BAB II

LANDASAN TEORI

2.1. Tinjauan Jurnal

Berdasarkan jurnal yang dijadikan sebagai acuan penulis skripsi ini, maka dijadikan beberapa jurnal sebagai referensinya.

> Menurut Hidayatullah, dkk (2013:26) Dengan perkembangan teknologi smartphone saat ini dapat dimanfaatkan untuk membuat inovasi media pembelajaran yang disebut *m-learning* (*mobile learning*) telah menjadi sebuah cara belajar baru yang memungkinkan pembelajaran dapat dilakukan secara mobile dengan memanfaatkan *device* bergerak, khusunya telepon genggam (*handphone*).

> Kerja sama yang baik ketat dan lancar akan memberikan hasil yang memuaskan. Dalam hal ini bagian Housekeeping Departmen mempunyai hubungan kerja dengan bagian (departemen) yang lain diantaranya dengan bagian Kantor Depan Hotel (Front Office Departement) menurut Agus Sulistiyono (2010: 184).

2.2. Konsep Dasar Program

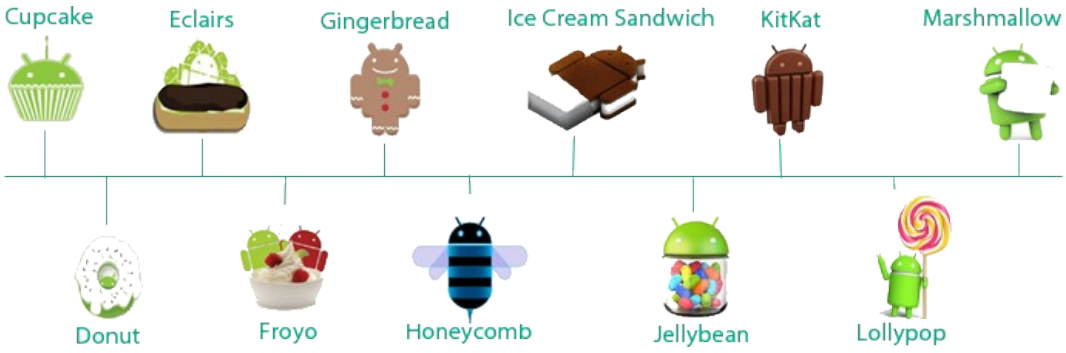
Konsep dasar program yang digunakan dalam pembuatan android adalah :

a. Android

> Menurut Masruri (2012:2) Android merupakan sistem operasi berbasis Linux untuk perangkat *mobile* yang dikutip dari situs Wikipedia. Android merupakan sistem operasi gratis dan open source, jadi Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan suatu aplikasi sendiri yang mampu berjalan di atas peranti Android.

Hal itulah yang menjadikan Android mampu bersaing di tengah keramaian *smartphone* Blackberry dan iPhone yang lebih dahulu meramaikan pasaran.

Keunikan dari sistem operasi Android ini adalah dari nama yang diberikan pada tiap versinya yang dimana nama tersebut menggunakan nama sebuah makanan penutup disebuah restoran. "Tidak ada penjelasan resmi dari Google mengenai mengenai penamaan versi Android, namun Google hanya ingin tampil sedikit berbeda dalam hal ini" , ungkap Randal Sarafa juru bicara Google.



http://techtrickle.com/wp-content/uploads/2015/09/history-of-android-operating-system.png

Gambar II.1 Logo Android

Berikut ini perjalanan versi sistem operasi Android yang telah di berikan Google:

a.Android Versi Beta

Android versi beta ini dirilis pada tanggal 5 November 2007, di mana Android pertama kali muncul dan masih belum di gunakan oleh orang seperti saat ini.

b.Android versi 1.0

Android versi 1.0 ini dirilis pada tanggal 23 November 2008, ponsel pertama yang menggunakan sistem operasi, yaitu HTC G1 (Dream), pada Android 1.0 ini belum ada codename yang diberikan oleh Google. Fitur - fitur yang diberikan Android 1.0 adalah sebagai berikut:

1) Sudah terintegrasi dengan Gmail.
2) Tersedia layanan Google Talk.
3) Terdapat media player, dan masih banyak lagi fitur yang lainnya.

c.Android Versi 1.1

Android versi 1.1 ini dirilis pada tanggal 9 Maret 2009. Pada versi ini telah dilakukan perbaikan - perbaikan terhadap sejumlah permasalahan yang ditemukan pada Android versi 1.0 sebelumnya dan pada versi ini Google belum juga memberikan nama kode. Berikutnya ini fitur - fitur yang diberikan Android versi 1.1:

1) Pembaruan estetis pada semua aplikasi.
2) Penambahan fitur pada Google Maps.
3) Tersedia jam alarm.
4) Tersedia pencarian suara *voice search*.
5) Pengirim pesan dengan Gmail serta pemberitahuan email.

d.Android Versi 1.5

1) Android versi 1.5 ini dirilis oleh Google pada tanggal 30 April 2009. Pada versi inilah Google pertama kali memberikan nama kode atau *codename* pada versi Android, yaitu dengan nama *Cupcake* yang artinya Kue mangkuk.
2) Penambahan bluetooth A2DP dan AVRCP.
3) Dapat menonton dan mengunggah video melalui YouTube.
4) Keyboard virtual baru yang dilengkapi dengan *text prediction*.
5) Browser dengan kecepatan yang lebih tinggi.
6) Terdapat animasi perpindahan layar.
7) Tersedia banyak widget yang menarik.
8) Dapat membuat folder pada *home screen* sehingga lebih mudah mengelompokkan aplikasi.

e.Android Versi 1.6

Android versi 1.6 ini dikeluarkan oleh Google pada tanggal 15 September 2009. Kali ini Google memberikan kode nama *Donut*, yaitu sebuah nama makanan yang sudah tidak asing lagi di telinga kita dan pada versi ini pula telah hadir Android Market baru.

Android 1.6 dirilis untuk memperbaiki kesalahan reboot pada sistem operasi serta perubahan fitur pada antarmuka kamera dan integrasi pencarian yang lebih baik.

Berikut ini fitur yang diberikan Android versi 1.6:

1) Mendukung ukuran layar yang lebih besar.
2) Adanya gesture framework untuk mekanisme input.
3) *Google free turn by turn navigation*.
4) Penambahan fitur pada *voice search*.
5) Dukungan terhadap teknologi CDMA/EVDO, 802.1x, VPN, serta *text-to-speech engine*.

f. Android Versi 2.0 dan Android Versi 2.1

Android versi 2.0 ini dirilis pada tanggal 26 oktober 2009 selang waktu 3 bulan Google telah merilis kembali sistem operasi Android dengan versi 2.1 tepatnya pada tanggal 12 Januari 2010.

Kedua versi tersebut telah diberi nama kode yang sama, yaitu Eclair di mana *Eclair* merupakan makanan penutup yang biasanya dihidangkan bersama secangkir kopi hangat pada sebuah restoran.

Berikut ini fitur yang disuguhkan Android versi 2.0 dan 2.1:

1) Tampilan browser yang baru dan didukung HTML5.
2) Dukung *Multi Touch*.
3) Perubahan *interface* atau tampilan.
4) Adanya *Live Wallpaper*.

5) Dukungan terhadap teknologi bluetooth 2.1.

6) Peningkatan kelas *MotionEvent*.

7) *Android Market share*.

8) Google Maps 3.1.2.

g.Android Versi 2.2

Android versi 2.2 ini telah dirilis oleh google pada tanggal 20 Mei 2010 dengan nama kode Froyo, *codename*tersebut merupakan singkatan dari *Frozen Yogurt* yang merupakan sebuah makanan beku yang dibuat dari yogurt.

Pada Android versi ini sangat banyak sekali penambahan fitur - fitur baru dan perbaikan yang cukup besar pengaruhnya terhadap ponsel Android.

Berikut ini fitur serta perbaikan yang diberikan Android Froyo:

1) Mendukung Adobe Flash 10.1.

2) Optimasi umum terhadap kecepatan, memori, dan kemampuan Android.

3) Instalasi aplikasi pada memori eksternal.

4) Penambahan fitur update otomatis pada *Android Market.*

5) Implementasi JIT (*Justi In Time*)

6) Upload file yang mendukung browser dengan animasi GIF.

7) Perubahan pada launcher aplikasi, dengan adanya shortcut untuk mengakses phone dan browser.

8) Dan masih sangat banyak lagi.

h.Android Versi 2.3

Android versi 2.3 ini dikeluarkan oleh Google pada tanggal 6 Desember 2010. Kali ini Google memberikan *codenameGingerbread* yang memiliki arti Roti Jahe.

Gingerbread sering dibuat untuk merayakan liburan di akhir tahun di Amerika Serikat dan biasanya dibuat dengan bentuk menyerupai seperti manusia. Pada versi ini Android secara resmi mendukung beberapa perangkat keras baru, seperti kamera yang lebih dari satu, perangkat *Near Field Communication* (NFC), sensor *gyroscope*, dan sensor barometer.

Berikut ini fitur yang ditawarkan Android versi 2.3;

1) Perubahan tampilan yang sangat luar biasa.

2) Mendukung *Video Call*.

3) *Improved copy* dan *paste.*

4) *Multitouch virtual keyboard* yang dirancang ulang.

5) Dukungan native SIP VoIP *Telephony*.

i. Android Versi 3.0 dan Android Versi 3.1

Android versi 3.0 diluncurkan Google pada tanggal 22 Februari 2011 dengan memberikan nama kode *Honeycomb* yang berarti Sarang Madu, tak lama kemudian Google merilis versi terbarunya, yaitu Versi 3.1 dengan *codename* yang masih sama.

Android versi ini khusus di rancang untuk digunakan pada perangkat tablet yang memiliki ukuran antara 5-10 inci. Di bawah ini merupakan fitur yang ditawarkan oleh Android versi 3.0 dan 3.1:

1) *Visual MultitaskingSwitching.*
2) Adanya tampilan baru yang disebut dengan *Holo Theme*.
3) Desktop 3D dengan tampilan widget baru yang dapt di-resize.
4) Mendukung *Multi Core Processor* dengan mendukung GPU (*Gaphic Processing Unit*).

j.Android Versi 4.0

Android versi 4.0 dirilis pada tanggal 19 Oktober 201 yang diberikan nama *Ice Cream Sandwich*, versi ini merupakan Android keluaran dari Google yang membawa fitur *Honeycomb* untuk smartphone.

Berikut ini merupakan fitur yang diberikan oleh Android versi *Ice Cream Sandwich*:

1) *Face Unlock*, yaitu fitur yang memungkinkan pengguna membuka kunci dengan identifikasi wajah.
2) Penambah fitur *social networking* dan internet.

3) Kamera yang dilengkapi dengan aplikasi edit foto.

4) *Data usage*, yang berfungsi untuk membatasi suatu aplikasi yang sedang berjalan.

5) Penambahan fitur *multitasking*.

6) Mencari email secara *offline* serta berbagi informasi dengan mengguna-kan NFC.

7) Aplikasi terpadu kontak jaringan sosial.

8) *Favorite Tray*, berfungsi untuk membuat *shortcut* aplikasi *favorite*.

k.Android Versi 4.1

Sistem operasi besutan Google dengan logo sebuah robot berwarna hijau ini tak henti - henti mengeluarkan versi terbarunya untuk memperbaiki kekurangan yang ada pada versi sebelumnya. Setelah pihak Google telah mengeluarkan Android dengan versi *Ice Cream Sandwich*kini terdapat satu versi terbarunya lagi, yakni versi *Jelly Bean*. Pada versi kali ini ada perubahan yang membuat para pengguna akan meresakan sensasi yang luar biasa, akan tetapi sayangnya Android dengan versi ini masih hanya bisa ditemui atau digunakan pada Samsung Google Nexus 7 dan Samsung Galaxy S3 resmi mendapatkan update sistem operasinya.

Jelly Bean terkenal dengan sebuah faster OS, hal ini yang menjadi pembeda dengan versi lain karna hal tersebut membuat smartphone akan

menjadi lebih ringan apabila sedang menjalankan aplikasi secara bersamaan. Berikut ini kelebihan Android versi *Jelly Bean*:

1) Sistem keamanan menggunakan ASLR, dimana ASLR merupakan teknologi untuk meningkatkan sisi keamanan dalam sistem memori pada sebuah perangkat.

2) *Jelly Bean*telah menambahkan sebuah aplikasi yang dapat menghemat baterai ponsel anda.

3) Terdapat aplikasi *Google Now* yang merupakan aplikasi berbasis lokasi yang di klaim mampu memberikan banyak informasi kepada penggunanya.

4) Tentunya masih sangat banyak lagi fitur - fitur yang sangat menarik yang diberikan Android versi *Jelly Bean.*

Apabila kita cermati pemberian nama kode di tiap versi android oleh google diatas telah urut abjad, yakni C = *Cupcake*, D = *Donut*, E = *Eclair*, F = *Froyo*, G = *Gingerbread* , H = *Honeycomb*, I = *Ice Cream Sandwich*, J = *Jelly Bean*. Nama - nama tersebut semuanya merupakan nama makanan pada sebuah restoran, Google memang kali ini ingin sedikit tampil beda.

Akan tetapi yang menjadi pertanyaan kemana huruf A dan B yang tidak dipakai pada pemberian nama kode versi Android? Menurut juru bicara Google keduanya tersebut tidak terkenal pada sebuah makanan sehingga tidak di pakai oleh Google.

b. HTML5

HTML5 adalah versi html terbaru yang digunakan untuk membuat desain antarmuka bagi perangkat mobile apps. HTML5 saat ini masih terus dikembangkan dan merupakan proyek kerja sama antra *Worls Wide Web Consortium* (W3C) dan *Web Hypertext Application Technologi Working Group* (WHATWG)("Belajar Pemrograman Android," 2015).

WHATWG mengerjakan tentang web form dan aplikasi, sementara W3C mengerjakan tentang XHTML 2.0. Pada tahun 2006, diputuskan bahwa HTML5 mulai dikembangkan.

Beberapa rule untuk HTML5 adalah :

a. Fitur - fitur baru berbasis HTML, CSS, DOM, dan JavaScript.

b. Mengurangi kebutuhan terhadap plugin eksternal (misalnya *flash*).

c. *Error handling* yang lebih baik.

d. Lebih banyak *markup* untuk menggantikan *scripting*.

e. HTML5 harus bersifat tidak bergantung pada perangkat (*device independent*).

f. Proses pengembangannya harus terbuka ke umum.

c. MIT APP Inventor 2

App Inventor adalah sistem perangkat lunak untuk membuat aplikasi pada perangkat Android. Uniknya, App Inventor dibuat tidak seperti sistem pengembangan aplikasi biasa, di mana seorang programmer harus menuliskan baris - baris kode program, melainkan dengan interaksi visual berbasis grafis. Dalam kaitan ini, inventor dapat disebut sebagai

sistem terpadu untuk pengembangan aplikasi berbasis blok - blok grafis (dalam istilah asing:*blocks language*). Jika Anda telah mengenal *Scratch*(*http://scratch.mit.edu*), App Inventor bekerja dengan cara yang kurang - lebih sama, tetapi untuk platform perangkat bergerak yang berbasis Android (meliputi berbagai jenis ponsel dan komputer tablet).

Istilah App Inventor dan Inventor digunakan sebagai sinonim. Inventor diuji coba pertama kali pada kalangan pada Juli 2010, kemudian dirilis ke publik pada Desember tahun yang sama. Hingga saat buku ini ditulis, Inventor masih dalam versi beta. Pengembangan Inventor dimotivasi oleh keyakinan dan perspektif edukasi yang kuat bahwa pembelajaran aktif pemrograman (secara visual) dapat menjadi wahana untuk memicu ide-ide baru dan kreatif, bahkan untuk orang-orang yang tidak atau belum mengenal pemrograman komputer sebelumnya. Dan hasil uji coba memang telah munujukan hasil yang sangat positif.

Seiring dengan perubahan kebijakan internal Google untuk menutup proyek GoogleLabs, dimana App Inventor diinkubasi dan dikembangkan, pada akhir 2011 kode sumber Inventor dijadikan proyek *open-source*. Sedangkan layan Inventor sendiri dipindah-tangankan ke MIT (*Massachusetts Institute of Technology*). Layanan publik Inventor yang sebelumnya tersedia pada alamat *http://appinventor.googlelabs.com* atau *http://appinventorbeta.com* sudah tidak bisa diakses lagi. MIT menyediakan layanan Inventor pada alamat *http://appinventor.mit.edu* pada kuartal pertama 2012 (bulan Maret 2012).

Interaksi programmer dengan App Inventor hampir sepenuhnya melalui antarmuka visual dengan operasi *drag-and-drop*. Secara sistem, App Inventor terdiri dari dua komponen, yaitu *server* dan *client*. *Server* App Inventor berfungsi menyimpan semua aset program dan memberikan layanan lainnya yang terkait dengan manajemen berkas aplikasi (*project*). Sedangkan sisi *client* adalah aplikasi yang berhubungan langsung dengan programmer (pembuat aplikasi).

Sisi *client* terdiri dari tiga bagian berikut:

a. App Inventor ***Designer,*** merupakan aplikasi web untuk mendesain tampilan visual (*GUI-Graphical User Interface*) dan memilih komponen yang diperlukan;

b. App Inventor ***Blocks Editor***, merupakan aplikasi Java untuk mendefinisikan *behaviour*(perilaku) dari tiap komponen yang dengan menyusun blok-blok program secara visual (Seolah-olah seperti menyusun potongan-potongan *puzzle*);

c. Perangkat Android yang diperlukan untuk mengetes dan menjalankan aplikasi. Jika perangkat Android tidak tersedia, kita dapat menggunakan ***emulator.*** *Emulator* merupakan perangkat lunak yang berjalan diatas komputer dan mengemulasikan/berperilaku seperti perangkat sesungguhnya. Android SDK menyertakan emulator berbasis QEMU (*http://qemu.org*) yang dapat menjalankan sistem Android

sesungguhnya. Emulasi masih digunakan untuk beberapa layanan, seperti telepon, SMS, atau layanan lokasi.

Membuat aplikasi dengan Inventor dimulai dengan membuka *Designer* untuk merancang bentuk tampilan aplikasi. Tahap ini tidak berbeda dengan proses perncangan GUI pada umumnya, Bahkan cenderung lebih sederhana dibandingkan perncang GUI untuk aplikasi desktop. Kemudian, dengan *blocks Editor*kita harus mendefinisikan alur dan perilaku dari aplikasi tersebut. Pada tahap ini kita memilih dan menyusun blok-blok yang diperlukan. *Blocks Editor* hanya dapat dibuka melalui tombol yang tersedia pada *Designer*. Umtuk melihat secara *live* tampilan visual dan efek penyusunan blok, emulator atau perangkat harus disambungkan terlebih dahulu. Fasilitas penyambungan ke perangkat atau *emulator* tersedia melalui jendela *Blocks Editor*.

d. *Unified Modelling Language* (UML)

Menurut Chonoles dalam widodo (2011:6), mengemukakan bahwa "*UML* singkatan dari *Unified Modeling Language* yang berarti bahasa pemodelan standar, yang berarti*UML* memiliki sintaks dan semantik.

UML awalnya termotivasi oleh keinginan untuk membakukan sistem notasi yang berbeda dan pendekatan untuk desain perangkat lunak yang dikembangkan oleh Grady Booch, Ivar Jacobson dan James Rumbaugh di Rational *Software* di 1994-1995, dengan pengembangan lebih lanjut yang dipimpin oleh mereka melalui tahun 1996.

Sebuah bahasa model adalah sebuah bahasa yang mempunyai konsep tatanan atau aturan penulisan serta secara fisik mempresentasikan dari sebuah sistem. Seperti halnya UML adalah sebuah bahasa standard untuk pengembangansebuah *software* yang dapat menyampaikan bagaimana membuat dan membentuk model-model, tetapi tidak menyampaikan apa dan kapan model seharusnya dibuat yang merupakan salah satu proses implementasi pengembangan *software*.

UML tidak hanya merupakan sebuah bahasa pemrograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemrograman visual saja, namun juga dapat secara langsung dihunbungkan ke berbagai bahasa pemrograman, seperti Java, C++, Visual Basic, atau bahkan dihungungkan secara langsung ke dalam sebuah object-oriented database. Begitu juga mengenai pendokumentasian dapat dilakukan seperti ; *requirements*, arsitektur, *design*, *source code*, *project plan*,*test*, dan *prototypes*.

Dalam UML sendiri terdapat beberapa diagram yaitu :

1) *Use Case Diagram*

Use case diagram adalah suatu model yang sangat fungsional dalam sebuah sistem yang menggunakan *actor* dan *use case*. Sedangkan pengertian *use case* sendiri adalah layanan atau fungsi-fungsi yang tersedia pada sistem penggunaannya. *Use case diagram* menggambarkan efek fungsionalitas yang telah diharapkan oleh sistem. *Use case diagram* dapat

sangat membantu ketika sedang menyusun *requietment* sebuah sistem, mengkomunikasikan sebuah rancangan aplikasi dengan konsumen, serta meracangan *test case* untuk semua *feature* yang ada pada sistem.

2) *Sequence Diagram*

Sequence diagram adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaski antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan. *Sequence diagram* digunakan untuk menggambarkan *scenario* atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian untuk menghasilkan *output* tertentu. *Sequence diagram* diawali dari proses dan perubahan apa saja yang terjadi secara internal dan *output* yang dihasilkan.

3) *Activity Diagram*

Activity Diagram adalah versi UML untuk sebuah *flowchart*. *Activity diagram* digunakan untuk menganalisa proses. Sebuah *activity diagram* bukan sebuah *tool* yang sempurna untuk menganalisis masalah dari sistem. Sebagai *tool* untuk menganalisis, pemrogram tidak ingin untuk mulai memecahkan masalah dilevel teknis dengan membuat *class*, tetapi dengan menggunakan *activity diagram* untuk mengerti masalah dan menyaring proses yang terdapat dalam sistem.

Setiap *activity diagram* selalu mempunyai satu *initial state. Initial node* yang digambarkan dengan simbol lingkaran padat,merupakan titik yang mengawali *activity diagram*. *Activity diagram* dapat diakhiri dengan

memberikan *activity final diagram* yang digambarkan dengan lingkaran padat dengan mempunyai cincin dibagian luarnya.

4) *Class Diagram*

Class diagram adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket, dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* diagram juga menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.

Class juga memiliki 3 area pokok (utama) yaitu : nama,atribut,dan operasi. Nama berfungsi untuk member identitas pada sebuah kelas, atribut fungsinya adalah untuk member karakteristik pada data yang dimiliki suatu objek di dalam kelas, sedangkan operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek .

5) *Deployment Diagram*

*Diagram Deployment*adalah pandangan yang terkait dengan penyebaran fisik aplikasi. Hal ini termasuk persoalan *layout* jaringan dan lokasi komponen-komponen dalam jaringan. *Deployment View* berisikan prosesor-prosesor, peralatan-peralatan, proses-proses dan hubungan antar prosesor dan antar perlatan. Semua informasi ini digambarkan dalam suatu *deployment Diagram*. Hanya ada satu *deployment Diagram* dalam setiap *system*, sehingga hanya satu *deployment* dalam setiap model. Suatu *deployment diagram* menampilkan semua titik (*node*) dalam suatu jaringan, hubungan antar mereka, dan proses-proses yang dijalankan pada

masing-masing *node*.

2.3 Metode Algoritma

Algoritma yang digunakan dalam pembuatan aplikasi tersebut adalah metode algoritma Boolean, yaitu merupakan metode algoritma yang berisi *true* dan *false*, yang dimana digunakan untuk penanda sebuah proses benar atau salah jawaban dari pertanyaan kuis yang disediakan penulis.

2.4 Pengujian Software

Metode pengujian software yang akan penulis gunakan ada 2, yaitu metode *white box texting* dan *black box texting.*

a.*White Box Texting*

White Box Testing adalah salah satu jenis testing yang lebih berkonsentrasi terhadap "isi" dari perangkat lunak itu sendiri. Jenis ini lebih banyak berkonsentrasi kepada *source code* dari perangkat lunak yang dibuat.

Teknik yang terdapat di*White Box Testing* :

1) *Decision (branch) Coverage*

Sesuai dengan namanya, teknik testing ini fokus terhadap hasil dari tiap skenario yang dijalankan terhadap bagian perangkat lunak yang mengandung percabangan (*if...then...else*).

2) *Condition Coverage*

Teknik ini hampir mirip dengan teknik yang pertama, teatapi dijalankan terhadap percabangan yang dianggap kompleks atau percabangan majemuk. Hal ini biasanya dilakukan jika dalam sebuah perangkat lunak memiliki banyak kondisi yang dijalankan dalam satu proses sekaligus.

3) *Path Analysis*

Merupakan teknik *testing* yang berusaha menjalankan kondisi yang ada dalam perangkat lunak serta berusaha mengoreksi apakah kondisi yang dijalankan telah sesuai dengan alur diagram yang terdapat dalam proses perancangan.

4) *Execution Time*

Pada teknik ini, perangkat lunak berusaha dijalankan atau diekseskusi kemudian dilakukan pengukuran waktu pada saat *input* dimasukkan hingga *output* dikeluarkan. Waktu eksekusi yang dihasilkan kemudian dijadikan bahan evaluasi dan analisa lebih lanjut untuk melihat apakah perangkat lunak telah berjalan sesuai dengan kondisi yang dimaksud oleh *tester*

5) *Algorithm analysis*

Teknik ini umumnya jarang dilakukan jika perangkat lunak yang dibuat berjenis sistem informasi. Sebab teknik ini membutuhkan kemampuan matematis yang cukup tinggi darri para *tester,* karena di dalamnya berusaha melakukan analisa terhadap algoritma yang diimplementasikan pada perangkat lunak tersebut.

b. *Black Box Testing*

*Black Box Testing*adalah tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Sehingga para *tester*memandang perangkat lunak seperti layaknya sebuah "kotak hitam" yang tidak penting dilihat isinya, tapi cukup dikenai proses *testing.*

Teknik yang terdapat di*Black Box Testing*:

1) *Equivalence Partitioning*

   Pada teknik ini, tiap inputan data dikelompokkan ke dalam grup tertentu, yang kemudian dibandingkan *outputnya*.

2) *Boundary Value Analysis*

   Merupakan teknik yang sangat umum digunakan pada saat awal sebuah perangkat lunak selesai dikerjakan. Pada teknik ini, dilakukan inputan yang melebihi dari batasan sebuah data.

3) *Cause Effect Graph*

Dalam teknik ini, dilakukan proses *testing* yang menghubungkan sebab dari sebuah *input* dan akibatnya pada *output* yang dihasilkan.

4) *Random data Selection*

Seperti namanya , teknik ini berusaha melakukan proses inputan data dengan menggunakan nilai acak. Dari hasil inputan tersebut kemudian dibuat sebuah tabel yang menyatakan validitas dari *output* yang dihasilkan.

5) *Feature Test*

Pada teknik ini, dilakukan proses *testing*terhadap spesifikasi dari perangkat lunak yang telah selesai dikerjakan.

2.5 Peralatan Pendukung

Untuk mewujudkan sistem yang telah dirancang memerlukan sarana pendukung, yakni berupa komponen-komponen yang sangat berperan penting dalam menunjang penerapan sistem terhadap pengolahan data.

a. *Hardware*

Dalam pembuatan program ini penulis menggunakan sebuah *personal computer* dengan spesifikasi sebagai berikut :

1) *Harddisk* : 500 GB

2) Processor : Intel(R) Core(TM) i3-2348M

CPU @ 2.30GHz (4 CPUs), ~2.3GHz

3) RAM : 6.00 GB

4) Kartu Grafis : AMD Radeon HD 7610M

b. *Software*

Dalam pembuatan program ini penulis menggunakan beberapa aplikasi pendukung sebagai berikut :

1) Sistem operasi *personal computer* : Windows 7

2) Sistem operasi pada *handphone* : Android 5.1.1

3) Pembuat Aplikasi : MIT App Inventor 2