



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

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

Mulyono (2010:1) menjelaskan, “Komputer adalah mesin penghitung elektronik yang cepat dan dapat menerima informasi input digital, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya, dan menghasilkan output berupa informasi.”

Sutarman (2009:4) menjelaskan, “Komputer adalah alat yang dapat membaca *input* data dan mengolahnya sesuai dengan program yang ditetapkan untuk menghasilkan informasi yang merupakan *output* hasil pemrosesan *input* data.”

Dari pengertian tersebut, dapat disimpulkan bahwa “komputer merupakan suatu alat bantu untuk membaca *input* data secara elektronik kemudian memproses datanya berdasarkan urutan intruksi atau program yang tersimpan sehingga menghasilkan output berupa informasi.”

2.1.2. Pengertian Perangkat Lunak (Software)

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

Sutarman (2012:14) menjelaskan, “*Software* adalah kumpulan program-program komputer yang memungkinkan *hardware* memproses data.”



Siallagan (2009:3) menjelaskan, “*Software* adalah program-program komputer yang berguna untuk menjalankan atau mengoperasikan suatu pekerjaan sesuai dengan yang dikehendaki.”

Dari pengertian tersebut, dapat disimpulkan bahwa “*software* adalah kumpulan program komputer yang berguna untuk menjalankan suatu pekerjaan yang memungkinkan hardware memproses data.”

2.1.3. Pengertian Program

Sutarman (2012:3) menjelaskan, “Program adalah barisan perintah/instruksi yang disusun sehingga dapat dipahami oleh komputer dan kemudian dijalankan sebagai barisan perhitungan numerik, dimana barisan perintah tersebut berhingga, berakhir, dan menghasilkan *output* .”

Siallagan (2009:3) menjelaskan, “Program dapat dianalogikan sebagai instruksi atau perintah-perintah untuk mengoperasikan atau menjalankan *hardware*”

Dari pengertian tersebut, dapat disimpulkan bahwa “program merupakan kumpulan dari perintah-perintah atau intruksi-intruksi yang disusun sehingga dapat dipahami komputer kemudian dijalankan secara bersamaan dalam satu kesatuan yang menghasilkan sebuah *output*.”

2.1.4. Konsep Dasar Sistem

2.1.4.1. Pengertian Sistem

Sutarman (2009:5) menjelaskan, “Sistem adalah kumpulan elemen yang saling berhubungan dan berinteraksi dalam suatu kesatuan untuk menjalankan suatu proses pencapaian suatu tujuan utama.”

Sutabri (2012:10) menjelaskan, “Sistem diartikan sebagai kumpulan atau himpunan dari unsur, komponen, atau variable yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu.”

Dari pengertian tersebut, dapat disimpulkan bahwa “sistem adalah kumpulan elemen atau variable yang terorganisir yang saling berhubungan dan



berinteraksi dalam satu kesatuan untuk menjalankan suatu proses dalam pencapaian suatu tujuan utama.”

2.1.4.2. Karakteristik Sistem

Sutabri (2012:20) menjelaskan karakteristik sistem adalah sebagai berikut :

a. **Komponen Sistem (*Components*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling berkerjasama membentuk satu kesatuan.

b. **Batasan Sistem (*boundary*)**

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya.

c. **Lingkungan Luar Sistem (*Environment*)**

Bentuk apapun yang ada di luar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem disebut lingkungan luar sistem.

d. **Penghubung Sistem (*interface*)**

Media yang menghubungkan sistem dengan subsistem lain disebut penghubung sistem atau interface.

e. **Masukan Sistem (*Input*)**

Energi yang di masukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*)

f. **Keluaran Sistem (*Output*)**

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain seperti sistem informasi.

g. **Pengolah Sistem (*Proses*)**

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran, contohnya adalah sistem akuntansi.



h. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

2.1.4.3. Klasifikasi Sistem

Sutabri (2012:22), menjelaskan bahwa Sistem dapat diklasifikasikan menjadi beberapa bagian yaitu:

a. Sistem abstrak dan sistem fisik

Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik, misalnya sistem teologia, yaitu sistem yang berupa pemikiran hubungan antara manusia dengan tuhan, sedangkan sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem produksi, sistem penjualan, sistem administrasi personalia dan lain sebagainya.

b. 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 malam, pergantian musim. Sedangkan sistem buatan manusia merupakan sistem yang melibatkan interaksi manusia dengan mesin yang disebut *human machine sistem*. Sistem informasi berbasis komputer merupakan contoh *human machine sistem* karena menyangkut penggunaan komputer yang berinteraksi dengan manusia.

c. Sistem determinasi dan sistem probabilistik

Sistem yang beroperasi dengan tingkah laku yang dapat diprediksi disebut sistem *deterministic*. 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 *probabilistic*.

2.1.4.4. Prinsip Pengembangan Sistem

Sutabri (2012:54), sewaktu akan membangun atau mengembangkan sistem informasi, ada beberapa prinsip yang tidak boleh dilupakan yaitu sebagai berikut:

- a. Sistem yang dikembangkan adalah untuk manajemen.

Setelah sistem selesai dikembangkan, maka yang akan menggunakan informasi dari sistem ini adalah manajemen, sehingga sistem harus dapat memenuhi kebutuhan manajemen.

- b. sistem yang dikembangkan adalah investasi modal yang besar.

untuk pengembangan sistem membutuhkan dana yang tidak sedikit apalagi menggunakan teknologi yang mutakhir.

- c. Sistem yang dikembangkan memerlukan orang yang terdidik.

Manusia lah yang berperan atas keberhasilan suatu sistem, baik dalam proses pengembangan, penerapan, maupun dalam operasinya.

- d. Tahapan kerja dan tugas yang harus dilakukan dalam pengembangan system.

Proses pengembangan sistem umumnya melibatkan beberapa tahapan kerja dan melibatkan beberapa personal dalam bentuk suatu tim untuk mengerjakan.

- e. Proses pengembangan sistem yang tidak harus urut.

Tahapan kerja pengembangan sistem merupakan langkah – langkah yang harus dilakukan. Langkah ini dapat saja tidak berurutan, tetapi dapat dilakukan secara bersama-sama.

- f. Takut membatalkan proyek.

Pada umumnya merupakan pantangan untuk membatalkan proyek yang sedang berjalan. Keputusan untuk meneruskan atau membatalkan suatu proyek memang harus didasarkan evaluasi yang cermat.

- g. Dokumentasi harus ada untuk pedoman dalam pengembangan system.

Kegagalan untuk membuat suatu dokumentasi kerja merupakan salah satu hal yang sering terjadi dan merupakan kesalahan kritis yang dibuat oleh analisis sistem.



2.1.5. Pengertian Komputerisasi

Asropudin (2009:20) menjelaskan, “Komputerisasi merupakan pemakaian komputer sebagai alat bantu penyelesaian tugas sebagai pengganti penyelesaian pekerjaan secara manual.”

Salim (2013:331) menjelaskan, “Komputerisasi adalah penggunaan komputer (dl menghitung, mengolah data, dsb) secara besar-besaran.”

Dari pengertian diatas, dapat disimpulkan bahwa “komputerisasi adalah penggunaan komputer sebagai alat bantu dalam kegiatan pengolahan data sebagai pengganti pekerjaan secara manual.”

2.1.6. Pengertian Pendistribusian

Haryono (2009:194) menjelaskan, “Pendistribusian adalah kegiatan pembagian/penyaluran barang-barang kebutuhan pokok (seperti sembako) oleh pemerintah kepada masyarakat (terutama pada masa darurat) atau kepada pegawai-pegawai.”

2.1.7. Pengertian Beras Miskin (Raskin)

Salim (2013:92) menjelaskan, “Beras adalah padi yang telah terkupas kulitnya (yang menjadi nasi setelah ditanak).”

Salim (2013:391) menjelaskan, “Miskin adalah tidak berharta benda; serba kekurangan (berpenghasilan sangat rendah).”

Dari pengertian tersebut, dapat disimpulkan bahwa “Raskin (beras miskin) adalah subsidi pangan dalam bentuk beras yang diperuntukkan bagi rumah tangga berpenghasilan rendah sebagai upaya dari pemerintah untuk meningkatkan ketahanan pangan dan memberikan perlindungan sosial pada rumah tangga sasaran.

2.1.8. Pengertian Judul Secara Keseluruhan

Sistem Komputerisasi Pendistribusian Beras untuk Warga Miskin pada Kelurahan Karang Jaya Palembang adalah suatu sistem terkomputerisasi yang dibuat untuk memudahkan Kelurahan Karang Jaya dalam melakukan pengolahan



data raskin dengan menggunakan bahasa pemrograman *Java Netbeans 7.4* dan *database MySQL*.

2.2. Teori Khusus

2.2.1. Pengertian OOP (*Object Oriented Programming*)

Sukamto dan Shalahuddin (2013:100) menjelaskan, “*OOP* adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.”

Siallagan (2009:149) menjelaskan, “*Object Oriented Programming (OOP)* adalah suatu pemrograman yang berorientasi objek.”

Nugroho (2005:6) menjelaskan, “*OOP* 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 pengertian tersebut, dapat disimpulkan bahwa “*OOP* adalah suatu metode pemrograman yang berorientasi kepada objek yang digunakan untuk mempermudah pengembangan program.

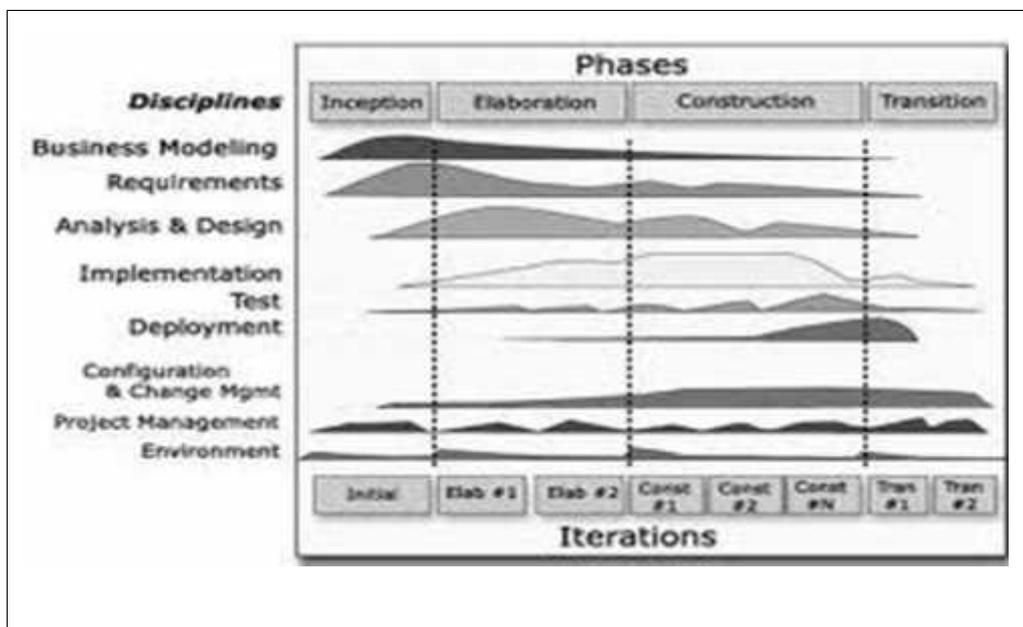
2.2.2. Pengertian Metode RUP (*Rational Unified Process*)

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 utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak. Gambar dibawah 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:



1. **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*.
2. **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* (RUP)

Pada penggunaan kedua standard tersebut diatas yang berorientasi objek (*object oriented*) memiliki manfaat yakni:



1. *Improve productivity*

Standard ini dapat memanfaatkan kembali komponen-komponen yang telah tersedia/dibuat sehingga dapat meningkatkan produktifitas

2. *Deliver high quality system*

Kualitas sistem informasi dapat ditingkatkan sebagai sistem yang dibuat pada komponen-komponen yang telah teruji (*well-tested* dan *well-proven*) sehingga dapat mempercepat *delivery* sistem informasi yang dibuat dengan kualitas yang tinggi.

3. *Lower maintenance cost*

Standar ini dapat membantu untuk menyakinkan dampak perubahan yang terlokalisasi dan masalah dapat dengan mudah terdeteksi sehingga hasilnya biaya pemeliharaan dapat dioptimalkan atau lebih rendah dengan pengembangan informasi tanpa standard yang jelas.

4. *Facilitate reuse*

Standard ini memiliki kemampuan yang mengembangkan komponen-komponen yang dapat digunakan kembali untuk pengembangan aplikasi yang lainnya.

5. *Manage complexity*

Standard ini mudah untuk mengatur dan memonitor semua proses dari semua tahapan yang ada sehingga suatu pengembangan sistem informasi yang amat kompleks dapat dilakukan dengan aman dan sesuai dengan harapan semua manajer proyek IT/IS yakni *deliver good quality software within cost and schedule time and the users accepted*.

2.2.2.1. Fase RUP (*Rational Unified Process*)

1. Inception

1.1. Menentukan Ruang lingkup proyek

1.2. Membuat 'Business Case'

1.3. Menjawab pertanyaan "apakah yang dikerjakan dapat menciptakan 'good business sense' sehingga proyek dapat dilanjutkan"



2. Elaboration
 - 1.1. Menganalisa berbagai persyaratan dan resiko
 - 1.2. Menetapkan 'base line'
 - 1.3. Merencanakan fase berikutnya yaitu construction
 3. Construction
 - 1.1. Melakukan sederetan iterasi
 - 1.2. Pada setiap iterasi akan melibatkan proses berikut: analisa desain, implementasi dan testing
 4. Transistion
 - 1.1. Membuat apa yang sudah dimodelkan menjadi suatu produk jadi
 - 1.2. Dalam fase ini dilakukan:
 - 1.2.1. Beta dan performance testing
 - 1.2.2. Membuat dokumentasi tambahan seperti; training, user guides dan sales kit.
 - 1.2.3. Membuat rencana peluncuran produk ke komunitas pengguna
- (Stefan,2011:01)

2.2.3. Pengertian UML (*Unified Modelling Language*)

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

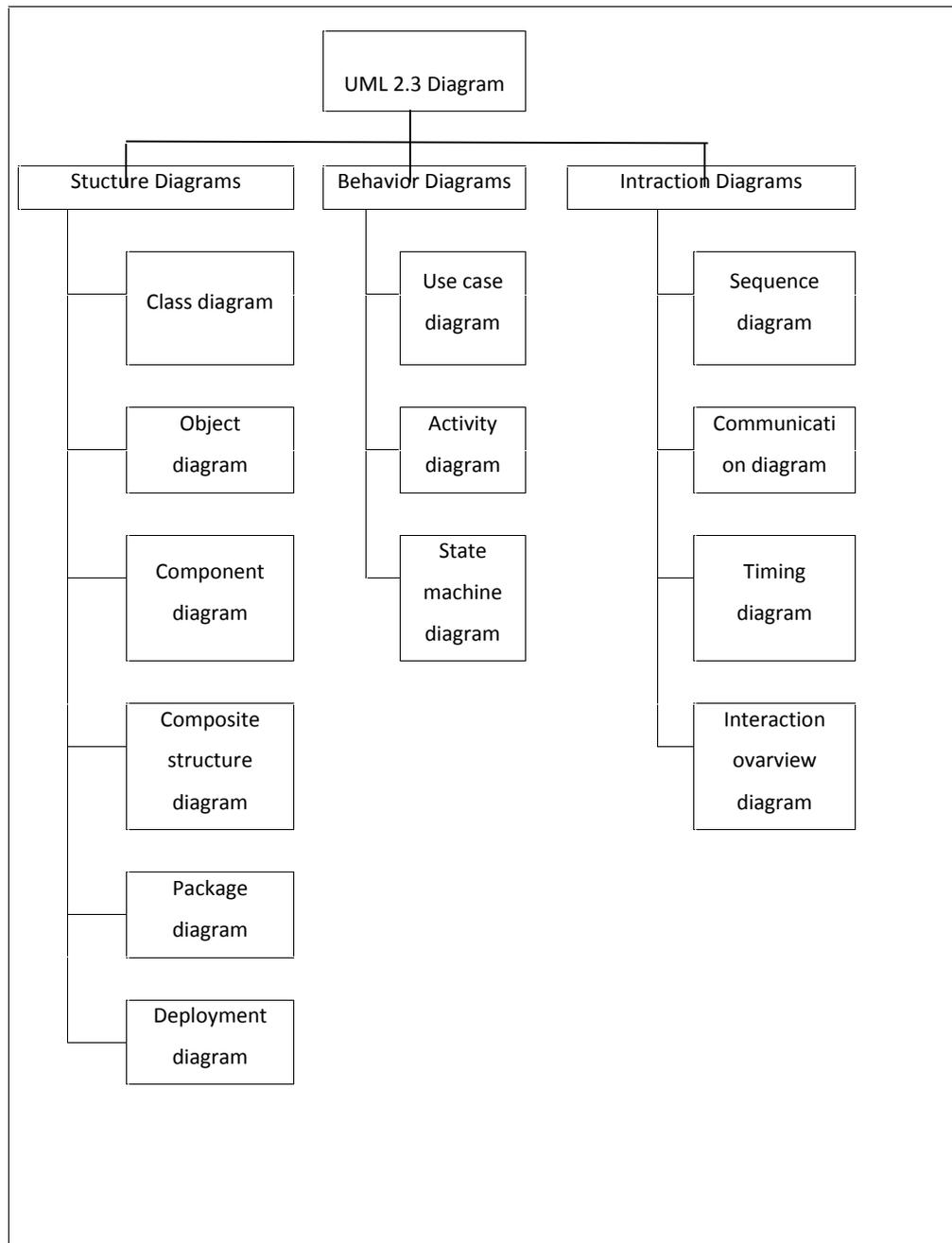
Widodo dan Herlawati (2011:6) menjelaskan, “UML (*Unified Modeling Language*) adalah bahasa pemodelan standar yang berarti UML memiliki sintaks dan semantik. Membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Misalnya elemen pada model – model yang kita buat berhubungan satu dengan lainnya harus mengikuti standar yang ada.

Dari pengertian tersebut, dapat disimpulkan bahwa “UML (*Unified Modeling Language*) adalah bahasa pemodelan standar yang digunakan untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”



2.2.4. Diagram pada UML (*Unified Modelling Language*)

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.1.



Gambar 2.1 Diagram UML



Berikut penjelasan singkat dari pembagian kategori tersebut :

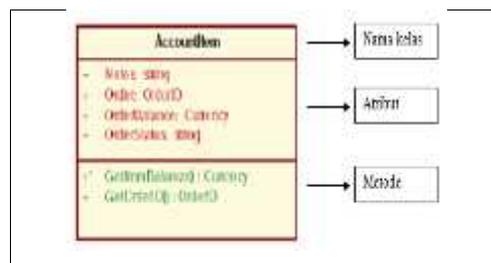
1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behaviour diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.2.5. Pengertian *Class Diagram*

Sukamto dan Shalahuddin (2013:141) menjelaskan, “*Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem.

Class memiliki 3 area pokok :

1. Nama kelas
2. Atribut
3. Metode



Gambar 2.2 *Class Diagram*

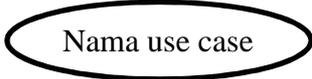
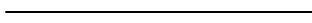
2.2.6. Pengertian *Use Case Diagram*

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



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

Tabel 2.1 Use case diagram

No	Simbol	Deskripsi
1	<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .
2	<p>Aktor / <i>actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar 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 diawal frase nama aktor.
3	<p>Asosiasi / <i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	<p>Ekstensi / <i>extend</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip inheritance pada pemrograman berorientasi objek; biasanya tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan
5	<p>Generalisasi / <i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum – khusus) antar dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	<p><i>Include</i></p>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

(Sumber : Sukanto dan Shalahuddin)



2.2.7. Pengertian Activity Diagram

Sukanto dan Shalahuddin (2013:161) menjelaskan, “*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak, yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol – simbol yang ada pada diagram aktivitas :

Tabel 2.2 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	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber : Sukanto dan Shalahuddin)

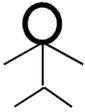
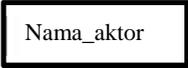
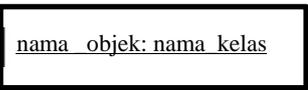
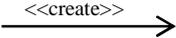
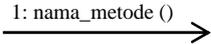
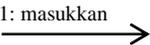
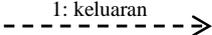
2.2.8. Pengertian Sequence Diagram

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



Berikut adalah simbol – simbol yang ada pada diagram sekuen :

Tabel 2.3 *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 itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal 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 lain, arah panah mengarah pada objek yang dibuat.
6	<p>Pesan tipe call</p>  <p>1: nama_metode ()</p>	Menyatakan suatu obejek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
7	<p>Pesan tipe send</p>  <p>1: masukkan</p>	Menyatakan bahwa suatu objek mengirimkan data / masukan/ informasi ke objek lainnya, arah panah mengara pada objek yang dikirim.
8	<p>Pesan tipe return</p>  <p>1: keluaran</p>	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.



No	Simbol	Deskripsi
9		Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

(Sumber : Sukanto dan Shalahuddin)

2.3. Teori Program

2.3.1. Pemrograman Java

2.3.1.1. Pengertian Pemrograman Java

Siallagan (2009:13) menjelaskan, “Java adalah bahasa Pemrograman Berorientasi Objek (PBO) atau *Object Oriented Programming (OOP)*.”

Westriningsih (2012:2) menjelaskan, “Java adalah suatu jenis teknologi pemrograman yang dikembangkan oleh Sun Microsystem.

Dari pengertian tersebut, dapat disimpulkan bahwa “java merupakan bahasa pemrograman berorientasi objek yang bersifat netral, tidak bergantung pada suatu platform.”

2.3.1.2. Pengelompokkan Tipe Data dalam Java

Westriningsih (2012:4), Bahasa pemrograman java menggunakan tipe data yang tetap sehingga dalam menulis kode program harus mendeklarasikan variabel terlebih dahulu. Pemrograman java memiliki delapan tipe data yaitu :

1. Byte : tipe data yang dapat menampung data berukuran 8 bit di dalam memori dengan jangkauan antara -128 sampai dengan 127, biasanya digunakan untuk mewakili bilangan bulat.
2. Short : tipe data yang dapat menampung data berukuran 16 bit di dalam memori dengan jangkauan antara -32.768 sampai dengan 32.767, biasanya digunakan untuk mewakili bilangan bulat.



3. Int : tipe data yang dapat menampung data berukuran 32 bit di dalam memori dengan jangkauan antara -2.147.483.364 sampai dengan 2.147.483.647, biasanya digunakan untuk mewakili bilangan bulat.
4. Long : tipe data yang dapat menampung data berukuran 64 bit di dalam memori dengan jangkauan antara -9.223.372.036.854.775.808 sampai dengan 9.223.372.036.854.775.807, biasanya digunakan untuk mewakili bilangan bulat.
5. Float : tipe data yang dapat menampung data berukuran 32 bit IEEE di dalam memori, biasanya digunakan untuk mewakili bilangan pecahan.
6. Double : tipe data yang dapat menampung data berukuran 64 bit di dalam memori dengan jangkauan antara -2.147.483.364 sampai dengan 2.147.483.647, biasanya digunakan untuk mewakili bilangan bulat.
7. Boolean : tipe data yang digunakan yang hanya memiliki dua buah kemungkinan yaitu **True** atau **False**.
8. Char : tipe data yang dapat menampung data berukuran 16 bit yang digunakan untuk mewakili simbol pada himpunan karakter.

2.3.1.3. Mendeklarasikan Variabel

Berikut ini bentuk umum cara mendeklarasikan variabel di dalam bahasa

Java:

Tipe namaVariabel;

Tipe namaVariabel1, namaVariabel2;

Contoh:

```
int nilai1;
```

```
//deklarasi beberapa variabel
```

```
Tipe_data nama_variabel1, nama_variabel2,...;
```



2.3.1.4. Operator dalam Java

1. Operator Sederhana

Operator sederhana merupakan operator penugasan berupa tanda sama dengan (=). Hal ini berfungsi untuk memberikan nilai pada variabel. Contoh operator sederhana : `int nilai1=1;`

2. Operator Aritmatika

Operator aritmetika adalah operator yang berfungsi mengerjakan komputasi numerik.

Tabel 2.4 Operator Aritmetika

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

(Sumber : Westriningsih)

3. Operator Unary

Operator unary merupakan operator yang berfungsi mengerjakan beberapa operator matematis seperti ekspresi bilangan negatif.

Tabel 2.5 Operator Unary

Operator	Kegunaan
+	Mendeklarasikan nilai bilangan positif
-	Mendeklarasikan nilai bilangan negatif
++	Menambah nilai variabel dengan 1
--	Mengurangi nilai variabel dengan 1
!	Mendeklarasikan kebalikan dari kemungkinan

(Sumber : Westriningsih)

4. Operator Relasional

Operator relasional merupakan operator yang berfungsi membandingkan operand-operand secara kuantitatif.

**Tabel 2.6** *Operator Relasional*

Operator	Keterangan
==	Sama dengan
!=	Tidak sama dengan
>	Lebih besar
<	Lebih kecil
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

(Sumber : Westriningsih)

5. Operator Kondisi

Operator kondisi merupakan operator yang berfungsi membandingkan operator boolean.

Tabel 2.7 *Operator Logika*

Operator	Keterangan
&&	Operasi AND
	Operasi OR

(Sumber : Westriningsih)

6. Operator Bit Shift an Bitwise

Operator bit shift dan bitwise merupakan operator yang berfungsi untuk operasi bit.

Tabel 2.8 *Operator Bit Sshift dan Bitwise*

Operator	Keterangan
&	Operator bitwise AND
	Operator bitwise OR
^	Operator <i>bitwise</i> XOR
~	Operator <i>bitwise</i> NOT
>>	Operator <i>shift right</i> (geser ke kanan sebanyak <i>n</i> bit)
>>>	Operator <i>shift right zero fill</i>
<<	Operator <i>shift left</i> (geser ke kiri sebanyak <i>n</i> bit)

(Sumber : Westriningsih)

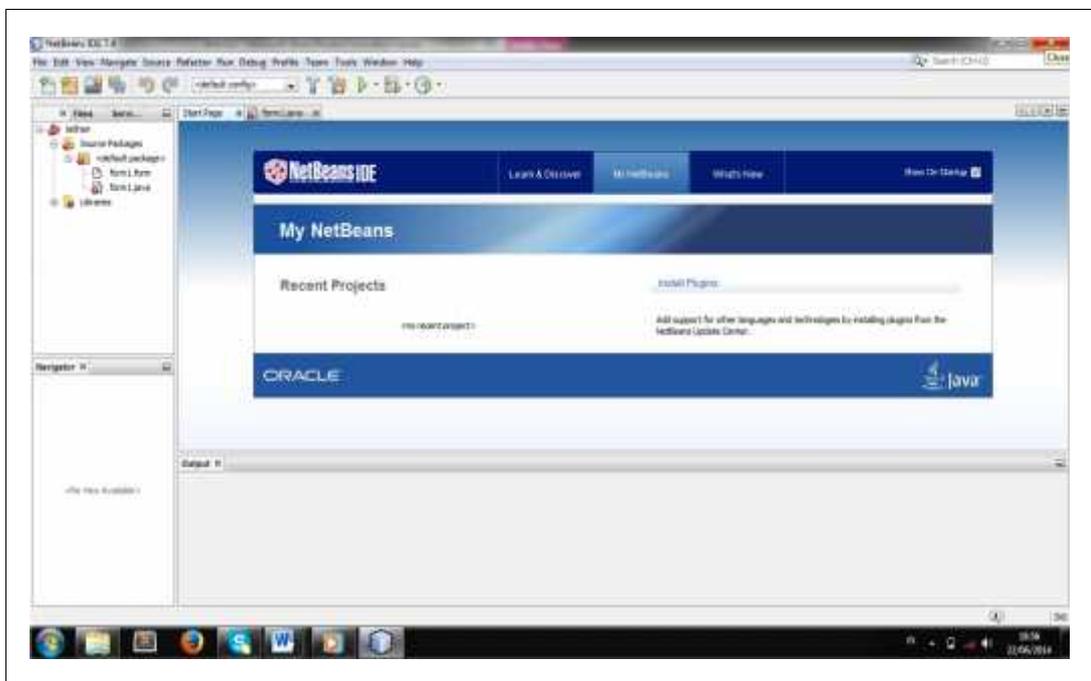


2.3.1.5. Mengenal Netbeans IDE

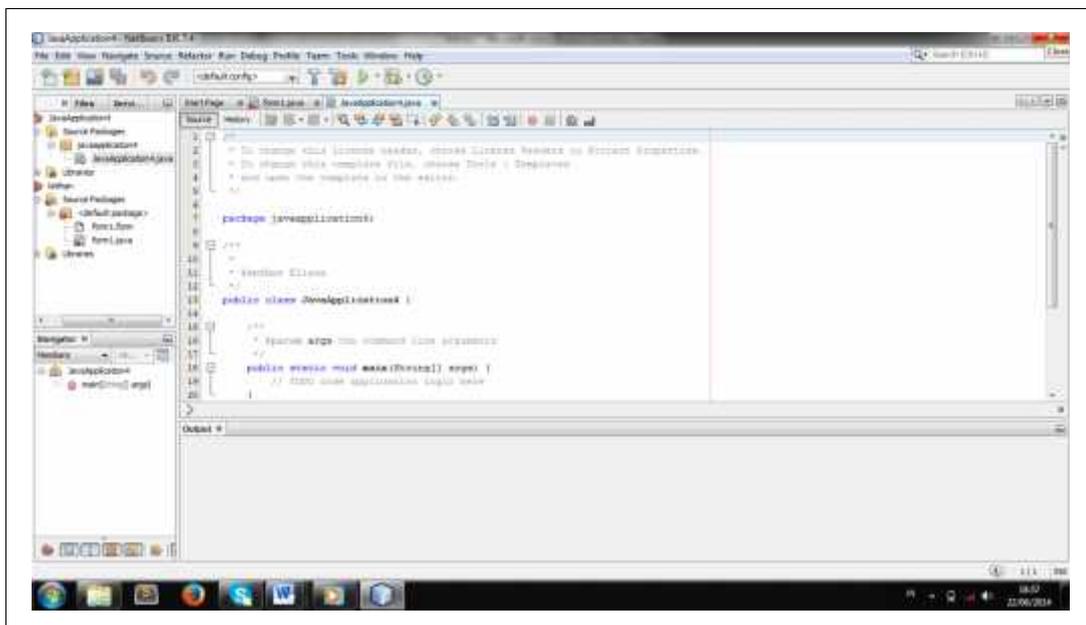
Siallagan (2012:28) menjelaskan, “Netbeans merupakan salah satu IDE yang dikembangkan dengan bahasa pemrograman java. Netbeans adalah sebuah perangkat lunak open source sehingga dapat digunakan secara gratis untuk keperluan komersial maupun nonkomersial yang didukung oleh Sun Microsystem.”

2.3.1.6. Tampilan Netbeans IDE 7.4

Berikut beberapa tampilan dari Netbeans IDE 7.4:



Gambar 2.3 Tampilan Awal Netbenas IDE 7.4



Gambar 2.4 Tampilan Project Editor Netbeans 7.4

2.3.1.7. Fitur-fitur baru Netbeans IDE

Pada Netbeans 7 ini telah dilengkapi dengan fitur-fitur highlight yang lebih sempurna daripada versi-versi sebelumnya. Fitur-fitur tersebut diantaranya adalah :

1. JDK 7

Pada Netbeans 7 telah mendukung penggunaan JDK 7 meskipun Netbeans 7 sendiri dapat digunakan untuk membuat program java dengan minimum JDK versi 6.

2. WebLogic Server

Pada Netbeans 7 deployment dan streamline ke weblogic lebih cepat selain itu pada Netbeans versi ini JSF telah terintegrasi dengan server library sehingga lebih praktis untuk digunakan.

3. Oracle Database

Pada Netbeans 7 telah dilengkapi dengan fasilitas wizard connection untuk oracle serta panduan instalasi JDBC driver.



4. GlassFish

Pada Netbeans 7 ini telah mendukung GlassFish 3.2 serta dilengkapi dengan fasilitas restart dan log viewer.

5. Java

Pada Netbeans 7 ini juga mendukung Maven 3 yang sudah terintegrasi dengan Junit 4.8.2 dan dilengkapi dengan visual costumizer untuk GridBagLayout.

6. Java EE

Java EE merupakan platform yang digunakan untuk membangun aplikasi bisnis pada perusahaan besar seperti sistem basisdata terdistribusi, e-commerce, ataupun management transaksi. Komponen dasar dari fitur ini adalah Enterprise JavaBeans (EJB) dan Java Persistence API (JPA).

7. Bahasa Pemrograman Web

Pada Netbeans 7 telah dilengkapi dengan aplikasi java web, aplikasi java web ini digunakan untuk membangkitkan halaman web interaktif dari berbagai bahasa pemrograman web seperti html, xml, ataupun javascript.

8. PHP

Pada Netbeans 7 ini telah dilengkapi dengan beberapa fitur yang dapat digunakan untuk membangun aplikasi berbasis bahasa pemrograman PHP.

9. C/C++

Pada Netbeans 7 ini telah dilengkapi dengan beberapa fitur yang dapat digunakan untuk membangun aplikasi berbasis bahasa pemrograman C/C++.

2.3.2. Pengertian Basis Data (*Database*)

Priyadi (2014:2) menjelaskan, “Basis Data adalah sekumpulan fakta berupa representasi tabel yang saling berhubungan dan disimpan dalam media penyimpanan secara digital.”

Sukanto dan Shalahuddin (2013:43) menjelaskan, “Basis Data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.



Pada intinya basis data merupakan media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.”

Dari pengertian tersebut, dapat disimpulkan bahwa “basis data adalah media untuk menyimpan sekumpulan data yang sudah diolah dan membuat informasi tersedia saat dibutuhkan.”

2.3.3. Pengertian Tabel

Priyadi (2014:3) menjelaskan, “Tabel direpresentasikan menjadi suatu bentuk segiempat berupa matriks, yang terdiri dari kolom dan baris.”

Heryanto (2012:3) menjelaskan, “Tabel adalah kumpulan dari kolom atau *field* dan baris atau *record*.”

Dari pengertian diatas, dapat disimpulkan bahwa “tabel adalah kumpulan dari kolom dan baris.”

Berikut adalah tampilan dari tabel :

id	nama	username	password	bidang	date_created	date_updated
5	678591089	yenni	e10adc3949ba59abbe56e05720883e	Kepegawaian	2014-05-26	2014-05-26
8	4980092	nika	e10adc3949ba59abbe56e05720883e	Keuangan	2014-05-26	2014-05-26
9	123456789	eka	fc9a9207412e5aa7be0c42b8c93759	Umum	2014-05-26	2014-05-26
10	49013038	syawal	25d55aed283aa400af464c76d713c07ad	Personal	2014-05-26	2014-05-26

Gambar 2.5 Tampilan Tabel

2.3.4. Pengertian *Field* (Kolom)

Priyadi (2014:3) menjelaskan, “Kolom merupakan representasi untuk sebuah nama *field* yang pembacaan datanya dilakukan secara vertikal.”

Heryanto (2012:3) menjelaskan, “*Field* atau kolom merupakan atribut dari tabel.”

Dari pengertian tersebut, dapat disimpulkan bahwa “kolom atau *field* adalah atribut dari tabel yang pembacaan datanya dilakukan secara vertikal.”

Berikut adalah tampilan dari *field* :



id	nama	username	password	bidang	date_created	date_updated
5	578591089	yenni	e10adc3949ba59abbe56e05722bf683e	Kepegawaian	2014-05-21	2014-05-26
6	4800092	nika	e10adc3949ba59abbe56e05722bf683e	Keuangan	2014-05-26	2014-05-26
9	123456789	eka	fbaa2077412b5da7be0c422bdc93759	Umum	2014-05-26	2014-05-26
10	48013038	ajawali	25e55ad283aa400af464c76d713cd7ad	Personal	2014-05-26	2014-05-26

Gambar 2.6 Tampilan Field

2.3.5. Pengertian Record (Baris)

Priyadi (2014:4) menjelaskan, “Baris merupakan representasi untuk sebuah *record* yang pembacaan datanya dilakukan secara horizontal. Satu baris pada sebuah tabel merupakan data yang dimiliki oleh satu *record*. Nilai-nilai yang dimiliki oleh sebuah *record* merupakan gabungan dari semua *field* yang terdapat dalam tabek tersebut.”

Heryanto (2012:3) menjelaskan, “*Record* atau baris merupakan isi dari tabel.”

Dari pengertian tersebut, dapat disimpulkan bahwa “baris atau *record* adalah isi dari tabel yang pembacaan datanya dilakukan secara horizontal.”

Berikut adalah tampilan dari *record* :

id	nama	username	password	bidang	date_created	date_updated
5	578591089	yenni	e10adc3949ba59abbe56e05722bf683e	Kepegawaian	2014-05-21	2014-05-26
6	4800092	nika	e10adc3949ba59abbe56e05722bf683e	Keuangan	2014-05-26	2014-05-26
9	123456789	eka	fbaa2077412b5da7be0c422bdc93759	Umum	2014-05-26	2014-05-26
10	48013038	ajawali	25e55ad283aa400af464c76d713cd7ad	Personal	2014-05-26	2014-05-26

Gambar 2.7 Tampilan Record

2.3.6. Pengertian MySQL

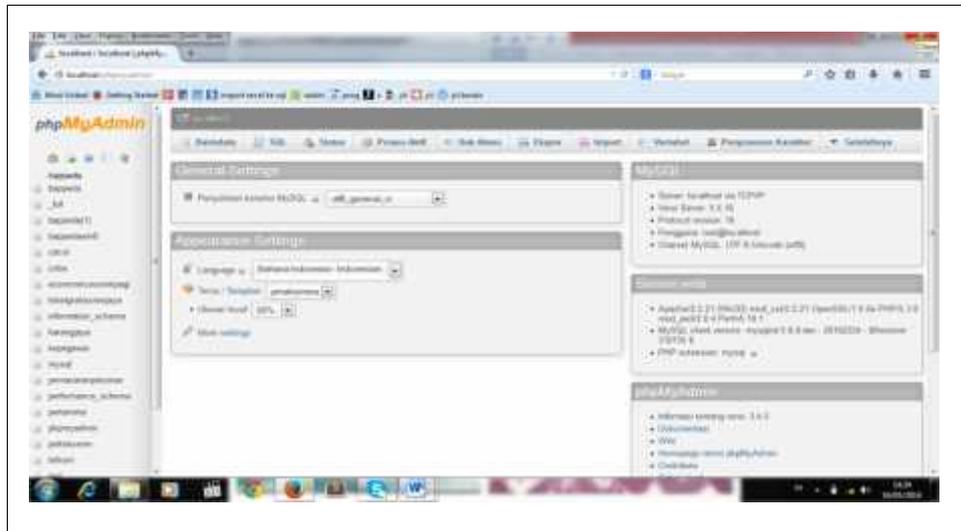
Badiyanto (2013:57) menjelaskan, “*MySQL* merupakan sebuah *database* server *SQL multiuser* dan *multi threaded*.”



Saputra (2013:14) menjelaskan, “*MySQL* merupakan *database storage engine* yang paling banyak digunakan oleh *web developer* karena sifatnya yang *free*, alias gratis.”

Kadir (2008:348) menjelaskan, “*MySQL* salah satu jenis *database server* yang sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *databasenya*. Selain itu, ia bersifat open source (Anda tidak perlu membayar untuk menggunakannya) pada berbagai platform (kecuali untuk jenis Enterprise, yang bersifat komersial.”

Dari pengertian tersebut, dapat disimpulkan bahwa “*MySQL* adalah sebuah *database server SQL multiuser* dan *multi threaded* yang sifatnya gratis.”



Gambar 2.8 Tampilan *MySQL*