



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Sistem Informasi

Teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen Sistem informasi adalah suatu sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan dan juga untuk menjalankan operasional perusahaan, di mana sistem tersebut merupakan kombinasi dari orang-orang, teknologi informasi dan prosedur-prosedur yang terorganisasi.

Menurut Jogiyanto (2010:697), menjelaskan bahwa sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdas. (Jogiyanto, 2010:697).

Menurut Sutabri (2012:38), juga berpendapat bahwa sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan strategi dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu.

Dari definisi diatas maka penulis menyimpulkan sistem informasi merupakan sekumpulan prosedur di dalam organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan.

2.1.2. Pengertian Sistem Informasi Geografis

Sistem informasi geografis adalah suatu sistem informasi yang digunakan untuk memasukkan, menyimpan, memanggil kembali, mengolah, menganalisis,



dan menghasilkan data yang mempunyai referensi geografis atau lazim disebut data geospasial, yang berfungsi sebagai pendukung pengambilan keputusan dalam perencanaan dan pengelolaan penggunaan lahan, sumber daya alam, lingkungan, transportasi, fasilitas kota, dan pelayanan umum lainnya.

Menurut Marjuki (2014:1), menjelaskan bahwa sistem informasi geografis adalah suatu komponen yang terdiri dari perangkat keras, perangkat lunak, data geografis dan sumberdaya manusia yang bekerja bersama secara efektif untuk memasukan, menyimpan, memperbaiki, memperbaharui, mengelola, memanipulasi, mengintegrasikan, menganalisa dan menampilkan data dalam suatu informasi berbasis geografis.

Menurut Falah (2015:1), juga berpendapat bahwa sistem informasi geografis adalah sistem informasi yang berfungsi untuk mengelola data yang berupa informasi keruangan (spasial).

Sehingga, sistem informasi geografis merupakan sistem berbasis komputer yang digunakan untuk menyimpan, memanipulasi, dan menganalisis informasi geografi.

2.1.3. Pengertian Aplikasi

Aplikasi merupakan suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna serta suatu perangkat lunak yang siap pakai dengan menjalankan intruksi-intruksi dari user atau pengguna, aplikasi banyak diciptakan guna membantu berbagai keperluan seperti untuk laporan, percetakan dan lain-lain.

Menurut Asropudin (2013:6), menjelaskan bahwa aplikasi adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Ms-Word, Ms-Excel.

Menurut Budiman (2014:2), juga berpendapat bahwa aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju. Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu.



Dari beberapa kesimpulan diatas aplikasi merupakan program siap pakai yang dibuat untuk mengerjakan tugas-tugas tertentu yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

2.1.4. Pengertian Web

Website sering juga disebut *web*, dapat diartikan suatu kumpulan-kumpulan halaman yang menampilkan berbagai macam informasi teks, data, gambar diam ataupun bergerak, data animasi, suara, video maupun gabungan dari semuanya, baik itu yang bersifat statis maupun yang dinamis, yang dimana membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau hyperlink.

Menurut Rini Sovia dan Jimmy Febio (2011:87), menjelaskan bahwa Website adalah keseluruhan halaman – halaman web yang terdapat dalam sebuah domain yang mengandung informasi.

Sibero (2014:09), juga berpendapat bahwa *web* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia dan lainnya pada jaringan internet.

Jadi, website merupakan sejumlah halaman web yang memiliki topik saling terkait, terkadang disertai pula dengan berkas gambar, video, atau jenis berkas lainnya.

2.1.5. Pengertian Penentuan

Penentuan sering disebut proses untuk menentukan pencarian lokasi yang dicari dan sebagai penetapan pencarian lokasi. Penentuan suatu lokasi juga harus tepat sasaran karena lokasi yang tepat akan memberikan berbagai keuntungan.

Menurut Kamus Besar Bahasa Indonesia, Penentuan adalah proses, cara, perbuatan menentukan; penetapan; pembatasan (arti dan sebagainya).

Jadi, penentuan merupakan suatu untuk menentukan pencarian lokasi yang dicari karena lokasi yang tepat akan memberikan berbagai keuntungan.



2.1.6. Pengertian Rute

Rute merupakan jalan yang akan dilalui seseorang atau kelompok dengan arah – arah tertentu untuk menuju suatu tempat yang telah direncanakan orang lain dan kumpulan ruas jalan yang menghubungkan satu tempat dengan tempat lain secara menerus.

Menurut Kamus Besar Bahasa Indonesia, rute adalah jarak atau arah yang harus diturut (ditempuh, dilalui).

Warpani (2002:53), juga berpendapat bahwa rute adalah lintasan menunjuk pada ruas jalan yang dilalui kendaraan umum yang melayani trayek bersangkutan; lintasan.

Sehingga rute merupakan jarak atau arah yang harus di turuti dan lintasan menunjuk pada ruas jalan yang dilalui kendaraan umum yang melayani trayek bersangkutan yang telah direncanakan orang lain.

2.1.7. Pengertian Lokasi

Lokasi merupakan suatu letak atau tempat dimana fenomena geografi terjadi. Tetapi kita tak akan dapat mengabstrasikan lebih jauh bagaimana karakteristik ruang tersebut sebelum dideskripsikan tentang lokasinya. Lokasi ini akan memberikan penjelasan lebih jauh tentang tempat atau daerah yang bersangkutan

Menurut Kamus Besar Bahasa Indonesia, lokasi adalah tempat.

Fure (2013:274), juga berpendapat bahwa lokasi adalah faktor yang sangat penting dalam bauran eceran, pemilihan lokasi yang tepat dan strategis pada sebuah gerai atau toko akan lebih sukses dibandingkan gerai lainnya yang berlokasi kurang strategis.

Dari beberapa definisi diatas dapat disimpulkan lokasi merupakan letak atau tempat dimana fenomena geografi terjadi dan memberikan penjelasan lebih jauh tentang tempat atau daerah yang bersangkutan.



2.1.8. Pengertian *Tenant* (Penyewa)

Tenant (*penyewa*) merupakan orang yang mengambil atau telah mengambil barang-barang dari pemilik di bawah perjanjian sewa beli, untuk pemakaian sementara suatu benda, baik bergerak maupun tidak bergerak, dengan pembayaran suatu harga tertentu yang menyewa barang atau benda dari pihak yang menyewakan.

Menurut Kamus Besar Bahasa Indonesia, penyewa adalah orang yang menyewa.

Jadi, Tenant (*penyewa*) merupakan orang yang menyewa barang atau benda dari pihak yang menyewakan.

2.1.9. Pengertian Inkubator Bisnis Politeknik Negeri Sriwijaya

Inkubator Bisnis Politeknik Negeri Sriwijaya adalah sebuah pusat dilingkungan polsri sebagai perwujudan keinginan tim perintis dalam upaya menggalakkan kegiatan kewirausahaan (*entrepreneurship*) dilingkungan polsri.

2.1.10. Pengertian Politeknik Negeri Sriwijaya

Politeknik Negeri Sriwijaya, adalah perguruan tinggi negeri yang terdapat di kota Palembang, Sumatera Selatan, Indonesia. Politeknik Negeri Sriwijaya, dahulunya bernama Politeknik Negeri Sriwijaya secara resmi dibuka pada tanggal 20 September 1982.

2.1.11. Pengertian Aplikasi Web Penentuan Rute Lokasi Tenant Inkubator Bisnis Politeknik Negeri Sriwijaya.

Pengertian Aplikasi Web Penentuan Rute Lokasi Tenant Inkubator Bisnis Politeknik Negeri Sriwijaya adalah aplikasi yang dibangun untuk memudahkan pembeli dalam mencari lokasi tenant Inkubator Bisnis Politeknik Negeri Sriwijaya terdekat dari lokasi pembeli tersebut.



2.2. Teori Khusus

2.2.1. Pemograman Berorientasi Objek

2.2.1.1. Pengertian Pemograman Berorientasi Objek

Nugroho (2005:6), menjelaskan bahwa paradigma berorientasi objek adalah cara yang berbeda dalam memandang aplikasi-aplikasi. Dengan pendekatan berorientasi objek, para pengembang membagi aplikasi-aplikasi besar menjadi objek-objek, yang mandiri satu terhadap yang lainnya.

Sukanto dan Shalahuddin (2013:100), menjelaskan bahwa berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Dari beberapa definisi diatas dapat disimpulkan bahwa pemograman berbasis objek adalah kumpulan objek yang dapat mengorganisasikan perangkat lunak.

2.2.1.2. Ciri Pemograman Berorientasi Objek

Siallagan (2009:149), menjelaskan bahwa ciri-ciri atau karakteristik pemograman berorientasi objek, antara lain:

a. Abstraksi (*Abstraction*)

Abstraksi adalah pengabstrakan atau melakukan seleksi terhadap aspek-aspek tertentu suatu masalah. Abstraksi digunakan untuk menyembunyikan kerumitan dari suatu proses. Sebagai contoh, dalam membuat suatu sistem, ada tombol-tombol yang dapat digunakan. Operator atau pengguna tidak perlu berpikir tentang pembuatan tombol tersebut, tetapi yang penting mereka dapat menggunakannya.

b. Pembungkusan (*Encapsulation*)

Pembungkusan sering pula disebut pengkapsulan. Artinya, proses membuat paket (memaketkan) data objek bersama dengan metode-metodenya. Berdasarkan kode program, proses memisahkan aspek-aspek objek dilakukan dengan pembungkusan. Proses pembungkusan itu sendiri merupakan cara atau mekanisme untuk melakukan abstraksi.



c. Pewarisan (*Inheritance*)

Pewarisan adalah memberikan atau mewariskan sesuatu kepada keturunan berikutnya. Misalnya, seorang anak pasti akan mewarisi beberapa sifat atau perilaku yang dimiliki oleh ibu/bapaknya. Dalam konteks ini, suatu kelas dalam program dapat diturunkan menjadi kelas-kelas baru lainnya yang akan mewarisi beberapa sifat atau perilaku dari kelas induknya.

d. Polimorfisme (*Polymorphism*)

Polimorfisme adalah suatu kejadian ketika objek dapat mengungkap banyak hal melalui satu cara yang sama.

2.2.2. Metodologi RUP (*Rational Unified Process*)

Rosa dan Shalahudin (2011:30) *Rational Unified Process (RUP)* merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri khas metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak. Gambar dibawah ini menunjukkan secara keseluruhan arsitektur yang dimiliki RUP.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language (UML)*.

Melalui gambar dibawah dapat dilihat bahwa RUP memiliki, yaitu:

a. Dimensi Pertama

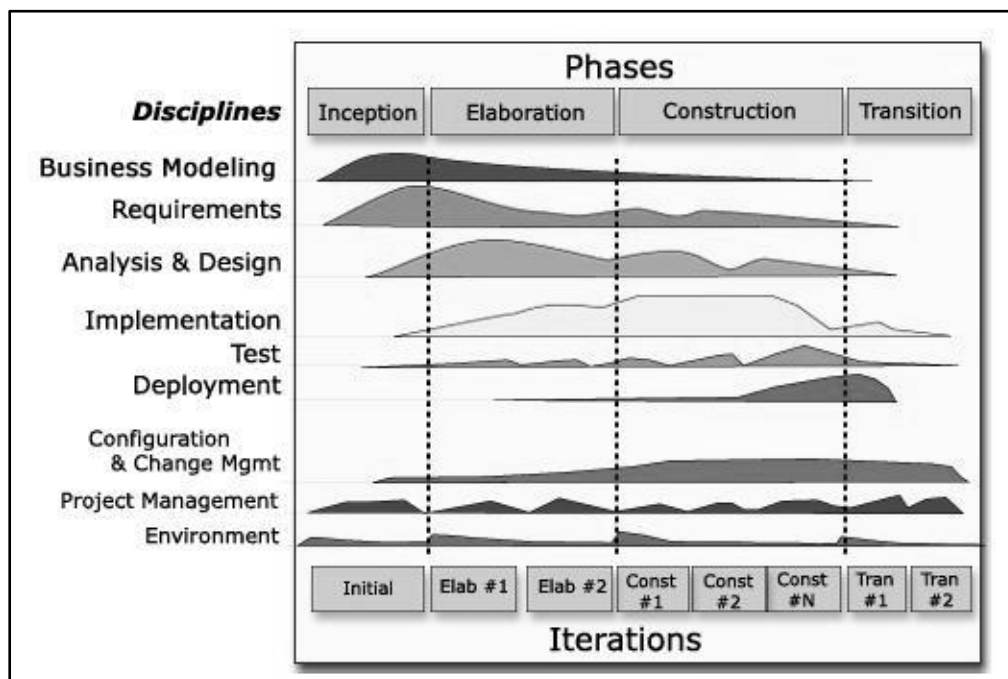
Digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari *phase* selanjutnya. Setiap *phase* dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.



b. Dimensi Kedua

Digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing, what, how, dan when*. Dimensi ini terdiri atas:

Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration, dan Change Management, Project Management, Environment.



Gambar 2.1 Arsitektur Rational Unified Process

2.2.2.1. Penerapan Tahap Metodologi Pengembangan Perangkat Lunak dengan RUP

Rosa dan Shalahudin (2011:32) dalam *Rational Unified Process* terdapat empat tahap pengembangan perangkat lunak yaitu:

1. *Inception*

Pada tahap ini pengembangan mendefinisikan batasan kegiatan, melakukan analisis kebutuhan *user*, dan melakukan perancangan awal perangkat lunak



(perancangan arsitektural dan *use case*). Pada akhir fase ini prototipe perangkat lunak versi *Alpha* harus sudah dirilis.

2. *Elaboration*

Pada tahap ini dilakukan perancangan perangkat lunak mulai dari menspesifikasikan fitur perangkat lunak hingga perilsan prototipe versi *Betha* dari perangkat lunak.

3. *Construction*

Pengimplementasian rancangan perangkat lunak yang telah dibuat dilakukan pada tahap ini. Pada akhir tahap ini, perangkat lunak versi akhir yang sudah disetujui administrator dirilis beserta dokumentasi perangkat lunak.

4. *Transition*

Instanlasi, deployment dan sosialisasi perangkat lunak dilakukan pada tahap ini.

2.2.2.2. Aliran Kerja Utama *RUP*

Rosa dan Shalahudin (2011:32) adapun aliran kerja utama pada Metodologi *RUP* adalah sebagai berikut:

1. Pemodelan Bisnis (*Bussines Modeling*)

Mendeskripsikan struktur dan proses-proses bisnis organisasi.

2. Kebutuhan (*Requirement*)

Mendefinisikan kebutuhan perangkat lunak dengan menggunakan metode *use case*.

3. Analisis dan Perancangan (*Analysis and Design*)

Mendeskripsikan berbagai arsitektur perangkat lunak dari berbagai sudut pandang.

4. Implementasi (*Implementation*)

Menuliskan kode-kode program, menguji, dan mengintegrasikan unit-unit programnya.

5. Pengujian (*Test*)

Mendeskripsikan kasus uji, prosedur, dan alat ukur pengujian.



6. Deployment

Menangani konfigurasi sistem yang akan diserahkan.

2.2.2.3. Aliran Kerja Pendukung RUP

Adapun aliran kerja pendukung RUP adalah sebagai berikut:

1. Manajemen konfigurasi dan perubahan (*configuration and change management*)
mengendalikan perubahan dan memelihara artifak-artifak proyek.
2. Manajemen proyek (*Project Management*)
Mendeskripsikan berbagai strategi pekerjaan dengan proses yang berulang.
3. Lingkungan (*Environment*)
Menangani infrastruktur yang dibutuhkan untuk mengembangkan sistem.

2.2.3. Unified Modelling Language (UML)

Prabowo dan Herlawati (2011:6), menjelaskan bahwa UML singkatan dari *Unified Modeling Language* yang berarti bahasa permodelan standar.

Munawar (2005:19), menjelaskan bahwa UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek.

UML menyediakan bahasa pemodelan visual yaitu proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Dengan menggunakan pemodelan UML ini, pengembang dapat melakukan:

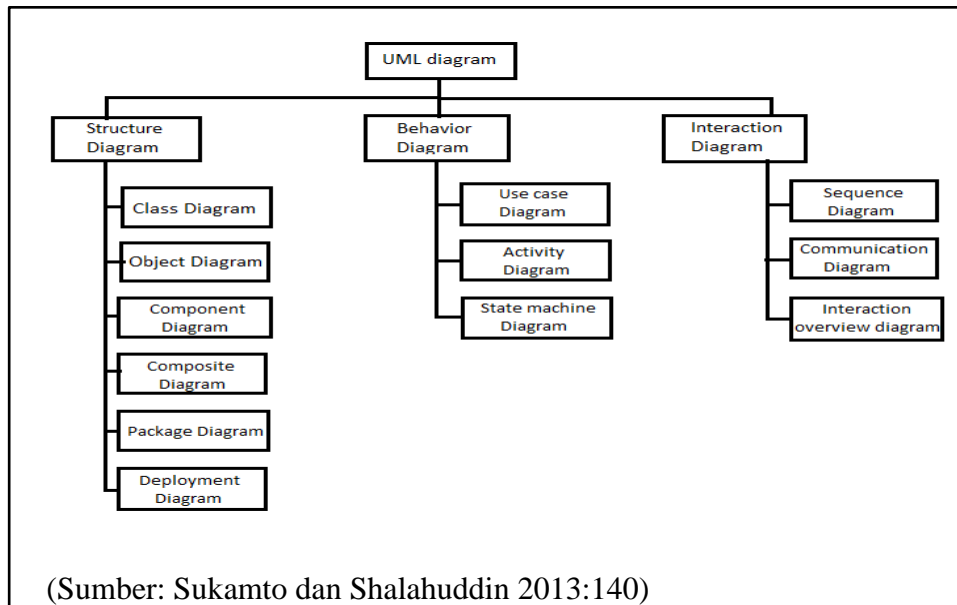
- a. tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
- b. Penelaahan bagaimana objek-objek dalam sistem saling mengirimkan pesan (*message*) dan saling bekerjasama satu sama lain.
- c. Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya.

Dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang.



2.2.3.1. Macam-macam Diagram UML

Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



Gambar 2.2 *Macam-macam Diagram UML*

Berikut ini penjelasan singkat dari pembagian kategori tersebut.

a. Structure Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

b. Behavior Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

c. Interaction Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.




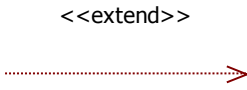


2.2.4. Use Case Diagram



Munawar (2005:63), menjelaskan bahwa *Use Case Diagram* adalah deskripsi fungsi dari sebuah *system* dari perspektif pengguna.

Sukanto dan Shalahuddin (2013:155), menjelaskan bahwa *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat.

Tabel 2.1 Simbol-simbol dalam *Use case*

No	Simbol	Deskripsi
1.	Use Case 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2.	Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	Asosiasi / <i>association</i> 	Komunitas antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
4.	Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tamhanan itu; biasanya <i>use case</i> yang menjadi <i>extend</i> -nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.



5.	Generalisasi / <i>Generalization</i> 	Hubungan generalisasi dan spesifikasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6.	<i>Include</i> <<include>> 	Relasi use case tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini; <i>include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan.

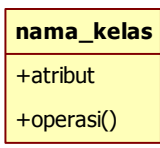

(Sumber: Sukamto dan Shalahuddin,2013:155)

2.2.5. Class Diagram






Widodo dan Herlawati (2011:3), menjelaskan bahwa *class diagram* adalah penggambaran satu set objek yang memiliki atribut dan *behavior* yang sama.

Sukamto dan Shalahuddin (2013:141), menjelaskan bahwa diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Tabel 2.2 Simbol-Simbol *Class Diagram*

No	Simbol	Deskripsi
1.		Kelas pada struktur sistem.
2.		Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.

Lanjutan Tabel 2.2 Simbol-Simbol *Class Diagram*

3.	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (whole-part).

(Sumber: Sukamto dan Shalahuddin, 2013:146)


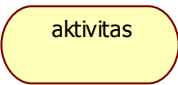
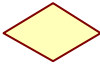


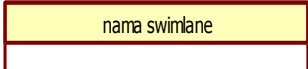

2.2.6. Activity Diagram

Munawar (2005:109), menjelaskan bahwa *activity diagram* adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus.

Sukamto dan Shalahuddin (2013:161), menjelaskan bahwa diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.



Tabel 2.3 Simbol-simbol dalam *Activity Diagram*

No	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber: Sukamto dan Shalahuddin,2013:162)


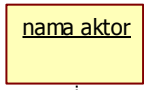

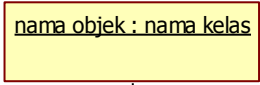


2.2.7. *Sequence Diagram*

Munawar (2005:87), *Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan message (pesan) yang diletakkan diantara obyek-obyek ini di dalam use case.



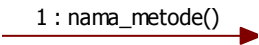
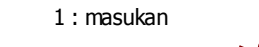

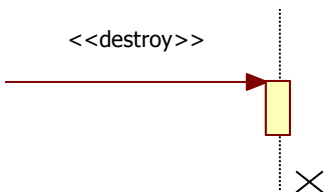
Sukanto dan Shalahuddin (2013:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.

Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>nama aktor</p> <p>tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
2.	<p>Garis hidup / <i>lifeline</i></p> 	menyatakan kehidupan suatu objek.
3.	<p>Objek</p>  <p>nama objek : nama kelas</p>	Menyatakan objek yang berinteraksi pesan.
4.	<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
5.	<p>Pesan tipe create</p> <p><<create>></p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.



Lanjutan Tabel 2.4 Simbol-simbol dalam *Sequence Diagram*

6.	Pesan tipe call 	Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/ metode, karena ini memanggil operasi/ metode maka operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	Pesan tipe send 	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe return 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
9.	Pesan tipe destroy 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy.

(Sumber: Sukamto dan Shalahuddin,2013:165)

2.2.8. Collaboration Diagram





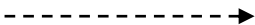
Munawar (2005:101), menjelaskan bahwa *Collaboration diagram* adalah diagram yang menunjukkan *message-message* obyek yang dikirimkan satu sama lain.

Seperti *sequence diagram*, *collaboration diagram* juga digunakan untuk memperlihatkan aliran-aliran pada *use case*. Sementara *sequence diagram*



berurutan menurut waktu, *collaboration diagram* berfokus pada relasi-relasi yang terjadi antara objek yang satu dengan objek-objek yang lainnya. Nugroho (2005:93). Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan (*message*). Nugroho (2005:19).

Tabel 2.5 *Simbol-simbol Collaboration Diagram*

No.	Simbol	Keterangan
1		<i>Note</i>
2		<i>Class</i>
3		<i>Interface</i>
4		<i>Association</i>
5		<i>Dependency & Realize</i>

2.3. Teori Program

2.3.1. Pemrograman Java

2.3.1.1. Pengertian Pemrograman Java

Siallagan (2009:13), menjelaskan bahwa Bahasa pemrograman java adalah bahasa pemrograman berorientasi objek (PBO) atau Object Oriented Programming (OOP). Java bersifat netral, tidak bergantung pada suatu platform, dan mengikuti prinsip WORA (Write Once and Run Anywhere).



Asropudin (2013:52), menjelaskan bahwa Java adalah bahasa pemrograman untuk menciptakan isi yang aktif dalam halaman web, juga dapat dijalankan dalam semua komputer.

Enterprise (2015:1), menjelaskan bahwa java merupakan bahasa pemrograman yang berbasis objek.

Lengkong (2015:21), menjelaskan bahwa java adalah bahasa berorientasi objek yang dapat digunakan untuk pengembangan aplikasi mandiri, aplikasi berbasis internet, serta aplikasi untuk perangkat – perangkat cerdas yang dapat berkomunikasi lewat internet atau jaringan komunikasi.

2.3.1.2. Pengelompokan Tipe Data dalam Java

Sutanta (2005:422), menjelaskan bahwa Bahasa *Java* mengenal tipe data yang mirip dengan bahasa C++. Tipe ukuran memori yang dibutuhkan, dan batasan nilai data dalam java adalah sebagai berikut:

Tabel 2.6 *Tipe Data Dalam Java*

No.	Tipe data		Ukuran Memori	Batasan Nilai
1	<i>Integer</i>	<i>Int</i>	4 byte	-2.147.486.648 s/d 2.147.486.6
		<i>Short</i>	2 byte	-32.768 s/d 32.767
		<i>Long</i>	8 byte	-9.223.372.036.854.775.808L s/d 9.223.372.036.854.775.807L
		<i>Byte</i>	1 byte	-128 s/d 127
2	<i>floating point</i>	<i>Float</i>	4 byte	± 3.40282347E+38F (7 digit signifikan)
		<i>double</i>	8 byte	± 1.79769313486231570E+308 (15 digit signifikan)



3	karakter dan <i>string</i>	<i>Char</i>	1 karakter	Sebuah objek string dan manipulasinya
		<i>String</i>	Banyak karakter	

2.3.1.3. Mendeklarasi Variabel

Berikut ini bentuk umum cara mendeklarasi variabel pada bahasa *java*:

Tipe namaVariabel;

Tipe namaVariabel;

Contoh:

int lebar,tinggi;

float hasil;

2.3.1.4. Operator dalam *Java*

a. Operator Aritmatika

“Operator aritmatika adalah operator-operator yang digunakan untuk mengoprasikan perhitungan (aritmatika). Bahasa pemograman *java* menyediakan operator-operator aritmatika untuk memanipulasi variabel data.” Menurut Siallagan (2009:50).

Tabel 2.7 Operator Aritmatika

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus (sisabagi)

b. Operator Relasional

“Operator relasional adalah operator hubungan (relasi) yang membandingkan kedua nilai *operand* dan hasilnya berupa nilai *boolean*, yaitu benar (*true*) atau salah (*false*).” Menurut Siallagan (2009:65).

**Tabel 2.8** *Operator Relasional*

Operator	Keterangan
==	Sama dengan (membandingkan bukan penugasan)
!=	Tidak sama dengan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan

c. Operator Logika/*Boolean*

“Operator logika adalah operator yang digunakan terhadap *operand* bertipe *Boolean* yang hasilnya benar (*true*) atau salah (*false*).” Menurut Siallagan (2009:51).

Tabel 2.9 *Operator Logika*

Operator	Keterangan
&	Logika AND
	Logika OR
^	Logika XOR
!	Logika NOT

2.3.2. Mengenal Eclipse IDE

Lengkong dkk (2015:21), menyatakan bahwa eclipse IDE adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform independent*).

Berikut ini adalah sifat dari eclipse :

1. *Multi-platform* : Target sistem operasi Eclipse adalah *Microsoft WindowsLinux, Solaris, AIX, HP-UX* dan *Mac OS X*.
2. *Multi-language* : Eclipse dikembangkan dengan bahasa pemrograman *Java*, akan tetapi eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti *C/C++, Cobol, Python, Perl, PHP*, dan lain sebagainya.



3. *Multi-role* : selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktifitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan *open source*. *Open source* berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang disebut *plug-in*.

2.3.3. MySQL

2.3.3.1. Pengertian MySQL

Shalahuddin (2013:46), menjelaskan bahwa SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.

Prasetio (2014: 29), menjelaskan bahwa *MySQL* adalah sebuah database, dan dapat anda bayangkan seperti sebuah tempat penyimpanan.

Sibero (2014:97), menjelaskan bahwa *MySQL* atau dibaca 'My Sekuel' dengan adalah suatu *RDBMS (Relation Database Management System)* yaitu aplikasi sistem yang menjelaskan fungsi pengolahan data.

Berikut tipe-tipe data pada *MySQL*:

**Tabel 2.10.** Tipe Data pada *MySQL*

No	Tipe Data	Ukuran (Bytes)
1.	TINYINT	1
2.	SMALLINT	2
3.	MEDIUM	3
4.	INT	4
5.	BIGINT	8
6.	FLOAT(p)	4
7.	FLOAT	4
8.	DOUBLE	8
9.	REAL	8
10.	DECIMAL (L,D)	L + 1
11.	NUMERIC (L,D)	L + 1
12.	BIT (L)	8
13.	DATE	3
14.	TIME	3
15.	DATETIME	8
16.	TIMESTAMP	4
17.	YEAR	1
18.	CHAR (L)	L
19.	BINARY(L)	L + 1
20.	VARCHAR (L)	L + 1
21.	VARBINARY (L)	L + 1
22.	TINYBLOB	L + 1
23.	TINYTEXT	L + 1
24.	BLOB	L + 2
25.	TEXT	L + 2
26.	MEDIUMBLOB	L + 3
27.	MEDIUMTEXT	L + 3
28.	LOB	L + 4
29.	LONGTEXT	L + 4
30.	ENUM ('nilai1',...)	1 – 2
31.	SET ('nilai1',...)	1 – 8