam suatu perusahaan. Berikut adalah bentuk struktur organisasi kandatel :



Gambar 2.2 Struktur Organisasi Kandatel

2.2 Landasan Teori

2.2.1 Pengertian Sistem

Menurut Jerry Fith Gerald; *sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu*.

2.2.1.1 Elemen Sistem

Elemen Sistem yaitu kumpulan komponen yang saling berkaitan dan bekerja sama untuk mencapai suatu tujuan tertentu. Suatu sistem dapat terdiri dari beberapa sub-sub sistem dan sub-sub tersebut dapat pula terdiri dari beberapa sub-sub sistem yang lebih kecil.

Seperti yang telah dikatakan diatas bahwa suatu sistem mempunyai elemen-elemen penyusun, diantaranya adalah :

1. Komponen-komponen (Components).

   Setiap sistem baik sistem dalam skala besar maupun dalam skala kecil sekalipun memiliki komponen-komponen atau elemen-elemen. Komponen-komponen ini saling berhubungan dan bekerja sama sehingga tercapai suatu kesatuan fungsi dari sistem, sehingga sistem dapat mencapai tujuannya.

2. Penghubung Sistem (System Interface).

   Penghubung sistem merupakan media perantara antara subsistem yang satu dengan subsistem yang lainnya. Melalui penghubung sistem ini, maka subsistem dapat saling memberi dan menerima sehingga terjalin kerjasama dan dapat membentuk datu kesatuan fungsi dari sistem.

3. Lingkungan Luar (Environtment).

   Lingkungan luar dari sistem adalah segala sesuatu yang berada diluar batas sistem. Lingkungan luar ini bisa juga berupa ekosistem dimana sistem tersebut berada, walaupun keberadaanya diluar sistem tapi lingkungan luar dapat mempengaruhi sistem. Adanya ketidakserasian antara lingkungan luar dengan sistem dapat menyebabkan terganggunya fungsi sistem. Oleh karena itu harus

senantiasa tercipta keharmonisan antara sistem dan lingkungan luarnya.

4. Batas Sistem (Boundary).

   Merupakan daerah pemisah antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memberikan ruang lingkup yang jelas dari suatu sistem. Dengan adanya ruang lingkup yang jelas dari suatu sistem tersebut, maka kita dapat memisahkan dan membedakan suatu sistem dengan sistem yang lainnya maupun sistem dari lingkungan luar.

5. Masukan Sistem (System Input).

   Masukan sistem adalah bahan atau energi yang dimasukan ke dalam sistem. Energi ini dimasukan ke dalam sistem untuk diproses oleh sistem sesuai dengan fungsi dari sistem agar dapat menghasilkan keluaran.

6. Keluaran Sistem (System Output).

   Keluaran sistem merupakan hasil dari pengolahan suatu sistem. Keluaran ini tentunya diharapkan dapat berguna sesuai dengan tujuan dari sistem. Selain sebagai hasil akhir, sebagai keluaran bisa juga dijadikan masukan untuk sistem lainnya.

7. Pengolahan Sistem (System Processing).

   Pengolahan sistem adalah mesin atau mekanisme yang digunakan untuk mengubah masukan menjadi keluaran. Pengolahan memiliki peranan yang penting, karena disinilah proses perubahan dan pendayagunaan masukan terjadi sehingga menghasilkan keluaran yang sesuai dengan tujuan sistem.

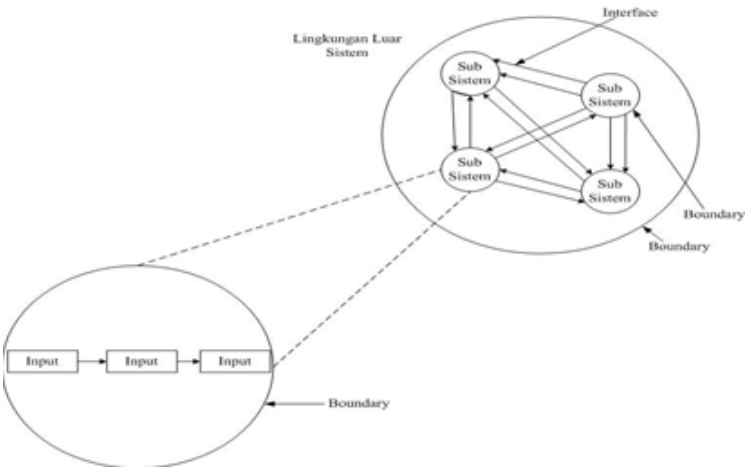
8. Sasaran Tujuan.

   Suatu sistem pasti mempunyai tujuan (goal) atau sasaran (objective). Tujuan

merupakan hal akhir yang ingin dicapai oleh suatu sistem, sedangkan sasaran merupakan hal-hal yang menjadi objek dan titik fokus untuk meraih tujuan. Suatu sistem bisa dikatakan berhasil atau menjalankan fungsinya bila berhasil mencapai sasaran dan tujuan dari sistem tersebut.

2.2.1.2 Karakteristik Sistem

Karakteristik atau sifat-sifat suatu sistem dapat dilihat seperti pada Gambar 2.1 berikut ini :



Gambar 2.3 Karakteristik Sistem

Suatu sistem mempunyai karakteristik/sifat-sifat tertentu yaitu mempunyai :

a. Komponen (Component)

   Terdiri dari sejumlah komponen yang saling berinteraksi dan bekerja sama membentuk satu kesatuan. Komponen-komponen dapat terdiri dari beberapa subsistem/subbagian, dimana setiap subsistem tersebut memiliki fungsi khusus dan akan mempengaruhi proses sistem secara keseluruhan.

b. Batas Sistem (Boundary)

Merupakan daerah yang membatasi antara satu sistem dengan sistem lainnya/dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu batas suatu sistem menunjukan ruang lingkup (scope) dari sistem tersebut.

c. Lingkungan Luar Sistem (Environtments)

Adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar dapat bersifat menguntungkan dan merugikan. Lingkungan yang menguntungkan harus tetap dijaga dan dipelihara, sebaliknya lingkungan yang merugikan harus ditahan dan dikendalikan jika tidak ingin terganggu kelangsungan hidup sistem.

d. Penghubung (Interface)

Merupakan media penghubung antar subsistem, yang memungkinkan sumber-sumber daya mengalir sari satu subsistem ke subsistem lainnya. Keluaran (output) dari satu subsistem akan menjadi masukan (input) untuk subsistem lainnya melalui penghubung disamping sebagai untuk mengintegrasikan subsistem-subsistem menjadi satu kesatuan.

e. Masukan (Input)

Adalah energi yang dimasukkan ke dalam sistem, yang dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input). Masukan perawatan : energi yang dimasukkan supaya sistem dapat beroperasi, sedangkan masukan sinyal adalah energi yang diproses untuk mendapatkan keluaran. Sebagai contoh di dalam sistem komputer, program

adalah maintenance input yang digunakan untuk mengoperasikan komputer dan data adalah signal input untuk diolah menjadi informasi.

f. Keluaran (Output)

Adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain. Misalnya untuk sistem komputer, panas yang dihasilkan merupakan keluaran yang tidak berguna dan hasil sisa pembuangan sedangkan informasi adalah keluaran yang dibutuhkan.

g. Pengolah (Process)

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan mengubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan lain menjadi keluaran berupa barang jadi. Sistem akuntansi akan mengolah data-data transaksi menjadi laporan-laporan keuangan dan laporan lain yang dibutuhkan oleh manajemen.

h. Sasaran (Objectives) atau tujuan (goal)

Suatu sistem pasti mempunyai tujuan/sasaran. Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil jika mengenai sasaran/tujuannya.

2.2.1.3 Klasifikasi Sistem

Klasifikasi sistem dibagi menjadi empat, yaitu :

1. Sistem Abstrak (*abstract system*) dan sistem fisik (phsycal system).

   Sistem abstrak adalah sistem yang berupa pemikiran/ide-ide yang tidak tampak secara fisik. Contoh : Sistem Teologia.

2. Sistem Alamiah (*natural system*) dan sistem buatan manusia (human made system).

   a. Sistem alamiah : sistem yang terjadi melalui proses alam dan tidak dibuat oleh manusia. Contoh : Sistem Perputaran Bumi.

   b. Sistem buatan manusia : sistem yang dirancang oleh manusia dan melibatkan interaksi antara manusia dengan mesin. Contoh Sistem Informasi.

3. Sistem Tertentu (*closed system*) dan sistem terbuka (open system)

   a. Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi, interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti sehingga keluarannya dapat diramalkan. Contoh : Sistem Komputer melalui program.

   b. Sistem tak tentu : sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.

4. Sistem Tertutup (closed system) dan sistem terbuka (open system)

   a. Sistem Tertutup : sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya (kenyataanya tidak ada system yang benar-benar tertutup) yang ada hanyalah *relatively closed system*.

b. Sistem Terbuka : sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar atau subsistem yang lainnya, sehingga harus memiliki sistem pengendalian yang baik.

2.2.2 Pengertian Informasi

Informasi merupakan salah satu sumber daya yang paling utama yang dimiliki oleh suatu organisasi apapun jenis organisasi tersebut.

Informasi: *data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat.* Jadi ada suatu proses transformasi data menjadi suatu informasi.

Dari pengertian-pengertian tersebut dapat disimpulkan bahwa : Informasi adalah hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi penerimanya dimana akan menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat bantu untuk pengambilan keputusan.

2.2.2.1 Kualitas Informasi

Informasi berhubungan dengan keputusan dan keputusan yang diambil akan berhasil sesuai dengan informasi yang berkualitas. Adapun syarat-syarat agar informasi berkualitas yaitu :

1. Akurat.

   Sebuah informasi akan dikatakan akurat jika informasi tersebut tidak bias atau menyesatkan, bebas dari kesalahan dan harus jelas kemana arah tujuannya.

2. Tepat Waktu.

   Informasi sebagai dasar untuk menentukan keputusan haruslah memiliki

ketepatan dalam waktu.

3. Relevan.

   Informasi dikatakan berkualitas jika relevan bagi pemakainya. Informasi akan relevan jika memberikan manfaat bagi pemakainya. Relevansi informasi untuk tiap – tiap orang satu dengan yang lainnya berbeda.

4. Sumber.

   Informasi yang baik adalah informasi yang mempunyai sumber yang jelas.

2.2.3 Pengertian Sistem Informasi

Sistem informasi adalah sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, membantu dan mendukung kegiatan operasi, bersifat manajerial dari suatu organisasi dan membantu mempermudah penyediaan laporan yang diperlukan. (Erwan Arbie).

Sistem informasi adalah data yang dikumpulkan, dikelompokkan dan diolah sedemikian rupa sehingga menjadi sebuah satu kesatuan informasi yang saling terkait dan saling mendukung sehingga menjadi suatu informasi yang berharga bagi yang menerimanya. (Tafri D. Muhyuzir).

Dari kedua pengertian diatas, sistem informasi dapat didefinisikan sebagai suatu kesatuan dari berbagai informasi yang saling berkaitan dan berinteraksi satu sama lainnya untuk keperluan dan tujuan tertentu.

Sistem Informasi juga dapat didefinisikan sebagai :

a. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan

informasi.

b. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi.

c. Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

d. Kumpulan dari sub-sub sistem, baik fisik maupun non-fisik yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mengolah data menjadi informasi.

Adapun kegiatan sistem informasi adalah sebagai berikut :

1. Input, yaitu menggambarkan suatu kegiatan untuk menyediakan data untuk proses.
2. Proses, yaitu menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.
3. Penyimpanan, yaitu suatu kegiatan untuk memelihara dan menyimpan data.
4. Output, yaitu suatu kegiatan untuk menghasilkan laporan dari suatu proses informasi.
5. Kontrol, yaitu suatu aktivitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan.

Sedangkan komponen sistem informasi terdiri dari :

1. Perangkat keras (hardware), terdiri dari komputer, printer, jaringan
2. Perangkat lunak (software).
3. Data, merupakan komponen dasar informasi.
4. Manusia (user).

2.2.4 Metode Analisis dan Perancangan Terstruktur

Didalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analisis sistem sebagai berikut :

1. Identify, yaitu mengidentifikasi masalah.
2. Understand, yaitu memahami kerja dari sistem yang ada.
3. Analyze, yaitu menganalisis sistem.
4. Report, yaitu membuat laporan hasil analisis.

2.2.5 Pemodelan Sistem

Dalam suatu proses pengembangan software, analisa dan rancangan telah merupakan terminologi yang sangat tua. Pada saat masalah ditelusuri dan spesifikasi dinegosiasikan, dapat dikatakan bahwa kita berada pada tahap rancangan. merancang adalah menemukan suatu cara untuk menyelesaikan masalah, salah satu tool/model untuk merancang pengembangan software yang berbasis object-oriented adalah UML. Alasan mengapa UML digunakan adalah, pertama, scalability dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan komplek. Kedua, dynamic modeling, dapat dipakai untuk pemodelan sistem dinamis dan real time. Sebagaimana dalam tulisan pertama, penulis menjelaskan konsep mengenai obyek, OOA&D (Obyek Oriented Analyst/ Design) dan pengenalan UML, maka dalam tulisan kedua ini lebih ditekankan pada cara bagaimana UML digunakan dalam

merancang sebuah pengembangan software yang disertai gambar atau contoh dari sebuah aplikasi.

2.2.6 Basis Data

Basis data (*database*), atau sering pula dieja basisdata, adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS). Sistem basis data dipelajari dalam ilmu informasi.

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi

matematika). Dalam model ini, hubungan antar tabel diwakili denga menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Istilah *basis data* mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai *sistem manajemen basis data* (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak administrator dan programer menggunakan istilah basis data untuk kedua arti tersebut.

Perangkat lunak basis data yang banyak digunakan dalam pemrograman dan merupakan perangkat basis data aras tinggi (*high level*):

Tabel 2.2 perangkat lunak basis data

1.	DB2	13.	FoxPro
2.	Microsoft SQL Server	14.	Visual FoxPro
3.	Oracle	15.	Arago
4.	Sybase	16.	Force
5.	Interbase	17.	Recital
6.	XBase	18.	dbFast
7.	Firebird	19.	dbXL
8.	MySQL	20.	Quicksilver
9.	PostgreSQL	21.	Clipper
10.	Microsoft Access	22.	FlagShip
11.	dBase III	23.	Harbour
12.	Paradox	24.	Visual dBase

Selain perangkat lunak pada tabel 2.1, terdapat juga perangkat lunak pemrograman basis data aras rendah (*low level*), diantaranya:

1. Btrieve
2. Tsunami Record Manager

DBMS (Database Management System) adalah software yang menangani semua akses ke basis data. Secara konsep apa yang terjadi adalah sebagai berikut :

1. user melakukan pengaksesan basis data untuk informasi yang diperlukannya menggunakan suatu bahasa manipulasi data, biasanya disebut SQL.
2. DBMS menerima *request* dari user & menganalisa request tersebut
3. DBMS memeriksa skema eksternal user, pemetaan eksternal/konseptual, skema konseptual, pemetaan konseptual/internal, & struktur penyimpanan.
4. DBMS mengeksekusi operasi-operasi yang diperlukan untuk memenuhi permintaan user.

2.2.7 *Unified Modeling Lenguage* (UML)

Unified modeling language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem peranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Dari berbagai penjelasan rumit yang terdapat di dokumen dan buku-buku UML. Sebenarnya konsep dasar UML bisa kita rangkumkan dalam Tabel berikut :

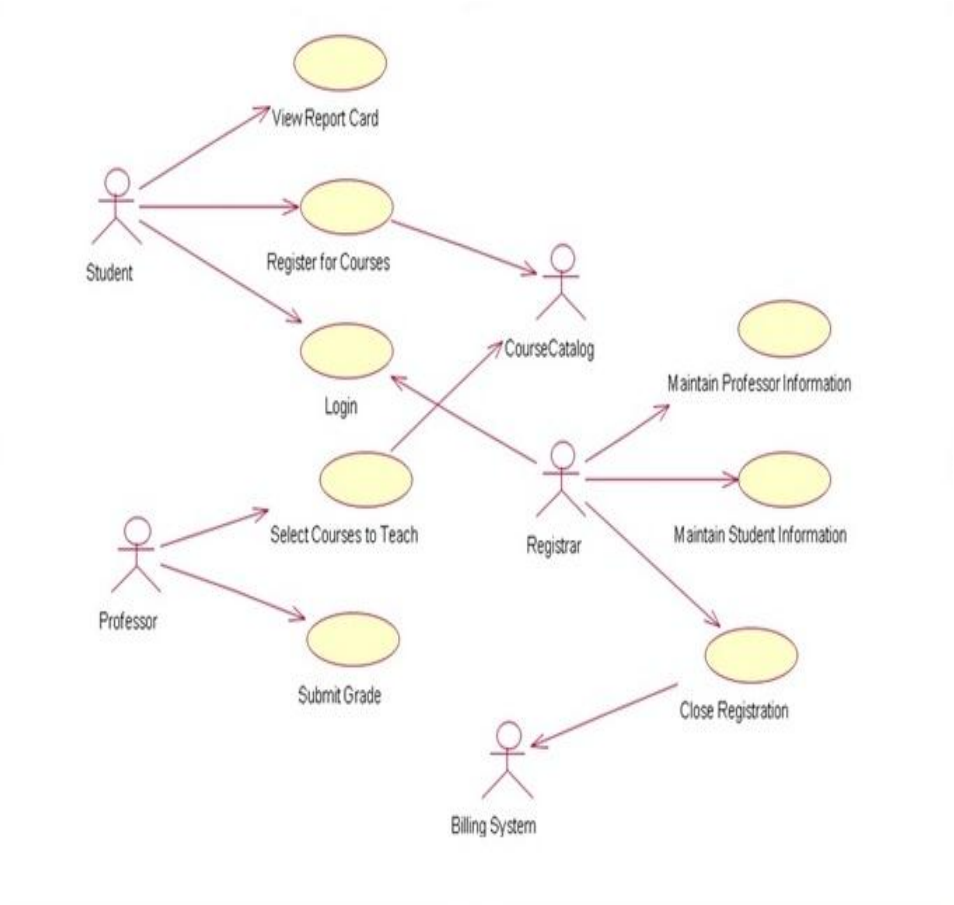
Tabel 2.4 Konsepsi Dasar UML

Major Area	*View*	*Diagrams*	*Main Concepts*
struktural	Static view	Class diagram	Class, association, generalization, dependency,

			realization, interface.
	Use case view	Use case diagram	Use case, actor, association, extend, include, use case generalization
	Implementaion view	Component diagram	Componen, interface, dependency, realization
	Deployment view	Deployment diagram	Node, component, dependency, location
Dynamic	State machine view	Statechart diagram	State, event, transition, action
	Activity view	Activity diagram	State, activity, completion transition, fork, join
	Interaction view	Sequence diagram	Interaction, object, message, activation
		Colaboration diagram	Collaboration, interaction, collaboration role, message
Model management	Model management view	Class diagram	Peckage, sub system, model
Extensibility	All	All	Constraint, stereotype, taged values.

2.2.7.1 Use Case Diagram

Sebuah *use case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Setiap use case menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur internalnya.



Gambar 2.4 Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem.

2.2.7.1.1 Aktor

Sebuah aktor mencirikan suatu bagian *outside user* atau susunan yang berkaitan dengan user yang berinteraksi dengan sistem [Rumbaugh, Booch, da Jacobson 1999]. Dalam model use case, aktor merupakan satu-satunya kesatuan eksternal yang berinteraksi dengan sistem. Sebuah aktor sering kali merupakan manusia (*human*

user). Pada sejumlah sistem informasi, manusia adalah satu-satunya aktor. Dan mungkin saja dalam sistem informasi, seorang aktor bisa saja menjadi suatu sistem eksternal. Pada aplikasi real-time dan distribusi, sebuah aktor bisa saja menjadi satu perangkat eksternal I/O atau sebuah alat pengatur waktu. Perangkat eksternal I/O dan pengatur waktu aktor secara khusus lazimnya berada dalam real-time yang tersimpan dalam sistem (*real-time embedded systems*), sistem berinteraksi dengan lingkungan eksternal melalui sensor dan aktuator.

Primary actor (aktor utama) memprakarsai sebuah use case. Jadi, suatu primary aktor memegang peran sebagai proaktif dan yang memulai aksi dalam sistem. Aktor lainnya yang berperan sebagai *secondary* aktor bisa saja terlibat dalam use case dengan menerima *output* dan memberikan *input*. Setidaknya satu actor harus mendapatkan nilai dari use case. Biasanya adalah *primary* aktor (actor utama). Bagaimanapun, dalam *real-time embedded systems*, *primary* aktor dapat berperan sebagai perangkat eksternal I/O atau pengatur waktu, penerima utama dari *use case* bisa menjadi *secondary human* aktor yang menerima sejumlah informasi dari sistem.



Gambar 2.5 aktifitas actor dan use case

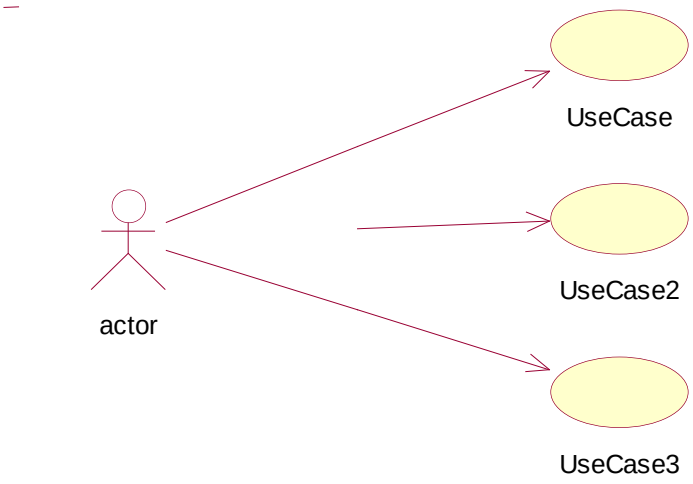
2.2.7.1.2 Identifikasi Use Case

Sebuah use case dimulai dengan masukan/input dari seorang aktor. Use case

merupakan suatu urutan lengkap kejadian-kejadian yang diajukan oleh seorang aktor, dan spesifikasi interaksi antara aktor dengan sistem. Use case yang sederhana hanya melibatkan satu interaksi/hubungan dengan sebuah aktor, dan use case yang lebih kompleks melibatkan beberapa interaksi dengan aktor. Use cases yang lebih kompleks juga melibatkan lebih dari satu aktor.

Untuk menjabarkan use case dalam sistem, sangat baik bila dimulai dengan memperhatikan aktor dan actions/aksi yang mereka lakukan dalam sistem. Setiap use case menggambarkan suatu urutan interaksi antara aktor dengan sistem. Sebuah use case harus memberikan sejumlah nilai pada satu aktor.

Kemudian, kebutuhan fungsional sistem dijelaskan dalam use case yang merupakan suatu spesifikasi eksternal dari sebuah sistem.



Gambar 2.6 interaksi actor dan use case

Urutan utama use case menjelaskan urutan interaksi yang paling umum antara aktor dan sistem. Dan mungkin saja terdapat cabang-cabang urutan use case utama, yang mengarah pada berkurangnya frekuensi interaksi antara actor dengan sistem.

Deviasi-deviasi dari urutan utama hanya dilaksanakan pada beberapa situasi, contohnya jika aktor melakukan kesalahan input pada sistem. Ketergantungan pada aplikasi kebutuhan, alternatif ini memecahkan use case dan kadang-kadang bersatu kembali dengan urutan utama. Cabang-cabang alternative digambarkan juga dalam use case.

2.2.7.1.3 Pendokumentasian Model Use Case

Use case didokumentasi dalam *use case model* sebagai berikut:

1. *Use Case Name.*

   Setiap use case diberi nama.

2. *Summary.*

   Deskripsi singkat use case, biasanya satu atau dua kalimat.

3. *Dependency.*

   Bagian ini menggambarkan apakah use case yang satu tergantung pada use case yang lain, dalam arti apakah use case tersebut termasuk pada use case yang lain atau malah memperluas use case lain.

4. *Actors*

   Bagian ini memberikan nama pada actor dalam use case. Selalu terdapat use case utama (primary use case) yang memulai use case. Disamping itu terdapat juga secondary use case yang terlibat dalam use case. Contohnya, dalam use case Withdraw Funds, ATM Customer adalah actor-nya.

5. *Preconditions*

   Satu atau lebih kondisi harus berjalan dengan baik pada permulaan use case; contohnya mesin ATM yang tidak jalan, menampilkan pesan Selamat Datang.

6. *Deskripsi*

   Bagian terbesar dari use case merupakan deskripsi naratif dari urutan utama use case yang merupakan urutan yang paling umum dari interaksi antara aktor dan sistem. Deskripsi tersebut dalam bentuk input dari aktor, diikuti oleh respon pada sistem. Sistem ditandai dengan sebuah kotak hitam (black box) yang berkaitan dengan apa yang sistem lakukan dalam merespon input aktor, bukan bagaimana internal melakukannya.

7. *Alternatif-alternatif*

   Deskripsi naratif dari alternatif merupakan cabang dari urutan utama. Terdapat beberapa cabang alternatif dari urutan utama. Contohnya, jika rekening customer terdapat dana yang tidak sesuai, akan tampil permohonan maaf dan menolak kartu.

8. *Postcondition*

   Kondisi yang selalu terjadi di akhir use case, jika urutan utama telah dilakukan; contohnya dana customer telah ditarik.

9. *Outstanding questions*

   Pertanyaan-pertanyaan tentang use case didokumentasikan untuk didiskusikan dengan para user.

2.2.8 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan

layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

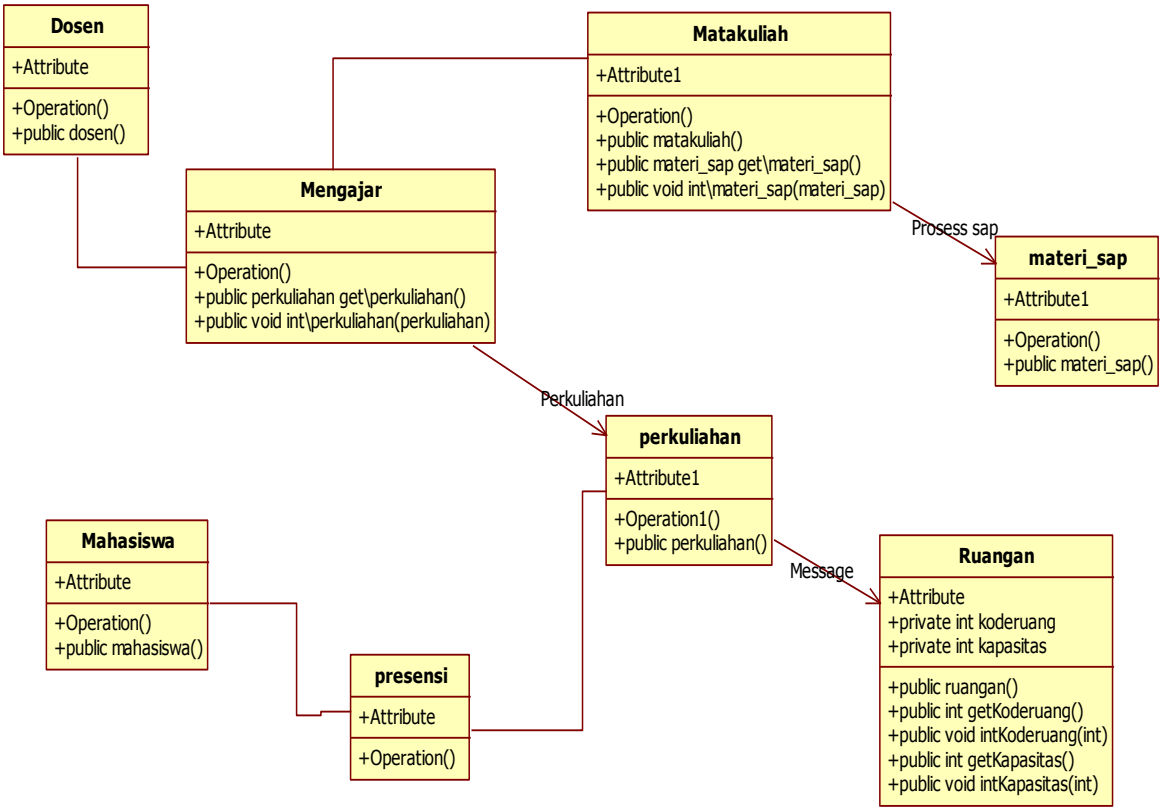
Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment,* pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama(dan stereo type)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut :

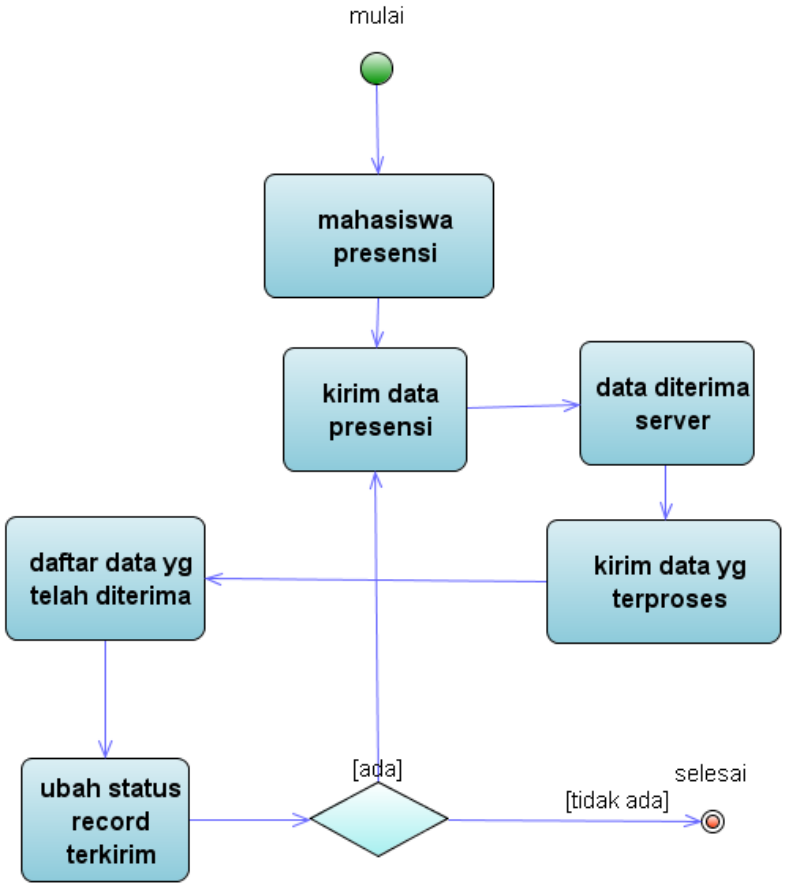
1. Private, tidak dapat dipanggil dari luar class yang bersangkutan
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
3. Public, dapat dipanggil oleh siapa saja



Gambar 2.7 Class Diagram

2.2.9 StateChart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).



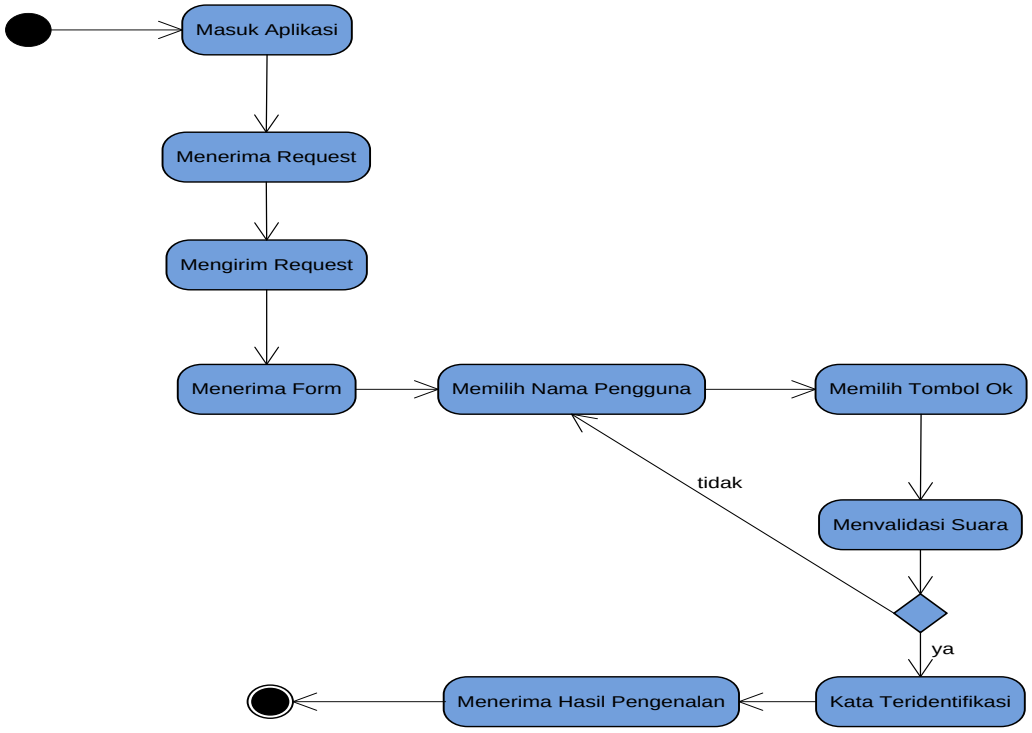
Gambar 2.8 StateChart Diagram

2.2.10 *Activivty Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan

behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.



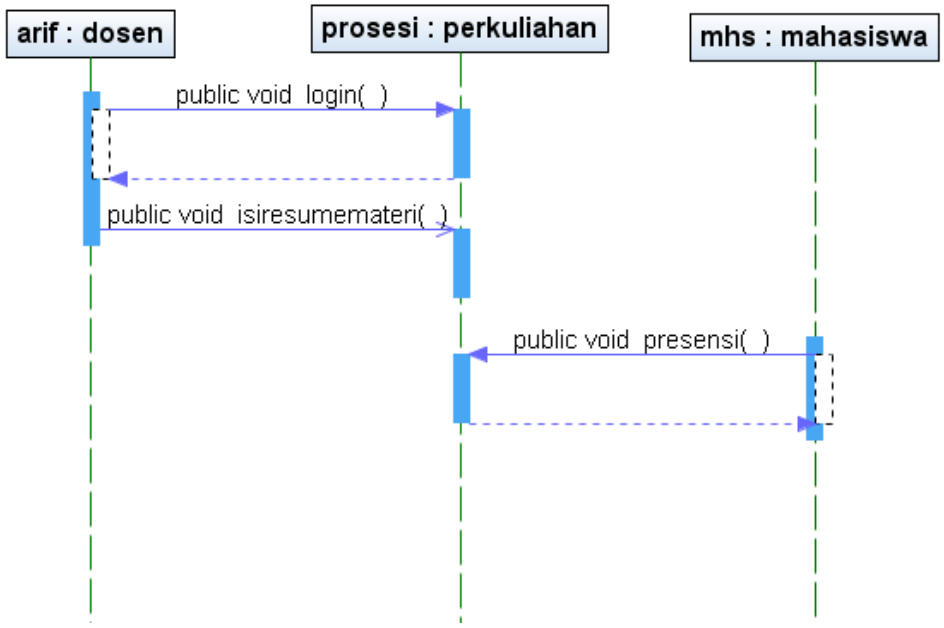
Gambar 2.9 Activity Diagram

2.2.11 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. *Sequence* diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah event untuk

menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.



Gambar 2.10 Sequence Diagram

2.3 Metode Pengembangan Perangkat Lunak

Merupakan metode yang menggambarkan "perkembangan perangkat lunak sebagai sebuah kesatuan dari pemakai ke pengembang dan ke teknologi" disebut dengan "Prescriptive" karena Menentukan sekumpulan elemen proses (Aktivitas, Aksi, tugas, produk kerja, jaminan kualitas, dll untuk setiap proyek), setiap model proses juga menentukan alur kerjanya.

Pada saat kerja bergerak maju menuju sebuah sistem yang lengkap, keadaan yang digambarkan secara rekursif diaplikasikan kepada kebutuhan pemakai dan

spesifikasi teknis P/L pengembang Saat ini, prescriptive memberikan jawaban secara definitive untuk masalah pengembangan P/L dalam setiap perubahan lingkungan komputasi

2.3.1 Model Skuensial Linear

Pada Laporan Kerja Praktek ini menggunakan metode penggunaan perangkat lunak siklus kehidupan klasik atau model air terjun. Model ini mengusulkan sebuah pendekatan kepada perkembangan software yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas – aktivitas sebagai berikut :

1. Rekayasa dan pemodelan sistem/informasi

   Karena sistem merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke software tersebut. Pandangan sistem ini penting ketika software harus berhubungan dengan elemen-elemen yang lain seperti software, manusia, dan database. Rekayasa dan anasisis system menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta disain tingkat puncak. Rekayasa informasi mancakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisis kebutuhan Software

   Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khusunya pada software. Untuk memahami sifat program yang dibangun, analis harus memahami

domain informasi, tingkah laku, unjuk kerja, dan interface yang diperlukan. Kebutuhan baik untuk sistem maupun software didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

Desain software sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur software, representasi interface, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi software yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi software.

4. Generasi Kode

Desain harus diterjemahkan kedalam bentuk mesin yang bias dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

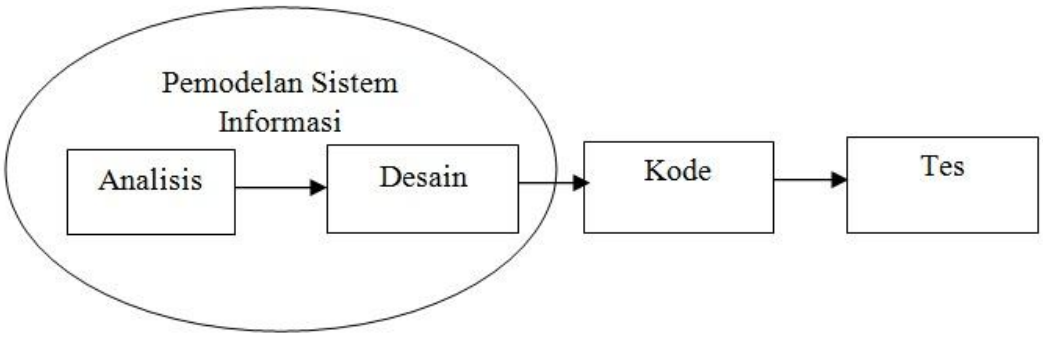
5. Pengujian

Sekali program dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal software, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan – kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Pemeliharaan

Software akan mengalami perubahan setelah disampaikan kepada pelanggan

(perkecualian yang mungkin adalah software yang dilekatkan). Perubahan akan terjadi karena kesalahan – kesalahan ditentukan, karena software harus disesuaikan untuk mengakomodasi perubahan – perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan software mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.



Gambar 2.11 Model Skuensial Linear (model metodologi Waterfall)

Masalah yang kadang terjadi ketika model sekuensial linier diaplikasikan adalah :

1. Jarang sekali proyek nyata mengikuti aliran sekuensial yang dianjurkan oleh model. Meskipun model linier bisa mengakomodasi iterasi, model ini melakukannya dengan cara tidak langsung. Sebagai hasilnya, perubahan – perubahan dapat menyebabkan keraguan pada saat tim proyek berjalan.

2. Kadang – kadang sulit bagi pelanggan untuk menyatakan semua kebutuhannya secara eksplisit. Model linier sekuensial memerlukan hal ini dan mengalami kesulitan untuk mengakomodasi ketidakpastian natural yang ada pada bagian awal beberapa proyek.
3. Pelanggan harus bersifat sabar. Sebuah versi kerja dari program – program kerja itu tidak akan diperoleh sampai akhir waktu proyek dilalui. Sebuah kesalahan besar, jika tidak terdeteksi sampai program yang bekerja tersebut dikaji ulang, bisa menjadi petaka.

2.3.2 Java

2.3.2.1 Sejarah Java

Sejarah java berawal pada tahun 1991 ketika perusahaan *Sun Micro System* melalui *Green Project,* yakni projek penelitian untuk membuat bahasa yang akan digunakan pada chip – chip untuk *device intelegent consumer electronic* .Bahasa tersebut bersifat *multi-platform*, yang tidak tergantung pada vendor yang memanufaktur chip tersebut.

Dalam penelitiannya, projek *Green* berhasil membuat prototipe semacam PDA (*Personal Data Asisten*) yang dapat berkomunikasi antara satu dengan yang lain.

Karena pada awalnya di tujukan untuk pemograman *device* kecil, java memiliki karakteristik berukuran kecil, efisien dan portabel untuk berbagai *hardware*.

2.3.2.2 Keunggulan Java

Java memiliki beberapa keunggulan bila dibandingkan dengan bahasa pemograman lainnya. Berikut beberapa aspek yang dibahas :

1. Java bersifat sederhana dan relatif mudah

   Java dimodelkan sebagian dari bahasa C++, namun dengan memperbaiki beberapa karakteristik C++, seperti mengurangi kompleksitas beberapa fitur, penambahan fungsionalitas, serta penghilangan beberapa aspek pemicu ketidakstabilan sistem pada C++.

2. Java berorientasipada objek (Object Oriented)

   Java membagi program menjadi objek – objek, kemudian memodelkan sifat dan tingkah laku masing – masing.

3. Java bersifat distribusi

   Sistem komputerisasi java cenderung terdistribusi, mulai dari *workstation client*, *e-mail server*, *database server* dan sebagainya.

4. Java bersifat *Multiplatform*

   Java bersifat multiplatform, yakni dapat diterjemahkan olehjava interpreter pada berbagai sistem operasi.

5. Java bersifat *MultiThread*

   Thread adalah proses yang dapat dikerjakan oleh program dalam suatu waktu. Java bersifat *MultiThread*, artinya dapat mengerjakan beberapa proses dam waktu yang hampir bersamaan.

2.3.3 NetBeans

NetBeans adalah salah satu aplikasi IDE yang digunakan programmer untuk menulis, mengompile, mencari kesalahan, dan menyebarkan program.netbeans ditulis dalam bahasa java namun dapat juga mendukung bahasa pemrogramman lain.

Platform NetBeans adalah framework yang dapat digunakan kembali

(reusable) untuk menyederhanakan pengembangan aplikasi desktop. Ketika aplikasi berbasis platform NetBeans dijalankan, kelas Main dari platform dieksekusi. Modul-modul yang tersedia ditempatkan di sebuah registry di dalam memori, dan tugas startup modul dijalankan. Secara umum, kode modul dimuatkan ke dalam memori hanya ketika ia diperlukan.Aplikasi dapat menginstal modul secara dinamis. Aplikasi dapat memasukkan modul Update Center untuk mengijinkan pengguna aplikasi men-download digitally-signed upgrade dan fitur-fitur baru secara langsung ke dalam aplikasi yang berjalan. Penginstalan kembali sebuah upgrade atau rilis baru tidak memaksa pengguna untuk men-download keseluruhan aplikasi lagi.

Platform NetBeans menawarkan layanan-layanan yang umum bagi aplikasi desktop, mengijinkan pengembang untuk fokus ke logika yang spesifik terhadap aplikasi. Fitur-fitur yang disediakan oleh platform NetBeans:

1. Manajemen antarmuka (misal: menu & toolbar)
2. Manajemen pengaturan pengguna
3. Manajemen penyimpanan (menyimpan dan membuka berbagai macam data)
4. Manajemen jendela
5. Wizard framework (mendukung dialog langkah demi langkah)

2.3.4 MySQL

Mysql adalah sebuah perangkat lunak pembuat database yang bersifat terbuka atau *open source* dan berjalan disemua *platform* baik Linux maupun Windows, MySQL merupakan program pengakses database yang bersifat *network* sehingga dapat digunakan untuk aplikasi *Multi User* (Pengguna Banyak).

MySQL adalah *Relational Database Management System* (RDBMS) yang

didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial.

MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase*.