BAB II
TINJAUAN PUSTAKA

Berikut ini adalah beberapa tinjauan pustaka yang menjadi pokok bahasan dalam pembuatan skripsi yaitu sistem operasi Android, *Java Depelopment Kit* (JDK), Android SDK, Android Studio, Gardu Listrik, pemrograman berorientasi objek, pemrograman Java, *Unified Modelling Language* (UML), Basis Data, Oracle, Internet, JSON (*JavaScript Object Notation*) dan Pengujian *Black Box*.

2.1 Sistem Operasi Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware,* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi. Android dipuji sebagai platform mobile pertama yang lengkap, terbuka, dan bebas.

Sejauh ini Android telah melalui cukup banyak pembaruan sejak pertama dirilis. Tabel 2.1 menunjukkan berbagai versi Android beserta dengan tanggal rilis, nama kode dan level API (*Application Programming Interface*)[1].

Tabel 2.1 Versi Android

Versi Android	Tanggal Rilis	Nama Kode	Level API
1.0	23 September 2008	-	1
1.1	9 Februari 2009	-	2
1.5	30 April 2009	Cupcake	3
1.6	15 September 2009	Donut	4
2.0/2.0.1/2.1	26 Oktober 2009	Éclair	5/6/7
2.2	20 Mei 2010	Froyo	8
2.3/2.3.2/2.3.3/2.3.7	6 Desember 2010	Gingerbread	9/10
3.0/3.1/3.2	22 Februari 2011	Honeycomb	11/12/13
4.0/4.0.2/4.03/4.04	19 Oktober 2011	Ice Cream Sandwich	14/15
4.1/4.2/4.3	9 Juli 2012	Jelly Bean	16/17/18
4.4	31 Oktober 2013	Kitkat	19
5.0	3 November 2014	Lolipop	21
6.0	September 2015	Marshmallow	23
7.0	-	Noughat	

Level API menyatakan suatu bilangan unik yang digunakan untuk mengidentifikasi *application programming interface* (API) yang digunakan pada suatu versi Android. Dengan perkataan lain, setiap versi Android ditandai dengan sebuah level API. Peranti yang digunakan untuk mengembangkan aplikasi Android mencakup *Java Development Kit* (JDK), Android SDK dan Android Studio [2].

2.1.1 Java Development Kit

Java Depelopment Kit (JDK) adalah perangkat pengembangan aplikasi Java mutlak diperlukan untuk membuat aplikasi Android, mengingat aplikasi Android itu berbasis Java. Sebagaimana diketahui, Java adalah salah satu bahasa pemrograman yang biasa digunakan untuk membuat aplikasi. Namun perlu diketahui, tidak semua pustaka dalam Java digunakan di Android. Sebagai contoh Android tidak menggunakan *Swing* [2].

2.1.2 Android SDK

Android SDK adalah kumpulan software yang berisi mengenai pustaka, *debugger* (alat pencari kesalahan program), *emulator* (peniru perangkat bergerak), dokumentasi, kode contoh, dan panduan. Keberadaan *emulator* dapat membuat dan menguji aplikasi Android tanpa harus mempunyai perangkat kerasa berbasis Android [2].

2.1.3 Android Studio

Android Studio adalah sebuah IDE (*Integrated Development Environment*) untuk Android *Development* yang diperkenalkan Google pada acara Google I/O 2013. Android Studio merupakan pengembangkan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan *Ant*, Android Studio menggunakan *Gradle* sebagai *build environment*. Fitur-fitur lainnya adalah sebagai berikut [3]:

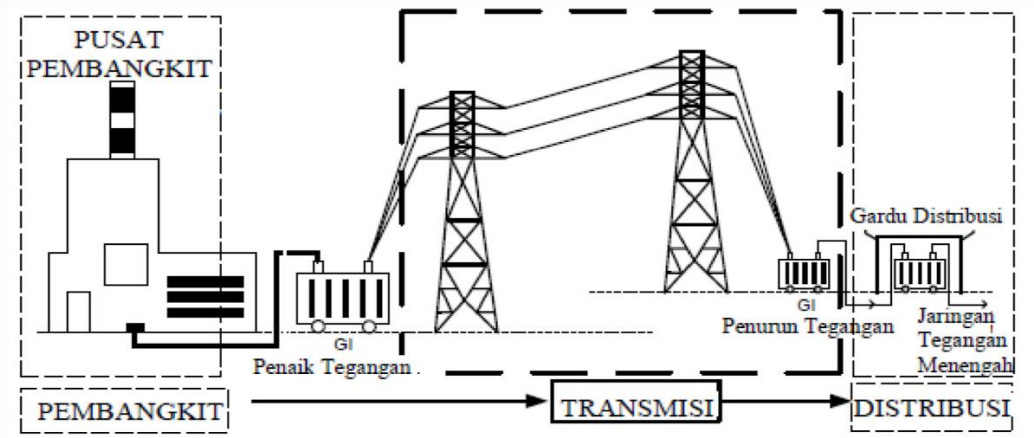
a. Menggunakan *Gradle-based build system* yang fleksibel.
b. Bisa mem-*build multiple* APK (*Application Package*).
c. *Template support* untuk Google *Services* dan berbagai macam tipe perangkat.
d. Layout *editor* yang lebih bagus.

e. *Built-in support* untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.
f. *Import library* langsung dari Maven *repository*.
g. *Emulator* yang cepat dan kaya fitur.
h. Mendukung C++, NDK dan Kotlin.

2.2 Gardu Listrik

Gardu Listrik adalah suatu instalasi yang terdiri dari peralatan listrik yang berfungsi untuk :

a. Mengubah tenaga listrik tegangan tinggi yang satu ke tegangan tinggi yang lainnya atau tegangan menengah.
b. Pengukuran, pengawasan, operasi serta pengaturan pengamanan sistem tenaga listrik.
c. Pengaturan daya ke Gardu-Gardu Induk lain melalui tegangan tinggi dan Gardu-Gardu Distribusi melalui gawai tegangan menengah.



Gambar 2.1 Sistem Jaringan Gardu Listrik

Gardu Listrik dapat di klasifikasikan menjadi dua hal, diantaranya sebagai berikut:

1. Menurut Lokasi dan Fungsi

   Merurut lokasinya di dalam sistem tenaga listrik, fungsi dan tegangannya (tinggi,menengah atau rendah) maka gardu listrik dapat dibagi:

a. Gardu Induk.

Gardu Induk adalah gardu listrik yang mendapatkan daya dari satuan transmisi atau sub-transmisi suatu sistem tenaga listrik untuk kemudian menyalurkannya ke daerah beban (industri, kota dan sebagainya) melalui saluran distribusi primer.



Gambar 2.2 Gardu Induk

b. Gardu Distribusi.

Gardu Distribusi adalah gardu listrik yang mendapatkan daya dari saluran distribusi primer yang menyalurkan tenaga listrik ke pemakai dengan tegangan rendah.



Gambar 2.3 Gardu Distribusi

2. Menurut Penempatan Peralatannya

   Menurut penempatannya, gardu listrik dapat dibagi:

   a. Gardu Induk pemasangan dalam.

      Gardu Induk dimana semua peralatannya *(switchgear*, isolator dansebagainya) di pasang di dalam gedung/ruangan tertutup.

   b. Gardu Induk pemasangan luar.

      Gardu Induk dimana semua peralatannya (*switchgear,* isolator dan sebagainya) di tempatkan di udara terbuka [4].

2.3 Pemrograman Berorientasi Objek (Object Oriented Programming)

Pemrograman Berorientasi Objek (*Object Oriented Programming*) merupakan suatu paradigma pemrograman yang secara mendasar berbeda dengan pemrograman procedural. Pemrograman Berorientasi Objek (selanjutnya disingkat PBO) memiliki fokus pada konsep pemrograman objek yang memiliki atribut dan fungsi yang digunakan untuk melakukan manipulasi terhadap atribut tersebut.

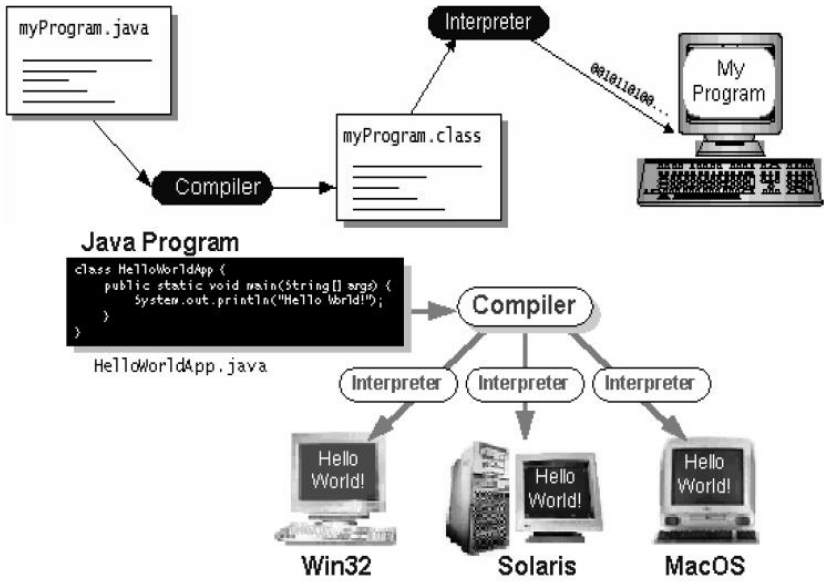
Secara spesifik, pengertian berorientasi objek berarti bahwa mengorganisasi perangkat lunak sebagai kumpulan objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemrograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu [5]:

1. Identifikasi
2. Klasifikasi
3. *Polymorphism* (polimorfisme)
4. *Inheritance* (pewarisan)

2.4 Pemrograman Java

Java adalah program yang sangat populer di lingkungan pendidikan saat ini, hampir di seluruh kampus memiliki satuan ajar pemrograman ini. Hal yang paling mendasar dan harus dikuasai oleh Programer Java adalah memasang library (kelas dalam bentuk biner). Karena java yang dirancang secara orientasi objek, tentunya program yang terdiri atas kesatuan antar objek yang membentuk aplikasi. Sehingga sebuah objek tertentu bisa digunakan dalam beberapa aplikasi yang berbeda. Objek

ini biasanya bersifat umum penggunaannya, misalkan objek kalender, objek *driver* basis data, objek *report* dan lain-lain [6].



Gambar 2.4 Cara Kerja Pemrograman Java

2.5 UML

UML (*Unified Modeling Language*) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak berorientasi objek. UML memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema basis data, dan komponen-komponen yang diperlukan dalam sistem perangkat lunak. UML juga merupakan sebuah bahasa standar untuk pengembangan sebuah perangkat lunak yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model yang seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan perangkat lunak. UML terdiri atas pengelompokkan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram tersebut menggambarkan permasalahan maupun solusi dari suatu model. UML mempunyai 9 diagram, yaitu; *use-case, class, object, state, sequence, collaboration, activity, component,* dan *deployment* diagram [7].

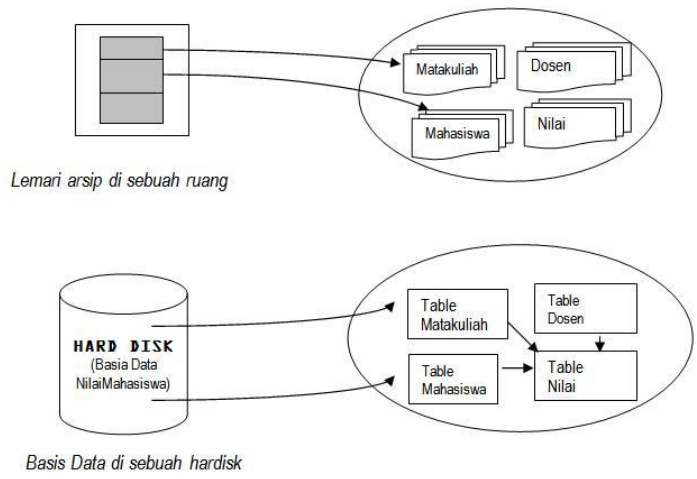
2.6 Basis Data

Basis Data terdiri atas 2 kata, yaitu Basis dan Data. Basis diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah refresentasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya.

Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi, berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Basis data dan lemari arsip sesungguhnya memiliki prinsip dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip menggunakan lemari sebagai media penyimpanan, maka basis data menggunakan media penyimpanan elektronis seperti disk (disket dan *harddisk*).



Gambar 2.5 Lemari Arsip dan Basis Data

Yang perlu d2ngat adalah bahwa tidak semua bentuk penyimpanan data secara elektronis bisa disebut basis data. Yang sangat ditonjolkan dalam basisdata adalah pengaturan/pemilaha/pengelompokkan/pengorganisasian data yang akan kita simpan sesuai fungsi/jenisnya. Pemilahan/ pengelompokan ini dapat berbentuk sejumlah file/ tabel terpisah atau dalam bentuk pendefinisian kolom-kolom/*field-field* data dalam setiap file/tabel. Tujuan dibangunnya Basis Data adalah sebagai berikut [8]:

1. Kecepatan & kemudahan (*speed*)
2. Efisiensi ruang penyimpanan (*space*)
3. Keakuratan (*accuracy*)
4. Ketersediaan (*availability*)
5. Kelengkapan (*completeness*)
6. Keamanan (*security*)
7. Kebersamaan pemakaian (*sharability*)

2.7 Oracle

Oracle Corporation yang berdiri pada tahun 1977 sebelumnya mengembangkan database Oracle sebagai perangkat lunak yang dipakai oleh pihak militer US. Namun seiring perkembangan database Oracle juga dipergunakan dan dijual ke pihak sipil atau bisnis. Dengan demikian citra militer seperti tangguh dan siap dalam segala situasi juga turut melekat dalam database Oracle. Oracle sendiri tidak hanya bermain dalam ranah database namun juga dalam ranah aplikasi dan perangkat keras. Database Oracle memiliki berbagai edisi, antara lain [9]:

1. Standard One edisi berbayar dengan batasan maksimal dua prosesor.
2. Standard edisi berbayar dengan batasan maksimal empat prosesor.
3. Enterprise edisi berbayar tanpa batasan prosesor.
4. Express edisi gratis dengan batasan data maksimal 4 Gigabyte (4.294.967.296 Byte)
5. Untuk versi terakhir, 11.2.0.6, database Oracle tersedia untuk hampir semua platform, baik Windows, Linux maupun Unix, mesin 32 bit maupun mesin 64 bit.

2.8 Internet

Istilah Internet berasal dari kata *internetworking*. *Internetworking* sendiri bisa diartikan sebagai *network* dari *networ*k, yang berarti kumpulan dari jaringan-jaringan yang menghubungkan komputer dari sistem yang berbeda-beda. Jadi, secara singkat dapat dikatakan bahwa Internet adalah kumpulan berbagai sistem jaringan komputer di dunia yang terkoneksi satu sama lain dan dapat saling berkomunikasi satu sama lain [10].

2.9 JSON (JavaScript Object Notation)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemprograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemprograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.[11]

2.10 Pengujian Black Box

Pengujian *black box* berfokus pada keperluan penelusuran kesalahan fungsional dari *software*. Ujicoba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya [12]:

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan *interface*.
3. Kesalahan dalam struktur data.
4. Kesalahan performa.
5. Kesalahan inisialisasi dan terminasi