BAB 2
LANDASAN TEORI

2.1 Perangkat Lunak

Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumen perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sebuah program komputer tenpa tersosiasi dengan dokumentasinya maka belum dapat, disebut perangkat lunak (*software*). Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak. Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.

Sistem perangkat lunak berarti sebuah sistem yang memiliki komponen berupa perangkat lunak yang memiliki hubungan satu sama lain untuk memenuhi kebutuhan pelanggan (*customer*). Pelanggan (*customer*) adalah orang atau organisasi yang memesan atau membeli perangkat lunak (*software*) dari pengembang perangkat lunak atau bisa dianggap bahwa pelanggan (*customer*) adalah orang atau organisasi yang sukarela mengeluarkan utnuk memesan, atau membeli perangkat lunak. *User* atau pemakai perangkat lunak adalah orang yang memiliki kepentingan untuk memakai atau menggunakan perangkat lunak untuk memudahkan pekerjaannya. [3]

Secara umum perangkat lunak ini dapat diklasifikasikan menjadi 2 bagian [4] yaitu:

1. Perangkat Lunak Sistem Operasi

   Perangkat lunak sistem operasi merupakan perangkat lunak yang berfungsi untuk mengkonfigurasikan komputer agar dapat menerima berbagai perintah dasar yang diberikan sebagai masukan. Perangkat lunak sistem juga merupakan sekumpulan program yang ditulis untuk melayani program-program yang lain. Contohnya: *MS-DOS, LINUX, UNIX, FREE, BSD, OS/2, SUN, OS (JAVA), WINDOWS, MACINTOSH, NOVELL,* dan lain sebagainya.

2. Perangkat Lunak Aplikasi

Perangkat lunak aplikasi merupakan program siap pakai yang digunakan untuk aplikasi dibidang tertentu. Perangkat lunak aplikasi yang membantu *user* sehingga dapat bekerja lebih efektif, dan efisien. Aplikasi ini dibagi atas beberapa bisang, antara lain:

a. *Business & Office Application*

Aplikasi perkantoran dipergunakan untuk menyelesaikan pekerjaan kantor seperti mengetik (*Word processing, document managament*) mambuat tabel kerja (*Worksheet*), membuat Database sederhana (*MS. Access*) ataupun mengolah image/citra sederhana. Contohnya aplikasi perkantoran yang terpaket komplit: *Microsoft Office,Koffice, Star Office, Abiword, Open Office, MS Office for MAC, dan lain sebagainya.*

Aplikasi bisnis lainnya juga meliputi aplikasi pengolahan data keuangan, misalnya (*Personal Finance Application & Business Acounting Aplication*). Contohnya: Aplikasi perhitungan beraca, rugi laba perusahaan, Aplikasi perhitungan pajak personal & perusahaan, Aplikasi manajemen proyekk, aplikasi persediaan stok barang dan penjualan, dan lai sebagainya.

b. *Database Application*

Aplikasi yang digunakan dalam pengolahan data baik yang berukuran kecil maupun besar bisa digunakan secara *stand-alone* (tunggal) maupun dalam sisitem berbasis jaringan lokal *client server* maupun *webbase* (intranet maupun internet). Contohnya: *MS SQL, FoxBase, Oracle, MySQL, PostgresSQL, SQL Server, dan lain sebagainya.*

c. *Grafics Design Application*

Aplikasi yang digunakan untuk membuat sketsadua dimensi untuk desain ruangan, perabotan, mesin-mesin model industri, juga model tubuh manusia, hewan, dan lain sebagainya. Contohnya: *AutoCad, Pro Desain, CorelDraw, Adobw Ilistrator,* dan lain sebagainya.

Ada juga yang memasukan dalam golongan: aplikasi *image, audio, video manipulator*, yaitu aplikasi yang digunakn untuk mengolah citra, suara hingga gambar 3D untuk pembuatan animasi, pengolahan *image, photo, audio, video editing,* dan lain sebagainya. Cotohnya: *Adobe (Photoshop, After Effect, Premiere, & Audition), Corel (Draw, RAVE, Photo Paint), Ulead Media Studio, GIMP (In Linux), Cool Edit Pro, Sound Force, Mecromedia (Flash, Director), 3D Studio Max,* dan lain sebagainya.

d. *Antivirus & Utility Tools*

Software ini digunakan untuk operasi minor (*Background*) yang bekerja di belakang sistem, berguna untuk meningkatkan kinerja sistem operasi atau aplikasi dan juga performa *hardware.* Terdapat 2 jenis utilitas didalam sistem operassi:

1. *Utilitas Internal,* sudah termasuk dalam paket yang di instalasi dalam sistem operasi seperti *scandisk* di *Windows* atau perintah *fsck (file system check)* di OS Linux.
2. *Utilitas eksternal,* yang harus diinstalasi kemudian pada OS, seperti *System Optimizer PC Tools, Data/File Recovery, PC Maintenance, Internet Utilities, Norton*

System Work, Registry Clean & Repair, Hardware Repair.

Sementara untuk proteksi virus dan antivirus, contohnya adalah antvirus produk *Mc Affee,Grisoft, Norton, PandaSoft,* dan lain sebagainya.

e. *Development Tools*

Development Tools merupakan sistem bahasa pemograman komputer yang dilengkapi dengan *compiler* tertentu sebagai sumber media penyusunan program aplikasi. Contohnya: *QuickBasic, Clipper, Visual Basic, Turbo Pascal, Visual C++, GCC, Turbo Assembler, Phyton, KDEvelopt, Gdekstp,* dan lain sebagainya.

Termasuk dalam jenis aplikasi ini adalah *web development tools,* yaitu aplikasi yang dgunakan untuk membuat/mendesain tampilan website (*hompage*) di internet, seperti *Front Page, Macromedia Dreamweaver, PHPedit, WebPage Editor,* dan lain sebagainya.

f. *Communication Application*

Commuication Application merupakan aplikasi yang digunakan untuk membangun komunikasi antara satu komputer dengan komputer yang lain dalam jaringan komputer lokal maupun global. Contohnya: *E-mail& Fax Application, Remote Access, Facebook chat, YM, GTalk, MSN, Terminal Emulation, Instant Messaging, Voice Recognition, Web Browser Internet Telephony, dan lain sebagainya.*

g. Jenis Aplikasi lain

1. *Multimedia Player*, seperti: *Winamp, XMMS, Power DVD, JetAudio, Windows Media Player*, dan lain sebagainya.
2. *Hacker Tools*, seperti:*Find Password, Keygen/Crack Software License, Scanning Port*, dan lain sebagainya
3. *Network & Security Application*, seperti: *Firewall, Intruder Detections, Network Monitoring*, dan lain sebagainya.
4. *PC Games Application, dan masih banyak lagi.*

2.2 Basis Data

Sistem basis data adalah sistem telekomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersdia saat dibutuhkan. Pada intinya basis data adalah media untuk penyimpanan data agar dapat diakses dengan mudah dan cepat.

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Management System* (DBMS) [3]. Kebutuhan basis data dalam sistem informasi meliputi masukan, menyimpan, dan mengambil data serta membuat laporan berdasarkan data yang telah disimpan.

2.2.1 *Database Management System (DBMS)*

Database Management System (DBMS) atau dalam bahasa Indonesa sering disebut sebagai Sistem Manajemen Data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal dapat menyediakan fasilitas untuk mengelola akses data, mampu menangani integrasi data, Dam mampu menangani *backup* data.

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi *Relational DBMS* yaitu DMBS

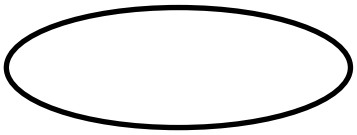
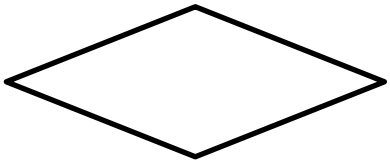
berbasis *relasional model*. *Relasional model* pertama kali dikembangkan oleh Edgar J. Codd pada tahun 1970. Secara sederhana *relasional model* dapat dipahami sebagai suatu mdel yang memandang data sebagai sekumpulan tabel yang saling terkait. Pada tahun 1980an mulai berkembang *Object Oriented DBMS (OODBMS).* OODBMS berkembang seiring dengan perkembangan teknologi pemrograman berorientasi objek. Secara umum dapat diartikan bahwa OODBMS merupakan DBMS yang memandang data sebagai suatu objek.

Terdapat beberapa macam DBMS versi komersial yang paling sering digunakan di dunia saat ini, diantaranya *Oracle, Microsoft SQL Server, IBM DB2*, dan *Microsoft Access.* Sedangkan DBMS versi *open source* yang cukup berkembang dan sering digunakan saat ini adalah diantaranya *MySQL, PostgreSQL, Firebird,* dan *SQLite.* Hampir semua DBMS mengadopsi SQL sebagai bahasa untuk mengelola data pada DBMS [8].

2.2.2 *Entity Relationship Diagram* (ERD)

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagra* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi Chen (dikembangkan oleh Peter Chen), Barker (dikembangkan oleh Richard Barker, ian Palmer, Harry Ellis), notasi Crows Foot, dan beberapa notasi lain, namun yang banyak digunakan adalah notasi dari Chen. Berikut adalah simbol-simbol yang dignakan pada ERD dengan notasi Chen [3].

Tabel 2.1 Simbol-Simbol yang Terdapat Pada ERD

Nama	Simbol	Deskripsi
Entitas/ *Entity*		Entitas merupakan kata inti yang akan disimpan; bakal tabel pada basis data
Atribut		*Field* atau kolom data yang butuh disimpan dalam suatu entitas
Relasi		Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi/ *Association*		Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian.

2.2.2.1 Entitas

Entitas adalah suatu objek (baik nyata maupun abstrak) di dunia nyata yang dapat dibedakan dari objek lain berdasarkan karakteristik yang dimilikinya. Misalnya: pegawai suatu perusahaan, barang yang dijual di suatu toko,

mahasiswa, dosen, mata kuliah, jurusan, dan sebagainya. Dalam basis data, kita harus menentukan entitas apa saja yang informasinya perlu disimpan [18].

2.2.2.2 Atribut

Properti/atribut merupakan suatu informasi yang ingin disimpan dari suatu entitas. Misalnya pada entitas mahasiswa, beberapa atribut yang mungkin adalah NIM, nama mahasiswa, tanggal lahir, alamat, nomor telepon, dan sebagainya. Untuk membedakan antara data pada satu entitas, biasanya adalah suatu atribut digunakan sebagai *identifier* yang dapat membedakan antara satu data dengan data lainnya. Atribut ini disebut *primary ey*. Misalnya pada entitas mahasiswa, atribut NIM digunakan sebagai *primary key*, karena NI merupakan identitas unik dari seorang mahasiswa (tidak ada dua mahasiswa yang memiliki NIM sama).

Pertimbangkan pemilihan atribut mana yang dijadikan sebagai *primary key* tergantung dari beberapa hal, antara lain:

1. Atribut harus unik, itu artinya untuk objek/barang yang berbeda, atribut tersebut pasti beda juga.
2. Nilai atribut tersebut jarang berubah.
3. Mudah diingat dan biasanya memiliki format knsisten

   Berikut adalah contoh penggambaran entitas dengan atributnya. Perhatikan bahwa atribut *primary key* memiliki garis bawah.

Tabel 2.2 Entitas dengan Atributnya

NIM	Nama_Mhs	JK	Alamat_Mhs	No_Telp
10110001	Sonna	P	Jl. Dago 12	081212128899
10110002	Bakana	L	Jl. Gado 35	081312119999
…	…	…	…	…

Pada beberapa kasus, suatu atribut dapat mengacu ke atribut pada tabel lain (atau atribut lain pada tabel itu sendiri). Atribut yang diacu oleh

atribut lain disebut *foreign key*. Foreign key adalah suatu cara untuk menjaga integritas data [18].

2.2.2.3 Relasi

Relasi adalah asosiasi yang menyatakan keterhubungan antarentitas. Misalnya [18]:

Relasi **Mengajar** yang menghubungkan entitas **Dosen** dengan **Mata kuliah.** Relasi ini menyatakan bahwa seorang Dosen hanya boleh mengajar banyak mata kuliah dan satu mata kuliah.

Relasi memliki derajat keterhubungan. Derajat keterhubungan antarentitas pada suatu relasi seperti pada contoh di atasdisebut dengan kardinalitas. Terdapat tiga jenis kardinalitas:

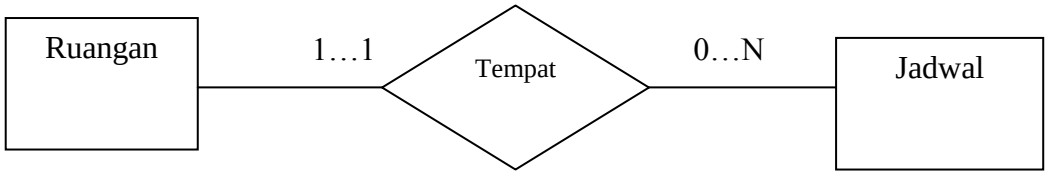
1. Satu ke satu (1-1), misalnya relasi **Registrasi.**

   Contoh gambar:



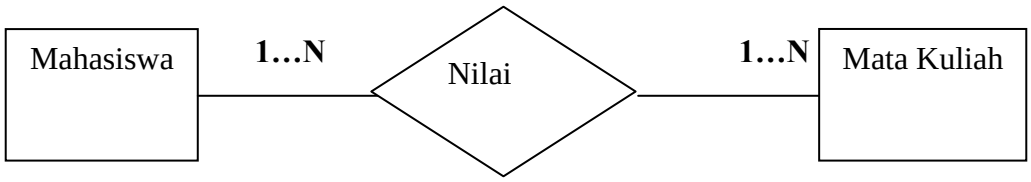
Gambar 2.1 Relasi Registrasi

2. Satu ke banyak / Banyak ke satu (1-N), misalnya relasi **Tempat**.



Gambar 2.2 Relasi Tempat

3. Banyak ke banyak (N-N)



Gambar 2.3 Relasi Nilai

2.2.3 *Structured Query Language (SQL)*

Structured Query Language (SQL) adalah bahasa yang digunakan untuk mengelola data pada DBMS. Meskipun SQL diadopsi da diacu sebagai bahasa standar oleh hampir sebagian besar RDBMS yang beredarsaat ini, tetapi tidak semua stanar yang tercantum dalam SQL diimplementasikan oleh seluruh DBM tersebut. Sehingga terkadang terdapat perbedaan prilaku (hasil yang ditampilkan) oleh DBMS yang berbeda padahal *query* yang dimasukan sama.

Berikut ini adalah contoh pengaksesan data pad DBMS dengan SQL yang secara umum terdiri dari 4 hal sebagai berikut[3]:

1. Memasukan data (*insert*)

```
INSERT INTO
    Tabel_mahasiswa
(nim, nama, tanggal_lahir)
VALUES (‘10115001’, ‘Risa’,
‘1996);
```

Query diatas digunakan untuk memasukan data mahasiswa dengan NIM 10115001 nama Risa dan lahir tanggal 1 januari 1996 ke tabel “Table_mahasisswa”.

2. Mengubah data (*Update*)

```
UPDATE Tabel_mahasiswa
SET tanggal_lahir =’1994-02-04’
WHERE nim = ‘10115045’;
```

Query diatas digunakan untuk mengubah data tanggal lahir mahasiswa dengan NIM 10115045 menjadi 4 Maret 1994 data tabel “Tabel_mahasiswa” .

3. Menghapus data(*delete*)

```
DELETE FROM
Tabel_mahaiswa
WHERE nim = ‘10113189’;
```

Query diatas digunakan untuk menghapus data mahasiswa dengan NIM =10113189 dari “Tabel_mahasiswa”.

4. Menampilkan data (select)

```
SELECT nim,nama
FROM Tabel_mahasiswa
WHERE nim = ‘10114129’;
```

Query diatas digunakan untuk menampilkan data mahasiswa yang tersimpan dalam tabel “Tabel_mahasiswa” dengan NIM 10114129.

2.2.4 MySQL

MySQL adalah salah satu apikasi DBMS yang sudah sangat banyak digunakan oleh para pemrogram aplikasi web. Contoh DBMS lainnya adalah: PostgreSQL (*freeware*), SQL Server, MS Access dari microsoft, DB_2 dari IBM, Oracle dari Oracle Corp, Dbase, FoxPro, dan sebagainya.

Kelebihan dari MySQL adalah gratis, handal, selalu di-*update* dan banya forum yang memfasilitasi para pengguna jika meiliki kendala. MySQL juga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah [18].

1. Tipe Data Numerik

   MySQl menggunakan seluruh tipe data numerik standar ANSI. Berikut ini adalah tipe data numerik yang biasanya digunakan beserta penjelasannya.

Tabel 2.3 Tipe Data Numerik

Tipe Data	Deskripsi
INT	Nilai integer yang bisa bertanda atau tidak. Jika bertanda, maka rentang yang diperbolehkan adalah -2147483648 sampai 2147483647, sedangkan jika tidak bertanda maka rentangnya dari 0 sampai 4294967295
TINYINT	Nilai integer yang sangat kecil. Rentangnya -128 sampai 127 untuk yang bertanda dan 0 sampai 255 untuk yang tidak bertanda
SMALLINT	Nilai integer yang sangat kecil. Rentangnya -31768 sampai 32767 untuk yang bertanda dan 0 sampai 65535 untuk yang tidak bertanda
MEDIUMINT	Nilai integer dengan ukuran sedang. Rentangnya -8388608 sampai 8388607 atau 0 sampai 16777215
BIGINT	Nilai integer dengan ukuran besar. Rentangnya -9223372036854775808 sampai 9223372036854775807 atau 0 sampai 18446744073079551615
FLOAT (M,D)	Bilangan pecahan dengan panjang (termasuk jumlah desimal) M dan jumlah desimal D. Presisi desimalnya bisa sampai 24 digit. Defaultnya float (10,2). Bilangan float selalu bisa bertanda

DOUBLE(M,D)	Bilangan pecahan dengan presisi dua kali lipat, panjang (termasuk jumlah desimal) M dan jumlah desimal D. Presisi desimalnya bisa sampai 53 digit. Defaultnya Double (16,4). Bilangan Double selalu bisa bertanda. Sinonim dari Double adalah Real.
DECIMAL(M,D)	Bilangan pecahan harus didefinisikan M dan D-nya. Setiap desimal membutuhkan tempat 1 byte. Sinonim dari Decimal adalah Numeric.

2. Tipe Data Tanggal dan Waktu

   Berikut adalah tipe data tanggal dan waktu didalam MySQL.

Tabel 2.4 Tipe Data Tanggal dan Waktu

Tipe Data	Deskripsi
DATETIME	Kombinasi tanggal dan waktu dengan format YYYY-MM-DD HH:MM:SS dan rentang data antara 1000-01-01 00:00:00 sampai dengan rentang 9999-12-31 23:59:59. Contoh: pukul 10:00 agi pada tanggal 17 Agustus 1945 akan disimpan sebagai 1945-08-17 10:00:00
TIMESTAMP	Sebuah penanda waktu antara 1 Januari 1970 tengah malam sampai dengantahun 2037. Formatnya mirip dengan DATETIME tetapi tanpa pembatas diantara angkanya. Contoh: pukul 10:00 pagi pada tanggal 17 Agustus 1945 akan disimpan sebagai 19450817100000
TIME	Menyimpan waktu dalam format HH:MM:SS. Contoh pukul 10:00 akan disimpan menjadi 10:00:00
YEAR(M)	Menyimpan data tahun dalam format 2 atau 4 digit. Jika M diisi dengan nilai 2, maka rentang tahunnya dari 1970-2069 sedangkan jika M diisi dengan nilai 4 maka YEAR bisa bernilai 1901 sampai dengan 2155. Default nilai M adalah 4

3. Tipe Data String

Berikut ini adalah tipe data string yang paling umum didalam MySQL.

Tabel 2.5 Tipe Data String

Tipe Data	Deskripsi
CHAR	String dengan ukuran tetap. Ukurannya antara 1 sampai 255 karakter. Ukuran ditentukan dengan nilai M. Contoh: CHAR(6)
VARCHAR	String dengan ukuran bervariasi. Ukurannya antara 1 sampai 255 karakter. Contoh: VARCHAR(25)
TEXT	String dengan ukuran maksimum 65535 karakter. String yang tersimpan didalam TEXT dianggap tidak case sensitive. Untuk kapasitaas yang lebih kecil bisa menggunakan TINYTEXT dengan kapasitas maksimal 255 karaktersedangkan untuk kapasitas yang lebih besar bisa menggunakan MEDIUMTEXT (Maksimal 16777215 karakter) dan LONGTEXT (maksimal 4294967295 karakter)
BLOB	Binary Large Objects (BLOB) adalah tipe data untuk menyimpan data binary dalam jumlah besar. Biasanya digunakan untuk menyimpan citra. Untuk penyimpanan datayang lebih kecil bisa menggunakan TINYBLOB (maksimal 255 karakter) untuk kapasitas yang lebih besar menggunaan MEDIUMBLOB (maksimal 16777215 karakter) dan LONGBLOB (maksimal 4294967295 karakter)
ENUM	Enumerasi atau sebuah list (daftar). Jadi misalnya anda ingin bahwa nilai terbatas hanya boleh dengan nilai tertentu saja maka andabisa membuat sebuah daftar. Misalnya saja nilai itu hanya bisa terdiri dari A-E, maka anda bisa membuatnya menjad ENUM (‘A’,’B’,’C’,’D’,’E’)

2.3 *Object Oriented Programming (OOP)*

Object Oriented Programming atau yang disingkat OOP adalah paradigma atau teknik pemrograman yang yang berorientasi kepada objek. Semua data dan fungsi didalam OOP dibagi dalam kelas-kelas atau objek-objek.

Kenapa mampelajari *Object Oriented Programming* (OOP)? Pada umumnya perusahaan besar saat ini banyak menggunakan teknik pemrograman OOP dalam pembuatan *software*-nya, selain itu juga ketika kita memahami teknik pemrograman dengan OOP maka kita akan lebih mudah mempelajari framework CodeIgniter.

Pada logika pemrograman struktural dimana setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya (contohnya pemrograman normal PHP). Dengan adanya *Object Oriented Programming(OOP)* maka setiap objek berdiri sendiri (*Independen*), karena konsep OOP sendiri adalah semua pemecahan masalah dibagi ke dalam objek. Selain itu data dan fungsi-fungsi yang akan mengoprasikannya digabungan menjadi satu kesatuan yang dapat disebut sebagai objek [18].

2.3.1 Perancangan Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memanang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.

Ada banyak cara untuk mengebstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem. Saat megabstraksikan dan memodelkan objek, data dan proses-proses yang dimilki oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan.

Sistem berorientasi objek merupakan sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi, setiap komponen dalam sistem tersebut dapat mewarisi atribut, sifat, dan komponen lainnya yang dapat berinteraksi atau sama lain.

Terdapat beberapa konsep utama ada metodologi berorientasi objek, diantaranya: [3]

1. *Class* (Kelas)

Kelas merupakan kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dari himpunan objek yang sama yang mungkin lahir dari kelas tersebut. Sebuah kelas akan mempunya sifat (atribut), kelakuan (operasi/metode), hubungan (relationship), dan arti. Suatu kelas dapat diwariskan dari kelas yang lain, dimana atribut dari kelas semula dapat diwariskan ke kelas yang baru.

```
Contoh : [Modifier] class <nama_kelas>
{
    <class body>
}
```

Modifier adalah digunakan untuk menentukan hubungan suatu unsur kelas dengan unsur kelas lainnya. Dan modifier sendiri memiliki beberapa jenis menurut aksesnya, yaitu :

a. Public : semua unsur yang terdapat dalam suatu class (method,object,dll) bisa diakses secara bebas oleh semua class lain yang berada dalam satu package ataupun tidak.
b. Protected : semua unsur yang terdapat dalam suatu class (method,object,dll) bisa diakses secara oleh semua class lain yang berada dalam satu package dan class bagian/turunan dari class awal meski berbeda package.
c. Default : semua unsur yang terdapat dalam suatu class (method,object,dll) bisa diakses secara oleh semua class lain yang berada dalam satu package.
d. Private : semua unsur yang terdapat dalam suatu class (method,object,dll) bisa diakses oleh class itu sendiri.

2. *Object* (Objek)

Objek merupakan abstraksi dari suatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

Untuk membuat objek, menggunakan perintah *new* dengan sebuah nama class yang akan dibuat sebagai instance dari class tersebut.

Contoh:

```
String str = new String();
Random r = new Random();
Pegawai p2 = new pegawai();
Date hari = new Date();
```

3. *Abstraction* (Abstraksi)

Abstraksi atau kelas abstrak merupakan prinsip untuk mempresentasikan dunia nyata yang kompleks menjadi suatu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

Contoh berikut akan menjelaskan penggunaan class abstrak dalam pemrograman Objek dalam Java. Kelas abstrak yang dibuat adalah kelas Animal, kemudian dengan dua jenis variabel, jumlah kaki dan warna. Ditambahkan method berdasarkan cara jalan dan suaranya :

```
public abstract class Hewan {
private int jmlKaki;
private Color warna;
public abstract void jalan();
public abstract void suara();
{
```

4. *Encapsulation* (Enkapsulasi)

   Enkapsulasi merupakan pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek untuk meyembunyikan implementsi dari objek sehingga objek lain tidak mengetahui cara kerjanya.

   Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

5. *Inheritance* (Pewarisan)

   Pewarisan merupakan mekanisme yang memunginkan suatu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagian dari dirinya.

   Contoh: terdapat class sepedadan sepeda gunung. Sepeda termasuk superclass. Sepeda gunung termasuk subclass. Hal ini dikarenakan sepeda gunung memiliki variabel dan method yang dimilki oleh sepeda.

6. *Polymorphism* (Polimorfisme)

   Polimorfisme merupakan kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

   Contoh: terdapat kelas A yang diturunkan menjadi kelas B,C dan D. Dengan konsep polimorfisme, kita dapat menjalankan method-method yang terdapat pada kelas B,C dan D hanya dari objek yang di instansiasi dengan kelas A.

2.4 Unified Modeling Language (UML)

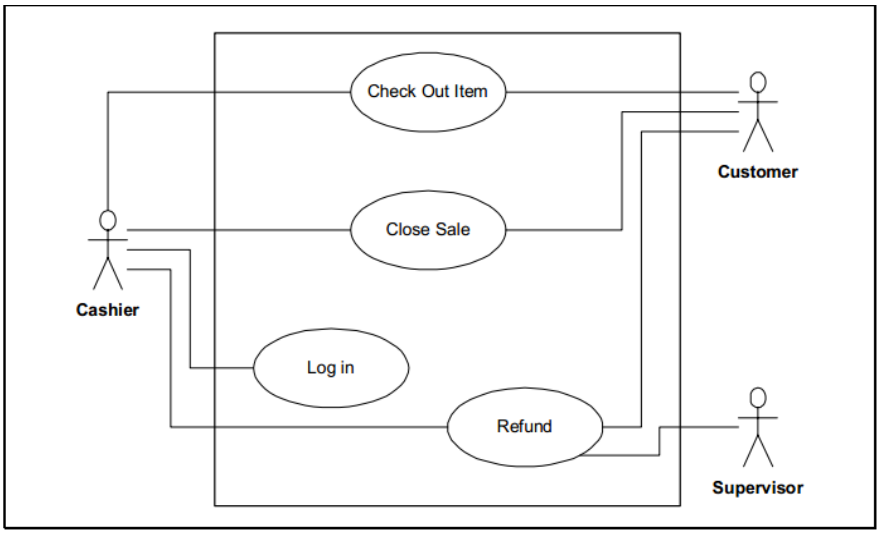
Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasikan, menspesifikasikan, mambangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*) [7]. UML sendiri juga memberikan standar penulisan sebuah sistem *blueprint*, yang meliputi konsep bisnis proses, penuisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem *software*. *Unified Modeling Language* (UML) juga merupakan sebuah bahasa yang telah menjadi standar dalam industri untu visualisasi, merancang, dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Terdapat beberapa diagram yang umum digunakan dalam pemodelan UML diantaranya: [3]

1. *Use Case Diagram*

   Use Case Diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem yang akan dibuat. *Use Case* mendeskripsiskan sebuah iteraksi antar satu atau lebih aktor dengan sistem dengan sistem yang akan dibuat. Secara kasar *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Berikut adalah contoh gambar *use case diagram*[16]*:*

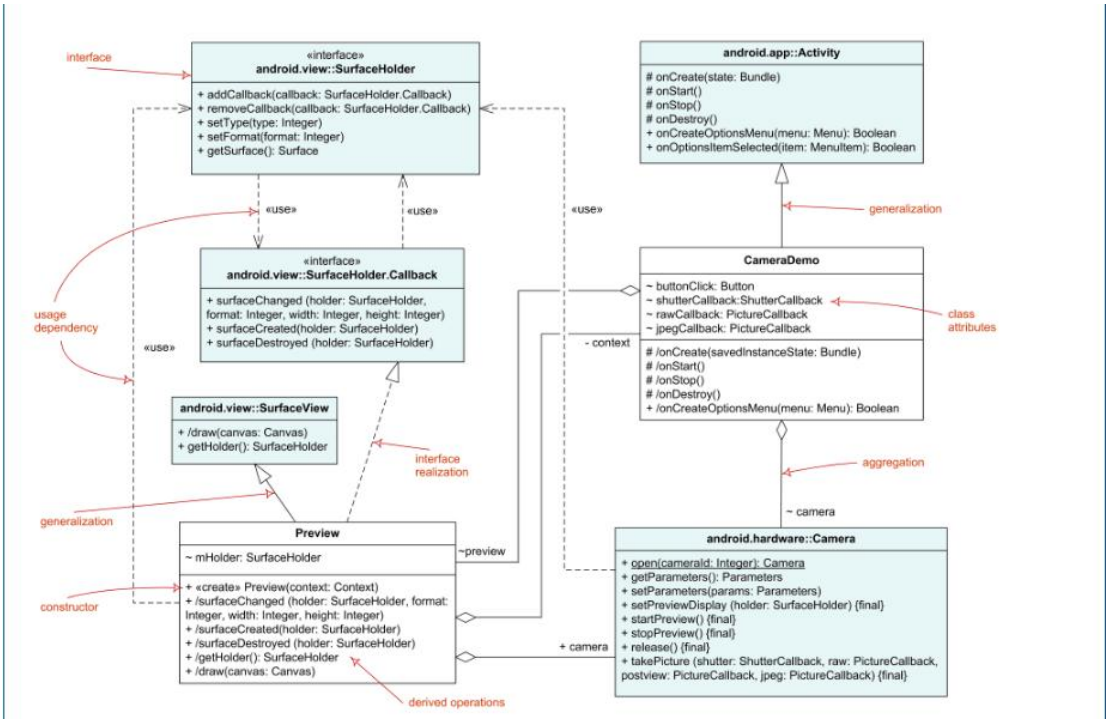


Gambar 2.4 Use Case Diagram

2. *Class Diagram*

Class Diagram digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki atribut dan metode (operasi). Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas. Metode (operasi) fungsi-fungsi yang dimiliki suatu kelas. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki kelas *main,* kelas *view,* kelas *controller,* serta kelas *model*.

Berikut adalah contoh gambar *class diagram*[16]:

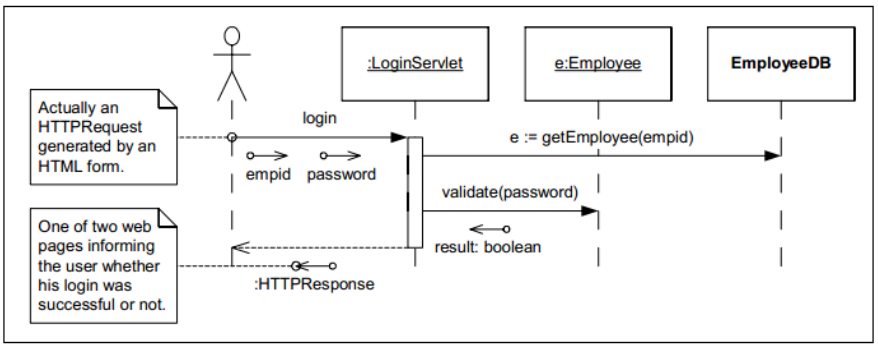


Gambar 2.5 Class Diagram

3. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan kelakukan objek pada *use case* dengan mendeskripsiskan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan sekuen maka harus diketahui objek-objek yang terlihat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case.*

Berikut adalah contoh gambar *sequence diagram* [16]:

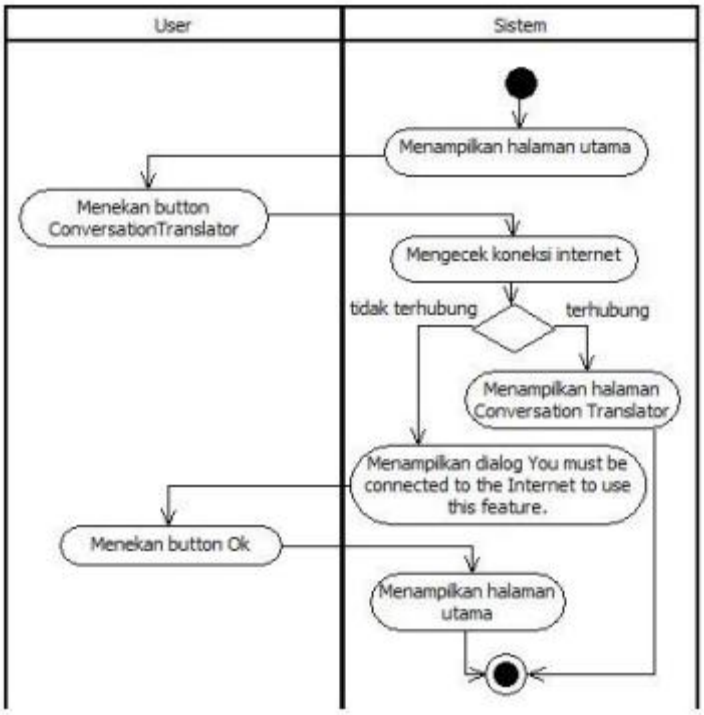


Gambar 2.6 Sequence Diagram

4. *Activity Diagram*

Activity Diagram digunakan untuk menggambarkan airan kerja (*worflow*) 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 dilakuka aktor, melainkan aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah contoh gambar *Activity Diagram* [8]:



Gambar 2.7 Activity Diagram

5. *Component Diagram*

Component Diagram dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah kmponen *user interface* yang menangani fungsi-fungsi proses bisnis, komponen data yang menangani manipulasi data, dan komponen *security* yang menangani keamanan sistem. Komponen lebih terfokus pada penggolongan secara umum fungsi-fungsi yang diperlukan.

2.5 *Image Processing* (Pengolahan Citra)

Pengolahan citra atau *image processing* adalah suatu sistem dimana proses dilakuan dengan masukan (*input*) berupa citra (*image*) dan hasilnya (*output*) juga berupa citra (*image*). Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputer yang memungkinkan manusia apat mengambil nformasi dari suatu citra maka *image processing* tidak dapat dilepaskan dengan bidang *computer vision.* [9]

2.5.1 *Gray-Scale*

Grayscale adalah warna-warna piksel yang berada dalam rentang gradasi warna hitam dan putih. Intensitas *grayscale* biasanya disimpan sebagai data citra 8 bit atau 256 intensitas warna *gray* dari nilai 0 (hitam) dan 255 (putih) [4]. Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian *RED = GREEN = BLUE*. Nilai tersebut digunakan untuk menunjukan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih.

Gray-scale berguna untuk melakukan perubahan suatu gambar full color (RGB) menjadi suatu citra grayscale (gambar keabuan), metode yang umum digunakan, yaitu:

Tabel 2.5 Contoh warna dan kode desimal R,G, B

	Nama Warna	Kode Hexa #RRGGBB	Kode Desimal R,G,B
	maroon	#800000	(128,0,0)
	dark red	#8B0000	(139,0,0)
	brown	#A52A2A	(165,42,42)
	firebrick	#B22222	(178,34,34)
	crimson	#DC143C	(220,20,60)
	red	#FF0000	(255,0,0)
	tomato	#FF6347	(255,99,71)
	coral	#FF7F50	(255,127,80)
	indian red	#CD5C5C	(205,92,92)
	light coral	#F08080	(240,128,128)
	dark salmon	#E9967A	(233,150,122)
	salmon	#FA8072	(250,128,114)
	light salmon	#FFA07A	(255,160,122)
	orange red	#FF4500	(255,69,0)
	dark orange	#FF8C00	(255,140,0)
	orange	#FFA500	(255,165,0)
	gold	#FFD700	(255,215,0)
	dark golden rod	#B8860B	(184,134,11)
	golden rod	#DAA520	(218,165,32)
	pale golden rod	#EEE8AA	(238,232,170)
	dark khaki	#BDB76B	(189,183,107)
	khaki	#F0E68C	(240,230,140)
	olive	#808000	(128,128,0)
	yellow	#FFFF00	(255,255,0)
	yellow green	#9ACD32	(154,205,50)
	dark olive green	#556B2F	(85,107,47)
	olive drab	#6B8E23	(107,142,35)
	lawn green	#7CFC00	(124,252,0)

	chart reuse	#7FFF00	(127,255,0)
	green yellow	#ADFF2F	(173,255,47)
	dark green	#006400	(0,100,0)
	green	#008000	(0,128,0)
	forest green	#228B22	(34,139,34)
	lime	#00FF00	(0,255,0)
	lime green	#32CD32	(50,205,50)
	light green	#90EE90	(144,238,144)
	pale green	#98FB98	(152,251,152)
	dark sea green	#8FBC8F	(143,188,143)
	medium spring green	#00FA9A	(0,250,154)
	spring green	#00FF7F	(0,255,127)
	sea green	#2E8B57	(46,139,87)
	medium aqua marine	#66CDAA	(102,205,170)
	medium sea green	#3CB371	(60,179,113)
	light sea green	#20B2AA	(32,178,170)
	dark slate gray	#2F4F4F	(47,79,79)
	teal	#008080	(0,128,128)
	dark cyan	#008B8B	(0,139,139)
	aqua	#00FFFF	(0,255,255)
	cyan	#00FFFF	(0,255,255)
	light cyan	#E0FFFF	(224,255,255)
	dark turquoise	#00CED1	(0,206,209)
	turquoise	#40E0D0	(64,224,208)
	medium turquoise	#48D1CC	(72,209,204)
	pale turquoise	#AFEEEE	(175,238,238)
	aqua marine	#7FFFD4	(127,255,212)
	powder blue	#B0E0E6	(176,224,230)
	cadet blue	#5F9EA0	(95,158,160)
	steel blue	#4682B4	(70,130,180)
	corn flower blue	#6495ED	(100,149,237)

	deep sky blue	#00BFFF	(0,191,255)
	dodger blue	#1E90FF	(30,144,255)
	light blue	#ADD8E6	(173,216,230)
	sky blue	#87CEEB	(135,206,235)
	light sky blue	#87CEFA	(135,206,250)
	midnight blue	#191970	(25,25,112)
	navy	#000080	(0,0,128)
	dark blue	#00008B	(0,0,139)
	medium blue	#0000CD	(0,0,205)
	blue	#0000FF	(0,0,255)
	royal blue	#4169E1	(65,105,225)
	blue violet	#8A2BE2	(138,43,226)
	indigo	#4B0082	(75,0,130)
	dark slate blue	#483D8B	(72,61,139)
	slate blue	#6A5ACD	(106,90,205)
	medium slate blue	#7B68EE	(123,104,238)
	medium purple	#9370DB	(147,112,219)
	dark magenta	#8B008B	(139,0,139)
	dark violet	#9400D3	(148,0,211)
	dark orchid	#9932CC	(153,50,204)
	medium orchid	#BA55D3	(186,85,211)
	purple	#800080	(128,0,128)
	thistle	#D8BFD8	(216,191,216)
	plum	#DDA0DD	(221,160,221)
	violet	#EE82EE	(238,130,238)
	magenta / fuchsia	#FF00FF	(255,0,255)
	orchid	#DA70D6	(218,112,214)
	medium violet red	#C71585	(199,21,133)
	pale violet red	#DB7093	(219,112,147)
	deep pink	#FF1493	(255,20,147)
	hot pink	#FF69B4	(255,105,180)

	light pink	#FFB6C1	(255,182,193)
	pink	#FFC0CB	(255,192,203)
	antique white	#FAEBD7	(250,235,215)
	beige	#F5F5DC	(245,245,220)
	bisque	#FFE4C4	(255,228,196)
	blanched almond	#FFEBCD	(255,235,205)
	wheat	#F5DEB3	(245,222,179)
	corn silk	#FFF8DC	(255,248,220)
	lemon chiffon	#FFFACD	(255,250,205)
	light golden rod yellow	#FAFAD2	(250,250,210)
	light yellow	#FFFFE0	(255,255,224)
	saddle brown	#8B4513	(139,69,19)
	sienna	#A0522D	(160,82,45)
	chocolate	#D2691E	(210,105,30)
	peru	#CD853F	(205,133,63)
	sandy brown	#F4A460	(244,164,96)
	burly wood	#DEB887	(222,184,135)
	tan	#D2B48C	(210,180,140)
	rosy brown	#BC8F8F	(188,143,143)
	moccasin	#FFE4B5	(255,228,181)
	navajo white	#FFDEAD	(255,222,173)
	peach puff	#FFDAB9	(255,218,185)
	misty rose	#FFE4E1	(255,228,225)
	lavender blush	#FFF0F5	(255,240,245)
	linen	#FAF0E6	(250,240,230)
	old lace	#FDF5E6	(253,245,230)
	papaya whip	#FFEFD5	(255,239,213)
	sea shell	#FFF5EE	(255,245,238)
	mint cream	#F5FFFA	(245,255,250)
	slate gray	#708090	(112,128,144)
	light slate gray	#778899	(119,136,153)

	light steel blue	#B0C4DE	(176,196,222)
	lavender	#E6E6FA	(230,230,250)
	floral white	#FFFAF0	(255,250,240)
	alice blue	#F0F8FF	(240,248,255)
	ghost white	#F8F8FF	(248,248,255)
	honeydew	#F0FFF0	(240,255,240)
	ivory	#FFFFF0	(255,255,240)
	azure	#F0FFFF	(240,255,255)
	snow	#FFFAFA	(255,250,250)
	black	#000000	(0,0,0)
	dim gray / dim grey	#696969	(105,105,105)
	gray / grey	#808080	(128,128,128)
	dark gray / dark grey	#A9A9A9	(169,169,169)
	silver	#C0C0C0	(192,192,192)
	light gray / light grey	#D3D3D3	(211,211,211)
	gainsboro	#DCDCDC	(220,220,220)
	white smoke	#F5F5F5	(245,245,245)
	white	#FFFFFF	(255,255,255)

$$X = 0,2 *R + 0,72*G + 0,07*B \qquad [2.1]$$

dimana :

X : Nilai *Greyscale*

R : Unsur warna merah

G : Unsur warna hijau

B : Unsur warna biru

Nilai yang dihasilkan dari persamaan diatas akan diinput ke masing-masing unsur warna dasar citra grayscale.

2.5.2 *Thresholding*

Secara umum proses thresholding terhadap citra grayscale bertujuan menghasilkan citra biner, secara matematis dapat ditulis sebagai berikut.

$$(x, y, f(x, y)) \rightarrow (x, y, g(x, y))$$

$$g(x, y) = \begin{cases} a_1 \, , f(x, y) < T \\ a_2 \, , \; f(x, y) \geq T \end{cases}$$

[2.2]

Dengan g(x,y) adalah citra biner dari citra grayscale f(x,y), dan T menyatakan nilai

threshold (Purba, D. 2010). Nilai T ditentukan dengan menggunakan metode thresholding global dan thresholding local.

2.5.3 OCR (Optical Character Recognition)

OCR (Optical Character Recognition) adalah aplikasi yang berfungsi untuk men scan gambar pada image dan dijadikan text, dan aplikasi ini juga bisa menjadi support /aplikasi tambahan untuk scanner. Dengan adanya OCR, Image yang bertulisan tangan, tulisan mesin ketik atau computer text, dapat dimanipulasi. Text yang discan dengan OCR dapat dicari kata per kata atau per kalimat. Dan setiap text dapat dimanipulasi, diganti, atau diberikan barcode.

2.5.4 Tesseract Library

Tessaract merupakan sebuah library yang dapat membantu aplikasi dalam mengenali karakter tertentu pada citra digital atau dapat disebut juga Tessaract Optical Characte r Recognition (OCR). Data yang digunakan berupa data latih yang telah disediakan oleh Tessaract OCR. Adaptive Classifier merupakan salah satu metode yang digunakan oleh Tessaract OCR. Adaptive Classifier merupakan bagian dari klasifikasi bentuk yang menggunakan normalisasi isotropic baseline/x-height.

2.6 Aplication Mobile

Aplikasi *mobile* adalah aplikasi yang memungkinkan untuk melakukan mobilitas dengan menggunakan perlengakapan seperti telepon genggam (*hadphone*), PDA, atau *smartphone*. Aplikasi *mobile* dapat mengakses dan menggunakan sebuah aplikasi secara nirkabel dengan menggunakan perangkat *mobile*, dimana data yang diperoleh hanya berupa teks sehingga tidak perlu membutuhan *bandwidth* yang terlalu besar. Data dalam bentuk teks tidak semenarik data dalam bentuk disertai dengan gambar. Namun penggunaan data teks adalah yang paling banyak digunakan saat ini. Penggunaan aplikasi mobile hanya memerlukan handphone yang sudah dilengkapi dengan fasilitas GPRS dan koneksinya [8].

Terdapat beberapa aspek yang harus diperhatikan khususnya perangkat keras dalam membangun sebuah aplikasi *mobile.* Selain itu pertimbangan terhadap keterbatasan pada piranti *mobile* punharus diperhatikan, antara lain:

1. Keterbatasan dari kecepatan *processor* dalam mengeksekusi program.
2. Keterbatasan memori penyimpanan RAM.
3. Perbedaan ukuran layar secara fisik dan resolusi pada masing-masing piranti.
4. Keterbatasan input pada masing –masing piranti *mobile*.
5. Ketahanan baterai yang berbeda pada setia piranti *mobile.*
6. Selain itu dari segi perangkat lunak (*software*), kompatibilitas dan *mobile* pendukung cukup berpengaruh dalam menjalankan sebuah aplikasi *mobile.*

2.7 Pengenalan Android

Android merupakan sisterm operasi *mobile* yang tumbuh di tengah sistem operasi lainnya yang berkembang dewasa ini. Sistem operasi lainya seperti Windows Mobile, i-Phone OS, Symbian dan masih banyak lagi juga menawarkan kekayaan isi dan keoptimalan berjalan di atas perangkat hardware yang ada. Akan tetapi, sistem operasi yang ada ini berjalan dengan memperioritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak

ketiga. Olehkarena itu, adanya keterbatasan distribusi aplikasi pihak ketiga untuk platform mereka. [5]

Android menawarkan sebuah lingkungan berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. *Application Programming Interface* (API) yang disediakan menawarkan akses ke *hardware*, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantinya dengan aplikasi pihak ketiga. [5]

2.7.1 Sejarah Android

A ndroid adalah sebuah sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Andoid menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri.

Awalnya Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti linak untuk ponsel atau *smartphone*. Kemudian untuk mengebangkan Andorid, dibentuklah *Open Handset Alliance*, kosorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. [6]

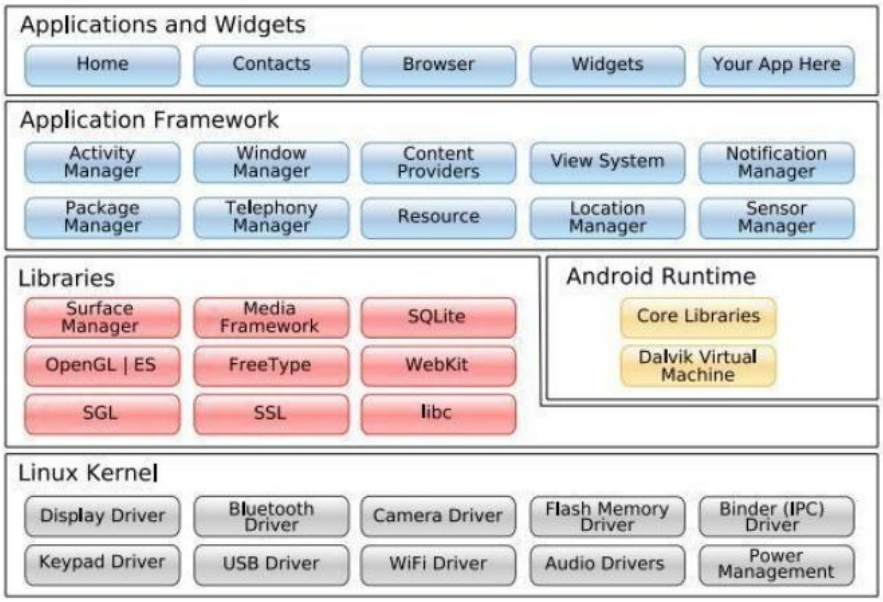
Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

2.7.2 Arsitektur Android

Arsitektur Android dapat digambarkan seperti pada gambar 2.8 dan secara garis besar arsitektur android dapat dijelaskan sebagai berikut:

a. *Application* dan *Widget*

Application dan *Widget* ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita laukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Hampir semua aplikasi ditulis menggunakan bahasa pemograman java. [5]



Gambar 2.8 Arsitektur Android

b. *Application Frameworks*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resource*, menjalankan *service background*, mengatur alarm, dan menambah status *notifications*, dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan (*reuse*). Sehingga bisa kita simpulkan *Application Frameworks* ini adalah layer

dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content providers* yang berupa sms dan panggilan telepon. Komponen-komponen yang termasuk didalam *Aplication Frameworks* adalah sebagai berikut:

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

c. *Libraries*

Libraries ini adalah layer dimana fitur-fitur android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya. Berjalan diatas kernel, laer ini meliputi berbagai library C/C++ inti seperti Libc SSl, serta:

1. *Libraries* media untuk pemutaran media audio dan video.
2. *Libraries* untuk manajemen tampilan.
3. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
4. *Libraries SQLit* untuk dukungan database.
5. *Libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*.
6. *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embedded webview*.
7. Libraries 3D yang mencakup implementasi OpenGL ES 1.0 API’s. [5]

d. *Android Run Time*

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Didalam *Android Run Time* dibagi menjadi dua bagian yaitu:

1. *Core Libraries*: Aplikasi android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan *Virlual Machine Java*, sehingga diperlukan sebuah libraries yang berfungsi untuk menterjemahkan bahasa Java/C yang ditangani oleh *Core Libraries*.
2. *Dalvik Virtual Machine*: Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dimana merupakan pengembangan yang mampu membuat *Linux Kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

e. *Linux Kernel*

Linux Kernel adalah layer dimana inti dari sistem operasi android itu berbeda. Berisi file-file sistem yang mengatur sistem processing memory, resource, drivers, dan sistem-sistem operasi android lainnya. *Linux Kernel* yang digunakan android adalah *Linux Kernel release* 2.6. [5]

2.7.3 Versi Android

Adapun versi android yang pernah dirilis adalah sebagai berikut [6]:

1. Android versi 1.1

   Pada tanggal 9 Maret 2009, Google merilis android versi 1.1 Andrid versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam, alarm, *voice searce* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.5 (*Cupcake*)

   Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan android dan SDK (*Software Development Kit*). Terdapat banyak pembaharuan yakni mampu merekam dan menonton video dengan modus kamera, mengupload video ke youtube dan gambar ke picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung otomatis ke headset Bluetooth.

3. Android versi 1.6 (*Donut*)

   Dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan *control applet VPN*. Fitur lainnya adalah kamera, camcorder, VPN, Gestures, CDMA/EVDO, *Text-to-Speech engine*, resolusi VWGA.

4. Android versi 2.0/2.1 (*Eclair*)

   Pada 3 Desember 2009 kembali diluncurkan ponse android dengan mengoptimalkan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukunga HTML 5, daftar kontak yang baru, dukungan kamera flash, dan Bluetooth 2.1.

5. Android versi 2.2 (*Froyo : Frozen Yoghurt*)

   Pada bulan Mei 2010 Android versi 2.2 diluncurkan. Android yang lebih kompleks dengan penambahan fitur-fitur terbaru seperti *Global Positioning System* (GPS), kompas, kamera dan *Accelerometer*.

6. Android versi 2.3 (*Gingerbread*)

   Pada bulan Desember 2010 Android versi ini diluncurkan. Hal-hal yang direvisi dari versi sebelumnya adalah kemampuan seperti berikut:

   a. *SIP-based VoIP*
   b. *Near Field Communication* (NCF)
   c. *Gyroscope dan sensor*
   d. *Multiple Cameras Support*
   e. *Mixable Audio Effects*
   f. *Download Manager*

7. Android versi 3.0 (*Honeycomb*)

   Android ini dirilis pada bulan Februari 2011 sebagai Android versi 3.0 yang terdapat fitur baru setelah versi sebelumnya.

8. Android versi 4.0 (*Ice Cream Sandwich*)

   Android ini dirilis pada tanggal 19 Oktober 2011, petinggi Google, Gabe Cohen menyatakan bahwa android versi 4. Secara teoritis kompatible dengan perangkat android 2.3x yang diproduksi pada saat itu.

9. Android versi 4.1 (*Jelly Bean*)

   Google mengumumkan Android 4.1 dalam konferensi Google I/O pada tanggal 27 Juni 2012, Jelly Bean adalah pembaharuan penting yang bertujuan meningkatkan fungsi dan kinerja antarmuka pengguna.

10. Android versi 4.4 (*Kitkat*)

    Google mengumumkan Android versi 4.4 Kitkat dinamai dengan izin dari Nestle dan Hearskey pada 3 esember 2013 dengan fitur UI yang lebih menarik.

11. Android versi 5.0 (*Lollipop*)

    Versi ini mempunyai tampilan UI yang lebih menarik, dengan meningkatkan daya hidupberani 30% lebih tahan lam.

2.7.4 *Java Development Kit (JDK)*

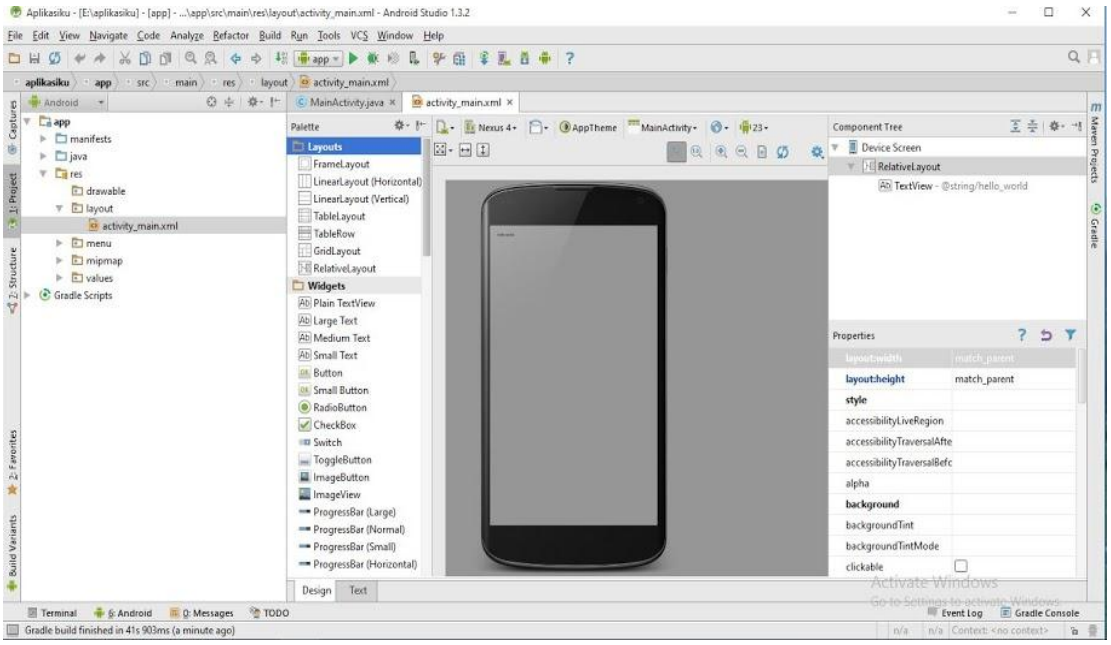
Java Depelopment Kit (JDK) adalah sebuah perangkat peralatan yang digunakan untuk membangun perangkat lunak dengan menggunakan bahasa pemograman Java. JDK berjalan diatas sebuah *virtual machine* yang dinamakan *Java Virtual Machine* (JVM).

Dokumentasi JDK berisi spesifikasi API, deskripsi fitur, panduan pengembang, referensi halaman untuk perkakas JDK dan utilitas, demo, dan *link* ke informasi terkait. Dokumentasi ini juga tersedia dalam bundel yang dapat diunduh dan diinstal pada komputer [10].

2.7.5 *Android Studio*

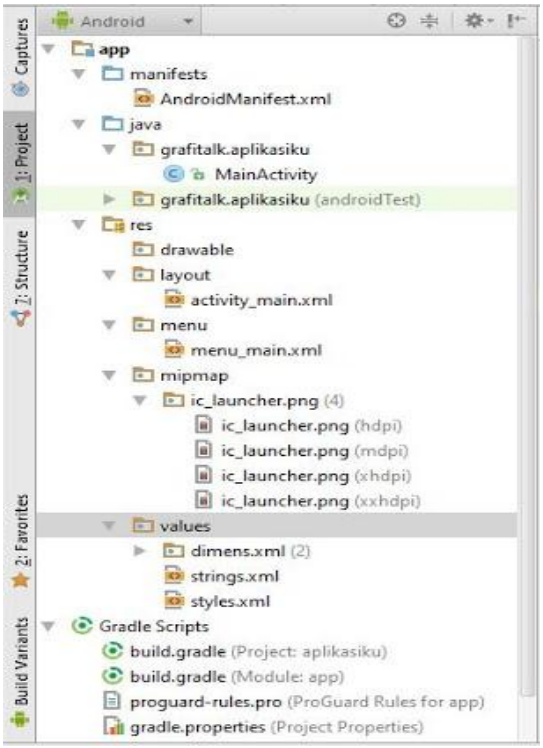
Android studio adalah lingkungan pengembangan terpadu (*IDE)* resmi untuk pengembangan platform android, hal itu disampaikan pada tanggal 16 mei 2013 Google I/O,android studio sudah tersedia secara bebas dibawah lisensi Apache 2.0.

Android studio pada awalnya tahap preview versi 0.1 yang dipakai pada tanggal 1 mei 2013 dan memasuki tahap beta pada bulan juni 2014 dan mulai stabil dirilis pada Desember 2014 dengan versi 1.0 ,berdasarkan jetBrains ‘IDEA Intellij Software, android studio dirancang khusus untuk pengembangan android yang tersedia untuk Windows,Mac OS X dan Linux sebagai pengganti Eclipse.



Gambar 2.9 Tampilan Aplikasi Android Studio

Android studio memiliki struktur menu yang dapat dilihat pada gambar dibawah ini.



Gambar 2.10 Struktur Android Studio

Berikut rincian dari struktur Android Studio :

a. App : berfungsi untuk menyimpan file file pendukung aplikasi yang akan diolah.
   1. *Manifest*
      tempat dimana konfigurasi konfigurasi pendukung project serta izin (permision) untuk mengakses hardware.
   2. Java
      tempat berdiamnya code/scripts java untuk menjalankan aplikasi yang akan dibuat
   3. *Res* :
      a. Drawable : tempat untuk menyimpan gambar atau icon pendukung aplikasi.
      b. *Layout* : tempat berkumpulnya design tampilan yang akan tampak pada tampilan aplikasi anda.
      c. *Menu* : berisi code berupa file xml yang didalamnya berisi array tentang menu yang akan tampil dalam project anda, misalkan file, new,open,close,dsb.
      d. *Mipmap* : tempat berkumpulnya icon launcher (icon utama aplikasi) yang ukurannya menyesuaikan device yang digunakan.
      e. *Values* : untuk menyimpan file .xml yang menggambarkan sumber dari berbagai jenis. Sebagai contoh folder berisi file strings.xml dengan definisi sumber string. Seperti yang akan anda lihat dari menambahkan bagian color , folder tata letak juga dapat berisi, misalnya, deskripsi warna. Di dalam file dimens.xml, terdapat string-string untuk konfigurasi dimensi aplikasi. Tidak jauh berbeda dengan dimens.xml maupun strings.xml, pada file styles.xml terdapat deklarasi variable untuk theme.

b. *Gradle scripts* : berfungsi sebagai configurasi aplikasi anda dengan device, serta menghubungkan library pendukung aplikasi yang anda buat.

2.7.6 *Android Software Development Kit (Android SDK)*

Android SDK adalah *tools* API (*Aplication programming Interface*) yang diperlukan untuk memulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemograman java. Android merupakan *subset* perangkat lunak untu ponsel yang meliputi sistem operasi, *middleware*, dan aplikasi kunci yang di *releas* oleh google. Saat ini disediakan android SDK sebagai alat bantu dan API untuk memulai mengembangkan aplikasi pada *platform* android menggunakan bahasa pemograman java [8].

1. *Framwork,* aplikasi yang mendukung penggantian komponen dan *reusable.*
2. *Virtual Dalvik Machine,* dioptimalkan untuk perangkat *mobile*.
3. *Integrate Browser,* berdasarkan *engine open source WebKit.*
4. *Grafis,* yang dioptimalkan dan didukung oleh *libraries* grafi 2D, grafis 3D berdasarkan spesifikasi OpenGL ES 1.0 (Opsional akselerasi *hardware*).
5. *SQLite,* untuk menyimpan data.
6. *Media Support*, yang mandukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC,AMR, JPG, PNG, GIF), GSM *Telephony* (tergantung *hardware*).
7. *Bluetooth,* EDGE, 3G, dan WiFi (tergantung *hardware*).
8. Camera, GPS, Kompas, dan *accelerometer* (tergantung *hardware*).
9. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat *emulator, tools* untuk *debugging, profil,* dan kinerja memori, dan *plugin* untuk IDE *Eclipse*.

2.7.7 *Android Development Tools (Android ADT)*

ADT adalah *plugin* yang didesain untuk IDE *Eclipse* yang memberikan kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE *Eclipse*. Dengan menggunakan ADT untuk *Eclipse* akan memudahkan dalam membuat aplikasi *project* android, membuat GUI aplikasi, dan menambahkan komponen-komponen lainnya., selain itu dapat digunakan untuk menjalankan

aplikasi menggunakan android SDK melalui *Eclipse*. Dengan ADT dapat melakukan pembuatan *package* android (.apk) yang digunakan dalam distribusi aplikasi android

2.8 Internet

Internet merupakan jaringan global komputer dunia, besar dan luas dimana setiap komputer saling terhubung satu sama lainnya dari negara ke negara lainnya di seluruh dunia dan berisi berbagai macam informasi, mulai dari *teks*, gambar, audio, video, dan lain sebagainya yang terhubung melaluiprotokol TCP/IP. Dalam mengatur integrasi dan komunikasi jaringan komputer ini menggunakan protokol yaitu TCP/IP. TCP (*Transmission Control Protocol*) bertugas untuk memastikan bahwa semua hubungan bekerja dengan benar, sedangkan IP (*Internet Protocol*) yang mentransmisikan data dari atu komputer ke komputer lain. TCP/IP secara umum berfungsi memilih rute terbaik transmisi data, memilh rute alternatif jika suatu rute tidak dapat digunakan, mengatur dan mengirimkan paket-paket pengiriman data [14].

2.8.1 Sejarah internet

Internet dan jaringan komputer adalah hasil evolusi dari *Arpanet*, sejauh proyek riset tingkat tinggi yang dimiliki iloh dDepartemen Pertahanan Amerika Serikat. Darpa mensponsori perkembangan jaringan yang menggunkan *Internet Protokol* (IP), *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP).

Pada mulanya *Aparnet* hanya menghubungkan 4 situs saja yaitu Stanford Research Istitute, University California, Santa Barbara, University of Utah, dimana mereka membentuk satu jaringan terpadu pada tahun 1969, dan secara umum Arpanet diperkenelkan pada bulan Oktober1972.

Tidak lama kemudian proyek ini berkembang pesat di seluruh daerah, dan semua universitas di negara tersebut ingin bergabung, sehingga membuat Aparnet kesulitan untuk mengaturnya. Oleh sebab itu Arpanet dipecah menjadi dua, yaitu “Milnet” untuk keperluan militer, dan “Arpanet” baru yang lebih kecil

untuk keperluan non-militer seperti, universitas-universitas. Gabungan kedua jaringan akhirnyadikenal dengan nama Darpa Internet, yang kemudian disederhanakan menjadi Internet,

Pada tahun 1982 istilah "Internet" pertama kali digunakan, dan TCP/IP diadopsi sebagai protokol universitas untuk jaringan tersebut. Pada tahun 1986 diperkenalkan sistem sistem nama domain, yang sekarang dikenal dengan DNS (*Domain Name System*) yang berfungsi untuk menyeragamkan sistem pemberian nama alamat di jaringan komputer. Singkatnya komputer hanya mengenali IP Address. Dengan DNS anda cukup mengingat www.google.com karena DNS nanti yang akan menerjemahkan menjadi 173.194.38.165 dan kemudian diakses oleh komputer kita [18].

2.9 HTML

HTML (*Hypertext Markup Language)* adalah bahasa *standard* yang digunakan untuk menampilkan halaman *web.* Yang bisa dilakukan dengan HTML yaitu [18]:

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halaman *web*.
3. Mampublikasikan halaman *web* secara *online*.
4. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via *web*.
5. Menambahkan objek-objek seperti citra, audio, video, animasi, java applet dalam halaman *web*.
6. Manampilkan area gambar (canvas) di *browser*.

2.10 PHP

PHP *Hypertext Preprocessor* merupakan sebuah bahasa *scripting* sisi server yang menjadi satu dengan HTML. Sebagian besar sintaknya mirip dengan bahasa C, java, dan perl. Tujuan bahasa ini diciptakan adalah untuk membantu pemrogram web dalam membuat halaman web dinamis.

Banyak tool yang dapat digunakan untuk membuat dokumen PHP, mulai dari *teks editor* biasa, seperti ***Notepad, Wordpad, Notpad ++, Editplus,*** dan lain-lain, sampai aplikasi populer untuk PHP, seperti ***Dreamweaver, PHP Designer***, dan sebagainya [21].

2.10.1 Sejarah PHP

Pada awalnya PHP merupakan pendekatan dari *Peronal Home Page* (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa seumpulan skrip yang digunakan untuk mengolah data formulir dari *web.*

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadikan sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. pada rilis 2.0 ini, *Interpreter* PHP sudah diimplementasikan dalam program C. didalam rilis ini juga ikut disertakan modul-modul ekstensi yag meningkatkan kemampuan PHP/FI secara signifikan.

Kemudian pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang *Interpreter* PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis *interpreter* baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 da singkatan PHP diubah menjadi akronim berulang *PHP: Hypertext Preprocessing.*

Pada pertengahan tahun 1999 Zend merilis *interpreter* PHP baru dan rilis tersebut dikenal dengan PHP 4.0 adalah PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini juga memasukan model pemograman berorientasi objek dalam PHP untuk menjawab perkembangan bahasa pemograman ke arah paraadigma berorientasi objek [18].

2.10.2 Kelebihan PHP

PHP mempunyai beberapa kelebihan, diantaranya sebagai berikut [18]:

1. PHP Berbasis *Server Side Scripting*

   Server Side Scripting pada PHP dapat bekerja jika ada tiga komponen berikut: PHP Parser (CGI atau *server* modul) , *web server* (contohnya Apache dalam XAMP), *web browser*. Hasil *output* PHP yang melewati *web server* dapat dilihat pada *web browser.*

2. *Command Line Scripting* pada PHP

   Kita dapat menggunakan PHP dengan dan tanpa *web server* atau *browser.* Dengan memanfaatkan PHP Parser saja kita sudah bisa menggunakan PHP pada *command line* (misalnya *shell bash* pada Linux, *task scheduler* pada Windows). Cara kerjanya hampir sama *script* PHP tersebuut digunakan untuk memproses sebuah *job* dan *task.*

3. Digunakan untuk Berbagai Macam *Platform* OS

   Contohnya: Linux, Unix (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, dan lain sebagainya.

4. Mendukung Berbagai Macam *Web Server*

   Contohnya: Apache, Microsoft Internet Information Server, Personal Web Server, Netscape andiPlanet servers, Oreilly Website Pro server, Caudium, Xitami OmniHTTPd, dan lain sebagainya.

5. *Object Oriented Programming* atau *Procedural*

   Program PHP dapat kita buat menggunakan teknik program *Procedural*, *Object Oriented Programming* (OOP).

6. *Output File* PHP pada XHTML, HTML, dan XML

   Dengan PHP kita tidak terbatasi oleh *output* HTML saja. PHP mampu menghasilkan gambar sebagai *output, file* bertipe PDF, bahkan Flash (menggunakan libswt dan Ming). Kita bisa juga dapat menerima *output* teks dengan mudah seperti penggunaan *output* pada XHTML, dan file XML lainnya. PHP dapat *autogenerate file-file* tersebut, menyimpan

dalam sistem *file* daripada mencetak, membentuk sebuah *cache server-side* untuk konten dinamis kita.

7. Mendukung Banyak RDMS (*Database*)

Salah satu fitur yang paling signifikan dalam PHP adalah dukungan penggunaan untuk berbagai *database*. Dibawah ini RDMS (*database engine*) yang sudah dapat diintegrasikan dengan PHP.

a. Adabas D Interbase SQL
b. dBase FrontBase SQLite
c. Ratu mSQL Padat
d. FilePro (Read-only) Direct MS-SQL Sybase
e. Hyperwave MySQL Velocis
f. IBM DB2 ODBC Unix dbm
g. Informix Oracle (OCI7 dan oci8)
h. Ingres Ovrimos

8. Mendukung Banyak Komunikasi

PHP juga memiliki dukungan untuk berkomunikasi dengan layanan lain menggunakan protokol seperti LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (pada Windows) dan lain sebagainya. Selain itu kita juga dapat membuka *raw network sockes* dan berinteraksi dengan menggunakan protokol lainnya. PHP juga memilki dukungan untuk pertukaran data WDDX kompleks, tidak semua mendukung namun hampir semua bahasa pemograman Web sudah mendukung pertukaran data ini. Untuk *Interconnection* sendiri, PHP memiliki dukungan untuk menginstansiasi objek PHP.

9. Pengolahan Teks yang Sangat Baik

PHP memiliki fitur pengolahan teks yang sangat baik, dari *POSIX Extended* atau *Perl regular expressions* untuk memparsing

dokumen dari XML. Untuk parsing dan mengakses dokumen XML, PP 4 mendukung standar SAX dan DOM, dan kita juga dapat menggunakan XSLT *extension* untuk mentransformasikan dokuman tersebut pada XML. Sedangkan PHP 5 sendiri merupakan standarisasi semua XML, exrension pada *solid base libxml2* dan memperluas set fitur menambahkan SimpleXML, dan dukungan XMLReader.

2.11 Google API

Google API merupakan bagian dari *framework* Google. Google menyediakan berbagai API (*Application Programming Interface*) yang sangat berguna bagi pengembang apikasi baik web, mobil, maupun *desktop* untuk memanfaatkan berbagai fitu yang disediakan oleg Google seperti: *AdSense, Search Engine, Translation, Youtube,* dan lain sebagainya.

API secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi untuk memanfaatkan fungsi atau fitur yang disediakan Google. Contohnya *developer* dapat menambahkan fitur Google Maps pada aplikasi yang sedang dibangun. Google API dapat dipelajari langsung melalui *Google Code*. Melalui *Google Code, developer* dapat belajar tenteng Google API dan dapat mengimplementasikan pada aplikasi yang sedang dikembangankan. Ada banyak API yang disediakan oleh Google, bebrapa diantaranya [13]:

1. *Language* API: memanfaatkan fitur *translation* yang dimiliki google.
2. *Earth* API: memanfaatkan fitur yang ada pada *Google Earth.*
3. *Maps* API: memanfaatkan fitur yang ada pada *Google Maps.*
4. *Search* API: memanfaatkan fitur pencarian pada *Google Search.*
5. *Visualization* API: membuat grafik maupun *chart* dengan Google API
6. *You Tube* API: memanfaatkan fitur yang ada pada *You Tube* misalnya untuk pencarian video.

2.12 Google Maps

Google Maps adalah layanan pemetaan berbasis web service yang disediakan oleh google dan bersifat gratis, yang memiliki kemampuan terhadap banyak layanan pemetaan berbasis web. Google Maps juga memiliki sifat server side, yaitu peta yang tersimpan pada server Google dapat dimanfaatkan oleh pengguna. Google Maps API adalah suatu library yang berbentuk javascript yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai kebutuhan [12].

2.13 GPS

GPS (Global Position System) adalah suatu sistem navigasi manggunakan lebih dari 24 satelit MEO (Medium Eart Orbit atau Middle Eart Orbit) yang mengelilingi bumi sehingga penerima-penerima sinyal di perukaan bumi dapat menangkap sinyalnya. GPS mengirimkan sinya gelobang mikro ke bumi. Sinyal ini diterima oleh alat penerima di permukaan, dan digunakan untuk menentukan letak, kecepatan, arah, dan waktu. Satelit mengorbit pada ketingian 12.000 mil diatas bumi dan mampu mengelilingi bumi dua kali dalam 24 jam. Satelit GPS secara kontinu mengirimkan sinyal radio digital yang mengandung data lokasi satelit dan waktu, pada penerima yang berhubungan. Satelit GPS dilengkapi dengan jam atom yang mempunyai ketepatan waktu satu per satu juta detik. Berdasarkan informasi ini, stasiun penerima mengetahui berapa lama waktu yang digunakan untuk mengirim sinyal sampai kepada penerima di bumi. Semakin lama waktu yang digunakan untuk sampai ke penerima, berarti semakin jauh posisi satelit dari stasiun penerima [11].