



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Sutanta (2011:01), menjelaskan bahwa komputer adalah alat elektronik yang mampu melaksanakan beberapa tugas, yaitu menerima *input*, memproses *input* sesuai dengan programnya, menyimpan perintah dan hasil pengolahan, serta menyediakan *output* dalam bentuk informasi.

Sutanta (2011:02), menjelaskan bahwa komputer adalah mesin penghitung elektronik yang cepat, dapat menerima *input* digital, kemudian memprosesnya sesuai dengan suatu program yang tersimpan (*stored program*) dan menghasilkan *output* berupa informasi.

Asropudin (2013:19), menjelaskan bahwa komputer merupakan alat bantu pemrosesan data secara elektronik dan cara pemrosesan datanya berdasarkan urutan instruksi atau program yang tersimpan dalam memori masing – masing komputer.

Dari ketiga pengertian tersebut, dapat disimpulkan bahwa komputer adalah alat elektronik atau mesin penghitung yang cepat, yang menerima input, memproses sesuai dengan perintah dan hasil pengolahan, dan menyediakan output berupa informasi.

2.1.2. Pengertian Data

Asropudin (2013:22), menjelaskan bahwa data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi.

Sutanta (2011:13), menjelaskan bahwa data didefinisikan sebagai bahan keterangan tentang kejadian – kejadian nyata atau fakta – fakta yang dirumuskan dalam sekelompok tertentu yang tidak acak, yang menunjukkan jumlah tindakan atau hal.

Jadi, data adalah kenyataan yang menggambarkan suatu kejadian atau fakta yang nyata berupa lambang maupun simbol , gambar, angka dan huruf.



2.1.3. Pengertian Pengolahan Data

Sutabri (2012:6), mengemukakan pengolahan data merupakan bahan mentah untuk diolah yang hasilnya kemudian menjadi informasi.

Ladjamudin (2005:9), menjelaskan bahwa pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan.

Kristanto (2003:8), menjelaskan bahwa pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk menjadi informasi yang memiliki kegunaan.

Jadi, pengolahan data adalah kegiatan penyimpanan data dan penangan data dalam waktu tertentu yang menggambarkan perubahan bentuk menjadi informasi yang memiliki kegunaan.

2.1.4. Pengertian Sistem

Sutabri (2012:6), menjelaskan sistem adalah sekelompok unsur yang erat hubungannya satu dengan yang lain yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Sutabri (2012:6), menjelaskan bahwa sistem merupakan bagian dari sistem yang lebih besar.

Dari kedua pengertian tersebut, dapat disimpulkan bahwa sistem adalah sekelompok unsur kumpulan komponen yang saling terkait untuk mencapai tujuan tertentu.

2.1.4.1. Karakteristik Sistem

Sutabri (2012:13), karakteristik sistem terdiri dari:

1. Komponen Sistem (*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem.



2. Batasan Sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut dengan masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklarifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lainnya.

7. Pengolahan Sistem (*Procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya.



2.1.4.2. Klasifikasi Sistem

Sutabri (2012:15), sistem dapat diklasifikasikan menjadi beberapa bagian yaitu:

1. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem sistem yang berupa pemikiran atau ide ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu suatu sitem yang berupa pemikiran tentang hubungan antara manusia dengan Tuhan; sedangkan sistem fisik merupakan sistem yang ada secara fisik, seperti sistem komputer, sistem produksi, dan lain sebagainya.

2. Sistem alamiah dan sistem buatan manusia

Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya sistem perputaran bumi, terjadinya siang dan malam, dan pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan hubungan manusia dengan mesin, yang disebut dengan *human machine system*. Sistem informasi berbasis komputer merupakan contohnya, karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

3. Sistem deterministik dan sistem probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem deterministik. Sistem komputer adalah contoh dari sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem yang bersifat probabilistik adalah sistem yang kondisi masa depannya tidak dapat diprediksi, karena mengandung unsur probabilitas.

5. Sistem tertutup dan sistem terbuka

Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh oleh lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa ada campur tangan dari pihak luar. Sedangkan sistem terbuka merupakan sistem yang berhubungan dan dipengaruhi oleh lingkungan luarnya, yang menerima masukan dan menghasilkan keluaran untuk subsistem lainnya.



2.1.5. Pengertian Informasi

Sutabri (2012:22), menjelaskan bahwa informasi adalah data yang telah diklarifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya.

Kristanto (2003:6), menjelaskan bahwa informasi adalah kumpulan data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerima.

Jadi, informasi adalah kumpulan data yang telah diolah menjadi bentuk yang lebih berarti bagi pemakai untuk digunakan dalam proses pengambilan keputusan.

2.2. Teori Khusus

2.2.1. Pemrograman Berorientasi Objek

2.2.1.1. Pengertian Pemrograman Berorientasi Objek

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

Nugroho (2005:6), 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.

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

2.2.1.2. Ciri Pemrograman Berorientasi Objek

Siallagan (2009:149), 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 penyembunyian 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. *Rational Unified Process (RUP)*

Sukamto dan Shalahuddin (2013:125), *RUP (Rational Unified Process)* adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*).

Siahaan (2012:183), *Rational Unified Process (RUP)* adalah salah satu kerangka kerja untuk melakukan proses rekayasa kebutuhan.



Dari beberapa definisi diatas penulis menyimpulkan bahwa *RUP* adalah tahapan pengembangan sistem secara iteratif khusus untuk pemrograman berorientasi objek.

2.2.2.1. Metodologi *Rational Unified Process (RUP)*

RUP (Rational Unified Process) menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language*. Melalui gambar dibawah dapat dilihat bahwa *RUP* memiliki, yaitu:

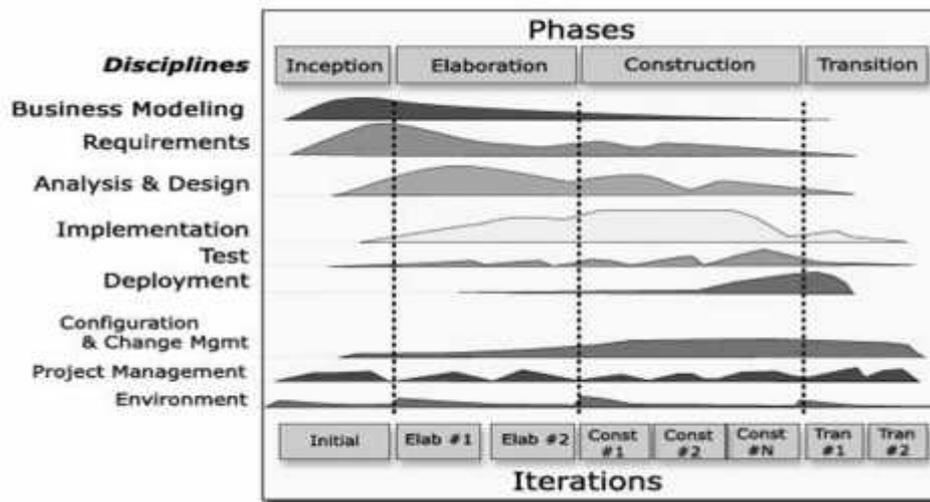
a. Dimensi Pertama

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:

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



Gambar 2.1. *Arsitektur Rational Unified Process (IBM, 2007)*

2.2.2.2. Fase *Rational Unified Process (RUP)*

RUP (Rational Unified Process) memiliki empat buah tahap atau fase yang dapat dilakukan secara iteratif. Berikut ini penjelasan untuk setiap fase pada *RUP (Rational Unified Process)*.

a. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

b. *Elaboration* (perluasan/perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem.

c. *Construction* (konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem.

d. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*.

Akhir dari keempat fase ini adalah produk perangkat lunak yang sudah lengkap. Keempat fase pada *RUP (Rational Unified Process)* dijalankan secara berurutan dan iteratif dimana setiap iterasi dapat digunakan untuk memperbaiki iterasi berikutnya.



2.2.2.3. Aliran Kerja Utama (*Rational Unified Process*) *RUP*

Adapun aliran kerja utama pada Metodologi *RUP* (*Rational Unified Process*) adalah sebagai berikut:

a. Pemodelan Bisnis (*Bussines Modeling*)

Mendeskripsikan struktur dan proses-proses bisnis organisasi.

b. Kebutuhan (*Requirement*)

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

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

Mendeskripsikan berbagai arsitektur perangkat lunak dari berbagai sudut pandang.

d. Implementasi (*Implementation*)

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

e. Pengujian (*Test*)

Mendeskripsikan kasus uji, prosedur, dan alat ukur pengujian.

f. *Deployment*

Menangani konfigurasi sistem yang akan diserahkan.

2.2.2.4. Aliran Kerja Pendukung *RUP*

Adapun aliran kerja pendukung *RUP* adalah sebagai berikut:

a. Manajemen konfigurasi dan perubahan (*configuration and change management*)

mengendalikan perubahan dan memelihara artifak-artifak proyek.

b. Manajemen proyek (*Project Management*)

Mendeskripsikan berbagai strategi pekerjaan dengan proses yang berulang.

c. Lingkungan (*Environment*)

Menangani infrastruktur yang dibutuhkan untuk mengembangkan sistem.



2.2.3. Unified Modelling Language (UML)

Sukamto dan Shalahuddin (2013:133), *UML (Unified Modelling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

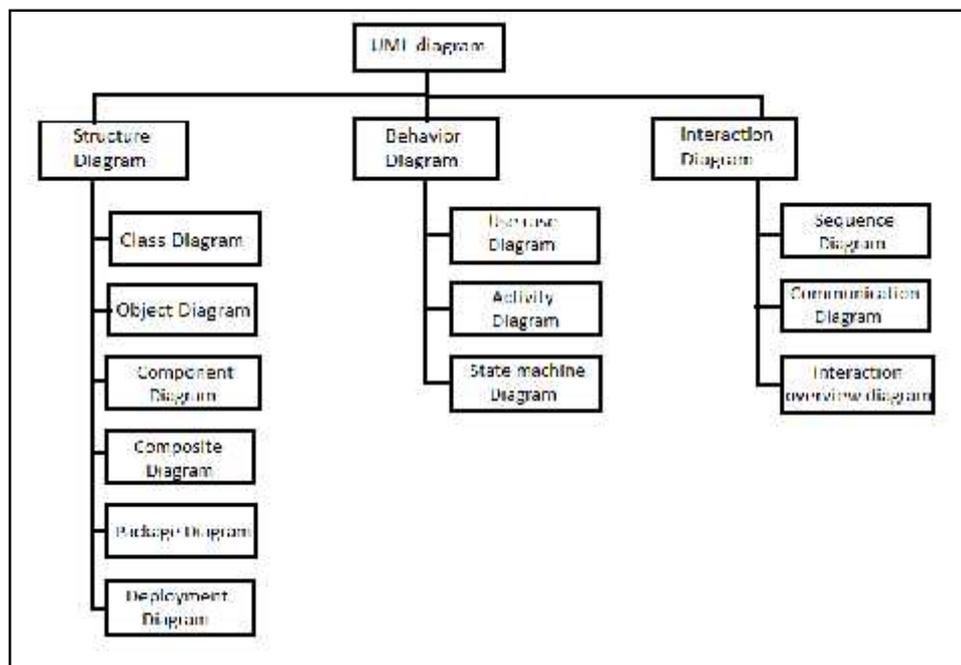
Siahaan (2012:184), *UML* adalah sebuah bahasa standar industri yang membantu komunitas kebutuhan perangkat lunak saling berkomunikasi.

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

Dari beberapa definisi di atas penulis menyimpulkan bahwa *UML (Unified Modelling Language)* adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek.

2.2.3.1. Macam-macam Diagram (*Unified Modelling Language*) UML

UML (Unified Modelling Language) 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* (Sukamto dan Shalahuddin 2013:133)



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. *Class Diagram*

Sukanto dan Shalahuddin (2013:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

1. Kelas main

Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas yang menangani tampilan sistem (*view*)

Kelas yang mendefinisikan dan mengatur tampilan ke pemakai.

3. Kelas yang diambil dari pendefinisian *use case* (*controller*)

Kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.

4. Kelas yang diambil dari pendefinisian data (*model*)

Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.



Sukanto dan Shalahuddin (2013:146), simbol-simbol yang ada pada diagram kelas:

Tabel 2.1 Simbol diagram kelas

No.	Simbol	Keterangan
1.	Kelas <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 5px auto;"> nama_kelas <hr style="border: 0; border-top: 1px solid black; margin: 2px 0;"/> + atribut <hr style="border: 0; border-top: 1px solid black; margin: 2px 0;"/> + operasi() </div>	Kelas pada struktur sistem
2.	Antarmuka / <i>interface</i> <div style="text-align: center; margin: 10px 0;">  nama_interface </div>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	Asosiasi / <i>association</i> <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	Asosiasi berarah / <i>directed association</i> <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	Generalisasi <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus)
6.	Kebergantungan / <i>dependency</i> <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna kebergantungan antarkelas
7.	Agregasi / <i>aggregation</i> <div style="text-align: center; margin: 10px 0;">  </div>	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>)



2.2.5. Use Case Diagram

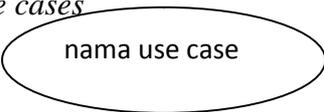
Sukanto dan Shalahuddin (2013:155), *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

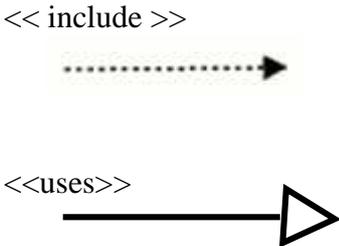
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Sukanto dan Shalahuddin (2013:156), simbol-simbol yang ada pada diagram *use case*:

Tabel 2.2 Simbol diagram *use case*

No.	Simbol	Keterangan
1.	<p><i>Use cases</i></p>  <p>nama use case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>
2.	<p>Aktor / <i>actor</i></p>  <p>nama aktor nama_interface</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 <i>actor</i>



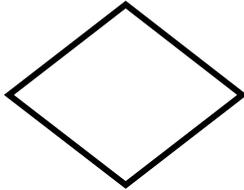
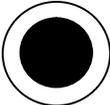
3.	Asosiasi / <i>association</i> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor
4.	Ekstensi / <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; ditambahkan, missal arah panah mengarah pada <i>use case</i> yang ditambahkan; 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 spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya
6.	Menggunakan / <i>include</i> / <i>uses</i> 	Relasi <i>use case</i> 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> .

2.2.6. Activity Diagram

Sukanto dan Shalahuddin (2013:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada diagram aktivitas:



Tabel 2.3 Simbol-simbol diagram aktivitas

No.	Simbol	Keterangan
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.	<p>Swimlane</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p style="text-align: center;">nama swimlane</p> </div> <p>atau</p> <div style="border: 1px solid black; padding: 5px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">n</td> <td></td> </tr> <tr> <td style="text-align: center;">a</td> <td></td> </tr> <tr> <td style="text-align: center;">m</td> <td></td> </tr> <tr> <td style="text-align: center;">a</td> <td></td> </tr> </table> </div>	n		a		m		a		<p><i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>
n										
a										
m										
a										

2.2.7. Sequence Diagram

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

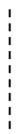
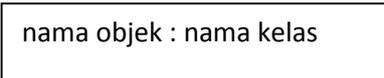
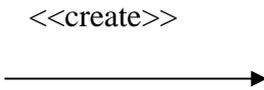
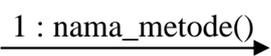
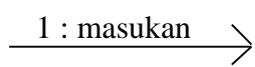
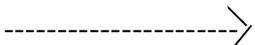
Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Sukanto dan Shalahuddin (2013:165), simbol-simbol yang ada pada diagram sekuen:

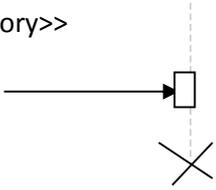
Tabel 2.4 Simbol-simbol diagram sekuen

No.	Simbol	Keterangan
1.	<p>Aktor</p> <div style="text-align: center;">  <p>aktor atau</p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p style="text-align: center;">nama aktor</p> </div> <p>tanpa waktu aktif</p> </div>	<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 <i>actor</i></p>



2.	Garis hidup / <i>lifeline</i> 	Menyatakan kehidupan suatu objek
3.	Objek 	Menyatakan objek yang berinteraksi pesan
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
5.	Pesan tipe create 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	Pesan tipe <i>call</i> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri
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 destrory <<destrory>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>
----	--	---

2.2.8. Kamus Data

Sukamto dan Shalahuddin (2013:73), menyatakan bahwa kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan).

Sukamto dan Shalahuddin (2013:73) menjelaskan simbol-simbol yang di gunakan dalam kamus data, yaitu :

Table 2.5 Simbol-simbol dalam Kamus Data

No	Simbol	Arti
1	=	disusun atau terdiri atas
2	+	Dan
3	[[]]	baik ...atau...
4	{ } ⁿ	n kali diulang/ bernilai banyak
5	()	data operasional
6	*...*	batas komentar

2.3. Pengertian Judul

2.3.1. Pengertian Aplikasi

Sutabri (2012:147), berpendapat bahawa aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

Asropuddin (2013:6), menyatakan aplikasi adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Ms-Word*, *Ms-Excel*.

Sugiar (2014:84), menjelaskan bahwa aplikasi adalah program yang dibuat untuk melaksanakan tugas tertentu yang dibutuhkan pengguna komputer.



Dari ketiga pengertian tersebut, dapat disimpulkan bahwa aplikasi adalah alat terapan atau program yang dibuat dan difungsikan secara khusus untuk melaksanakan tugas tertentu sesuai dengan kebutuhan *user*.

2.3.2. Pengertian *Monitoring*

Wahana Komputer (2005:274), menjelaskan bahwa *monitoring* adalah program yang mengatur seluruh sistem, memasukkan file apa yang diperlukan, menyimpan track dan lain-lain.

Wahana Komputer (2005:274), menjelaskan bahwa *monitoring* adalah kegiatan pemeriksaan suatu program dengan rutin diagnostik.

Dari kedua pengertian diatas dapat disimpulkan bahwa *monitoring* adalah program yang mengatur seluruh sistem dengan rutin diagnostik.

2.3.3. Pengertian Laboratorium

Mastika et al. (2014), menjelaskan bahwa laboratorium adalah tempat yang dilengkapi peralatan untuk melangsungkan eksperimen di dalam sains atau melakukan pengujian dan analisis.

Mastika et al. (2014), menjelaskan bahwa laboratorium adalah bangun atau ruangan yang dilengkapi peralatan untuk melangsungkan penelitian ilmiah maupun praktek pembelajaran.

Mastika et al. (2014), menjelaskan bahwa laboratorium merupakan tempat kerja untuk melangsungkan penelitian.

Dari ketiga pengertian tersebut, dapat disimpulkan bahwa laboratorium adalah tempat yang dilengkapi peralatan, bangun atau ruang yang dilengkapi peralatan untuk melangsungkan penelitian.

2.3.4. Aplikasi Monitoring Aktivitas Laboratorium Komputer Mahasiswa Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya

Penulis mendefinisika judul laporan akhir yang penulis buat bertujuan untuk membantu proses monitoring yang ada pada laborotorium jurusan Manajemen Informatika, khususnya pada aktivitas belajar mahasiswa pada saat



menggunakan laboratorium komputer yang dapat mengetahui aktivitas apa saja yang dilakukan mahasiswa pada proses belajar di laboratorium karena dosen pengajar dapat melihat *history* atau *record* dari kegiatan mahasiswa tersebut.

2.4. Teori Program

2.4.1. Basis Data (*database*)

Berikut ini ada beberapa definisi tentang Basis Data (*database*) yang disajikan oleh beberapa ahli :

Al Fatta (2007:10), *database* yaitu kumpulan data dan informasi yang diorganisasikan sedemikian rupa sehingga mudah diakses pengguna sistem informasi.

Sutanta (2011:35), *database* (basis data) merupakan sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar *record* dan rincian data terhadap objek tertentu.

Dari definisi diatas, dapat disimpulkan bahwa basis data (*database*) adalah kumpulan data yang saling berhubungan dari satu atau lebih organisasi.

2.4.2. *Visual Basic. Net 2010*

Darmayuda (2014:14), *Visual Basic.NET 2010* tidak jauh berbeda dengan *Visual Basic.NET* sebelumnya atau *Visual Basic 6.0 IDE (Interface Development Environment)*.

Adapun alat atau menu menu di *Microsoft Visual Basic.NET 2010* yaitu :

1. *Title Bar*, berfungsi untuk menampilkan nama project yang aktif atau sedang dikembangkan.
2. *Menu Bar*, berfungsi untuk pengelolaan fasilitas yang dimiliki oleh *Visual Basic.NET 2010*, sedangkan *Tool Bar*, berfungsi untuk melakukan perintah khusus secara cepat.
3. *Form*, adalah objek utama berfungsi untuk melakukan objek-objek yang terdapat pada *Toolbox* yang digunakan dalam melakukan perancangan sebuah tampilan program aplikasi.
4. *ToolBox*, berfungsi untuk menyediakan objek-objek atau komponen yang digunakan dalam merancang sebuah *form* pada program aplikasi.



5. *Solution Explorer*, berfungsi untuk menampilkan nama *project*, *file* konfigurasi beserta folder, *file-file* pendukung yang terdapat pada sebuah program aplikasi.
6. *Properties Window*, berfungsi untuk mengatur *properties-properties* pada objek (*setting object*) yang diletakan pada sebuah *form*.