



BAB II TINJAUAN PUSTAKA

2.1. Teori Umum

Dalam bab ini akan diterapkan mengenai landasan teori yang bertujuan untuk membahas permasalahan yang diambil. Sesuai judul dalam pembuatan Laporan Akhir, yang akan dibuat penulis terdiri dari pengertian-pengertian sebagai berikut :

2.1.1. Pengertian Perangkat Lunak

Ladjamudin (2013:20), *software* merupakan kumpulan dari perintah/fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu.

Sukamto dan Shalahuddin (2013:2), perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumen perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*).

Mulyono (2010:97), *software* adalah rangkaian instruksi yang memerintahkan komputer untuk melakukan tugas tertentu sesuai dengan perintah yang diberikan oleh seorang pengguna komputer.

Dari beberapa definisi diatas penulis menyimpulkan bahwa perangkat lunak adalah serangkaian instruksi elektronik yang dapat mengoperasikan suatu pekerjaan sesuai dengan kebutuhan.

2.1.2. Pengertian Program

Sutarman (2012:3), program adalah barisan perintah/instruksi yang disusun sehingga dapat dipahami oleh computer dan kemudian dijalankan sebagai barisan perhitungan numeric dimana barisan perintah tersebut berhingga, berakhir dan menghasilkan *output*.

Siallagan (2009:3), program dapat dianalogikan sebagai instruksi atau perintah-perintah untuk mengoperasikan atau menjalankan *hardware*.



Wahana Komputer (2005:348), program adalah serangkaian petunjuk yang berupa perintah-perintah yang disusun sedemikian rupa untuk melaksanakan suatu tugas yang akan dikerjakan oleh komputer.

Dari beberapa definisi diatas penulis menyimpulkan bahwa program adalah serangkaian instruksi yang dioperasikan untuk melaksanakan suatu tugas di dalam komputer.

2.1.3. Pengertian Data

Asropudin (2013:22), data adalah kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti.

Sutabri (2012:25), data merupakan bentuk mentah yang belum dapat bercerita banyak sehingga perlu diolah lebih lanjut.

Kristanto (2008:7), data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

Dari beberapa definisi diatas penulis menyimpulkan bahwa data adalah sekumpulan fakta yang harus diolah lebih lanjut untuk menghasilkan suatu informasi.

2.1.4. Pengertian Pengolahan Data

Ladjamudin (2013:9), pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan.

Sutarman (2012:4), pengolahan data adalah proses perhitungan/transformatasi data *input* menjadi informasi yang mudah dimengerti ataupun sesuai dengan yang diinginkan.

Kristanto (2008:8), pengolahan data adalah waktu yang digunakan untuk menggambarkan perubahan bentuk data menjadi informasi yang memiliki kegunaan.

Dari beberapa definisi diatas penulis menyimpulkan bahwa pengolahan data merupakan proses pengolahan data yang diinputkan untuk diubah menjadi informasi yang memiliki nilai dan manfaat bagi sekelompok orang tertentu.



2.2. Teori Judul

Teori judul adalah teori yang menjelaskan judul aplikasi pembuatan Laporan Akhir.

2.2.1. Pengertian Aplikasi

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

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

Wahana Komputer (2005:12), aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan atau menyelesaikan masalah-masalah khusus, misalnya program penggajian.

Dari beberapa definisi diatas penulis menyimpulkan bahwa aplikasi adalah perangkat lunak yang digunakan untuk mengerjakan masalah tertentu.

2.2.2. Pengertian Web

Asropudin (2013:109), sebuah kumpulan halaman (*webpages*) yang diawali dengan halaman muka (*homepages*) yang berisikan informasi, iklan, serta program interaksi.

Risnandar (2013:12), *web* adalah salah satu fitur dari internet di samping fitur-fitur lain seperti *e-mail*, *remote (SSH)*, dan *ftp*.

Wahana Komputer (2005:502), layanan penyedia informasi di internet yang berbasis grafis.

Dari beberapa definisi diatas penulis menyimpulkan bahwa *web* adalah koleksi sumber informasi kaya grafis yang saling berhubungan satu sama lain dalam internet yang lebih besar.

2.2.3. Pengertian Media

Simamora (2009:65), media adalah alat yang berfungsi menyampaikan pesan.



Media adalah alat dan bahan yang digunakan dalam proses pengajaran atau pembelajaran. (www.kbbi.web.id , diakses pada 31 Mei 2015 pukul 6:56 WIB)

Dari beberapa definisi diatas penulis menyimpulkan bahwa media adalah sebuah alat perantara yang digunakan untuk menyampaikan pesan.

2.2.4. Pengertian Pembelajaran

Simamora (2009:65), pembelajaran merupakan sebuah proses komunikasi antara peserta didik, pendidik, dan bahan ajar.

Pembelajaran adalah proses, cara, perbuatan yang menjadikan makhluk hidup belajar. (www.kbbi.web.id , diakses pada 31 Mei 2015 pukul 6:57 WIB)

Dari beberapa definisi diatas penulis menyimpulkan bahwa pembelajaran adalah suatu proses komunikasi agar makhluk hidup dapat belajar.

2.2.5. Media Pembelajaran

Simamora (2009:65), media pembelajaran adalah alat yang berfungsi untuk menyampaikan pesan pembelajaran.

Media pembelajaran merupakan pembawa pesan-pesan atau informasi yang bertujuan pembelajaran atau mengandung maksud-maksud pembelajaran.

Media pembelajaran memiliki tiga fungsi, yakni:

1. Media pembelajaran dapat mengatasi keterbatasan pengalaman yang dimiliki oleh para peserta didik. Pengalaman tiap peserta didik berbeda-beda, tergantung dari faktor-faktor yang menentukan kekayaan pengalaman anak, seperti ketersediaan buku, dan sebagainya.
2. Media pembelajaran dapat melampaui batasan ruang kelas. Banyak hal yang tidak mungkin dialami secara langsung di dalam kelas oleh para peserta didik tentang suatu obyek. Melalui penggunaan media yang tepat, maka semua obyek itu dapat disajikan kepada peserta didik.
3. Media pembelajaran memungkinkan adanya interaksi langsung antara peserta didik dengan lingkungannya.

Disamping tiga fungsi diatas, terdapat berbagai jenis media belajar, diantaranya:

1. *Media Visual* : grafik, diagram, *chart*, bagan, poster, kartun, komik.



2. Media Audial : radio, *tape recorder*, laboratorium bahasa, dan sejenisnya.
3. *Projected still media* : *slide*; *over head proyektor (OHP)*, *LCD Proyektor* dan sejenisnya.
4. *Projected motion media* : film, televisi, video (VCD, DVD, VTR), komputer dan sejenisnya.
5. *Study Tour Media* : Pembelajaran langsung ke obyek atau tempat study.

2.2.6. Pengertian Interaktif

Interaktif adalah kegiatan yang bersifat saling melakukan aksi, antar-hubungan, saling aktif. (www.kbbi.web.id , diakses pada 31 Mei 2015 pukul 6:57 WIB)

Wahana Komputer (2005:199), interaktif adalah kemampuan sistem/program yang bisa menanyakan sesuatu pada pengguna (menggunakan tanya jawab), kemudian mengambil tindakan berdasarkan respon tersebut.

Dari beberapa definisi diatas penulis menyimpulkan bahwa interaktif adalah proses hubungan dua arah dan mengambil tindakan atas respon tersebut.

2.2.7. Pengertian Aplikasi Web Media Pembelajaran Interaktif Berorientasi Objek di Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya

Merupakan suatu perangkat lunak yang berfungsi untuk menyampaikan pesan pembelajaran secara interaktif untuk mendukung proses pembelajaran di Jurusan Manajemen Informatika Politeknik Negeri Sriwijaya.

2.3. Teori Khusus

Teori khusus adalah teori yang berkaitan dengan sejumlah fakta-fakta yang bersifat partikular. Sesuai judul dalam pembuatan Laporan Akhir, teori khusus yang ditulis adalah sebagai berikut:

2.3.1. Pemrograman Berorientasi Objek

paradigma berorientasi objek adalah cara yang berbeda dalam memandang aplikasi-aplikasi.



2.3.1.1. Pengertian Pemrograman Berorientasi Objek

Sukamto 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.

2.3.1.2. 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 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.

2.3.2. *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.

2.3.2.1. 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

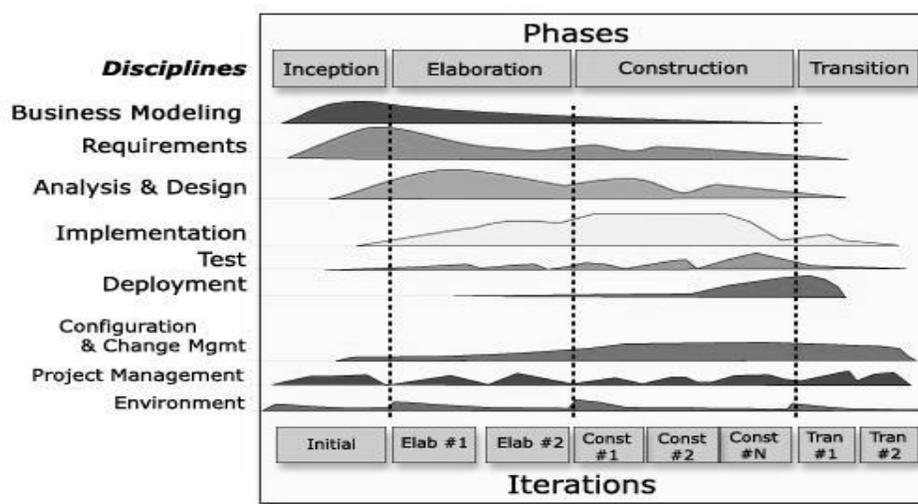
Digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari *phase* selanjutnya. Setiap *phase* dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaborationn*, *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 (IBM, 2007)*

2.3.2.2. 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.

2.3.2.3. 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*)

Mendesripsikan 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*)

Mendesripsikan berbagai arsitektur perangkat lunak dari berbagai sudut pandang.

d. Implementasi (*Implementation*)

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

e. Pengujian (*Test*)

Mendesripsikan kasus uji, prosedur, dan alat ukur pengujian.

f. *Deployment*

Menangani konfigurasi sistem yang akan diserahkan.

2.3.2.4. 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.

2.3.3. *Unified Modelling Language (UML)*

Sukanto 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.

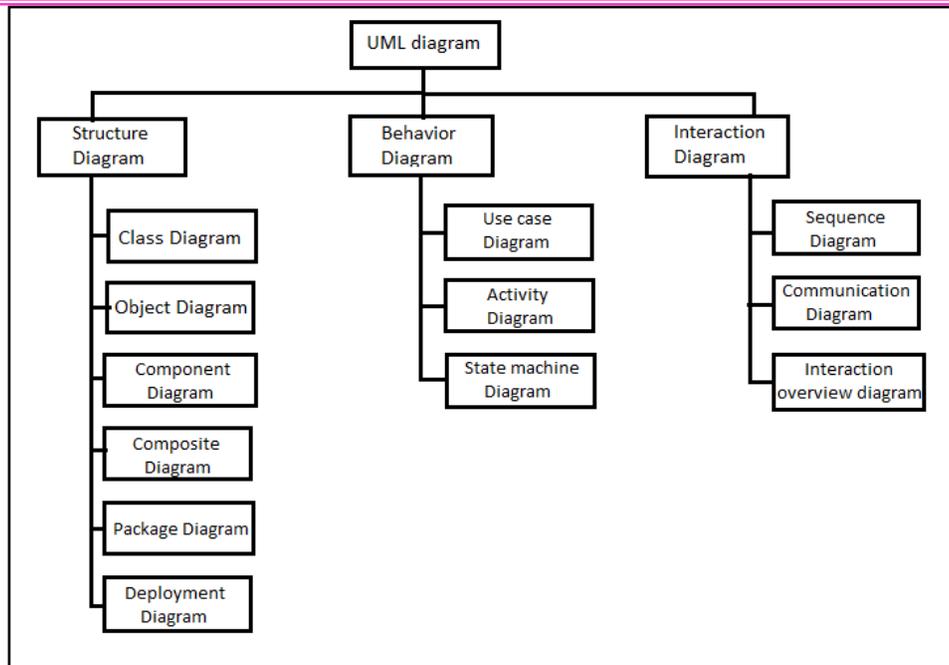
Siahaan (2012:184), *UML* adalah sebuah bahasa standar industri yang membantu komunitas kebutuhan perangkat lunak saling berkomunikasi.

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

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

2.3.3.1. *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 interkasi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.3.4. *Use Case Diagram*

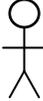
Sukamto dan Shalahuddin (2013:155), *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Nugroho (2005:59), diagram *use case* memperlihatkan hubungan-hubungan yang terjadi antara aktor-aktor dengan *use case-use case* dalam sistem.



Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1 Simbol-simbol Diagram *Use Case*

No.	Simbol	Nama	Keterangan
1.		Aktor	Merupakan seseorang atau sesuatu yang berinteraksi dengan sistem
2.		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar antar unit atau aktor
3.		Generalisasi	Menggambarkan hubungan generalisasi dan spesialisasi (umum-khusus) antara interaksi dalam objek

(Sumber : Nugroho, 2005:59)

2.3.5. *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.

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

Nugroho (2005:110), diagram kelas adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang kita kembangkan.

Nugroho (2005:111), kelas seperti juga objek, adalah sesuatu yang membungkus (*encapsulation*) informasi (baca:atribut) dan perilaku (baca:operasi) dalam dirinya. Berdasarkan pernyataan di atas, diketahui bahwasanya notasi kelas dalam *UML (Unified Modelling Language)* adalah sebagai berikut:



Nama kelas
Atribut
Operasi

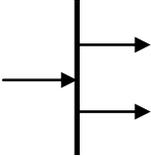
Gambar 2.3. Notasi kelas dalam UML (Nugroho, 2005:111)

2.3.6. Activity Diagram

Nugroho (2005:61), *Activity diagram* adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *use case*.

Berikut simbol-simbol yang sering digunakan dalam *activity diagram*:

Tabel 2.2 Simbol-simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		Titik Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.		Titik Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
3.		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
4.		<i>Fork</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas yang lebih dari satu



Lanjutan Tabel 2.2 Simbol-simbol <i>Activity Diagram</i>			
No.	Simbol	Nama	Keterangan
5.		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
6.		Percabangan	Pilihan untuk mengambil keputusan

(Sumber : Nugroho, 2005:61)

2.3.7. *Sequence Diagram*

Nugroho (2005:91), *sequence diagram* adalah *interaction diagram* yang memperlihatkan *event-event* yang berurutan sepanjang berjalannya waktu.

Tabel 2.3 Simbol-simbol *Sequence Diagram*

No.	Simbol	Nama	Keterangan
1.		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri.
2.		Entitas Kelas	Menggambarkan hubungan kegiatan yang akan dilakukan.
3.		<i>Boundary Class</i>	Menggambarkan sebuah penggambaran dari form.
4.		<i>Control Class</i>	Menggambarkan penghubung boundary dengan tabel.



Lanjutan Tabel 2.3 Simbol-simbol Sequence Diagram			
No.	Simbol	Nama	Keterangan
5.		<i>A Focus of Control and A Life Line</i>	Tempat menggambarkan tempat mulai dan berakhirnya sebuah <i>message</i> .
6.		Spesifikasi	Spesifikasi komunikasi dari antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi.

(Sumber : Nugroho, 2005:91)

2.3.8. Collaboration Diagram

Nugroho (2005:93), 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. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan (*message*).

Tabel 2.4 Simbol-simbol Collaboration Diagram

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

(Sumber : Nugroho, 2005:93)



2.4. Teori Program

Teori program adalah teori yang menjelaskan bahasa pemrograman yang diaplikasikan penulis dalam pembuatan Laporan Akhir.

2.4.1. Pemrograman Java

Pemrograman *java* adalah bahasa pemrograman berorientasi objek (PBO) atau *Object Oriented Programming (OOP)*.

2.4.1.1. Pengertian Pemrograman Java

Sukanto dan Shalahuddin (2013:103), *java* adalah bahasa pemrograman objek murni karena semua kode programnya dibungkus dalam kelas.

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

Siallagan (2009:13), 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)*.

2.4.1.2. Pengelompokan Tipe Data dalam Java

Sutanta (2005:422), 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 2.5 *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	$\pm 3.40282347E+38F$ (7 digit signifikan)
		<i>double</i>	8 byte	$\pm 1.79769313486231570E+308$ (15 digit signifikan)
3	karakter dan <i>string</i>	<i>Char</i>	1 karakter	Sebuah objek string dan manipulasinya
		<i>String</i>	Banyak karakter	

(*Sumber : Sutanta, 2005:422*)

2.4.1.3. Operator dalam Java

a. Operator Aritmatika

Siallagan (2009:50), operator aritmatika adalah operator-operator yang digunakan untuk mengoperasikan perhitungan (aritmatika). Bahasa pemrograman *java* menyediakan operator-operator aritmatika untuk memanipulasi variabel data

Tabel 2.6 Operator Aritmatika

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

(*Sumber : Siallagan, 2009:50*)

b. Operator Relasional

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

Tabel 2.7 Operator Relasional

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

**Tabel 2.7 Operator Relasional**

Operator	Keterangan
<	Lebih kecil
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan

(Sumber : Siallagan, 2009:65)

c. Operator Logika/Boolean

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

Tabel 2.8 Operator Logika

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

(Sumber : Siallagan, 2009:51)

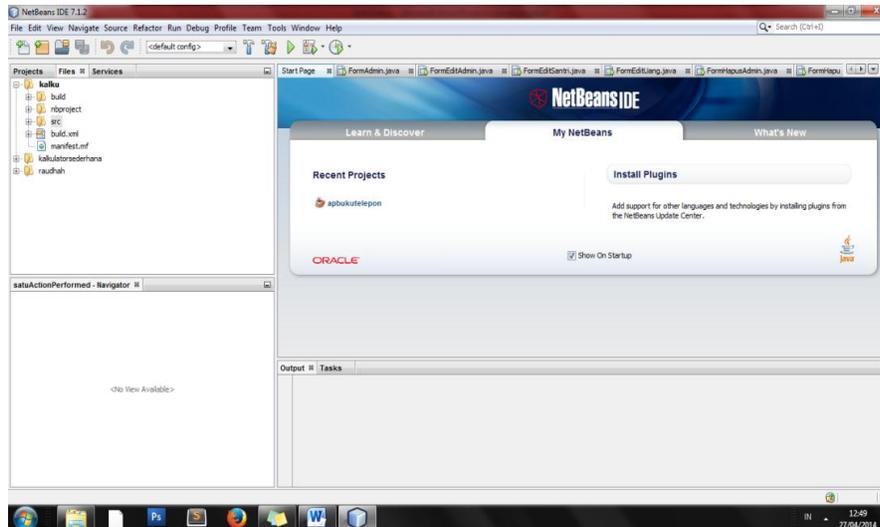
2.4.2. Mengenal Netbeans IDE

Netbeans mempunyai lingkup pemrograman yang terintegrasi dalam suatu perangkat lunak yang di dalamnya menyediakan pembagunan pemrograman *GUI*, *text editor*, *compiler*, dan *interpreter*. *Netbeans* adalah sebuah perangkat lunak *open source* sehingga dapat digunakan secara gratis untuk keperluan komersial maupun nonkomersial yang didukung oleh *Sun Microsystem*.

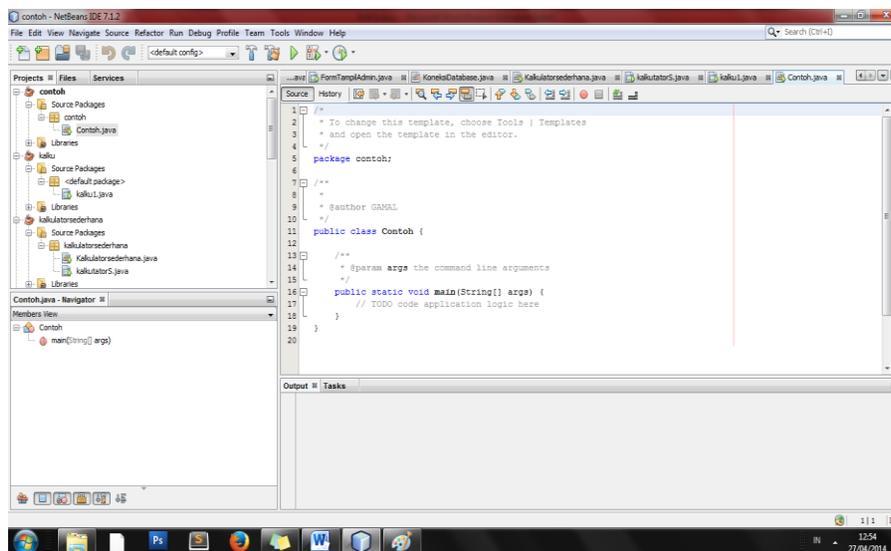


2.4.2.1. Tampilan Netbeans IDE 7

Berikut tampilan dari *Netbeans IDE 7*:



Gambar 2.4. Tampilan Awal Netbeans IDE 7

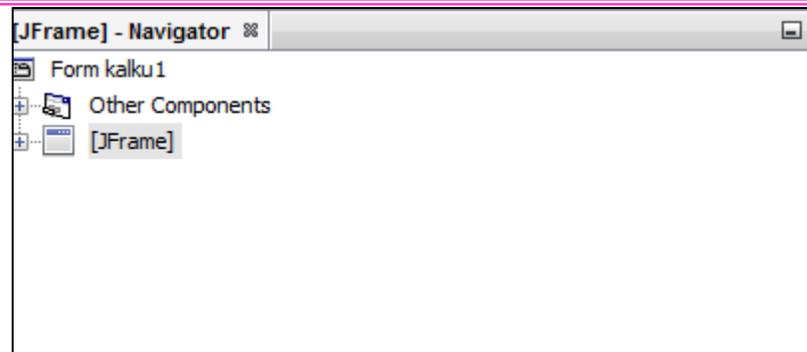


Gambar 2.5. Tampilan Project Editor Netbeans 7

2.4.2.2. Komponen-Komponen Netbeans IDE 7

a. Navigator Windows

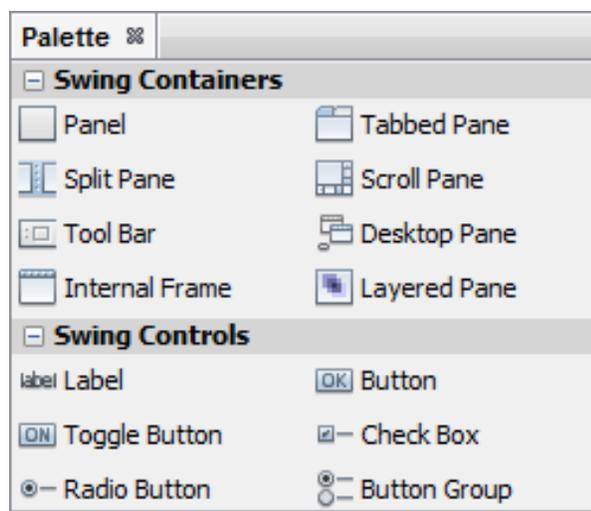
Navigator Windows merupakan jendela yang menampilkan pohon pewarisan dari semua komponen form yang dibuka seperti button, label, menu, timer, dan sebagainya.



Gambar 2.6. Tampilan Navigator Windows

b. *Palette Windows*

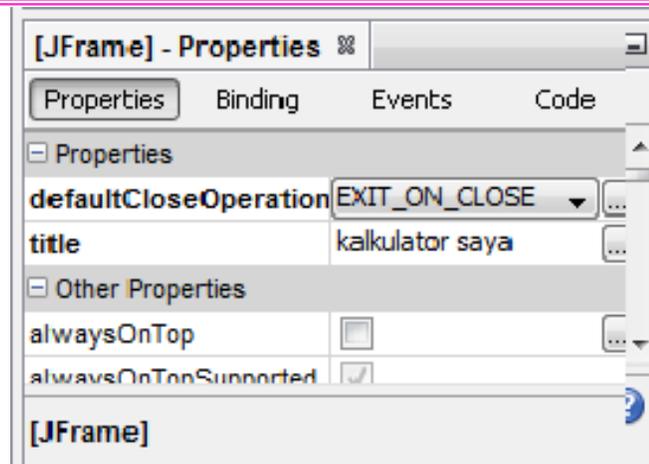
Palette Windows merupakan jendela yang menampilkan daftar semua komponen *swing* yang dapat dimasukkan ke dalam form seperti label, button, menu, dan lainnya.



Gambar 2.7. Tampilan Palette Windows

c. *Properties Windows*

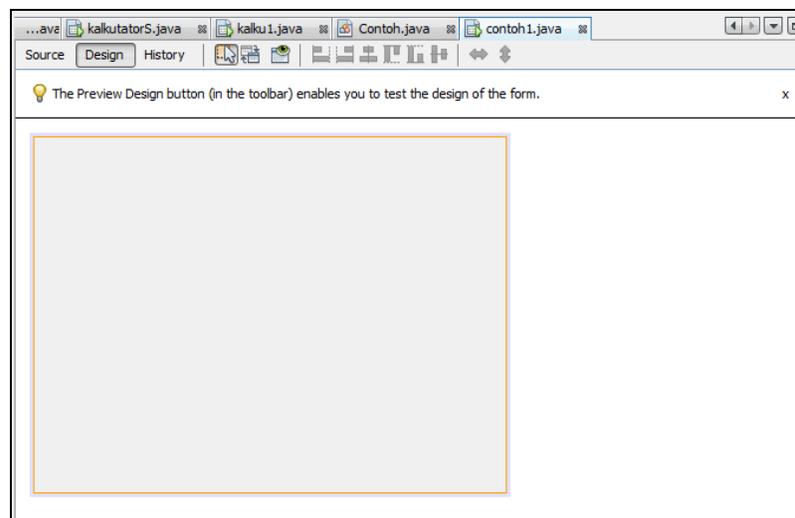
Properties Windows merupakan jendela yang menampilkan pengaturan yang dapat diubah memilih komponen yang akan dipergunakan.



Gambar 2.8. Tampilan Properties Windows

d. Desing Area

Desing area merupakan jendela yang digunakan untuk merancang antarmuka.



Gambar 2.9. Tampilan Design Area

e. Source Area

Source area merupakan jendela yang digunakan untuk menambahkan kode program pada pemrograman java.



```

Source Design History
1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23
2 | * To change this template, choose Tools | Templates
3 | * and open the template in the editor.
4 | */
5 |
6 | /**
7 |  *
8 |  * @author GAMAL
9 |  */
10 | public class contoh1 extends javax.swing.JFrame {
11 |
12 |     /**
13 |      * Creates new form contoh1
14 |      */
15 |     public contoh1() {
16 |         initComponents();
17 |     }
18 |
19 |     /**
20 |      * This method is called from within the constructor to initialize the form.
21 |      * WARNING: Do NOT modify this code. The content of this method is always
22 |      * regenerated by the Form Editor.
23 |      */

```

Gambar 2.10. Tampilan Source Area

2.4.3. MySQL

MySQL tergolong sebagai *database server* (*server* yang melayani permintaan terhadap *database*) yang andal, dapat menangani *database* yang besar dengan kecepatan tinggi.

2.4.3.1. Pengertian MySQL

Kadir (2008:02), MySQL atau dibaca mai-se-Kyu-el merupakan yang tergolong sebagai DBMS (*Database Management System*) yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dibuat untuk membuat MySQL), selain bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi.

2.4.3.2. Keunggulan MySQL

Sebagai *software DBMS*, MySQL memiliki sejumlah fitur seperti yang dijelaskan di bawah ini:

a. *Multiplatform*

MySQL tersedia pada beberapa *platform* (*Windows*, *Linux*, *Unix*, dan lain-lain).

b. Andal, cepat, dan mudah digunakan

MySQL tergolong sebagai *database server* (*server* yang melayani permintaan terhadap *database*) yang andal, dapat menangani *database* yang besar dengan



kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, dan sekaligus mudah untuk digunakan.

c. Jaminan keamanan akses

MySQL mendukung pengamanan *database* dengan berbagai kriteria penaksesan. Sebagai gambaran, dimungkinkan untuk mengatur *user* tertentu agar bisa mengakses data yang bersifat rahasia (misalnya gaji pegawai), sedangkan *user* lain tidak boleh.

d. Dukungan *SQL*

Seperti tersirat dalam namanya, *MySQL* mendukung perintah *SQL* (*Structured Query Language*). Sebagaimana diketahui, *SQL* merupakan standar pengaksesan *database* relasional. Pengetahuan akan *SQL* akan memudahkan siapa pun untuk menggunakan *MySQL*.