

BAB II

LANDASAN TEORI

2.1. Pengertian Sistem

Menurut Jogiyanto yang dikutip oleh Benny Indra Murphy, mengemukakan bahwa ‘Sistem bisa ditafsirkan sebagai kesatuan elemen yang memiliki keterkaitan. Beberapa elemen dapat digabung menjadi suatu unit, kelompok atau komponen sistem dengan fungsi tertentu. Komponen sistem ini bisa dilihat, dianggap, atau memang dirancang untuk berfungsi mandiri sebagai modul sistem (lepas dari sistem tetapi masih berkaitan dengan sistem pada mana modul ini menginduk)’. [1]

Sedangkan menurut Mulyadi yang dikutip oleh David Ardian Cahyono, mengemukakan bahwa ‘Suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lainnya, yang berfungsi bersama – sama mencapai tujuan tertentu’. [2]

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal data-item. Data adalah kenyataan yang menggambarkan suatu kejadian - kejadian dan kesatuan yang nyata.

2.1.1. Karakteristik Sistem

Menurut Abdul Hafid “Suatu sistem mempunyai karakteristik atau sifat – sifat tertentu yaitu:

1. Komponen – komponen

Komponen sistem atau elemen sistem dapat berupa:

- a. Elemen – elemen yang lebih kecil yang disebut sub-sistem, misalkan sistem komputer terdiri dari sub sistem perangkat keras, perangkat lunak dan manusia.
- b. Elemen – elemen yang lebih besar yang disebut *supra sistem*. Misalkan bila perangkat keras adalah sistem yang memiliki sub sistem CPU,

perangkat I/O dan memori, maka *supra sistem* perangkat keras adalah sistem komputer.

2. Batas Sistem

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan.

3. Lingkungan Luar Sistem

Lingkungan dari sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut.

4. Penghubung

Penghubung merupakan media perantara antar subsistem. Melalui penghubung ini memungkinkan sumber – sumber daya mengalir dari suatu subsistem ke subsistem lainnya”.[3]

2.1.2. Klasifikasi Sistem

Menurut Pujianto “Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya sebagai berikut ini:

1. Sistem diklasifikasikan sebagai sistem abstrak dan sistem fisik. Sistem abstrak adalah sistem yang berupa pemikiran atau ide – ide yang tidak tampak secara fisik. Misalnya sistem *teologia*, yaitu sistem yang berupa pemikiran hubungan antara manusia dan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik. Misalnya sistem komputer, sistem akuntansi, sistem produksi dan lain sebagainya.
2. Sistem diklasifikasikan sebagai sistem alamiah dan sistem buatan manusia. Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia melibatkan interaksi antara manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebut dengan *man-machine system*”.[4]

2.2. Definisi Informasi

Menurut Kusriani yang dikutip oleh Rizan Machmud, mengemukakan bahwa ‘Informasi adalah data yang sudah diolah menjadi sebuah bentuk yang berguna bagi pengguna yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi’.[5]

Deni Darmawan menjelaskan “5 ciri dari informasi yang bias memberikan makna bagi pengguna, diantaranya:

1. Kuantitas Informasi, dalam arti bahwa informasi yang diolah oleh suatu prosedur pengolahan informasi mampu memenuhi kebutuhan banyaknya informasi.
2. Kualitas Informasi, dalam arti bahwa informasi yang diolah oleh sistem pengolahan tertentu mampu memenuhi kebutuhan kualitas informasi.
3. Informasi Aktual, dalam arti bahwa informasi yang diolah oleh sistem pengolahan tertentu mampu memenuhi kebutuhan informasi baru.
4. Informasi yang Relevan, dalam arti bahwa informasi yang oleh sistem pengolahan tertentu mampu memenuhi kebutuhan informasi.
5. Ketepatan Informasi, dalam arti bahwa informasi yang oleh sistem pengolahan tertentu mampu memenuhi kebutuhan informasi”.[6]

Ciri-ciri dari informasi ini idealnya dimiliki oleh informasi yang dibutuhkan ketika kita akan merumuskan atau membuat kebijakan tertentu, sehingga tindakan atau aktivitas yang diambil sesuai dengan kebutuhan dan tujuan pemakaian informasi yang dimaksud.

2.2.1. Ciri-ciri Informasi

Berikut adalah penjelasan mengenai ciri – ciri informasi:

1. Benar Atau Salah
Informasi berhubungan dengan kebenaran terhadap kenyataan.
2. Baru
Informasi benar – benar baru bagi si penerima.

3. Tambahan

Informasi dapat memberpahari / memberikan perubahan terhadap informasi yang telah ada.

4. Korektif

Informasi dapat digunakan untuk melakukan koreksi terhadap informasi sebelumnya yang salah / kurang benar.

5. Penegas

Informasi dapat mempertegas informasi yang telah ada sehingga keyakinan terhadap informasi semakin meningkat.[7]

2.2.2. Kualitas Informasi

Menurut Hartono yang dikutip oleh S.Aditya, mengemukakan bahwa 'Kualitas dari suatu informasi tergantung dari 3 faktor, yaitu keakuratan, ketepatan waktu, dan kesesuaian.'[8]

1. Keakuratan

Informasi harus bebas dari kesalahan – kesalahan dan tidak bisa atau menyesatkan

2. Ketepatan Waktu

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai karena informasi merupakan landasan pengambilan keputusan.

3. Kesesuaian

Informasi tersebut mempunyai manfaat untuk pemakainya.

2.3. Sistem Informasi

Menurut Bin Ladjamudin yang dikutip oleh Reni Yulian, mengemukakan bahwa 'Sistem Informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi bersifat manajerial, kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan'.[9]

Sedangkan pengertian umum sistem informasi adalah: “Sebuah sistem yang terdiri atas rangkaian subsistem informasi terhadap pengolahan data untuk menghasilkan informasi yang berguna dalam pengambilan keputusan”.

Jadi dapat disimpulkan bahwa Sistem Informasi adalah kumpulan sub – sub yang saling berhubungan untuk melaporkan informasi sedemikian rupa untuk mencapai tujuan dalam sebuah organisasi.

2.4. Klasifikasi Sistem Informasi

Lely P mengklasifikasikan “Sistem informasi menjadi 7 bagian yaitu:

1. *Transaction Processing System* (TPS)

TPS adalah sistem informasi yang terkomputerisasi yang dikembangkan untuk memproses data dalam jumlah besar untuk transaksi bisnis rutin seperti daftar gaji dan inventaris.

2. *Office Automation Systems* (OAS) dan *Knowledge Work Systems* (KWS)

OAS dan KWS bekerja pada level *knowledge*. OAS mendukung pekerja data, yang biasanya tidak menciptakan pengetahuan baru melainkan hanya menganalisis informasi sedemikian rupa untuk mentransformasikan data atau memanipulasinya dengan cara – cara tertentu. KWS mendukung para pekerja profesional seperti ilmuwan, insinyur dan doktor dengan membantu menciptakan pengetahuan baru dan memungkinkan mendistribusikannya ke organisasi atau masyarakat.

3. Sistem Informasi Manajemen (SIM)

SIM mendukung spectrum tugas – tugas organisasional yang lebih luas, termasuk analisis keputusan dan pembuat keputusan.

4. *Decision Support Systems* (DSS)

DSS berfungsi untuk mendukung pembuat keputusan diseluruh tahap – tahapnya, meskipun keputusan aktual tetap wewenang eksklusif pembuat keputusan.

5. Sistem Ahli (ES) dan Kecerdasan Buatan (AI)

AI berfungsi untuk mengembangkan mesin – mesin yang berfungsi secara cerdas. Dua cara untuk melakukan riset AI adalah memahami bahasa

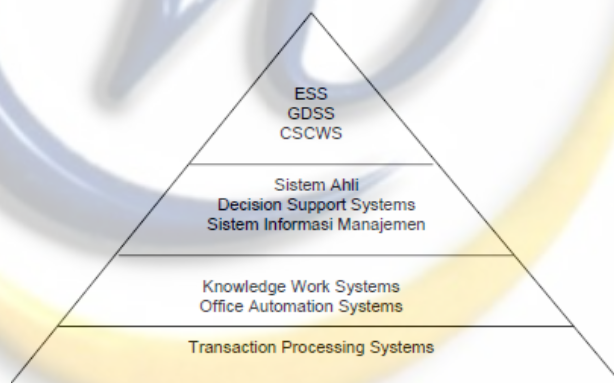
alamiahnya dan menganalisa kemampuannya untuk berfikir melalui masalah sampai kesimpulan logikanya.

6. *Group Decision Support Systems* (GDSS) dan *Computer-Support Collaborative Work Systems* (CSCW)

GDSS dimaksudkan untuk membawa kelompok bersama – sama menyelesaikan masalah dengan memberi bantuan dalam bentuk pendapat, kuesioner, konsultasi dan skenario. Sedangkan CSCW mendukung perangkat lunak yang disebut “*groupware*” untuk kolaborasi tim melalui komputer yang terhubung jaringan.

7. *Executive Support Systems* (ESS)

ESS tergantung pada informasi yang dihasilkan TPS dan SIM, ESS membantu eksekutif mengatur interaksinya dengan lingkungan eksternal dengan menyediakan grafik – grafik dan pendukung komunikasi di tempat – tempat yang bisa di akses seperti kantor”. [10]



Gambar II-1 klasifikasi sistem informasi

2.5. Pengertian Aplikasi

Aplikasi adalah suatu sub kelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Pengertian Aplikasi menurut Barry Pratama [11]:

1. Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas.
2. Aplikasi adalah sistem lengkap yang mengerjakan tugas spesifik.
3. Aplikasi basis data terdiri atas sekumpulan menu, formulir, laporan dan program yang memenuhi kebutuhan suatu fungsional unit bisnis / organisasi / instansi.

Menurut Ibis, aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya. Beberapa aplikasi yang digabung bersama menjadi suatu paket disebut sebagai suatu paket atau *application suite*. Aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. *Software application* adalah software program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu. *Software application* terdiri dari bahasa pemrograman (*programming language*), program aplikasi (*application program*), program paket atau paket aplikasi (*package program*), program utilitas (*utility program*), *games*, *entertainment*, dan lain-lain. Untuk mendukung operasi *software application* di atas, tugas pengguna komputer dibagi menjadi beberapa bagian yaitu sebagai Analis Sistem, Programmer, Operator, Administrator Database, Administrator Jaringan.

Aplikasi software yang dirancang untuk penggunaan praktisi khusus. Klasifikasi aplikasi menurut Barry Pratama dapat dibagi menjadi 2 (dua) yaitu:

1. Aplikasi software spesialis, program dengan dokumentasi tergabung yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi paket, dengan dokumentasi tergabung yang dirancang untuk jenis masalah tertentu.

2.6. Koperasi

Koperasi secara harfiah berasal dari Bahasa Inggris yaitu *Cooperation*. Kata *Cooperation* terdiri dari 2 suku kata, yang pertama adalah *co* (ko) yang berarti bersama. Sedangkan yang kedua adalah *operation* (operasi) yang berarti bekerja. Jadi *Cooperation* diartikan bekerja bersama. Menurut Johar (2009) ”koperasi merupakan usaha bersama yang berlandaskan asas kekeluargaan untuk meningkatkan kesejahteraan anggotanya, koperasi merupakan badan usaha yang mengorganisir pemanfaatan dan pendayagunaan sumber ekonomi anggota dengan dasar prinsip koperasi”. [12]

2.6.1. Simpanan

Pengertian simpanan sebagaimana dinyatakan dalam Peraturan Pemerintah tersebut adalah simpanan yang merupakan hutang bagi KSP/USP, sementara itu terdapat jenis simpanan lain dari anggota yang merupakan kekayaan bersih bagi KSP/USP, yaitu simpanan pokok dan simpanan wajib (bagi KSP). Pembahasan mengenai simpanan meliputi simpanan yang merupakan kekayaan bersih, yaitu simpanan pokok dan simpanan wajib serta simpanan yang merupakan hutang, yaitu tabungan dan simpanan

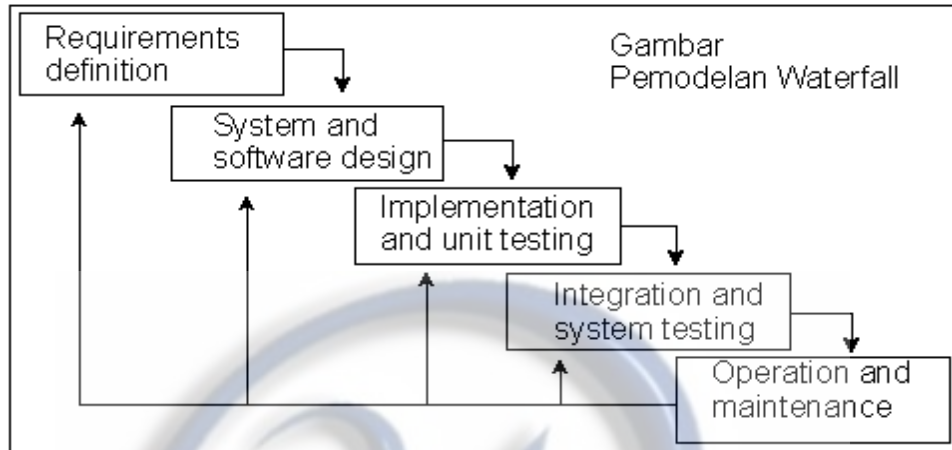
2.6.2. Pinjaman

Pinjaman adalah suatu kegiatan penyaluran dana (modal) koperasi kepada anggota koperasi yang bersangkutan maupun kepada koperasi lain dan atau anggotanya. Tujuan dari kegiatan pinjaman ini adalah untuk meningkatkan usaha dan kesejahteraan anggota. Untuk itu koperasi memberikan pinjaman kepada anggotanya dengan memperhatikan kelayakan dan sifat usaha itu sendiri, apakah termasuk usaha yang produktif atau tidak. (Tantik Suryani, Sri Lestari dan Wiwik Lestari)[13]

2.7. Metode *Waterfall*

Alfiasca, dkk menjelaskan “Metode *Waterfall* melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap

analisis, desain, *coding*, *testing*, dan *maintenance*. Disebut dengan *Waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya yaitu *requirement*'.



Gambar II-2 metode waterfall

Menurut Pressman yang dikutip oleh Alfiasca, dkk menjelaskan mengenai “Tahap – tahap yang ada dalam metode *Waterfall* seperti berikut:

1. ***Requirement Definition***

Proses pencarian kebutuhan diintensifkan data dan difokuskan pada *software*. Untuk mengetahui sifat dari program yang akan dibuat, maka para *software engineer* harus mengerti tentang domain informasi dari *software*.

2. ***System and Software Design***

Proses ini digunakan untuk mengubah kebutuhan – kebutuhan diatas menjadi representasi ke dalam bentuk *blueprintsoftware* sebelum *coding* dimulai.

3. ***Implementation and Unit Testing***

Desain harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap desain yang secara teknis akan dikerjakan oleh *programmer*.

4. ***Integration and System Testing***

Semua fungsi – fungsi *software* harus diuji cobakan, agar *software* bebas dari *error*, dan hasilnya harus benar – benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

5. *Operation and Maintenance*

Pemeliharaan suatu *software* diperlukan, termasuk di dalamnya adalah pengembangan, karena *software* yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada *error* kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur – fitur yang belum ada”. [16]

2.8. Konsep Berorientasi Objek

Menurut Sri Ernayati “Untuk menguasai pemrograman Java, harus mengerti dengan baik konsep pemrograman objek, ada 6 konsep yaitu:

1. Objek

Pada dasarnya semua benda yang ada di dunia nyata dapat dianggap sebagai sebuah objek. Jika perhatikan lebih lanjut, ada 2 karakteristik utama pada sebuah objek:

- a. Setiap objek memiliki atribut sebagai status yang kemudian akan disebut sebagai *state*.
- b. Setiap objek memiliki tingkah laku yang kemudian akan disebut sebagai *behavior*.

2. *Class*

Class berbeda dengan objek, *Class* merupakan prototipe yang mendefinisikan variabel - variabel dan metode – metode secara umum. Sedangkan objek pada sisi lain merupakan instansiasi dari suatu kelas.

3. Enkapsulasi

Dalam sebuah objek yang mengandung variabel – variabel dan metode – metode, dapat ditentukan hak akses pada sebuah variabel atau metode dari objek dalam bagian yang terlindungi inilah yang disebut enkapsulasi. Jadi enkapsulasi adalah bungkusan pelindung program dan data yang sedang di olah.

4. Pewarisan

Pewarisan merupakan menurunkan atribut dan metode pada sebuah *class* yang diperoleh dari *class* yang telah terdefinisi tersebut. Setiap *subclass* akan mewarisi *state* dan *behavior* dari *superclass*-nya.

5. Polimorfisme

Kata polimorfisme yang berarti satu objek dengan banyak bentuk yang berbeda, adalah konsep sederhana dalam bahasa pemrograman berorientasi objek yang berarti kemampuan dari suatu variabel referensi objek untuk memiliki aksi berbeda bila metode yang sama dipanggil, dimana aksi metode tergantung dari tipe objeknya.

6. Interface

Merupakan device yang digunakan untuk komunikasi antar objek yang berbeda yang tidak memiliki hubungan apapun. *Interface* bisa dikatakan sebagai protokol komunikasi antar objek tersebut”. [17]

2.9. Unified Modelling Language (UML)

Eneng Surya menyebutkan bahwa “*UML* merupakan sitem arsitektur yang bekerja dalam OOAD (*Object-Oriented Analysis Design*) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasikan *artifact* (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses rekayasa software. *UML* merupakan bahasa pemodelan yang paling sukses dari tiga metode *OO* yang telah ada sebelumnya, yaitu *Booch*, *OMT*, dan *OOSE*”. [18]




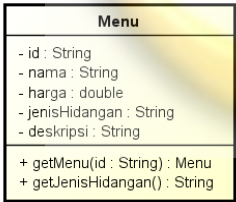


Menurut Mhd. Rozahi Istanbul “*UML* bukanlah suatu proses melainkan bahasa pemodelan secara grafis untuk menspesifikasikan, memvisualisasikan, membangun, dan medokumentasikan sebuah rancangan sistem perangkat lunak”. [19]

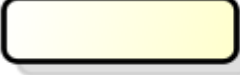


Mhd. Rozahi Istanbul menyebutkan bahwa “Pemodelan dengan menggunakan *UML* dapat di lakukan dengan cara sebagai berikut:

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan

2. Penelahaan bagaimana objek – objek dalam sistem saling berinteraksi, mengirimkan pesan dan saling bekerjasama satu sama lain
3. Menguji apakah sistem perangkat lunak sudah berfungsi seperti seharusnya.
4. Dokumentasi sistem perangkat lunak untuk keperluan – keperluan tertentu dimasa yang akan datang”. [19]

Table II-1 Daftar Simbol UML

Gambar Simbol	Nama Simbol	Keterangan
	<i>Usecase</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Package</i>	Simbol yang memberikan batasan dan komentar yang dikaitkan pada suatu elemen atau kumpulan elemen
	<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	<i>Control</i>	Mengkordinasikan aktifitas dalam sistem
	<i>Entity</i>	Kelas yang berhubungan data dan informasi yang dibutuhkan oleh sistem

Gambar Simbol	Nama Simbol	Keterangan
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	<i>Boundary</i>	Kelas yang memodelkan interaksi antar satu atau lebih aktor dengan sistem

UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya, yaitu:

2.9.1. Use Case Diagram

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”.

Table II-2 Simbol Use Case Diagram


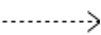
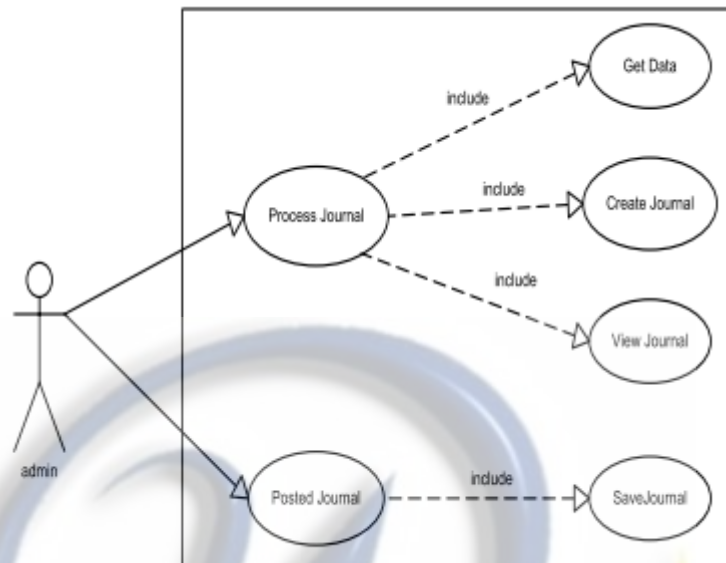
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

Table II-3 Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
10		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).

Mhd. Rozahi Istanbul menjelaskan bahwa “*Use Case Diagram* adalah kumpulan urutan interaksi antara *user* dengan sistem untuk mencapai suatu tujuan

dimana *Use Case* menggambarkan kebutuhan fungsional suatu sistem tanpa menampilkan struktur internal sistem”.

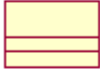



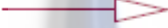





Gambar II-3 Contoh Use Case Diagram

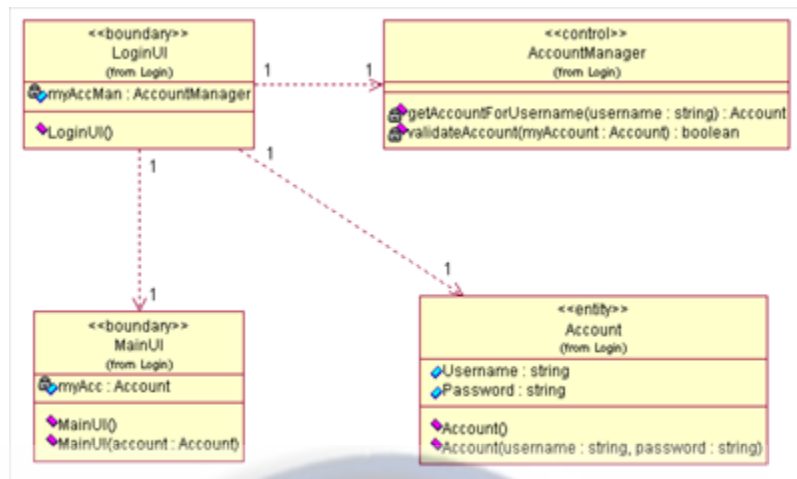
2.9.2. *Class Diagram*

Class Diagram menggambarkan elemen dari spesifikasi dimulai dengan kelas, obyek, dan hubungan mereka, dan beralih ke dokumen arsitektur loGIS dalam suatu sistem.

Table II-4 Simbol Class Diagram

Gambar	Nama	Fungsi
	Class	Menambahkan kelas baru pada diagram
	Interface	Menambahkan kelas antarmuka (<i>interface</i>) pada diagram
	Association	Menggambar relasi asosiasi
	Association class	Menghubungkan kelas asosiasi (<i>association class</i>) pada suatu relasi asosiasi
	Generalization	Menggambarkan relasi generalisasi
	Realize	Menggambarkan relasi realisasi
	Aggregation	Menggambarkan relasi agregasi
	Actor	Menggambarkan aktor pada diagram kelas

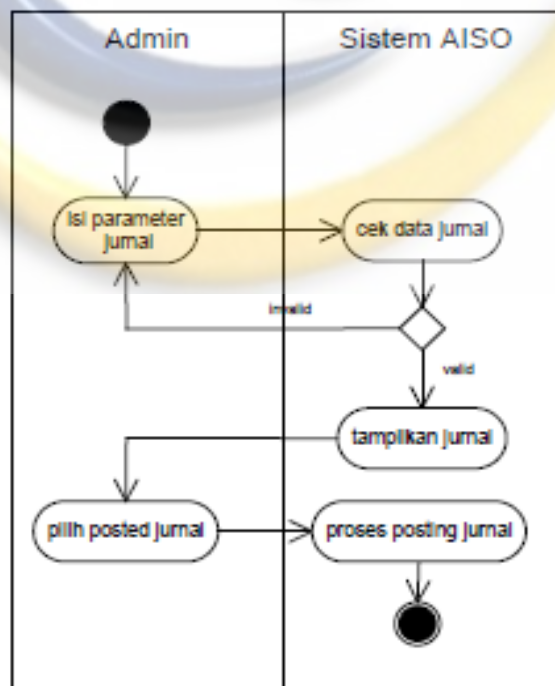
Mhd. Rozahi Istanbul menjelaskan bahwa “*Class Diagram* adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket – paket yang ada dalam sistem perangkat lunak untuk dikembangkan dan memberikan gambaran statis mengenai sistem perangkat lunak dan relasi – relasi di dalamnya”. [19]



Gambar II-4 Contoh Class Diagram

2.9.3. Activity Diagram

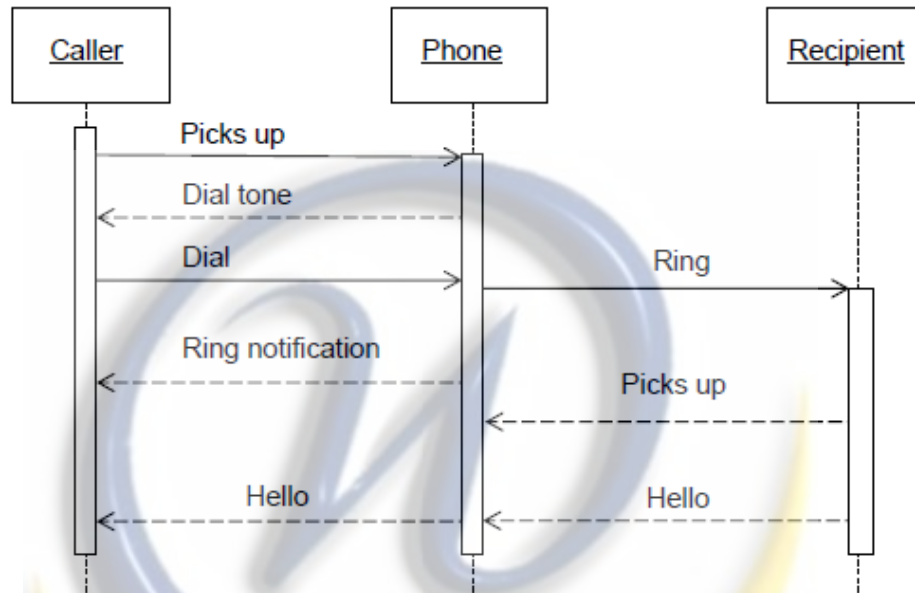
Mhd. Rozahi Istambul menjelaskan bahwa “*Activity Diagram* adalah representasi secara grafis dari proses dan *controlflow* dan berfungsi untuk memperlihatkan alur dari satu aktivitas ke aktivitas yang lain serta menggambarkan perilaku yang kompleks”. [19]



Gambar II-5 Contoh Activity Diagram

2.9.4. Sequence Diagram

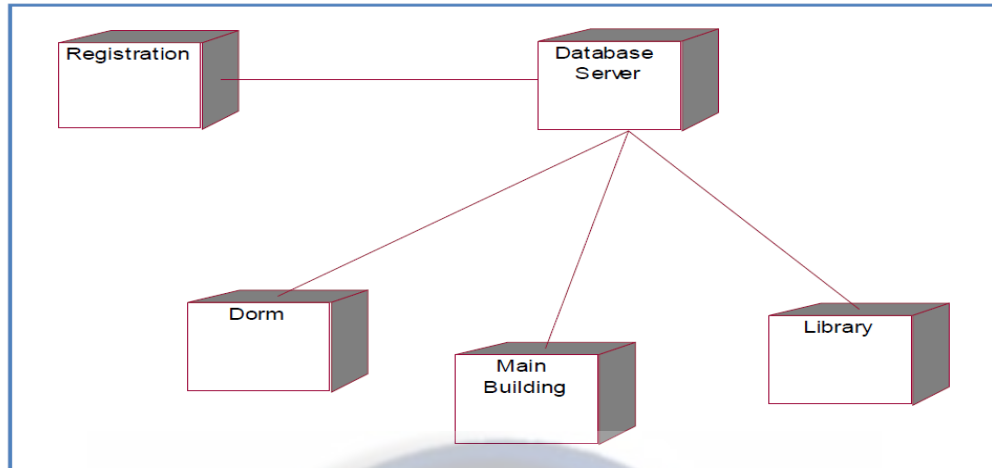
Mhd. Rozahi Istambul menjelaskan bahwa “*Sequence Diagram* adalah diagram yang digunakan untuk menggambarkan *event* yang dilakukan *actor eksternal* pada sistem atau *inter system* dilihat dalam satu *use case*”. [19]



Gambar II-6 Contoh Sequence Diagram

2.9.5. Deployment Diagram

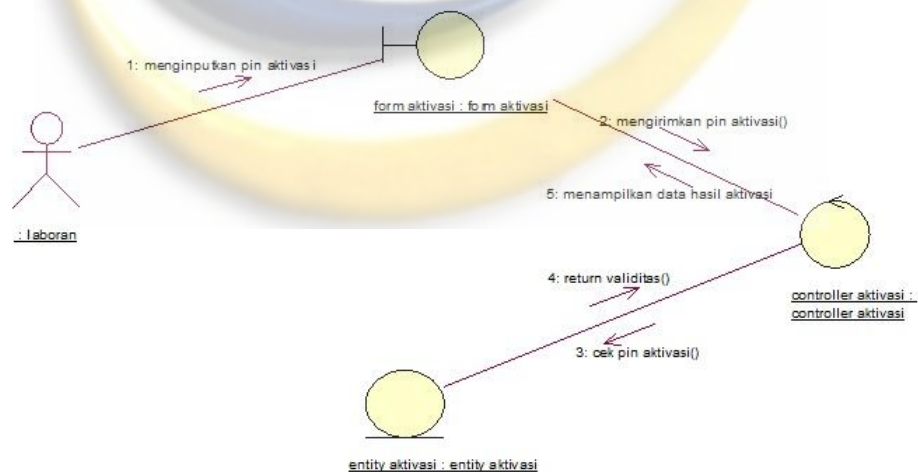
Mhd. Rozahi Istambul mendefinisikan “*Deployment Diagram* sebagai gambaran tugas – tugas kongkrit dari setiap *node* atau *software* yang terlibat dalam jaringan sistem, menampilkan keseluruhan *node* dalam jaringan serta hubungan dari *node* – *node* tersebut termasuk proses – proses yang terlibat di dalamnya”. [19]



Gambar II-7 Contoh Deployment Diagram

2.9.6. Collaboration Diagram

Mhd. Rozahi Istambul menjelaskan bahwa “*Collaboration Diagram* punya peran penting ketika satu tugas di lakukan. Peran – peran dimainkan oleh instan yang berinteraksi, dan menyatakan hubungan komunikasi antara objek – objek yang menunjuknya”. [19]



Gambar II-8 Contoh Collaboration Diagram

2.9.7. State Chart Diagram

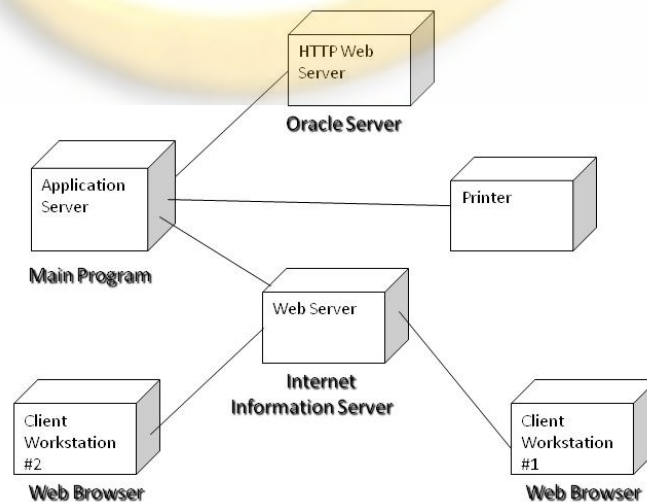
Mhd. Rozahi Istambul mendefinisikan “*State Diagram* sebagai perluasan dari *State Diagram* yang mendeskripsikan objek berupa *state – state* yang dimiliki, kejadian – kejadian yang berlangsung beserta transisi yang terjadi”. [19]



Gambar II-9 Contoh State Chart Diagram

2.9.8. Component Diagram

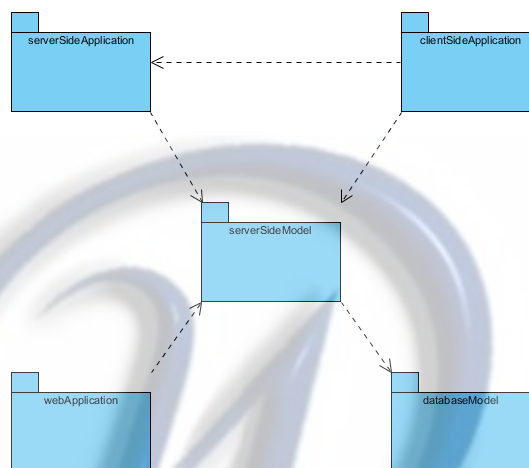
Mhd. Rozahi Istambul menjelaskan bahwa “*Component Diagram* menunjukkan organisasi dan kebergantungan di antara sekumpulan komponen. Diagram ini memodelkan pandangan implementasi fisik dari sistem”. [19]



Gambar II-10 Contoh Component Diagram

2.9.9. Package Diagram

Mhd. Rozahi Istambul menjelaskan bahwa “*Package Diagram* bertugas untuk menggambarkan dan mengelompokkan kelas – kelas yang terdapat pada perangkat lunak yang dibangun sesuai dengan fungsi atau subsistem aplikasi yang mempunyai ketergantungan satu sama lainnya”. [19]



Gambar II-11 Contoh Package Diagram

2.10. Tools

Tools yang digunakan dalam pembuatan aplikasi ini adalah sebagai berikut:

2.10.1. Visual Basic

Menurut Sunyoto *Microsoft Visual Basic* merupakan bahasa pemrograman yang bekerja dalam ruang lingkup *MS Windows*. Secara umum kemampuan Visual Basic adalah menyediakan komponen-komponen yang memungkinkan anda untuk membuat program aplikasi yang sesuai dengan tampilan dan cara kerja *Windows*. Peralatan yang disediakan oleh Visual Basic memberikan kemudahan bagi pemrograman untuk membuat program secara visual (*visual programming*), *visual programming* adalah metode dimana sebagian atau keseluruhan program dibuat dengan cara menggambarkan tampilan atau hasil akhir dan kemudian

meminta beberapa perangkat lunak membuat kode-kode program berdasarkan hasil akhir tersebut.[20]

2.10.2. Crystal Reports

Crystal Report merupakan Program yang dapat digunakan untuk membuat, menganalisis, dan menerjemahkan informasi yang terkandung dalam database atau program ke dalam berbagai jenis laporan yang sangat fleksibel. (Madkom dan penerbit Andi) [21]

2.10.3. Microsoft Access

Menurut Staven *Microsoft Access* adalah suatu piranti lunak dari *Microsoft Corporation* yang membantu proses pembuatan database. Sedangkan menurut Supardi *microsoft Access* salah satu pengolah database termudah dan handal, produk microsoft walaupun dalam penerapan program berorientasi objek mengalami kesulitan tetapi microsoft acces tercepat dan termudah dalam membuat program aplikasi bisnis.[22]

Selain tabel, sebuah file *database Access* juga berisi bermacam-macam obyek database yang lain diantaranya:

1. Queri untuk mengorganisasi data.
2. Forms untuk berinteraksi dengan data pada layar.
3. Reports untuk mencetak hasil.
4. Macros dan program Visual Basic untuk memperluas fungsionalitas dari aplikasi database.