



---

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Aplikasi

Asropudin (2013:6) mengemukakan bahwa, “*Application* adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Ms-Word*, *Ms-Excel*” Adapun pendapat lain dari Sutabri (2012:147), “Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya”.

Dari kedua pendapat di atas, maka dapat disimpulkan bahwa aplikasi adalah suatu perangkat lunak komputer yang memanfaatkan kemampuan komputer untuk melakukan suatu tugas yang diinginkan pengguna.

##### 2.1.2 Pengertian Peta

Menurut Kamus Besar Bahasa Indonesia, “peta merupakan gambar atau lukisan pada kertas dan sebagainya yang menunjukkan letak tanah, laut, sungai, gunung, dan sebagainya representasi melalui gambar dr suatu daerah yg menyatakan sifat seperti batas daerah, sifat permukaan dan denah”. Adapun pendapat lain dari Prahasta (2013:2) menyatakan bahwa “peta dapat diartikan sebagai sebuah (dokumen resmi mengenai) bentuk sajian (presentasi) atau gambaran (miniatur) mengenai unsur-unsur spasial (*features*) yang (pada umumnya) terdapat di permukaan bumi pada sebuah (media) bidang datar (atau yang telah “didatarkan”)”.

Dari kedua pendapat di atas, maka dapat disimpulkan bahwa peta adalah gambaran permukaan bumi pada bidang datar dengan skala tertentu melalui suatu sistem proyeksi.

##### 2.1.3 Pengertian Pemetaan

Menurut Kamus Besar Bahasa Indonesia, “pemetaan merupakan proses, cara, perbuatan membuat peta”. Adapun pendapat lain dari Mutia dan Firdaus



(2011:42) “Pemetaan dapat didefinisikan sebagai proses pengukuran, perhitungan dan penggambaran obyek-obyek di permukaan bumi dengan menggunakan cara dan atau metode tertentu sehingga didapatkan hasil berupa peta”.

Dari kedua pendapat di atas, maka dapat disimpulkan bahwa pemetaan adalah suatu proses untuk menghasilkan suatu peta.

#### **2.1.4 Pengertian Informasi**

Menurut McLeod dalam Yakub (2012:8), “Informasi (*Information*) adalah data yang diolah menjadi bentuk lebih berguna dan lebih berarti bagi yang menerimanya. Informasi juga disebut data yang diproses atau data yang memiliki arti”. Adapun pendapat lain dari Sutabri (2012:22) menyatakan bahwa, “informasi adalah data yang telah diklasifikasi atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan”.

Dari kedua pendapat di atas, maka dapat disimpulkan bahwa informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti sehingga dapat digunakan dalam proses pengambilan keputusan.

#### **2.1.5 Pengertian Puskesmas**

Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 75 Tahun 2014 Tentang Pusat Kesehatan Masyarakat Bab I Ketentuan Umum Pasal 1 menyatakan tentang Puskesmas sebagai berikut:

Pusat Kesehatan Masyarakat yang selanjutnya disebut puskesmas adalah fasilitas pelayanan kesehatan yang menyelenggarakan upaya kesehatan masyarakat dan upaya kesehatan perseorangan tingkat pertama, dengan lebih mengutamakan upaya promotif dan preventif, untuk mencapai derajat kesehatan masyarakat yang setinggi-tingginya di wilayah kerjanya.

#### **2.1.6 Pengertian Aplikasi Pemetaan dan Informasi Pusat Kesehatan Masyarakat (Puskesmas) Berbasis *Android* pada Dinas Kesehatan Kota Palembang**

Aplikasi Pemetaan dan Informasi Pusat Kesehatan Masyarakat (Puskesmas) Berbasis *Android* pada Dinas Kesehatan Kota Palembang merupakan aplikasi



yang dapat digunakan untuk mendapatkan informasi tentang Pusat Kesehatan Masyarakat (Puskesmas) yang ada di Kota Palembang sehingga dengan adanya aplikasi pemetaan dan informasi tersebut masyarakat dapat mencari lokasi puskesmas dan juga Dinas Kesehatan dapat menyebarkan informasi puskesmas yang ada di Kota Palembang dengan menggunakan sistem operasi *android*.

## 2.2 Teori Khusus

### 2.2.1 Metode *Rational Unified Process (RUP)*

Menurut Sukamto dan Salahuddin (2016:125) menjelaskan tentang *RUP (Rational Unified Process)* sebagai berikut:

*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*), *RUP* merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well structured*).

#### 2.2.1.1 Fase *Rational Unified Process (RUP)*

Menurut Sukamto dan Salahuddin (2016:128—131) menjelaskan tentang fase *RUP (Rational Unified Process)* sebagai berikut:

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

1. *Inception* (permulaan)  
Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).
2. *Elaboration* (perluasan/ perencanaan)  
Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).
3. *Construction* (konstruksi)  
Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari



*initial operation capability milestone* atau batas/tonggak kemampuan operasional awal.

#### 4. *Transition* (transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *initial operation capability milestone* atau batas/tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

### 2.2.2 *Unified Modelling Language (UML)*

Menurut Sukamto dan Shalahuddin (2016:133), “*UML (Unified Modelling Language)* adalah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”. Adapun pendapat lain dari Asropudin (2013:102) menyatakan bahwa “*UML* singkatan dari *Unified Modelling Language* ini adalah bahasa pemrograman yang digunakan untuk perangkat lunak berorientasi objek”.

Dari beberapa definisi di atas penulis menyimpulkan bahwa *UML (Unified Modelling Language)* adalah salah satu alat bantu yang dapat digunakan dalam bahasa pemrograman yang berorientasi objek dalam membuat analisis dan desain.

### 2.2.3 *Klasifikasi Diagram UML (Unified Modelling Language)*

Menurut Sukamto dan Shalahuddin (2016:140), *UML (Unified Modelling Language)* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori.

Berikut ini penjelasan singkat dari pembagian kategori *UML* sebagai berikut:

#### a. *Structure Diagram*

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

#### b. *Behavior Diagram*

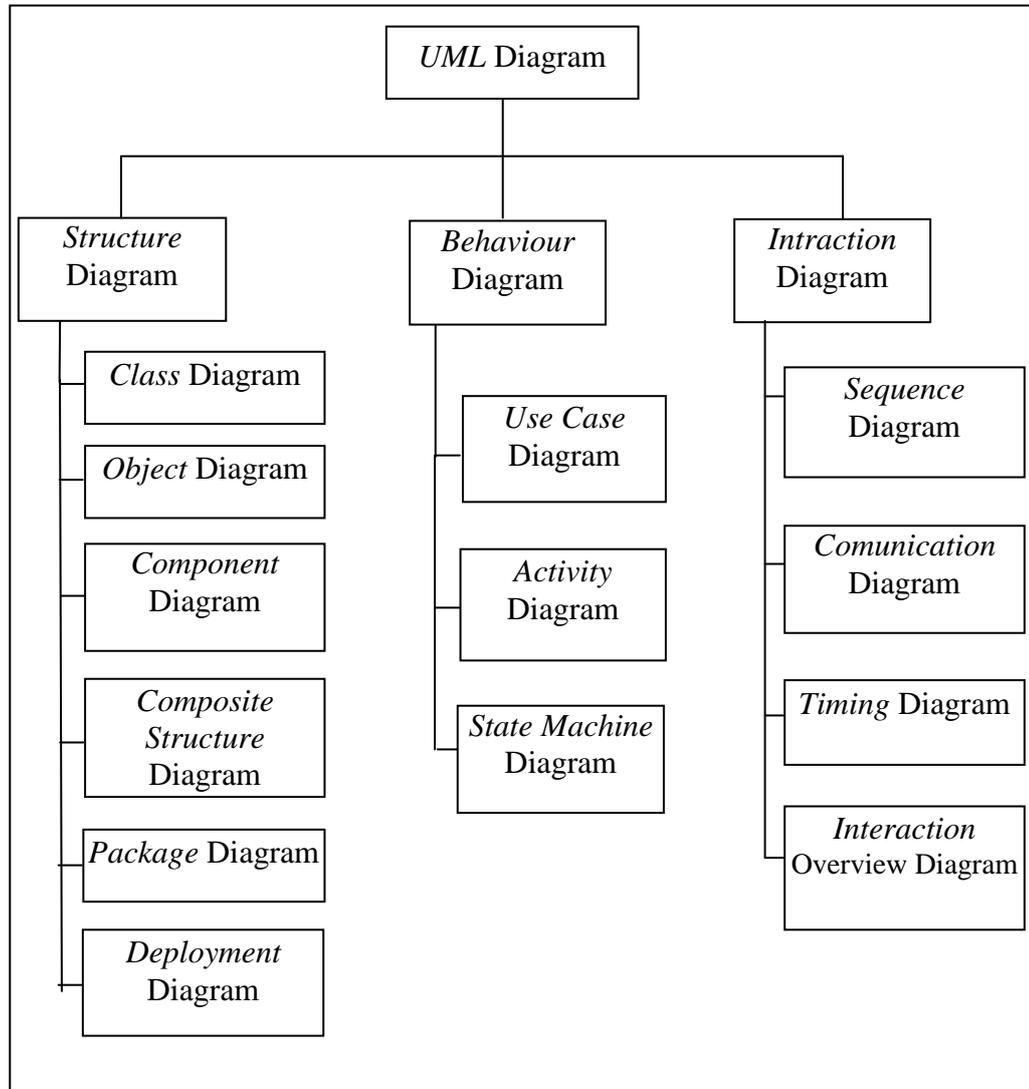
Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

#### c. *Interaction Diagram*

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



Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini



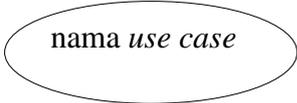
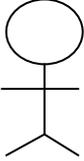
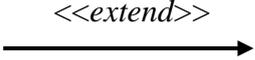
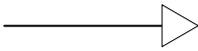
(Sumber: Sukamto dan Shalahuddin (2016:140))

**Gambar 2.1** Kalsifikasi Diagram UML (Unified Modelling Language)

#### 2.2.4 Use Case Diagram

Menurut Sukamto dan Shalahuddin (2016:155), *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. Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

Tabel 2.1 Simbol-simbol *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 di awal frase nama <i>use case</i>
2.	<p>Aktor / <i>actor</i></p>  <p>nama aktor nama_ <i>interface</i></p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
3.	<p>Asosiasi / <i>association</i></p> 	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case yang memiliki interaksi dengan aktor
4.	<p>Ekstensi / <i>extend</i></p> 	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.	<p>Generalisasi/<i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya

Lanjutan Tabel 2.1 Simbol-simbol *Use Case Diagram*

No	Simbol	Deskripsi
6.	<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p style="text-align: center;"> </p>	<p>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>.</p>

(Sumber: Sukamto dan Shalahuddin (2016:156-158))

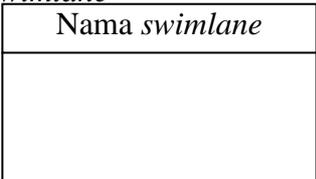
### 2.2.5 Activity Diagram

Menurut Sukamto dan Shalahuddin (2016:161), “*Activity Diagram* adalah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis yang ada pada perangkat lunak”.

Tabel 2.2 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.	<p>Status Awal</p> <p style="text-align: center;"> </p>	<p>Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.</p>
2.	<p>Aktivitas</p> <p style="text-align: center;"> </p>	<p>Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.</p>
3.	<p>Percabangan / <i>decision</i></p> <p style="text-align: center;"> </p>	<p>Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.</p>
4.	<p>Penggabungan / <i>join</i></p> <p style="text-align: center;"> </p>	<p>Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.</p>

Lanjutan Tabel 2.2 Simbol-simbol *Activity Diagram*

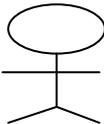
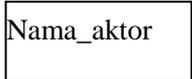
No	Simbol	Deskripsi
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	<i>Swimlane</i> 	<i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

(Sumber: Sukamto dan Shalahuddin (2016:162-163))

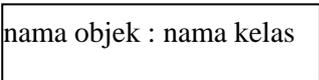
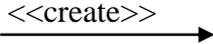
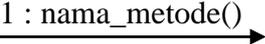
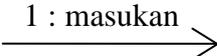
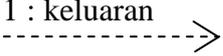
### 2.2.6 Sequence Diagram

Menurut Sukamto dan Shalahuddin (2016:165), “*Sequence diagram* adalah diagram yang menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu daur hidup objek dan message yang dikirimkan dan diterima antar objek”.

Tabel 2.3 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	Aktor  Aktor Atau  tanpa waktu aktif	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>

Lanjutan Tabel 2.3 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
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 <i>create</i> 	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 <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	Pesan tipe <i>return</i> 	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.

Lanjutan Tabel 2.3 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
9.	<p>Pesan tipe <i>destrory</i></p>	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> .

(Sumber: Sukamto dan Shalahuddin (2016:165-167))

### 2.2.7 Pengujian Perangkat Lunak

Menurut Sukamto dan Shalahuddin (2016:272) menjelaskan tentang pengujian perangkat lunak sebagai berikut:

Pengujian adalah satu *set* aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desai kasus uji yang spesifik dan metode pengujian. Secara umum pola pengujian perangkat lunak adalah sebagai berikut:

- Pengujian dimulai dari level komponen hingga integrasi antar komponen menjadi sebuah sistem.
- Teknik pengujian berbeda-beda sesuai dengan berbagai isi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan.
- Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (*Independent Test Group (ITG)*).
- Pengujian dan penirkutuan (*debugging*) merupakan aktivitas yang berbeda tetapi penirkutuan (*debugging*) harus diakomodasikan pada berbagai strategi pengujian.

#### 2.2.7.1 Back-Box Testing (Pengujian Kotak Hitam)

Menurut Sukamto dan Shalahuddin (2016:275) menjelaskan tentang *Black-box testing* sebagai berikut:

*Black-box testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan



dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji coba yang dilakukan adalah:

- a. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
- b. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalkan nama pemakai benar tapi kata sandi salah, atau sebaliknya atau keduanya salah.

## 2.3 Teori Program

### 2.3.1 Pengertian Basis Data (*Database*)

Menurut Sujatmiko (2012:40) mengemukakan bahwa, “Basis Data (*Database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data”. Adapun pendapat lain dari Yakub (2012:51) menyatakan bahwa, “Basis Data merupakan kumpulan data yang saling berhubungan (punya relasi). Relasi biasanya ditunjukkan dengan kunci (*key*) dari tiap file yang ada”.

Berdasarkan beberapa pendapat di atas dapat disimpulkan bahwa basis data merupakan sekumpulan data yang sudah disusun sedemikian rupa dengan ketentuan atau aturan tertentu yang saling berelasi sehingga memudahkan pengguna dalam mengelolanya juga memudahkan memperoleh informasi.

### 2.3.2 Pengertian *MySQL*

Menurut Nugroho (2013:94) menyatakan bahwa “*MySQL* adalah *software* atau program *database server*”. Adapun menurut buku terbitan Madcoms dengan judul *Adobe Dreamweaver CS6 dan PHP-MySql untuk Pemula* (2012:282) menyatakan bahwa “*MySQL* adalah salah satu *database* untuk *server* yang menggunakan *SQL* yang bersifat *free* (gratis atau tidak perlu membayar untuk menggunakannya), selain itu dapat berjalan di berbagai *platform* antara lain *Linux* dan *Windows*”



Berdasarkan kedua pendapat di atas dapat disimpulkan bahwa *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL*.

### 2.3.3 Sekilas Tentang *Android*

Dalam buku terbitan Wahana Komputer dengan judul *Step by Step menjadi Programmer Android* (20013:2) menjelaskan tentang *android* sebagai berikut:

Pertama kali *android* dikembangkan oleh perusahaan *Android.Inc*. Nama perusahaan inilah yang pada akhirnya digunakan sebagai nama proyek sistem operasi *mobile* tersebut, yaitu sistem operasi *Android*. Pada tahun 2005, sebagai bagian dari strategi untuk memasuki pasar *mobile*, google membeli *android* dan mengambil alih proses pengembangannya sekaligus *team developer Android*. Google menginginkan *android* untuk menjadi sistem operasi *open source* dan gratis, kebanyakan *code Android* dirilis dibawah lisensi *Open Source Apache* yang berarti setiap orang bebas untuk menggunakan dan mengunduh *source code Android* secara penuh. *Android* telah dikembangkan dan di-*update* beberapa kali semenjak dirilis pertamanya.

#### 2.3.3.1 Pengertian *Android*

Menurut Sujatmiko (2012:19) “*Android* adalah sistem operasi berbasis *linux* untuk telepon seluler seperti telepon pintar dan komputer tablet”. Adapun pendapat lain dalam buku terbitan Wahana Komputer dengan judul *Step by Step menjadi Programmer Android* (20013:2), menyatakan bahwa “*Android* adalah sebuah sistem operasi *mobile* yang berbasiskan pada versi modifikasi *linux*”.

Dari beberapa definisi di atas penulis menyimpulkan bahwa *android* adalah suatu sistem operasi yang berjalan pada *smatphone* saat ini yang berbasis *linux*.

#### 2.3.3.2 Versi *Android*

Dalam buku terbitan Wahana Komputer dengan judul *Step by Step menjadi Programmer Android* (20013:3) *android* terus berkembang dan hal itu ditandai dengan versinya sebagai berikut:

**Tabel 2.4** Versi *Android*

Versi	Nama	Tanggal Rilis	Level API
1.0	Tanpa nama	23 September 2009	1
1.1	Tanpa nama	9 Februari 2009	2

Lanjutan Tabel 2.4 Versi *Android*

Versi	Nama	Tanggal Rilis	Level API
1.5	<i>Cupcake</i>	30 April 2009	3
1.6	<i>Donut</i>	15 September 2009	4
2.0	<i>Éclair</i>	26 Oktober 2009	5
2.1	<i>Éclair</i>	12 Januari 2010	7
2.2	<i>Froyo</i>	20 Mei 2010	8
2.3	<i>Gingerbread</i>	6 Desember 2010	10
3.0	<i>Honeycomb</i>	22 Februari 2011	11
3.1	<i>Honeycomb</i>	10 Mei 2011	12
3.2	<i>Honeycomb</i>	15 Juli 2011	13
4.0	<i>Ice Cream Sandiwch</i>	19 Oktober 2011	14 (Versi 4.0.1-4.02) 15 (Versi 4.0.3-4.0.4)
4.1	<i>Jelly bean</i>	9 Juli 2012	16
4.2	<i>Jelly bean</i>	13 November 2012	17

### 2.3.4 Java

Menurut Kadir (2013:6-7) menjelaskan tentang *java* sebagai berikut:

Ekstensi *.java* digunakan pada kode sumber bahasa *java*. Contoh kode sumber bahasa *java* ditunjukkan di bawah ini:

```
package com.example.aplikasipertama;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class MainActivity extends Activity {
    @Override
    protected void onCreate (Bundle savedInstanceState) {
        super.onCreate (savedInstanceState);
        setContentView (R.layout.activity_main);
    }
    @Override
    public boolean onCreateOptionsMenu (Menu menu) {
        getMenuInflater().inflate (R.menu.main, menu);
        return true;
    }
}
```

### 2.3.5 XML

Menurut Kadir (2013:7) menjelaskan tentang *XML* sebagai berikut:

Ekstensi *.xml* digunakan pada data yang menggunakan format *eXtended Markup Language (XML)*. Contoh data dengan format *XML* ditunjukkan di bawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
```



```
<resources>
    <string name="app_name"> Aplikasi Pertamaku </string>
    <string name="action_settings"> Settings </string>
    <string name="hello_world"> Selamat Belajar Pemrograman Android!
</string>
</resources>
```

File .xml diawali dengan baris semacam berikut:

```
<?xml version="1.0" encoding="utf-8"?>
```

Data yang terkandung dalam file .xml sering kali dinyatakan dalam pasangan *tag*. Dalam hal ini, suatu tag (menyatakan suatu elemen) dapat mengandung banyak *tag*. Pada contoh diatas, pasangan tag `<string>` dan `</string>` menyatakan sebuah data. Tiga pasangan `<string>` dan `</string>` terdapat pada pasangan tag `<resource>` dan `</resource>`.

Adakalanya, elemen XML tidak mempunyai pasangan tag penutup. Pada keadaan seperti ni, elemen akan diakhiri dengan `</>`. Pada contoh berikut, `TextView` diakhiri dengan `</>`:

```
<TextView
    android:id="@+id/textViewNegara"
    android:layout_width="@+id/textViewNegara"
    android:layout_width="@+id/textViewNegara"
    android:text="Negara:" />
```

Suatu tag dapat mengandung satu atau beberapa atribut. Pada tag `<TextView>` di atas, `android:id` dan `android:layout_width` adalah nama atribut, sedangkan `"@+id/textViewNegara"` dan `"wrap_content"` adalah nilai atribut.

### 2.3.6 Pengertian PHP

Dalam buku terbitan Madcoms dengan judul *Adobe Dreamweaver CS6 dan PHP-MySQL untuk Pemula* (2012:206), "*PHP (Hypertext Preprocessor)* adalah salah satu bahasa pemrograman yang berjalan dalam sebuah *webserver* dan berfungsi sebagai pengolah data pada sebuah *server*". Adapun menurut Utomo (2014:1), "*PHP* adalah salah satu bahasa pemrograman *web server-side* yang populer dan banyak digunakan sampai saat ini." Adapun pendapat lain dari Nugroho (2013:153) menyatakan bahwa "*PHP* itu adalah bahasa program yang digunakan untuk membuat aplikasi berbasis *web* (*website*, *blog*, atau aplikasi *web*)".

Dengan demikian berdasarkan beberapa pendapat di atas dapat disimpulkan bahwa *PHP* adalah bahasa pemrograman *script server side* yang didesain untuk pengembangan *web*.



### **2.3.7 Pengertian *Dreamweaver***

Dalam buku terbitan Wahana Komputer dengan judul *Sistem Informasi Penjualan Online untuk Tugas Akhir* (2014:52) “*Adobe Dreamweaver* merupakan suatu program aplikasi yang digunakan untuk membuat dan membangun sebuah website baik secara grafis maupun dengan menuliskan kode sumber secara langsung.”

### **2.3.8 Pengertian *XAMPP***

Menurut Nugroho (2013:1), “*XAMPP* adalah paket program *web* lengkap yang dapat anda pakai untuk belajar pemrograman *web*, khususnya *PHP* dan *MySQL*”.