

RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID



**Disusun Oleh:
Wim Adhiatma
06.12.600**



**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014**

RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID

SKRIPSI

**Diajukan Untuk Memenuhi Salah Satu Persyaratan
Dalam Menyelesaikan Program Pendidikan Strata Satu (S-1)**



Disusun Oleh:

Wim Adhiatma

06.12.600

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014**

RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID

SKRIPSI

**Diajukan Untuk Memenuhi Salah Satu Persyaratan
Dalam Menyelesaikan Program Pendidikan Strata Satu (S-1)**



Disusun Oleh:

Wim Adhiatma

06.12.600

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2014**

LEMBAR PERSETUJUAN

RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID

SKRIPSI

Disusun dan diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai gelar Sarjana Teknik Strata Satu (S-1)

Disusun Oleh :
WIM ADHIATMA

06.12.600

Mengetahui,
Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT
NIP.P. 1030100385

Diperiksa dan Disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

Sotyhadi, ST
NIP.P.1039700309

PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : WIM ADHIATMA
NIM : 06.12.600
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer
MASA BIMBINGAN: SEMESTER GANJIL 2013/2014
JUDUL : RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Selasa
Tanggal : 18 Februari 2014
Dengan Nilai : 84,55 (A) *α*

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Sekretaris Majelis Penguji

Dr. Eng. Aryuanto S, ST, MT
NIP.P.1030800417

ANGGOTA PENGUJI

Dosen Penguji I

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Dosen Penguji II

Yuli Wahyuni, ST, MT
NIP.P.1031200456

RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID

WIM ADHIATMA (0612600)

Konsentrasi Teknik Komputer
Jurusan Teknik Elektro S-1
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang
Jl. Raya Karanglo Km 2 Malang
e-mail : wim_adhiatma@yahoo.com

ABSTRAK

Pada saat ini banyak sekali bermunculan game – game dari para *developer* game baik yang gratis atau pun yang berbayar. Pada kenyataannya *smartphone* berbasis android merupakan *platform* baru di dunia game. Maka penulis membuat aplikasi game main kata di *smartphone* berbasis *android* untuk memajukan industri game khususnya di Indonesia. Salah satu cara bersaing di industri game android yaitu membuat game ini dapat di *download* secara gratis dan dapat di mainkan oleh semua umur dengan syarat memiliki *smartphone* berbasis android.

Pada skripsi telah di buat aplikasi game berbasis android yang dapat di unduh secara gratis. Metode yang di gunakan dalam perancangan dan pengembangan game ini adalah pemrograman *Java*. Aplikasi game ini di buat menggunakan program *Eclipse Juno*.

Aplikasi game yang di buat dengan *Eclipse* tersebut menghasilkan sebuah aplikasi game yang dapat di main kan di *platform mobile* dan dapat di unduh secara gratis. Aplikasi game ini telah di uji di beberapa *smartphone android* dengan minimal sistem operasi *froyo 2.2* dan menghasilkan tampilan yang sama.

Kata kunci : *Android, game, smartphone, main kata, Eclipse*

ABSTRACT

At this time a lot of popping game - a game from game developers either free or paid. In fact it is an android based smartphone platform in the world of new games. The authors make an application to play word games on the android based smartphone to promote the gaming industry, especially in Indonesia. One way to compete in indrusti android games that make this game can be downloaded for free and can be mainkn by all ages with the condition have a smartphone based on Android.

In the thesis has made android -based gaming applications that can be downloaded for free . The method used in the design and development of this game is a Java programming. This game application created using Eclipse Juno program.

Gaming applications created with the Eclipse application resulted in a game that can be played right on the mobile platform and can be downloaded for free. This gaming app has been tested on several smartphones with android 2.2 froyo operating system minimum and produce the same appearance.

Keyword : *Android, game, smartphone, main kata, Eclipse*

KATA PENGANTAR

Puji Syukur kepada Allah SWT atas berkat dan rahmat-Nya, sehingga kamiselaku penyusun dapat menyelesaikan laporan skripsi ini yang berjudul **“RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID”** dapat terselesaikan.

Adapun maksud dan tujuan dari penulisan laporan ini merupakan salah satu syarat untuk dapat menyelesaikan studi dan mendapatkan gelar Sarjana Teknik di Jurusan Teknik Elektro Konsentrasi Teknik Komputer ITN Malang.

Penyusun menyadari tanpa adanya kemauan dan usaha serta bantuan dari berbagai pihak, maka laporan ini tidak dapat diselesaikan dengan baik. Oleh sebab itu, penyusun mengucapkan terima kasih kepada yang terhormat :

1. Bapak Ir. Soeparno Djiwo, MT. selaku Rektor ITN Malang.
2. Bapak Ir. Anang Subardi, MT. selaku Dekan Fakultas Teknologi Industri ITN Malang.
3. M. Ibrahim Ashari, ST, MT selaku Ketua Program Studi Teknik Elektro ITN Malang dan juga Dosen Pembimbing.
4. Bapak Ir. Yusuf Ismail Nakhoda, MT dan bapak Sotyohadi, ST selaku Dosen Pembimbing Skripsi.
5. Teman-teman ITN Malang dan semua pihak yang membantu dalam penyusunan Skripsi ini.

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna, untuk itu kritik dan saran dari semua pihak sangat kami harapkan untuk perbaikan penelitian ini.

Malang, 18 Febuari 2014

Penyusun

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PERSETUJUAN	ii
LEMBAR BERITA ACARA	iii
ABSTRAK	iv
KATA PENGANTAR	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3. Tujuan	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Penulisan	3
BAB II TINJAUAN PUSTAKA	6
2.1 Pemograman Java	6
2.2 Konsep Dasar <i>Game</i>	6
2.2.1 Pengertian <i>Game</i>	6
2.2.2 Definisi <i>Game</i>	8
2.2.3 Sejarah <i>Game</i>	10
2.2.4 Tipe Mesin <i>Game</i>	10
2.2.5 Genre <i>Game</i>	11
2.3 Pengertian Multimedia.....	10
2.3.1 Definisi Multimedia	14
2.4 Adobe Photoshop CS 5	15
2.4.1 Pengertian Adobe Photosop CS 5	15
2.4.2 Bagian-Bagian Photoshop	16

2.4.3 Tool Box Dalam Photoshop CS.....	18
2.5 Sistem Operasi Android.....	21
2.5.1 Pengertian Android	21
2.5.2 Versi Android	21
2.5.3 Sejarah Android	25
2.6 Eclipse (IDE).....	21
2.6.1 Pengertian Eclipse	27
2.6.2 Bagian – Bagian Eclipse	28
2.7 Android SDK.....	29
2.7.1 Pengertian Android SDK	29
2.7.2 ADT.....	30
2.7.3. Arsitektur Android	31
BAB III PERANCANGAN DAN ANALISA SISTEM	35
3.1 Analisis	35
3.1.1 Analisis SWOT	35
3.1.2 Analisis Kebutuhan Sistem	36
3.1.3 Analisis Kelayakan Sistem.....	39
3.2 Perancangan Game Main Kata	39
3.2.1 Keterangan Umum Game Main Kata	40
3.2.2 Deskripsi Tingkat Kesulitan Game Main Kata.....	42
3.2.3 Deskripsi Pemilihan Kategori Game Main Kata	45
3.2.4 Deskripsi Sistem Point Game Main Kata.....	52
3.2.5 Deskripsi Gameplay Pada Game Main Kata	53
3.2.6 Deskripsi Kebutuhan Minimum <i>SmartPhone Android</i>	54
BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM	55
4.1 Implementasi	55
4.1.1 Implementasi Sistem	55
4.1.1.1 Spesifikasi Kebutuhan Perangkat Keras.....	56
4.1.1.2 Spesifikasi Kebutuhan Perangkat Lunak	56
4.2 Implementasi Antarmuka.....	57
4.3 Pengujian.....	60
4.3.1 Skenario Pengujian Game.....	61

4.3.2 Kasus dan Hasil Pengujian.....	61
4.4 Tutorial Memainkan <i>Game</i> Pada <i>Smartphone Android</i>	76
4.4.1 Kuisisioner.....	76
BAB V PENUTUP	80
5.1 Kesimpulan.....	80
5.2 Saran-Saran	80
DAFTAR PUSTAKA	81
LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 <i>Photoshop CS5</i>	16
Gambar 2.2 <i>Toolbox Photoshop CS</i>	18
Gambar 2.3 Bagian <i>Eclipse</i>	28
Gambar 2.4 Android SDK	30
Gambar 2.5 Arsitektur Android	34
Gambar 3.1 <i>Flowchart Game Main Kata</i>	41
Gambar 3.2 <i>Flowchart Pemilihan Level Easy</i>	42
Gambar 3.3 <i>Flowchart Pemilihan Level Medium</i>	43
Gambar 3.4 <i>Flowchart Pemilihan Level Hard</i>	44
Gambar 3.5 <i>Flowchart Pemilihan Kategori Negara</i>	45
Gambar 3.6 <i>Flowchart Pemilihan Kategori Tokoh</i>	47
Gambar 3.7 <i>Flowchart Pemilihan Kategori Binatang</i>	49
Gambar 3.8 <i>Flowchart Pemilihan Kategori Anime</i>	51
Gambar 3.9 Gambar Papan Permainan	53
Gambar 4.1 Tampilan <i>Main Menu</i>	57
Gambar 4.2 Tampilan <i>Menu Rules</i>	58
Gambar 4.3 Tampilan <i>Menu Tingkat Kesulitan</i>	59
Gambar 4.4 Tampilan <i>Menu Kategori</i>	60
Gambar 4.5 Papan permainan di tingkat kesulitan <i>Hard dan Medium</i>	63
Gambar 4.6 Papan permainan pada tingkat kesulitan <i>Easy</i>	64
Gambar 4.7 Permainan pada system operasi <i>Froyo 2.2</i>	67
Gambar 4.8 Permainan pada system operasi <i>Gingerbread 2.3.3</i>	68
Gambar 4.9 Permainan pada system operasi <i>Honeycomb 3.2</i>	68
Gambar 4.10 Permainan pada system operasi <i>Jelly Bean 4.2.2</i>	69
Gambar 4.11 Permainan pada system operasi ICS 4.0.....	69
Gambar 4.12 Tampilan Samsung Wonder dan Main Menu Permainan.....	86
Gambar 4.13 Tampilan Samsung Wonder dan Main Menu Permainan.....	71
Gambar 4.14 Tampilan Samsung Galaxy S6810.....	72
Gambar 4.15 Tampilan Permainan pada Samsung Galaxy S6810.....	72

Gambar 4.16 Tampilan Samsung Galaxy Core.....	73
Gambar 4.17 Tampilan Permainan pada Samsung Galaxy Core.....	74
Gambar 4.18 Tampilan Advan Vandroid T5-A.....	75
Gambar 4.19 Tampilan permainan pada Advan Vandroid T5-A.....	75

DAFTAR TABEL

Tabel 3.1 JawabandanPetunjukKategori Negara	46
Tabel 3.2 JawabandanPetunjukKategoriTokoh	48
Tabel 3.3 JawabandanPetunjukKategoriBinatang.....	50
Table 3.4 JawabandanPetunjukKategoriAnime	52
Tabel 3.5Sistem Point.....	52
Tabel 3.6Kebutuhan Minimum Perangkat Smartphone	54
Tabel 4.1 TabelPerangkatKeras Yang Digunakan	56
Tabel 4.2 TabelPerangkatLunak Yang Digunakan	56
Tabel 4.3 SkenarioPengujian	61
Tabel 4.4 TabelPengujian <i>Menu</i> Utama	62
Tabel 4.5 PengujianPetunjuk di Tiap Tingkat Kesulitan.....	63
Tabel 4.6 Pengujian Menu JenisKategori	65
Tabel 4.7 Pengujian system point	65
Tabel 4.8 Pengujian Sound	66
Tabel 4.9 TabelSpesifikasi Galaxy Woder	70
Tabel 4.10 TabelSpesifikasi Galaxy Fame	71
Tabel 4.11 TabelSpesifikasi Galaxy Core	73
Tabel 4.12 TabelSpesifikasi Galaxy T5-5	74
Tabel 4.13 KuisisionerPertama.....	77
Tabel 4.14 Kuisisionerkedua	77
Tabel 4.15 KuisisionerKetiga	78
Tabel 4.16 KuisisionerKeempat	78
Tabel 4.17KuisisionerKelima.....	77
Tabel 4.18KuisisionerKeenam.....	79

BAB I

PENDAHULUAN

1.1 Latar Belakang

Judul *game* ini terinspirasi dari *game* yang sudah ada saat ini yang berjudul “*tebak kata*” versi 1.12 yang di kembangkan oleh APITUTE INTERACTIVE. Dimana *game* yang terdahulu tidak di sertakan tingkat kesulitan atau *level*, *backsound* pada permainan dan *point* atau bantuan. Yang membedakan dari *game* yang sudah ada yaitu terdapat tingkat kesulitan yang berbeda dan terdapat *backsound* pada permainan. Dari yang paling mudah yaitu *easy*, *medium*, *hard*. Di *game* ini juga terdapat *point system*, yang berguna sebagai bantuan apabila kita kesulitan menyelesaikan permainan. *Point* di dalam *game* ini di dapat dengan cara menjawab dengan benar pertanyaan yang di berikan. Selain itu *game* ini juga menyertakan jenis kategori pertanyaan, bertujuan agar pemain lebih mudah memilih jenis pertanyaan sesuai dengan kemampuan

Lalu terdapat pula fitur lain yaitu terdapatnya kategori pertanyaan. Di dalam fitur ini pemain berikan banyak pilihan kategori agar menambah keseruan dalam bermain *game* ini.

Penulis mencoba merancang pembuatan sebuah *game* hasil imajinasi penulis yaitu “*game main kata*” yang ditujukan bagi pemula atau maupun yang sudah mahir karena permainan ini sangat sederhana tetapi sangat asik bila sudah memainkan. Untuk itu penulis berusaha membuat tugas akhir dengan judul : “*main kata*” menggunakan *eclipse*.

1.2 Rumusan Masalah

Berdasarkan latar belakang pemilihan judul, maka yang menjadi permasalahan adalah :

1. Membangun sebuah permainan atau yang biasa kita sebut *game* dengan menggunakan sistem berbasis *android*.
2. Mengingat kosa kata lama dan mencari kosa kata baru agar permainan menjadi lebih menarik.
3. Mengembangkan *game main kata* menggunakan aplikasi *SDK* dan *Eclipse* dengan bahasa pemrograman *java*.

1.3 Tujuan dan Manfaat

Tujuan penyusunan skripsi ini adalah untuk mengembangkan *game main kata* yang dapat di mainkan di sistem *android mobile*.

Manfaat dari penyusunan skripsi ini, yaitu :

1. Peningkatan lebih lanjut tentang perangkat lunak game berbasis *android*
2. Mencoba mengingat dan menguji kemampuan tentang dunia seputar kita
3. Sebagai sarana hiburan yang cukup menarik.

1.4 Pembatasan Masalah

Pada pembuatan *game main kata* ini, penulis membuat ruang lingkup pembahasan atau batasan masalah sebagai berikut:

1. Hanya terdapat tiga tingkat kesulitan yaitu *easy, medium, hard* di dalam permainan.
2. Jumlah huruf pada total kata di setiap jawaban tidak lebih dari 30 huruf pada tingkat kesulitan *Easy, Medium, hard* dan di semua kategori.
3. Setiap *clue* yg di tampilkan tidak lebih dari 160 character.
4. *Game* ini hanya dapat di mainkan oleh 1 orang.

5. *Game main kata* ini dapat di jalankan pada perangkat android mobile dengan menggunakan minimum *android serie froyo 2.2* ke atas karena jika menggunakan yang sebelumnya sistem dan kemampuan *hardware* tidak menunjang.
6. Terdapat empat kategori yaitu kategori binatang, Negara, tokoh, dan anime.
7. Setiap menyelesaikan tingkat kesulitan *hard* akan mendapatkan 1 point dan hanya dapat di gunakan pada tingkat kesulitan *hard* saja

1.5 Metodologi Penyelesaian Masalah

Metode yang digunakan dalam pengembangan *game main kata* adalah :

1. Mempelajari cara mengembangkan *game main kata* berbasis android mobile.
2. Mempelajari cara pembuatan game menggunakan *SDK (Software Development Kit) dan Eclipse*
3. Membaca dan mempelajari buku-buku yang berhubungan dengan *SDK dan Eclipse*.
4. Merancang tampilan untuk *game main kata*.
5. Merancang suatu perangkat lunak *game main kata* menggunakan menggunakan bahasa pemograman *java*

1.6 Sistematika Penulisan

Pada penulisan skripsi ini terdiri atas lima pembahasan yaitu :

BAB I : PENDAHULUAN

Bab ini merupakan bagian pendahuluan dimana akan tercakup secara umum mengenai latar belakang penulisan laporan, ruang lingkup karya tulis skripsi ini, tujuan dan manfaat yang mau

dicapai, metodologi yang dipakai dalam penyusunan laporan dan sistematika penulisan yang digunakan.

BAB II : TINJAUAN TEORI

Bab ini berisi tentang teori – teori yang mendukung dan berhubungan dengan judul penulisan skripsi.

BAB III : METODOLOGI

Bab ini berisi prosedur percobaan dan menjelaskan metode penyelesaian yang dilakukan.

BAB IV : IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi penjelasan pembahasan program sesuai dengan permasalahan yang diambil dalam penulisan skripsi.

BAB V : PENUTUP

Bab ini berisi kesimpulan dan saran dari penulisan skripsi.

BAB II

TINJAUAN PUSTAKA

2.1 Pemrograman Java

2.1.1 Pengertian Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi *java* mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Saat ini *java* merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis *web*.^[8]

Untuk membuat program *Java*, Anda membutuhkan *JDK*. Proses instalasi *JDK* tersebut sangat mudah dan tidak membutuhkan pengetahuan tertentu. Namun untuk menggunakannya Anda perlu melakukan beberapa penyesuaian dengan sistem operasi Anda. Umumnya yang perlu Anda lakukan adalah memasukkan path ke direktori *JDK* Anda ke setting path pada sistem operasi Anda. Misalkan direktori *JDK* Anda adalah `C:\jdk1.4` maka pada *Windows 98* Anda cukup menambahkan baris perintah `SET PATH=C:\jdk1.4\bin` pada file `autoexec.bat` Anda. Untuk *Windows NT/2000/XP* Anda cukup menambahkan direktori

C:\jdk1.4\bin pada variabel path di *System Environment*. Caranya: klik kanan ikon *My Computer*, pilih *Properties*. Kemudian pilih tab *Advanced*. Lalu klik tombol *Environment Variables*, cari variabel path, kemudian tambahkan path direktori JDK Anda ke dalam variabel tersebut. Untuk *Linux*, tambahkan baris perintah `SET CLASSPATH=(direktori jdk Anda)` ke file profile Anda. Untuk mencoba JDK, ketikkan perintah `java` dan `javac` pada shell prompt (atau *DOS Command Prompt*). Jika perintah tersebut sudah dikenali maka program `java` atau `javac` akan menampilkan sintaks penggunaan. Untuk kemudahan dan berbagai fasilitas tambahan Anda dapat menggunakan *Integrated Development Environment* (IDE) untuk bahasa *Java* seperti *Visual Café* dari *Symantec* atau *JBuilder* dari *Borland*.^[8]

2.2 Konsep Dasar Game

2.2.1 Pengertian Game

Game berasal dari bahasa inggris yang memiliki arti dasar permainan. Permainan dalam hal ini merujuk pada pengertian “kelincahan intelektual” (*intellectual playability*). Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran sejauh mana *game* itu menarik untuk dimainkan secara maksimal. Kita selalu berpikir *game* merupakan suatu kegiatan yang dilakukan oleh anak-anak yang dapat menyenangkan hati mereka. Dengan kata lain, segala bentuk kegiatan yang memerlukan pemikiran, kelincahan intelektual dan pencapaian terhadap target tertentu dapat dikatakan sebagai *game*.^[9]

Berikut adalah dasar-dasar yang diperlukan dalam membuat *game* yang ditulis oleh detikinet.

1. Membuat *game* hanya memerlukan *scripting* sederhana. Tidak membutuhkan kemampuan programming yang tinggi.
2. Untuk mempermudah pengerjaan, dapat menggunakan program pembuatan *game*. Seperti *RPG Maker* atau *Game Maker*.

3. Bila ingin membuat *mobile game*, dibutuhkan kemampuan programming dasar seperti penguasaan bahasa pemrograman dan OOP.
4. Sesuaikan bahasa pemrograman dengan *mobile platform* yang diinginkan, misalkan *Android* menggunakan *Java* atau *IOS* menggunakan *Objective C*.
5. Bila anda mematok diri anda sebagai seorang programmer, usahakan bekerja sama dengan orang yang mempunyai kemampuan dalam membuat art asset. Kemampuan artist game yang dibutuhkan adalah animasi, desain karakter dan *environment art*.
6. Dengan kombinasi minimal 1 orang programmer dan 1 orang artist, kita sudah dapat membuat sebuah *game*

Dan juga selain ke 6 di atas yang saya ambil dari suatu web, sekarang saya akan menambahkan juga apa saja yang di butuhkan oleh 'a *maker game*' dalam membuat sebuah game , yaitu :

- Tentukan genre *game*
- Tentukan *tool* yang ingin di gunakan
- Tentukan *game play game*
- Tentukan *grafis game*
- Tentukan suara yang akan di gunakan
- Tentukan proses perencanaan akan seperti apa
- Tentukan Proses pembuatan
- Lakukan publishing

Dan tentu saja setelah menentukan langkah - langkah apa saja yang akan di butuhkan dalam sebuah *game* , sekarang saya akan menuliskan bahasa pemograman apa saja yang bisa di gunakan untuk membuat sebuah *game* , yaitu^[9]:

- C++ , yang saya tahu masih banyak *game* yang di buat oleh pemograman yang satu ini walaupun sudah bisa di bilang C++ adalah bahasa pemograman yang terlama di pakai dalam pembuatan sebuah *game*

- C# , bahasa pemrograman ini bisa juga di buat untuk pemrograman *game web*, tetapi C# lebih ke *unity3D*
- *Java* , bahasa pemrograman *java* bisa di buat untuk pemrograman *game* pada *mobile phone* , dan juga membuat desktop games
- *Actionscript3* , ini adalah yang paling umum dalam pembuatan *web games* berbasis *flash*.
- *objective - C* , bahasa pemrograman ini untuk membuat game berbasis *mac os* , karena *objective c* ini adalah bahasa pemrograman untuk *apple*.

Dan setelah itu saya akan menuliskan tentang jenis - jenis *game* , yaitu :

- *Action games* , ex : *sudoku*
- *Real Time Strategy (RTS)* ex : *WarCraft* , *DoTA*
- *Role Playing Games (RPG)* , mungkin ini lah *genre games favorite di indonesia* , ex : *Ragnarok* , *Final Fantasy dll* .
- *Real World Simulation* , ex : *Championship Manager*
- *Constrution and Management* , ex : *The Sims* , *sims city*
- *Slide Scrolling games* , ex : *mario boss* , *metal slug*
- *Adventure Games*
- *Puzzle Game*

2.2.2 Definisi Game

Berikut adalah beberapa definisi *game* oleh para ahli dari berbagai sumber:^[5]

1. *Game* memiliki "akhir dan cara": sebuah tujuan, hasil dan serangkaian peraturan untuk mencapainya. (David Parlett)^[5]
2. *Game* adalah aktivitas yang melibatkan keputusan pemain, mencapai tujuan dalam "konteks terbatas" [yaitu aturan] (Clark C. Abt)^[5]
3. *Game* memiliki enam sifat: "bebas" (bermain adalah pilihan, bukan kewajiban), "terpisah" (waktu dan tempat telah ditetapkan terlebih dahulu), memiliki hasil yang tidak pasti, "tidak produktif" (artinya tidak menghasilkan barang atau kekayaan perhatikan bahwa bertaruh

memindahkan kekayaan antar pemain tapi tidak menciptakannya), dipayungi sebuah aturan, dan "pura-pura" (dibarengi dengan kesadaran bahwa *game* bukan Kehidupan Nyata, tapi semacam realita yang terpisah tapi dimiliki bersama). (Roger Callois)^[5]

4. *Game* adalah "upaya sukarela untuk mengatasi rintangan yang tidak perlu". Definisi ini disukai mahasiswa saya. Terdengar agak beda tapi mencakup banyak konsep dari definisi sebelumnya: Definisi ini mencakup sukarela, ada tujuan dan aturan. Soal "rintangan yang tidak perlu" mensiratkan inefisiensi yang disebabkan oleh aturan secara sengaja contohnya, jika tujuan dari *Tic Tac Toe* adalah menulis tiga noktah secara berurutan dalam satu baris, kolom atau diagonal, cara paling mudah adalah menuliskannya pada giliran pertama dan merebut kertas dari lawan. Tapi hal itu tidak dilakukan, karena peraturan menghalangi hal itu..dan dari peraturan inilah muncul permainannya. (Bernard Suits)^[5]
5. *Game* memiliki empat sifat. Yaitu "sistem formal, tertutup (ini artinya *game* memiliki peraturan; "formal" berarti bisa didefinisikan, bukan mengenakan jas dan dasi); mengandung interaksi; melibatkan konflik; dan aman untuk dilakukan, paling tidak dibandingkan hal yang diwakilinya (contohnya, Football Amerika bukanlah hal yang benar-benar aman, cedera sering terjadi, tapi sebagai *game* ini adalah perwakilan abstrak dari perang, dan jelas lebih aman daripada menjadi seorang tentara di medan tempur). (Chris Crawford)^[5]
6. *Game* adalah "sebentuk karya seni di mana peserta, disebut Pemain, membuat keputusan untuk mengelola sumberdaya melalui *game* tokens demi mencapai tujuan." Definisi ini mencakup beberapa konsep yang tidak terlihat dalam definisi lain sebelumnya: *game* adalah karya seni, melibatkan pengambilan keputusan dan manajemen sumber daya, dan memiliki "tokens" (bena dalam *game*). Juga ada konsep yang sudah akrab soal tujuan. (Greg Costikyan)^[5]

7. *Game* adalah "sistem tempat pemain melakukan konflik bohongan, ditentukan oleh aturan, yang memberi hasil terukur" ("terukur" di sini artinya, sebagai contoh, ada konsep "menang" dan "kalah"). Definisi ini dari buku *Rules of Play* karya Katie Salen dan Eric Zimmerman.^[5]
8. *Game* atau permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah, biasanya dalam konteks tidak serius atau dengan tujuan refreshing. (Dawang Muchtar,2005)^[1]

2.2.3 Sejarah Game

Perkembangan *game* memang begitu pesat. Dunia *game* diawali dengan *console-console* pendahulu seperti Atari, Nintendo, Super Nintendo (SNES), dan SEGA yang menampilkan game-game 2 dimensi yang cukup sederhana namun untuk di zamannya banyak diminati oleh masyarakat.

Bahkan, perkembangan *game* ini juga ikut merambah ke komputer PC (*Personal Computer*), yang kini pun telah menjadi *console game* yang cukup diperhitungkan. Saat ini perkembangan *games* di komputer sangat cepat.^[4]

2.2.4 Tipe mesin Game

Berdasarkan jenis *platform* atau alat yang digunakan untuk memainkan sebuah *game*^[4], maka dapat digolongkan sebagai berikut:

1. Machine Arcade *Game*

Di Indonesia orang-orang jauh lebih mengenal dengan sebutan *díng dong*, biasanya berada di tempat ramai seperti mall dan memiliki *box* atau mesin yang memang khusus di design untuk jenis video games tertentu dan tidak jarang bahkan memiliki fitur yang dapat membuat pemainnya lebih menikmati dan terbawa suasana *game*, contoh menggunakan pistol, kursi khusus pembalap F1, menggunakan setir dan tuas presneleng dan tunggangan motor beserta transmisi visual.

2. *Console Game*

Video *game* rumahan yang menggabungkan mesin *game* (*console*) dengan televisi seperti *console* tercanggih yang menggunakan 256 bit, Sony Playstation 4 (rilis 2013), X-Box One dan Nintendo WII.

3. *Portable HandleGame*

Game yang di mainkan pada gadget yang bisa di bawa kemanapun, tentu saja menyenangkan karena bisa bermain *game* di manapun yang kita mau di waktu senggang dari kesibukan. Contoh dari *portable game* saat ini seperti *Sony PSP* dan *Nintendo DS*.

4. *Personal Computer Game*

Game yang dimainkan menggunakan perangkat personal komputer, kebanyakan *game* yang di buat untuk perangkat ini cenderung ber genre FPS (*First Person Shooter*) dan MMORPG *online*.

2.2.5 Genre Game

Game terbagi dalam beberapa gaya permainan yang lebih dikenal dengan nama *genre*. Genre *game* bisa terdiri dari sebuah genre saja atau bisa merupakan gabungan dari dua atau lebih *genre*. Perbedaan genre itu dilakukan untuk menemukan genre baru yang lebih menarik daripada biasanya^[2]. Berikut beberapa jenis genre yang biasa dimainkan, yaitu:

1. Action

Tipe *game* dengan fitur utama berupa banyaknya aksi di mana pemain harus memiliki keterampilan reaksi yang cepat untuk menghindari musuh atau menghindari rintangan. Pengembang *game* tipe ini perlu memastikan *game* yang dibuat dioptimasi sehingga pemain memiliki pengalaman bermain yang baik, yang tidak terganggu oleh *delay* proses yang lama.

2. Adventure

Tipe *game* yang umumnya membuat pemain harus berjalan mengelilingi suatu tempat yang terkondisi, seperti sebuah istana, gua yang berkelok, dan planet yang jauh. Pemain melakukan navigasi suatu area, mencari pesan-pesan rahasia, memperoleh obyek yang memiliki kemampuan yang bervariasi, bertempur dengan musuh, dan lain-lain. Untuk membuat *game* ini, diperlukan perencanaan yang akurat sehingga memiliki alur cerita yang menarik bagi pemain.

3. Sport

Tipe *game* yang berupa kompetisi antara dua pemain atau lebih, di mana pemain dapat berupa individual atau tim. Contoh *game* tipe ini antara lain sepakbola, bola basket, tenis, dan bilyard. Tergantung seberapa cepat permainan yang terjadi, aplikasi *game* perlu dioptimalkan.

4. Role Playing *Game* (RPG)

Tipe *game* yang seringkali berupa multi-player *game* di mana setiap pemain memiliki karakter dengan kemampuan, kekuatan, dan kelemahan yang spesifik. Para pemain saling berkompetisi, berinteraksi, dan bertempur satu sama lain. Tampilan grafis yang khas untuk setiap karakter pemain ditambah dengan *storyline* yang mendebarkan akan sangat menarik dan memberikan pengalaman yang berbeda di dalam bermain.

5. Platform

Tipe *game* yang mengharuskan pemain mengarahkan suatu obyek dengan melalui berbagai tahap atau tingkatan area untuk menyerang musuh dan menghindari terhadap serangan. Tipe *game* ini sedikit serupa dengan action *game*, tetapi aksinya tidak secepat action *game*. Teknik *collision detection* sangat sering dimanfaatkan pada tipe *game* ini.

6. Puzzle

Tipe *game* yang umumnya membuat pemain menggunakan kemampuan berpikirnya sebagai pengganti keterampilan reaksi yang cepat

karena terdapat rahasia yang perlu dipecahkan. *Game* ini lebih bersifat statis dibanding *action game*. Pembuatan *game* tipe ini seringkali ditunjang dengan algoritma *artificial intelligence*(AI).

7. Sandbox

Tipe *game* yang umumnya ditujukan untuk menjelajahi suatu kota atau tempat dan bebas berinteraksi dengan objek disekitarnya. Mungkin *game* ini bisa di bilang mirip RPG tapi *Sandbox* sangat berbeda dengan RPG *game*, perbedaannya adalah tidak adanya peningkatan level karakternya.

8. *First Person Shooter* (FPS)

Tipe *game* yang menggunakan sudut pandang orang pertama untuk membidik atau membunuh musuh, sehingga kita hanya melihat tangannya saja dan tidak melihat tubuh karakter yang dimainkan.

9. *Third Person Shooter* (THS)

Tipe *game* yang mirip dengan FPS, tapi bedanya tipe *game* ini menggunakan sudut pandang orang ke-3, sehingga kita bisa melihat seluruh tubuh karakter yang dimainkan.

10. Music

Tipe *game* yang menuntut pemainnya untuk menekan tombol sesuai dengan tombol yang ada di layar dengan diiringi dengan musik.

11. Fighting

Tipe *game* yang intinya harus menjatuhkan lawan tandangnya, entah itu dengan pukulan, tendangan, combo, maupun dengan jurus spesial.

12. Strategy

Tipe *game* yang mengharuskan pemainnya menggunakan taktik dan strategi untuk jeli dalam melihat setiap peluang, kelemahan musuh dan bijaksana dalam menggunakan sumber daya yang ada.

13. Simulation

Tipe *game* yang memberikan pengalaman atau interaksi sedekat mungkin dengan kendaraan yang aslinya, meskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut.

14. Racing

Tipe *game* yang tujuannya adalah mencapai garis finish dari suatu race, dalam *game* ini biasanya pemain dapat memilih dan membeli kendaraan, mendandani, mengupgrade mesin dsb.

2.3 Pengertian Multimedia

2.3.1 Definisi Multimedia

Multimedia merupakan pemanfaatan komputer untuk membuat dan menggabungkan *teks, grafik, audio, gambar bergerak (video dan animasi)* dengan menggabungkan *link dan tool* yang memungkinkan pemakai melakukan navigasi, berinteraksi, bereaksi dan berkomunikasi. Dalam definisi ini terkandung empat komponen penting multimedia. Komponen pertama adalah harus ada komputer yang mengkoordinasi apa yang di lihat yang di dengar yang berinteraksi dengan pengguna.^[10]

Disini dapat digambarkan bahwa *multimedia* adalah suatu kombinasi data atau media untuk menyampaikan suatu informasi sehingga informasi itu tersaji dengan lebih menarik.^[10]

“*Multimedia* adalah kombinasi dari komputer dan video (Rosch, 1996) atau Multimedia secara umum merupakan kombinasi tiga elemen, yaitu suara, gambar dan teks (McCormick 1996) atau *Multimedia* adalah kombinasi dari paling sedikit dua media input atau output dari data, media dapat audio (suara, musik), animasi, video, teks, grafik dan gambar (Turban dkk, 2002) atau Multimedia merupakan alat yang dapat menciptakan presentasi yang dinamis dan interaktif yang mengkombinasikan teks, grafik, animasi, audio dan gambar video (Robin dan

Linda, 2001). *Multimedia* adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, audio, gambar bergerak (video dan animasi) dengan menggabungkan link yang memungkinkan pemakai melakukan navigasi, berinteraksi, berkreasi dan berkomunikasi. Dalam definisi ini terkandung empat komponen penting multimedia. Pertama, harus ada komputer yang mengkoordinasikan apa yang dilihat dan didengar. Kedua, harus ada *link* yang menghubungkan pemakai dengan informasi. Ketiga, harus ada alat navigasi yang membantu pemakai menjelajah jaringan informasi yang saling terhubung. Keempat, multimedia menyediakan tempat kepada pemakai untuk mengumpulkan, memproses, dan mengkomunikasikan informasi dengan ide. Jika salah satu komponen tidak ada, bukan multimedia dalam arti luas namanya. Misalnya, jika tidak ada komputer untuk berinteraksi, maka itu namanya media campuran, bukan multimedia. Kalau tidak ada alat navigasi yang memungkinkan untuk memilih jalannya suatu tindakan maka itu namanya *film*, bukan multimedia. Demikian juga kita tidak mempunyai ruang untuk berkreasi dan menyumbangkan ide sendiri, maka nama televisi, bukan multimedia. Dari beberapa definisi di atas, maka multimedia ada yang *online (Internet)* dan multimedia ada yang *offline (tradisional)*.^[10]

2.4 Adobe Photoshop CS5

2.4.1 Pengertian Adobe Photoshop CS

Adobe Photoshop adalah software yang dibuat oleh perusahaan *Adobe System*, yang di khususkan untuk pengeditan foto atau gambar dan pembuatan *effect*. Perangkat lunak ini banyak digunakan oleh Fotografer Digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar. Meskipun pada awalnya *Photoshop* dirancang untuk menyunting gambar untuk cetakan berbasis-kertas, *Photoshop* yang ada saat ini juga dapat digunakan untuk memproduksi gambar untuk *World Wide Web*.

Software Adobe Photoshop merupakan *software* yang dapat digunakan pada aktivitas digital imaging dengan dilengkapi *tool-tool* fotografi yang powerful, dan memiliki terobosan kemampuan untuk menyeleksi gambar yang kompleks (*complex image selections*), memperoleh hasil yang realistis (*realistic painting*), dan memberikan sentuhan yang pintar (*intelligent retouching*).

Kelengkapan fitur yang ada di dalam *Photoshop* inilah yang akhirnya membuat *software* ini banyak digunakan oleh desainer grafis profesional. Dan mungkin juga sampai saat ini masih belum ada *software* desain grafis lain yang bisa menyamai kelengkapan fitur dalam *Photoshop*

2.4.2 Bagian - Bagian Photoshop

Berikut gambar 2.1 tampilan dari photoshop CS:



Gambar. 2.1 Photoshop CS5

Keterangan :

a. Menu Bar

Menu bar adalah menu pulldown yang berisi perintah-perintah dalam Photoshop seperti *menu File, Edit, Image, Layer, Select, Filter, View, Window, dan Help*. Baris menu ini terkelompok berdasarkan topiknya.

b. Toolbox

Toolbox adalah alat-alat yang digunakan untuk memodifikasi *image* (gambar atau foto). Alat-alat ini juga dikelompokkan menurut jenisnya.

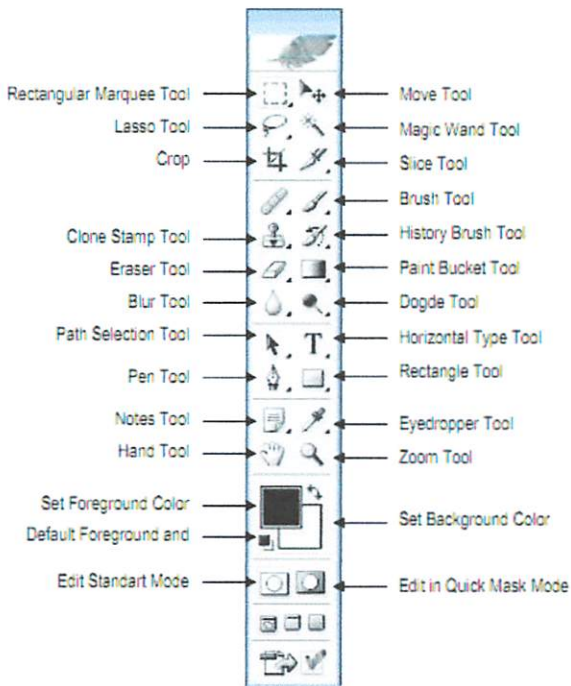
c. Canvas

Canvas adalah bidang yang digunakan sebagai tempat untuk meletakkan *image*. Biasanya ukuran canvas akan sama dengan ukuran *image*, tetapi dalam *Photoshop* kita dapat merubah ukuran canvas dan *image* sesuai dengan kebutuhan. Kalau kita memunculkan canvas baru biasanya ada tiga pilihan yaitu canvas yang putih, berwarna dan transparan.

d. Window Pallette

Window pallette adalah window yang berguna untuk memilih atau mengatur berbagai parameter pada saat menyunting *image* dalam *Photoshop*. Untuk menampilkan *Window Pallette* dapat kita lakukan dengan cara memilih menu *Window* kemudian pilih *pallette* yang dimunculkan.

2.4.3 Toolbox dalam Photoshop CS



Gambar. 2.2 ToolBox Photoshop CS

Keterangan :

a. *Rectangular Marquee Tool*

Digunakan untuk menyeleksi atau mem-blok bagian dari suatu *image* dengan bentuk seleksi persegi. Didalamnya ada *elliptical Marquee Tool*, *Single Row, Marquee Tool*, dan *Single Column Marquee Tool*.

b. *Lasso Tool*

Digunakan untuk menyeleksi atau mem-blok bagian dari suatu *image* dengan bentuk bebas dengan cara mendrag bagian *image* yang akan diseleksi. Di dalamnya ada *Poligonal Lasso Tool*, dan *Magnetic Lasso Tool*.

c. *Crop*

Digunakan untuk menyeleksi bagian dari suatu *image* dengan bentuk seleksi persegi dan membuang bagian yang tidak diinginkan.

d. *Clone Stamp Tool*

Digunakan untuk mengambil contoh dari suatu *image* dan membuat copynya dimanapun.

e. *Eraser Tool*

Digunakan untuk menghapus *image*. Ini juga dapat digunakan untuk mengembalikan sebuah *image* menjadi sebuah daerah awal.

f. *Blur Tool*

Digunakan untuk mengaburkan bagian dari suatu *image*.

g. *Path Selection Tool*

Digunakan untuk mempercerah bagian dari suatu *image*.

h. *Pen Tool*

Digunakan untuk merancang atau mengedit *path*.

i. *Notes Tool*

Digunakan untuk membuat catatan pada suatu bagian *image*.

j. *Hand Tool*

Digunakan untuk menggeser *image* jika lebar *image* melebihi ukuran *window*.

k. *Set Foreground Color*

Digunakan sebagai indikator warna *foreground* yang aktif dan sekaligus dapat digunakan untuk mengganti warna *foreground* dengan warna yang lain.

l. *Default Foreground and Background Color*

Digunakan untuk mengembalikan warna standart yaitu *foreground* hitam dan *background* putih.

m. *Edit Standart Mode*

Digunakan untuk membuat seleksi dan menampilkan standar painting dan editing.

n. *Move Tool*

Digunakan untuk memindah bagian dari *image* yang di blok atau layer diantara *image* atau *image* lainnya.

- o. *Magic Wand Tool*
Digunakan untuk membuat seleksi atau blok secara otomatis yang hampir sama dengan warna areanya.
- p. *Slice Tool*
Digunakan untuk membuat atau memblok setiap area *rectangular* dari sebuah *image*.
- q. *Brush Tool*
Digunakan untuk menggambar dengan warna *foreground* dengan menggunakan standar brushes atau *custom brushes*.
- r. *History Brush Tool*
Digunakan untuk menggambar di atas sebuah *image* dengan daerah sebelumnya yang telah ditentukan dalam rangka membenarkan kesalahan atau untuk menggambar di atas sebuah *image*.
- s. *Paint Bucket Tool*
Digunakan untuk memberi warna suatu bagian dari *image* dengan warna *foreground*.
- t. *Dodge Tool*
Digunakan untuk mencerahkan bagian dari suatu *image*.
- u. *Horizontal Type Tool*
Digunakan untuk membuat teks pada suatu *image*.
- v. *Rectangle Tool*
Digunakan untuk menggambar kotak pada suatu *image*.
- w. *Eyedropper Tool*
Digunakan untuk memilih warna *foreground* maupun *background* dari warna yang ada pada sebuah *image*.
- x. *Zoom Tool*
Digunakan untuk memperbesar atau memperkecil tampilan sebuah *image*. Perubahan tampilan menggunakan *zoom tool* ini tidak mempengaruhi ukuran pixel dalam sebuah *image*.
- y. *Set Background Color*

Digunakan sebagai indikator warna *background* yang aktif dan sekaligus dapat digunakan untuk mengganti warna *background* dengan warna yang lain.

z. *Edit in Quick Mask Mode*

Digunakan untuk merancang dan mengedit *selection mask* sementara.

2.5 Sistem Operasi Android

2.5.1 Pengertian Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan *platform* yang bersifat *open source* bagi para pengembang untuk menciptakan sebuah aplikasi. Awalnya, *Google Inc.* mengakuisi *Android Inc.* yang mengembangkan *software* untuk ponsel yang berada di *Palo Alto, California* Amerika Serikat. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, yaitu konsorsium dari 34 perusahaan *hardware, software,* dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.* Telepon pertama yang memakai sistem operasi Android adalah *HTC Dream*, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan *Android.* ^[2]

2.5.2 Versi Android

Pada 9 Februari 2009, Google merilis Android versi 1.1 dan *Android* sampai saat ini telah melakukan pembaharuan dan merilis *android* versi terbarunya, berikut versi-versi *android* :

1. Android versi 1.1 (Bender)

Android versi ini dilengkapi dengan pembaharuan :

- Estetis pada aplikasi.
- Jam alarm.
- *Voice search* (pencarian suara).

- Pengiriman pesan dengan Gmail.
- Pemberitahuan *email*

2. Android Versi 1.5 (Cupcake)

Pada 30 April 2009, Google kembali merilis telepon seluler dengan menggunakan *Android dan SDK (Software Development Kit)* dengan versi 1.5 (*Cupcake*) dibangun di atas *Linux Kernel 2.6.27*. Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini adalah: ^[2]

- Kemampuan merekam dan menonton video dengan modus kamera;
- Mengunggah *video* ke *youtube* dan gambar ke *Picasa* langsung dari telepon;
- Dukungan *bluetooth A2DP*;
- Kemampuan terhubung secara otomatis ke *headset bluetooth*, animasi layar dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

3. Android Versi 1.6 (Donut)

Donut (versi 1.6) dibangun di atas *Linuxkernel 2.6.29* dirilis pada 15 September 2009 dengan beberapa pembaharuan yaitu: ^[2]

- Menampilkan proses pencarian yang lebih baik dibanding sebelumnya;
- Penggunaan baterai indikator dan kontrol *applet VPN*.
- Galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus.
- Kamera, *camcorder* dan galeri yang diintegrasikan.
- CDMA / EVDO, 802.1x, VPN, *gestures*, dan *text-to-speech engine*;
- Kemampuan *dial contact*;
- Teknologi *text to change speech*;
- Pengaturan resolusi VWGA.

4. Android Versi 2.0/2.1 (Eclair)

Pada tanggal 26 Oktober 2009 kembali diluncurkan ponsel *Android* dengan versi 2.0/2.1 (Eclair) dibangun di atas *Linuxkernel 2.6.29*, perubahan yang dilakukan adalah: ^[2]

- Peningkatan *hardware*.
- Peningkatan *google maps* 3.1.2.
- Perubahan UI dengan *browser* baru dan dukungan HTML5.
- Daftar kontak yang baru.
- Dukungan *flash* untuk kamera 3,2 MP.
- *Digital zoom*, dan *bluetooth* 2.1.

5. Android Versi 2.2 (Froyo)

Android 2.2 atau yang biasa dikenal dengan *Froyo* dirilis pada tanggal 20 Mei 2010, *Froyo* dibangun di atas *Linux kernel 2.6.32*, *Froyo* memang merupakan versi terbaru dari sistem operasi *Android* yang telah dirilis oleh *google* untuk melengkapi versi terdahulu. Walaupun secara resmi telah dirilis oleh *google*, namun tidak semua ponsel *Android* dapat menggunakan *Froyo*. Pengguna masih harus menunggu notifikasi resmi yang dikeluarkan masing-masing vendor ponsel. ^[2]

6. Android Versi 2.3 (Gingerbread)

Pada tanggal 6 Desember 2010 *Google* merilis *Android 2.3* dengan sebutan *Gingerbread*, dibangun di atas *Linux Kernel 2.6.35* dengan beberapa pembaharuan sebagai berikut: ^[2]

- Perubahan *user interface*,
- Mendukung ukuran layar WXGA,
- Mendukung nativ SIP VoIP,
- Mendukung *WebM/VP8 playback video*, dan *AAC audio encoding*
- Audio efek baru seperti *reverb*, *equalization*, *headphone virtualization*, dan *bass boost*,
- Peningkatan grafis, *audio* dan input untuk pengembang game.

- Mendukung *Near Field Communication(NFC)*
- Peningkatan fungsi *copy-paste*

7. Android versi 3.0/3.1 (Honeycomb)

Android *Honeycomb* dirancang khusus untuk tablet. *Android* versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet pertama yang dibuat dengan menjalankan *Honeycomb* adalah *Motorola Xoom*.^[2]

8. Android versi 4.0 (ICS: Ice Cream Sandwich)

Versi dirilis pada 19 Oktober 2011 *Ice Cream*, tentu saja kita tahu karena ini adalah minuman atau tepatnya makanan yang sangat disukai terutama oleh anak kecil *Ice Cream* dipakai sebagai nama alias dari *Android* versi 4.0. *Smartphone* yang pertama kali menggunakan OS *Android* ini adalah *Samsung Galaxy Nexus*. Secara teori semua perangkat seluler yang menggunakan versi *Android* sebelumnya, *Gingerbread*, dapat di-update ke *Android Ice Cream Sandwich*. Namun sayangnya sampai saat ini kebanyakan *smartphone* yang menggunakan *Android* ICS merupakan *smartphone* kelas *high-end* yang dijual dengan harga cukup mahal. Mungkin karena alasan inilah distribusi versi *Android* satu ini tidak lebih dari 8% sampai pertengahan tahun 2012 ini.^[2]

9. Android versi 4.1 (Jelly Bean)

Sampai tulisan ini di tulis Versi *Android Jelly Bean* adalah versi *Android* yang terbaru. *Android Jelly Bean* yang diluncurkan pada acara *Google I/O* lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input *keyboard*, desain baru fitur pencarian, *UI* yang baru dan pencarian melalui *Voice Search* yang lebih cepat. Tak ketinggalan *Google Now* juga menjadi bagian yang diperbarui. *Google Now* memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat

mengetahui informasi cuaca, lalu-lintas, ataupun hasil pertandingan olahraga. Sistem operasi *Android Jelly Bean* 4.1 muncul pertama kali dalam produk tablet *Asus*, yakni *Google Nexus 7*.^[2]

Fitur-fitur pada *Android* antara lain adalah :

- *Framework* aplikasi, memungkinkan daur ulang dan penggantian komponen.
- *Browser* terintegrasi berbasis *engine Open Source WebKit* yang juga digunakan di browser *IPhone* dan *Nokia S60v3*.
- Rancangan *handset*. Platform disesuaikan dengan kebutuhan *VGA (Video Graphics Adapter)* yang lebih besar, *library grafik 2D dan 3D* yang berdasarkan pada spesifikasi *OpenGL ES 1.0* serta *layout smartphone* yang tradisional.
- *Multi-touch*. *Android* memiliki dukungan bawaan untuk *multi-touch* yang tersedia pada handset terbaru seperti *HTC Hero*.
Dukungan *hardware* tambahan. *Android* mendukung penggunaan kamera, layar sentuh, *GPS (Global Positioning System)*, pengukur kecepatan, *magnetometer*, *akselerasi 2D bit blits* (dengan orientasi hardware, scaling, konversi *format piksel*) dan akselerasi grafis 3D.

2.5.3 Sejarah Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis *linux* yang mencakup system operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya *google Inc* membeli *android Inc*, yang merupakan pendatang baru yang membuat peranti lunak untuk *posel/smartphone*. Kemudian untuk mengembangkan *android*, di bentuk lah *Open Handset Alliance* konsorium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *google, HTC, Intel, Motorola, Qualcomm. T-Mobile, dan Nvidia*.^[2]

Pada saat perilisan perdana *android*, 5 november 2007, *android* bersama *Open Handset Alliance* menyatakan mendukung pengembangan open source pada perangkat mobile. Di pihak lain, *google* merlis kode-kode android di bawah lisensi *apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.^[2]

Di dunia ini terdapat 2 jenis distributor sistem operasi *android*, pertama yang mendukung penuh dari *google* atau *Google Mail Services* (GMS) dan yang kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung *google* atau dikenal sebagai *Open Handset Distribution* (OHD)

Sekitar September 2007 *google* mengenalkan *nexus one*, salah satu jenis *smartphone* yang menggunakan *android* sebagai system operasinya. Telepon seluler ini di produksi oleh *HTC Corporation* dan tersedia di pasaran pada 5 januari 2010. Pada desember 2008, di umumkan anggota baru yang bergabung dalam program kerja *android* *ARM Holdings*, *Atheros Communication*, di produksi oleh *Asustek Computer INC*, *Garmin Ltd*, *Softbank sony Ericsson*, *Toshiba Corp*, dan *Vodafone Groip Plc*. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, *Android*, Perangkat *Mobile* yang merupakan modifikasi *kernel Linux 2.6*. Sejak *android* di rilis setelah dilakukan berbagai pembaharuan berupa perbaikan bug dan penambahan fitur baru.

Pada masa saat ini sebagian besar vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis *android*, vendor vendor itu antara lain *HTC*, *Motorola*, *Samsung*, *LG*, *HKC*, *Huawei*, *Archos*, *Webstation Camangi*, *Dell*, *Nexus*, *SciPhone*, *WayteQ*, *Sony Ericson*, *LG*, *Acer*, *Philips*, *T-Mobile*, *Nexian*, *IMO*, *Asus* dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi *android*, hal ini karena *android* itu adalah sistem operasi yang *open source* sehingga bebas di distribusikan dan di pakai vendor manapun.

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini *android* menjadi pesaing utama dari *Apple* pada sistem Operasi *Table PC*. Pesatnya pertumbuhan *android* selain factor yang di sebutkan di atas adalah karena *android*

itu sendiri adalah *platform* yang sangat lengkap baik itu sistem operasinya, *aplikasi* dan *tool* pengembangan, *Market* aplikasi android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga *android* terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *devise* yang ada di dunia. [2]

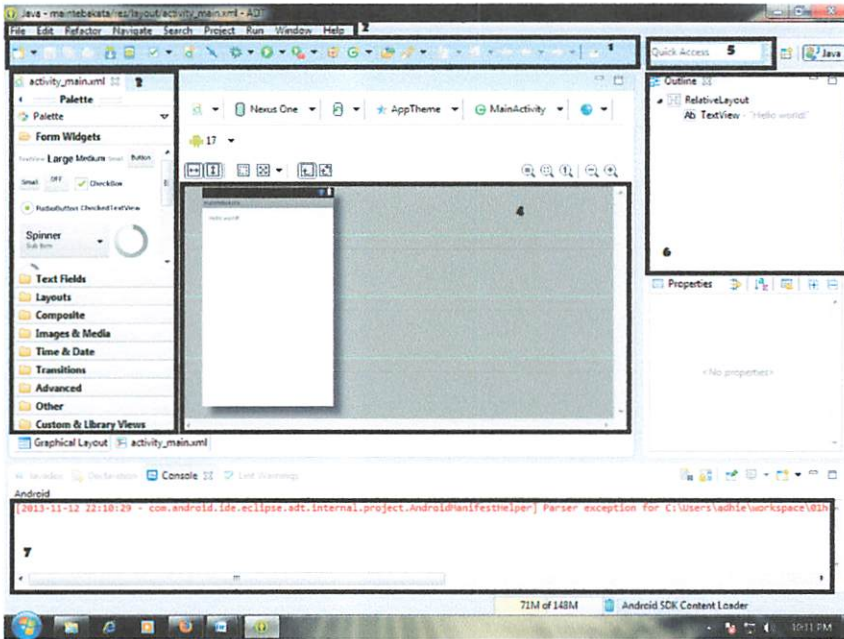
2.6 Eclipse (IDE)

2.6.1 Pengertian Eclipse

Eclipse Integrated Development Environment (IDE) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak. Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak. *Eclipse* adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independent*). Berikut ini adalah sifat-sifat dari *Eclipse* antara lain:

- *Multi-platform* : Target sistem operasi *Eclipse* adalah *Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X*.
- *Multi-language* dikembangkan dengan bahasa pemrograman *Java*, akan tetapi *Eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++, Cobol, Python, Perl, PHP* dan sebagainya.
- *Multi-role* : Selain sebagai IDE untuk pengembangan aplikasi, *Eclipse* dapat digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, uji perangkat lunak, pengembangan web dan sebagainya.

2.6.2 Bagian – Bagian Eclipse



Gambar 2.3 Bagian Eclipse

Keterangan :

1. Tools Bars

Tools Bar adalah alat untuk mengolah program yang sedang di kerjakan di dalam eclipse.

2. Menu bar

Menu Bar adalah *Menu Pulldown* yang berisi perintah-perintah di dalam *eclipse*. Menu Bar terdiri dari *File, Edit, Refactor, Navigate, Search, Project, Run, Windows, Help*.

3. Windows Pallete

Windows Pallete adalah windows yang berguna untuk memilih atau mengatur parameter pada saat mengerjakan program di dalam *eclipse*.

4. Screen

Screen atau layar adalah tempat untuk mengedit atau melihat program yang sedang kita kerjakan di dalam *eclipse juno*.

5. Quick Access

Quick Access atau *search* adalah tempat untuk mencari apa yang di dalam pemograman *eclipse* untuk mempercepat pekerjaan kita.

6. Outline

Outline adalah *history* yang terdapat pada aplikasi *eclipse*.

7. Error Bar

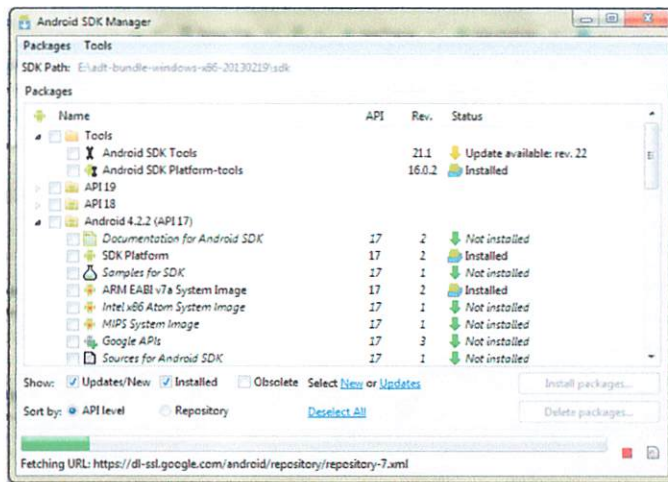
Error bar adalah tempat di mana di munculkan pesan *error* saat kita akan menjalankan program di dalam *eclipse*.

Aplikasi *Eclipse* ini menggunakan bahasa pemograman *Java*, maka dari pada itu metode yang di gunakan berasal dari bahasa pemograman *java*. Metode *random* yang ada dalam permainan menggunakan metode *java* yaitu `int I = 0; I < strSoal.size(); i++`. Dan untuk menampilkan semua konten dalam *game* menggunakan metode bahasa pemograman *Java*.^[2]

2.7 Android SDK

2.7.1 Pengertian Android SDK

SDK atau *Software Development Kit* adalah aplikasi dari *Android* yang memungkinkan pengembang untuk membuat aplikasi untuk *platform Android*. SDK *Android* mencakup sampel proyek dengan *source code*, perangkat pengembangan, *emulator*, dan direktori yang diperlukan untuk membangun aplikasi *Android*. Gambar *android SDK* sebagai berikut:^[2]



Gambar 2.4 Android SDK

2.7.2 ADT

Android Development Tools (ADT) adalah *plugin* untuk *Eclipse IDE* yang dirancang untuk memberikan Anda lingkungan yang terintegrasi untuk membangun aplikasi *Android*. ADT memperluas kemampuan *Eclipse* untuk mempermudah anda mengatur proyek-proyek baru *Android*, membuat GUI aplikasi, menambahkan komponen berdasarkan *Kerangka Android API*, debug aplikasi Anda menggunakan alat *SDK Android*, dan bahkan ekspor *file APL*. *File APK* untuk mendistribusikan aplikasi Anda. Dengan ADT juga kita dapat melakukan *package android (.apk)* yang di gunakan untuk distribusi aplikasi *android* yang kita rancang.^[2]

Mengembangkan aplikasi *android* dengan menggunakan ADT di *eclipse* sangat di anjurkan dan sangat mudah entuk memulai mengembangkan aplikasi *android*. Berikut adalah versi ADT untuk *eclipse* yang udah di rilis :^[2]

- ADT 12.0.0 (July 2011)
- ADT 11.0.0 (June 2011)
- ADT 10.0.1 (March 2011)
- ADT 10.0.0 (February 2011)
- ADT 9.0.0 (January 2011)

- ADT 8.0.1 (December 2010)
- ADT 8.0.0 (December 2010)
- ADT 0.9.9 (September 2010)
- ADT 0.9.8 (September 2010)
- ADT 0.9.7 (May 2010)
- ADT 0.9.6 (March 2010)
- ADT 0.9.5 (Desember 2009)
- ADT 0.9.4 (October 2009)

Semakin tinggi *platform* android yang kita gunakan, dianjurkan menggunakan ADT yang lebih baru, karena biasanya muncul *platform* baru diikuti oleh munculnya versi ADT yang terbaru. Untuk melakukan instalasi ADT di *eclipse* dapat dilakukan secara online maupun offline untuk mendownload ADT ini dapat dilakukan di <http://developer.android.com/sdk/eclipse-adt.html>.^[2]

2.7.3 Arsitektur Android

Secara garis besar arsitektur android dapat dijelaskan dan digambarkan sebagai berikut:

- Applications dan Widgets

Applications dan *Widgets* ini adalah layer di mana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien *email*, program sms, calendar, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman *java*.^[2]

- Application Frameworks

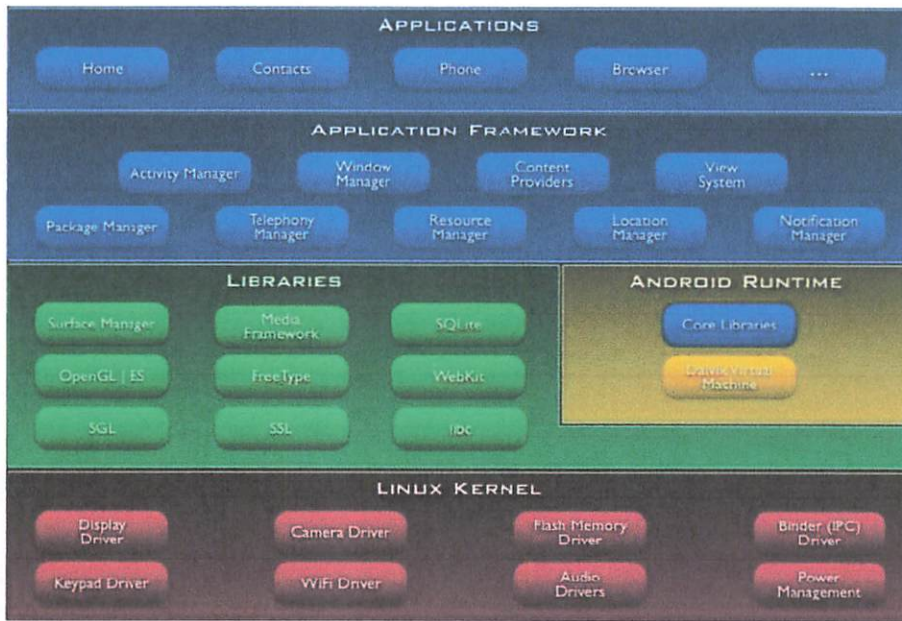
Android adalah “*Open Development Platform*” yaitu android menawarkan kepada pengembang atau member kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan

innovative. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status notification dan sebagainya. Pengembang memiliki akses penuh menuju API *framework* seperti yang di lakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi di rancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah di gunakan (*reuse*). Sehingga bisa kita simpulkan *application framework* ini adalah *layer* di mana para pembuat aplikasi melakukan pengembangan / pembuatan aplikasi yang akan di jalankan di sistem operasi android, karena pada *layer* inilah aplikasi dapat di rancang dat, seperti content *providers* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *application frameworks* adalah sebagai berikut: ^[2]

- a) Views
 - b) Conten Providers
 - c) Resource Manager
 - d) Notification Manager
 - e) Activity Manage
- Libraries
- Libraries* ini adalah *layer* dimana fitur-fitur *android* berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, *layer* ini meliputi berbagai *library C/C++* inti Libe dan SSL, serta: ^[2]
- A) *Libraries* media untuk pemutaran media audio dan video
 - B) *Libraries* untuk manajemen tampilan
 - C) *Librerries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D
 - D) *Libraries SQLite* untuk dukungan database

- E) *Libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*
 - F) *Libraries LiveWebcore* mencakup *modern web browser* dengan *engine embedded web view*
 - G) *Libraries 3D* yang mencakup implementasi *OpenGL ES 1.0 API's*
- **Android Run Time**
Layer yang membuat aplikasi *android* dapat di jalankan dimana dalam prosesnya menggunakan Implementasi *Linux. Dalvik Virtual Machine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi *android*. Di dalam *android Run Time* di bagi menjadi dua bagian yaitu :^[2]
 - *Core Libraries*: Aplikasi *Android* di bangun dalam bahasa *java*, sementara *Dalvik* sebagai virtual mesinnya bukan *virtual machine java*, sehingga di perlukan sebuah *libraries* yang di berfungsi untuk menerjemahkan bahasa *java/c* yang di tangani oleh *Core Libraries*.
 - *Dalvik Virtual Machine*: Virtual mesin berbasis register yang di optimalkan untuk menjalankan fungsi-fungsi secara efisien di mana merupakan pengembangan yang mampu membuat *linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.
 - **Linux Kernel**
Linux Kernel adalah layer di mana inti dari operating system dari *Android* itu berada. Berisi file-file sistem yang mengatur sistem *processing, memory, resource, driver*, dan sistem-sistem operasi kernel release 2.6. jika anda ingin mempelajari *linux karnel* oni, anda dapat mengunduh *linux kernel* secara gratis di situs www.kernel.org^[2]



Gambar 2.5 Arsitektur Android

BAB III

PERANCANGAN DAN ANALISA SISTEM

3.1 Analisis

Analisis sistem merupakan tahap yang paling penting dalam suatu pengembangan sebuah aplikasi, karena kesalahan pada tahap analisis sistem akan menyebabkan kesalahan pada tahap selanjutnya. Dengan adanya proses ini akan dihasilkan suatu gambaran sistem yang kemungkinan memiliki kesalahan-kesalahan ataupun kelemahan-kelemahan sehingga dimungkinkan dilakukan perbaikan.

3.1.1 Analisis SWOT

Dalam hal ini analisis menggunakan metode pengembangan sistem yaitu analisis SWOT (*Strength, Weakness, Opportunities, Threatens*) yaitu dimana dalam merancang sebuah sistem terlebih dahulu lebih baik melihat adanya kekuatan, kelemahan, peluang (kesempatan) dan ancaman (hambatan) itu sendiri. Kekuatan, kelemahan, peluang dan ancaman/hambatan ini jika dipahami dan dijadikan dasar pijakan akan dapat solusi yang tepat didalam mendapatkan sebuah perencanaan yang strategis.

a. Strength (kekuatan)

Strength merupakan kekuatan yang dimiliki oleh sistem baru yang dibuat. Kekuatan atau kelebihan dari game ini adalah dimana game “*Main Kata*” memiliki tampilan yang cukup familiar, sehingga orang akan mudah untuk memainkan permainan ini. Dengan mode pengoprasian secara *touch screen* memudah kan pengguna game memahami permainan ini.

b. *Weakness* (kelemahan)

Weakness merupakan kelemahan yang mungkin dimiliki oleh sistem baru. Dalam hal ini kelemahan yang dimiliki yaitu untuk menerbitkan *game* ini di *Play Store* harus membayar sejumlah uang sebanyak 25\$. Sehingga menjadi kendala tersendiri bagi pengembang. Selain itu untuk mendapatkan *game* ini pengguna perangkat android harus terhubung ke jaringan internet untuk mendownload *game* ini dengan koneksi minimal 3G untuk kelancaran *download*. Sedangkan jaringan 3G sendiri masih belum menjangkau tempat-tempat terpencil di Indonesia.

c. *Opportunities* (Kesempatan)

Opportunity merupakan peluang atau kesempatan yang dapat dilakukan untuk mengembangkan *game* ini. *Game* ini mempunyai gaya permainan yang simple dan mudah dimengerti oleh semua kalangan usia. Untuk menjalankan *game* ini membutuhkan *android* versi *froyo* 2.2 dan dapat di *download* secara gratis di *Play Store*.

d. *Threatens* (Ancaman)

Ancaman yang paling berpengaruh adalah di pandang rendah *game* buatan anak bangsa oleh masyarakat Indonesia, sehingga minat masyarakat untuk *game* asli karya anak bangsa dan memilih *game* buatan luar negeri. Padahal *game* karya anak bangsa sendiri mempunyai kualitas yang cukup menjanjikan. Hal seperti ini lah yang menjadi ancaman bagi pengembang.

3.1.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan cara untuk memahami apa yang harus dilakukan oleh sistem. Tujuan dari tahap analisis adalah memahami dengan baik kebutuhan dari sistem. Kebutuhan sistem dapat diartikan sebagai pernyataan tentang apa yang harus dikerjakan oleh sistem dan pernyataan tentang karakteristik yang harus dimiliki sistem.

3.1.2.1 Kebutuhan Fungsional

Kebutuhan fungsional berisi proses-proses apa saja yang nantinya dapat ditampilkan dan dilakukan oleh *game*. Adapun analisis kebutuhan sistem fungsional dalam *game main kata* meliputi :

- A. *Game* ini memiliki 3 level (*Easy, Medium, Hard*)
- B. *Game* ini memiliki katagori permainan
- C. Di dalam *game* ini menampilkan clue atau petunjuk untuk menjawab
- D. Di *level easy* pemain mendapatkan 3 *clue* atau petunjuk untuk menjawab.
- E. Di *level medium* pemain mendapatkan 2 *clue* atau petunjuk untuk menjawab.
- F. Di *level hard* pemain mendapatkan hanya 1 *clue* atau petunjuk untuk menjawab.
- G. Untuk melanjutkan ke soal berikutnya pemain harus menjawab dengan benar soal yang sedang di tampilkan.
- H. Terdapat sistem point untuk mempermudah menjawab pertanyaan dalam permainan.

3.1.2.2 Kebutuhan nonFungsional

Kebutuan nonfungsional mengidentifikasi batasan dari fasilitas yang disediakan oleh sistem. Kebutuhan nonfungsional bisa dilihat dari aspek operasional (*hardware, software* dan *brainware*) dan akses user.

1. Kebutuhan perangkat keras (*Hardware*)

Perangkat keras merupakan kebutuhan utama sistem dasar membuat *game*. Pembuatan *game* berbasis *android* di butuh kan spesifikasi yang cukup untuk memaksimalkan kinerja dan memudahkan pengembang dalam membuat *game*. Spesifikasi perangkat keras yang di butuhkan dalam membuat *Game Main kata* menggunakan *Eclipse Juno FREE* SDK adalah sebagai berikut :

- a. Processor: AMD Athlon(tm) 64 X2 Dual Core Processor 5000+ (2 CPUs), ~2.6GHz
 - b. Memory: 2048MB RAM
 - c. DirectX Version: DirectX 11
 - d. HDD 260 GB
 - e. 3G HSDPA Modem
 - f. DVD-Super Multi DL drive
 - g. Card name: ATI Radeon HD 4600 Series 1018 MB DDR2
 - h. Keyboard and Mouse
2. Kebutuhan Perangkat Lunak

Kebutuhan Perangkat Lunak yang dimaksud adalah kebutuhan *software* atau perangkat lunak dalam pembuatan *game* ini. Berikut perangkat lunak yang digunakan:

- a. Operating System: Windows 7 Ultimate 32-bit
 - b. [x64](#) Visual C++ 2008 packages dan [x64](#) Visual C++ 2010 packages
 - c. DirectX Version: DirectX 11
 - d. Adobe Photoshop CS5
 - e. Eclipse Juno Free SDK
3. Sumber Daya Manusia (*Brainware*)

Sumber daya manusia yang dibutuhkan dalam pembuatan *game Main Kata* antara lain :

A. Sistem Analisis

Berperan melakukan analisis terhadap permasalahan yang dihadapi, serta merancang solusi pemecahannya dalam bentuk program strategi.

B. Game Designer

Game designer terlibat dalam berbagai macam aspek dalam pengembangan *game*. Tugasnya antara lain, memberi tingkat

kesulitan atau level, judul *game*, aturan main dalam *game*, mendesain *level*, dan lain-lain.

3.1.3 Analisis Kelayakan Sistem

Analisis kelayakan adalah suatu analisis yang akan digunakan untuk menentukan kemungkinan apakah pengembangan proyek sistem multimedia layak diteruskan atau dihentikan.

a. Analisis Kelayakan Teknologi

Dari segi kelayakan teknologi *game* layak karena di peruntukan untuk sistem operasi *android* yang belakangan ini tengah berkembang pesat.

b. Analisis Kelayakan Hukum

Kelayakan hukum dapat dilihat berdasarkan legalitas *software* yang digunakan dan isi atau informasi yang dibangun. *Game* ini dikatakan layak hukum karena *software* yang digunakan bersifat legal, *open source* dan isi yang terkandung dalam *game* tidak menyimpang dari peraturan hukum yang berlaku dan dari segi content tidak mengandung unsur pornografi dan penipuan.

c. Dari segi operasional *game* ini di katakana layak karena masyarakat sudah mampu mengoperasikan *smartphone* yang berbasis *android* dengan lancar. Dan *game* ini tidak memerlukan koneksi *internet* untuk memainkannya, sehingga *game* ini dapat di mainkan di manapun.

3.2 Perancangan Game Main Kata

Pada perancangan sistem *game* “*Main Kata*”, ditulis sesuai pembuatan dokumen desain sebuah *game*. Dokumen desain sebuah *game* sebaiknya menerangkan semua perencanaan dan bahasan yang ada di dalam *game* yang akan dibuat. Dalam desain *game* “*Main Kata*” ini akan diterangkan hal – hal dibawah ini :

1. Keterangan umum *game* “*main kata*”
2. Deskripsi tingkat kesulitan *game* “*main kata*”

3. Deskripsi pemilihan kategori *game* “*main kata*”
4. Deskripsi sistem point pada *game* “*main kata*”
5. Deskripsi *gameplay* pada *game* “*main kata*”
6. Deskripsi kebutuhan *minimum smartphone android*

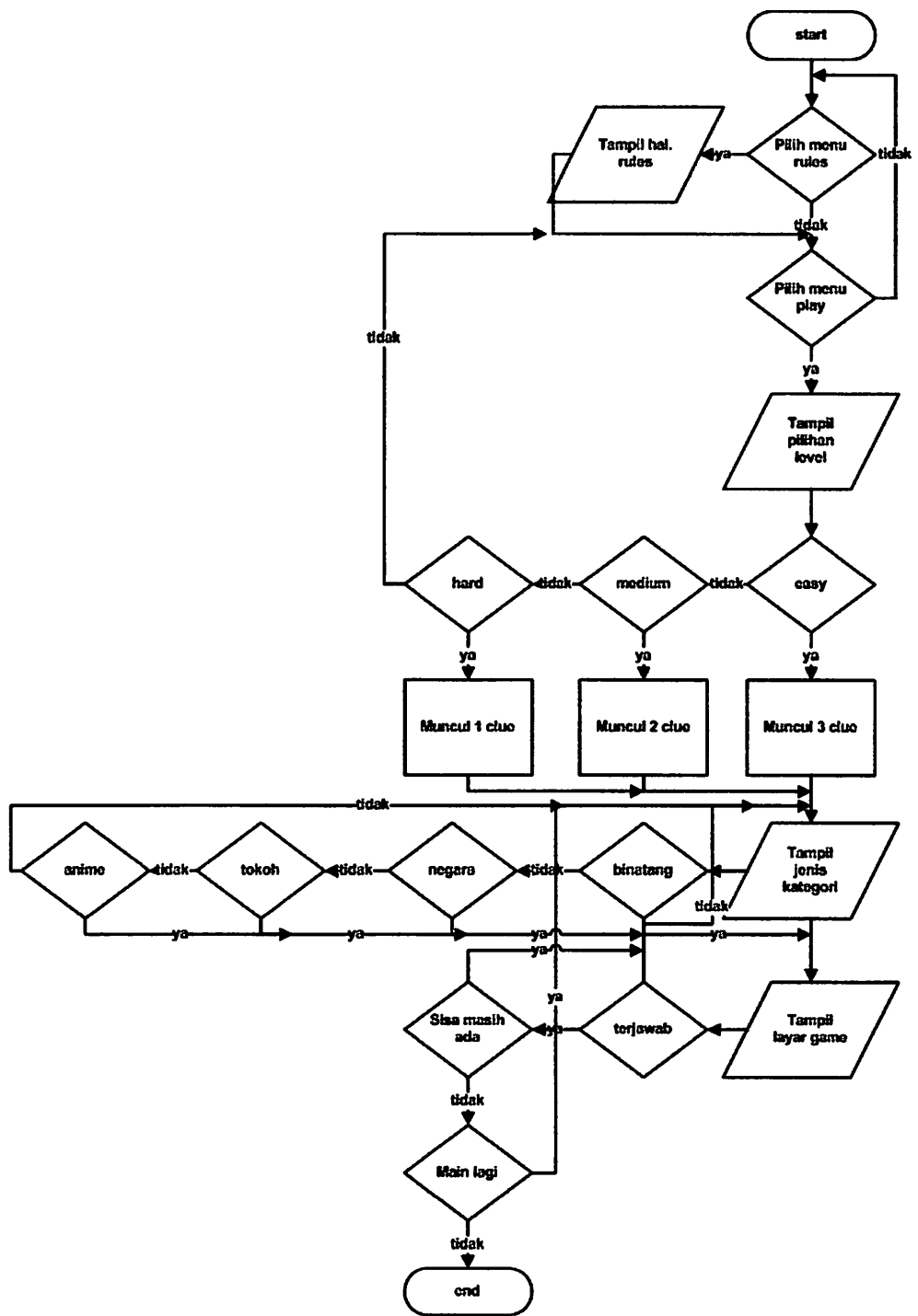
3.2.1 Keterangan Umum Game Main Kata

Game ini bertujuan untuk menghibur dengan gaya permainan yang *simple* dan *edukatif* sehingga tidak cepat bosan karena terdapat tingkat kesulitan yang berbeda mulai dari *easy, medium, hard*.

Ide pembuatan *game* ini sebenarnya terinspirasi dari *game* yang sudah ada sebelum nya, yaitu *game Tebak Kata*. Yang membedakan dengan *game main kata* dengan *game* sebelum nya yaitu di tambahkannya kategori tingkat kesulitan (*Level*). Di *game Tebak Kata* tidak di sertakan *level* sehingga permainan terasa monoton dan tidak ada tantangan yang lebih.

Di sini pembuat juga menyertakan kategori yang lebih variatif dari *game* yang sudah ada. Kategori yang akan di tampilkan meliputi kategori yang umum sampai yang lebih spesifik dan butuh wawasan yang luas untuk menjawab di tingkat kesulitan yang paling tinggi.

Alur permainan dari *game* ini dapat di lihat pada gambar flowchart di bawah ini :



Gambar 3.1 Flowchart game main kata

3.2.2 Deskripsi tingkat kesulitan game “main kata”

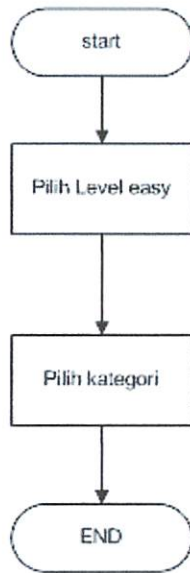
Game main kata ini memiliki beberapa tingkatan kesulitan atau biasa yang kita sebut *level*. Tujuan di penulias memberikan tingkat kesulitan agar pemain tidak cepat bosan dan ada tantangan tersendiri yang harus di selesaikan di tingkat kesulitan yang berbeda.

Berikut tingkat kesulitan yang ada di game main kata :

1. Level Easy

Level easy merupakan tingkat kesulitan atau *level* yang paling mudah untuk di mainkan. Dimana di tingkat kesulitan easy kita di beri tiga(3) *clue* atau petunjuk untuk menjawab soal di *game main kata*. Petunjuk tersebut terdapat serba serbi tentang seputar jawaban. Sehingga semakin banyak petunjuk yang di dapat maka akan semakin mudah untuk menjawab nya.

Alur pemilihan level *easy* pada game “main kata”



Gambar 3.2 Flowchart pemilihan level *Easy*

Untuk menampilkan *button easy* pada permainan menggunakan perintah pada eclipse `ImageButton btnEasy = (ImageButton) findViewById(id.btnEasy);`
`btnEasy.setOnClickListener(new View.OnClickListener() {`

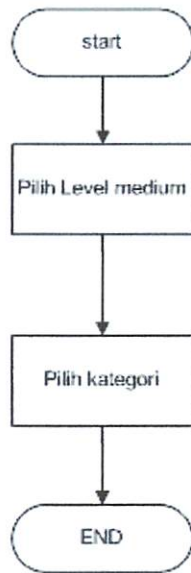
Dan untuk menghubungkan button easy pada permainan di butuh kan perintah

```
public void onClick(View v) {  
    // TODO Auto-generated method stub  
    startLevel("0");  
}
```

2. Level Medium

Level medium merupakan pilihan kedua yang ada di dalam game main kata. Pada tingkat kesulitan medium ini penulis akan memberikan dua(2) buah petunjuk atau clue. Pada level ini pemain di harus kan memikir lebih keras untuk menjawab soal pada game main kata.

Alur pemilihan level medium pada game “main kata”



Gambar 3.3 Flowchart pemilihan level Medium

Untuk menampilkan button medium pada permainan menggunakan perintah pada eclipse `ImageButton btnMed = (ImageButton) findViewById(id.btnMed);`
`btnMed.setOnClickListener(new View.OnClickListener() {`

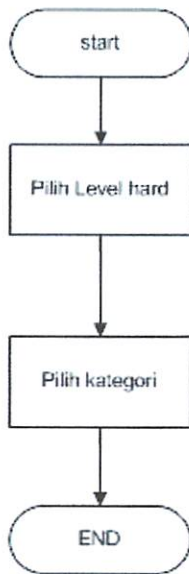
Untuk menghubungkan button medium pada permainan menggunakan perintah `public void onClick(View v) {`

```
// TODO Auto-generated method stub  
startLevel("1");
```

3. Level Hard

Level hard merupakan pilihan terakhir dan memiliki tingkat kesulitan paling tinggi di antara *level* yang lain. Pada tingkat kesulitan *hard* pemain hanya di beri satu(1) petunjuk atau *clue*. Dengan kata lain pemain harus berpikir sangat keras untuk menjawab soal soal pada permainan *main kata*.

Alur permainan pada level *hard* pada game "*main kata*"



Gambar 3.4 Flowchart pemilihan level *Hard*

Untuk menampilkan *button hard* pada permainan menggunakan perintah

```
ImageButton btnHard = (ImageButton) findViewById(id.btnHard);  
btnHard.setOnClickListener(new View.OnClickListener() {
```

Dan untuk menghubungkan *button hard* pada permainan menggunakan perintah

```
public void onClick(View v) {
```

```
// TODO Auto-generated method stub  
startLevel("2");
```

3.2.3 Deskripsi pemilihan kategori game main kata

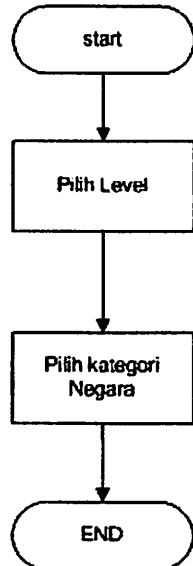
Di dalam *game main kata* terdapat *element* penting yang membedakan dengan *game* sejenis, yaitu terdapat kategori. Kategori ini berupa kumpulan soal yang telah di pilah pilah sesuai dengan kategori yang telah di tetapkan oleh penulis. Di dalam setiap kategori masing masing terdapat sepuluh(10) soal pertanyaan. Sehingga pemain dapat memilih kategori yang di inginkan sesuai kemampuan pemain.

Berikut beberapa jenis kategori yang ada dalam *game main kata* :

1. Kategori Negara

Di golongan kategori ini penulis memasukan beberapa Negara-negara terkenal yang sudah tidak asing lagi di telinga kita. Sehingga pemain dengan mudah menjawab soal yang ada di dalam *game*. Penulis menyarankan pemain bermain di tingkat kesulitan *medium* ke atas agar terdapat tantangan tersendiri dalam bermain.

Alur pemilihan kategori negara dalam *game "main kata"*



Gambar 3.5 Flowchart pemilihan kategori Negara

Tabel 3.1 Jawaban dan Petunjuk kategori Negara

NO	Jawaban	Petunjuk
1	Indonesia	Negara ini memiliki kepulauan terbanyak di dunia diapit dua benua dan dua samudera anggota PBB ke 60
2	Inggris	Sistem pemerintahannya berupa kerajaan Terkenal dengan sepak bolanya Tempat kelahiran David Beckham
3	Amerika	Sistem pemerintahan republik Di kenal dengan julukan paman sam presidennya bernama <i>Barrack Obama</i>
4	Arab Saudi	Negara dari timur tengah Terdapat universitas alazzar Mekah
5	Brazil	Menjadi seting tempat <i>fast farius 5</i> juara piala dunia terbanyak di dunia Tempat piala dunia 2014
6	China	Negara <i>superpower</i> yang ada di asia negeri tirai bamboo <i>Kung Fu</i>
7	Jepang	Produsen kendaraan Honda negeri sakura Terkenal dengan manga dan anime nya
8	Korea Selatan	Negara ini terbagi dua negeri gingseng terkenal dengan <i>boyband</i> nya
9	Italia	Negara ini terdapat kota air tempat di adakanya gladiator menara miring pisa
10	Afrika Selatan	Salah satu Negara di benua afrika <i>Nelson Mandela</i> pernah menjabat presiden Tempat piala dunia 2010

Untuk menampilkan *button* Negara pada permainan menggunakan perintah

```
ImageButton btnJenis1 = (ImageButton) findViewById(id.btnJenis_1);
```

```
btnJenis1.setOnClickListener(new View.OnClickListener() {
```

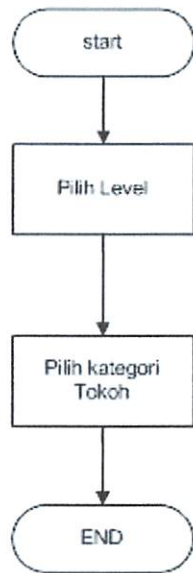
Untuk menghubungkan button Negara pada permainan menggunakan perintah

```
public void onClick(View v) {
    // TODO Auto-generated method stub
    startJenis("0");
}
```

2. Kategori Tokoh

Dalam kategori tokoh terdapat nama-nama dari tokoh dalam negeri maupun luar negeri. Diantaranya terdapat tokoh yang cukup populer di telinga kita. Pada kategori ini penulis menyarankan bermain di tingkat kesulitan *medium* ke bawah, untuk yang berwawasan luas boleh mencoba di tingkat kesulitan *hard*.

Alur pemilihan kategori Tokoh dalam game "main kata"



Gambar 3.6 Flowchart pemilihan kategori Tokoh

Tabel 3.2 Jawaban dan Petunjuk kategori Tokoh

NO	Jawaban	Petunjuk
1	Soekarno	lahir 6 juni 1901 di Surabaya
		Proklamator
		Presiden RI pertama
2	Alexander Graham Bell	lahir 3 maret 1847 di <i>Einburg</i>
		putra dari <i>Alexander Melville</i>

		Penemu pesawat Telepon
3	James Watt	Lahir 19 januari 1736 di <i>Greenock Skotlandia</i>
		meninggal 19 agustus 1819
		Penyempurna mesin uap
4	Mohandas Karamchand Gandhi	Lahir 2 oktober 1869 india
		Pemimpin sepiritul dan politikus dari india
		wafat 30 januari 1948
5	Mark Elliot Zuckerberg	Lahir <i>Westchester County</i> kota <i>New York</i>
		Lulusan <i>Harvard University</i>
		Penemu <i>Facebook</i>
6	Joanne Kathleen Rowling	Lahir 31 Juli 1965 di <i>Chipping Sodbury Inggris</i>
		lulusan <i>Universitas Exeter</i>
		Penulis Serial Harry Potter
7	Christopher Columbus	Lahir 30 Oktober 1451
		Wafat 20 Mei 1506
		Penemu Benua Amerika
8	Joseph John Thomson	Lahir 8 Desember 1856
		profesor fisika di <i>Cambridge</i> dan <i>Royal Institution</i>
		Penemu Elektron
9	Albert Einstein	Lahir 14 Maret 1879
		wafat 18 April 1955
		Penemu Teori Relativitas
10	Fujiko F. Fujio	Lahir 1 Desember 1933 Toyama
		Wafat 23 September 1996
		Penulis manga <i>Doraemon</i>

Untuk menampilkan *button* tokoh pada permainan menggunakan perintah `ImageButton`
`btnJenis2 = (ImageButton) findViewById(id.btnJenis_2);`

```
btnJenis2.setOnClickListener(new View.OnClickListener() {
```

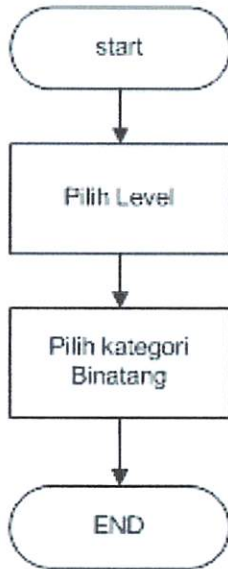
Untuk menghubungkan button tokoh pada permainan menggunakan perintah

```
public void onClick(View v) {
    // TODO Auto-generated method stub
    startJenis("1");
}
```

3. Kategori Binatang

Kategori ini merupakan kategori termudah menurut penulis. Di dalam kategori ini terdapat nama hewan yang sangat umum dan familiar sehingga mudah di tebak oleh pemain. Penulis menyarankan memilih tingkat kesulitan *medium* ke atas agar permainan tidak terlalu mudah.

Alur pemilihan kategori Binatang dalam game “main kata”



Gambar 3.7 Flowchart pemilihan kategori Binatang

Tabel 3.3 Jawaban dan petunjuk kategori Binatang

NO	Jawaban	Petunjuk
1	Katak	Hewan ini memakan serangga kecil
		Kalau masih bayi bernama kecebong
		Biasa hidup di air dan di darat
2	Jerapah	Hewan ini memakan tumbuh tumbuhan

		Letak makanannya di ujung dahan pohon berleher panjang
3	Sapi	Sering di sembelih untuk kurban
		Memakan rerumputan Di peras untuk diambil susunya
4	Nyamuk	berkembang biak di air yang tenang memiliki mulut yang tajam menghisap darah
5	Kangguru	Memiliki dua kaki dan dua tangan Memiliki kantong untuk menggendong bayi hewan asli Australia
6	Cendrawasih	Memiliki corak bulu yang menarik Binatang asli Indonesia Populasinya terdapat di Papua.
7	Cacing	Memiliki kelamin ganda Bertubuh lonjong hidup di tanah
8	Kelelawar	Aktif di malam hari Mampu memancarkan gelombang ultrasonic Logo Batman
9	Semut	Hidupnya berkelompok Hewan ini tergolong hewan kecil suka mengerubung serpihan makanan.
10	Ular	Bertubuh Panjang Memiliki lidah bercabang Berbisa

Untuk menampilkan *button* binatang pada permainan menggunakan perintah

```
ImageButton btnJenis3 = (ImageButton) findViewById(id.btnJenis_3);
btnJenis3.setOnClickListener(new View.OnClickListener() {
```

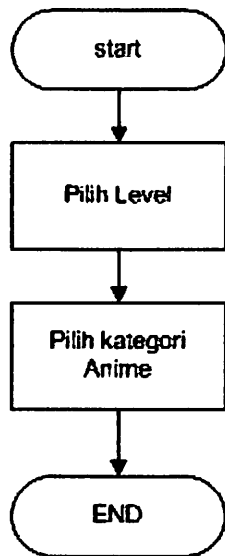
Untuk menghubungkan *button* binatang pada permainan menggunakan perintah

```
public void onClick(View v) {
    // TODO Auto-generated method stub
    startJenis("2");
}
```

4. Kategori Anime

Pada kategori kali ini merupakan kategori yang cukup sulit di karenakan jarang nya masyarakat mengetahui seputar *anime*. Penulis memberikan soal

seputar *character anime* Jepang. Penulis juga menyarankan bermain di tingkat kesulitan *medium* ke bawah agar dapat menjawab semua soal.



Gambar 3.8 Flowchart pemilihan kategori Anime

Tabel 3.4 Jawaban dan petunjuk kategori Anime

NO	Jawaban	Petunjuk
1	Uzumaki Naruto	Ninja Konoha
		Anak dari Hokage ke 4
		Memiliki jinkuruchi ekor 9
2	Monkey D. Dragon	Ketua <i>Revolusioner Army</i>
		Anak dari Monkey D. Garp
		Lahir di East Blue
3	Eren Jaegar	Pasukan <i>Racoon Corp</i>
		Tinggal di Benteng Sina
		Pemeran utama <i>Attack on Titan</i>
4	Conan Edogawa	Detektif Cilik Terkenal
		Tinggal di Chiba Tokyo
		Teman dekat Ran Mouri
5	Hatake Kakashi	Chunin dari Konoha
		Di juluki Ninja Copy
		Guru dari Naruto

6	Nobita	anak kelas 5 SD
		Memiliki kucing dari abad 21
		temen dakat dari Sizuka
7	Roronoa Zorro	Waktu kecil Rival dari Kusina
		Menjadi Pendekar pedang terhebat
		Anggota pertama luffy
8	Trafalgar Law	Lahir di <i>East Blue</i>
		Salah satu anggota Suchibukai
		Di juluki <i>Heart Pirate</i>
9	Light Yagami	Anak Kepala Polisi di Jepang
		Memiliki temen Shinigami bernama Ryuk
		Pemilik Death Note
10	Ryuzaki Lawliet	Detektif yang gemar makan gula
		Berinisial "L"
		Rival dari Light Yagami

Untuk menampilkan *button anime* pada permainan menggunakan perintah

```
ImageButton btnJenis4 = (ImageButton) findViewById(id.btnJenis_4);
```

```
btnJenis4.setOnClickListener(new
```

```
View.OnClickListener() {
```

Untuk menghubungkan button anime pada permainan menggunakan perintah

```
public void onClick(View v) {
```

```
// TODO Auto-generated method stub
```

```
startJenis("3");
```

3.2.4 Deskripsi sistem point game “main kata”

Sistem poin merupakan bantuan di dalam *game* untuk dapat menyelesaikan soal. Poin di dapat dari ketika kita berhasil menyelesaikan atau menjawab soal dalam *game* di tingkat kesulitan terlentu. Dalam *game main kata* penulis memberikan poin ketika pemain menyelesaikan di level *hard* , maka pemain mendapatkan satu(1) poin. Poin hanya dapat di gunakan pada *level hard* saja dan tidak dapat di gunakan pada *level* lain. Poin yang di dapat pada tingkat kesulitan *hard* dapat di simpan atau di akumulasi untuk di gunakan di kategori yang berbeda di tingkat kesulitan *hard*.

Berikut table sistem poin pada game “main kata” :

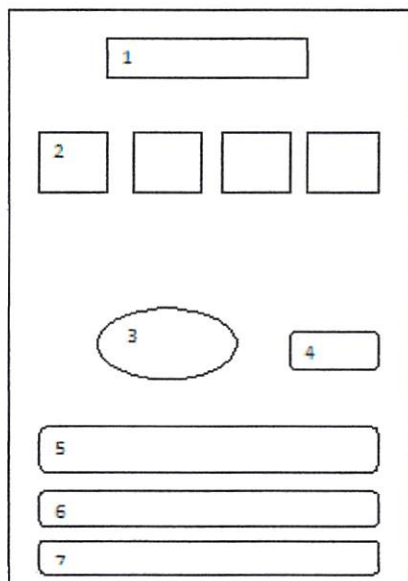
Tabel 3.5 sistem poin

level	Mendapatkan poin	Dapat menggunakan poin
EASY	0	Tidak
MEDIUM	0	Tidak
HARD	1	Ya

3.2.5 Deskripsi *gameplay* pada game “main kata”

Sebenarnya *game main kata* bukan *gameplay* baru melainkan *gameplay* yang telah ada pada jaman dulu. Pada tahun 1990-an kita mengenal *game* legendaries bernama *Tetris*, *game Tetris* ini merupakan cikal bakal *game* sejenis. *Gameplay* permainan *Tetris* ini di sebut dengan istilah *Boardgame*. Pada permainan *game main kata* berbeda dengan *game Tetris*, dimana pemain harus menebak atau menulis sebuah jawaban pada kolom yang telah di sediakan pada board atau papan pada permainan.

Selain itu juga pada *game* ini di terapkan sistem poin dimana poin tersebut dapat di gunakan untuk membantu menyelesaikan soal pada permainan. Poin di peroleh dengan menyelesaikan *level hard* dan hanya dapat di gunakan pada tingkat kesulitan *hard* saja.



Gambar 3.9 Papan permainan

Keterangan :

1. Kolom untuk menampilkan jenis kategori yang sedang di mainkan
2. Kolom untuk menjawab
3. Tombol “Tebak” untuk menjawab permainan
4. Tombol “Point” untuk menggunakan point
5. Kolom petunjuk pertama
6. Kolom petunjuk kedua
7. Kolom petunjuk ketiga

3.2.6 Deskripsi kebutuhan *minimum smartphone android*

Berikut kebutuhan minimum untuk memainkan *game* ini pada *smartphone android* :

Table 3.6 Kebutuhan minimum perangkat *smartphone*

NO	Nama	Spesifikasi
1	Type	TFT capacitive touchscreen, 256K colors
2	Size	240 x 320 pixels, 3.0 inches (~133 ppi pixel density)
3	Alert Type	Vibration, MP3 ringtones
4	Internal	180 MB, 290 MB RAM
5	OS	Android OS, v2.2 (Froyo)
6	CPU	830 MHz ARMv6

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap *game* yang dibangun. Tahapan ini dilakukan setelah analisis dan perancangan selesai dilakukan dan selanjutnya akan diimplementasikan kedalam *engine* pemrograman *Eclipse*. Setelah implementasi maka dilakukan pengujian terhadap *game*. *Game* yang telah dibangun di implementasikan untuk mengetahui apakah sistem di dalam *game* tersebut dapat berjalan sesuai dengan tujuan atau tidak.

4.1 Implementasi

Tahap implementasi merupakan tahap penciptaan *game* dan juga tahap kelanjutan dari kegiatan perancangan *game*. Tahap ini merupakan tahap dimana *game* siap untuk dioperasikan, yang terdiri dari penjelasan mengenali lingkungan implementasi, baik itu lingkungan perangkat keras, maupun lingkungan perangkat lunak, serta implementasi *game*.

4.1.1 Implementasi Sistem

Game main kata ini di implementasikan dengan menggunakan bahasa pemrograman *Java*. *Game* ini menggunakan Eclipse Juno yang dapat mengatur bahasa pemrograman *Java*. Dengan menggunakan *Eclipse* maka didapat beberapa keuntungan, seperti berbasis *open source*, fitur yang lengkap, dan dukungan banyaknya format pendukung pembuatan *game* berbasis android. Sistem - sistem pendukung yang di butuhkan untuk implementasi dan pengujian *game* yang telah dibua terbagi menjadi 2, yaitu kelompok perangkat keras dan kelompok piranti lunak.

4.1.1.1 Spesifikasi Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang digunakan untuk pembangunan *game* dijelaskan pada table 4.1

Tabel 4.1 Perangkat Keras Yang Di Gunakan

Jenis	Spesifikasi perangkat keras
Processor	AMD Athlon(tm) 64 X2 Dual Core Processor 5000+ (2 CPUs), ~2.6GHz
Memory (RAM)	2 GB DDR 2
Hard Disk	260 GB
Monitor	LCD 17"
VGA Card	ATI Radeon HD 4600 Series 1080 GB
Modem	Speed 3G HSDPA
Keyboard	-
Mouse	-

4.1.1.2 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk mengimplementasikan sistem dijelaskan pada tabel 4.2 :

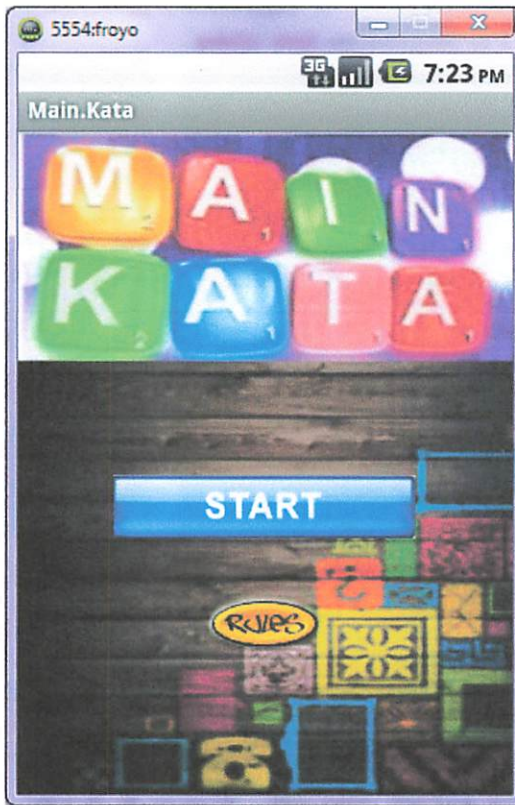
Tabel 4.2 Kebutuhan Perangkat Lunak

Jenis	Spesifikasi Perangkat Lunak
Sistem Operasi	Windows 7
Game Engine	Eclipse Juno
Direct X	DirectX 11
Aplikasi Tambahan	Java Development Kit (JDK) 7u21-windows-i586

Sistem operasi harus sudah terpasang pada komputer, agar *game* yang dibangun dapat dijalankan pada komputer tersebut. Jika pengguna ingin membuka projectnya maka terlebih dahulu harus memasang *Eclipse* dan komponen-komponen yang mendukungnya.

4.2 Implementasi Antarmuka

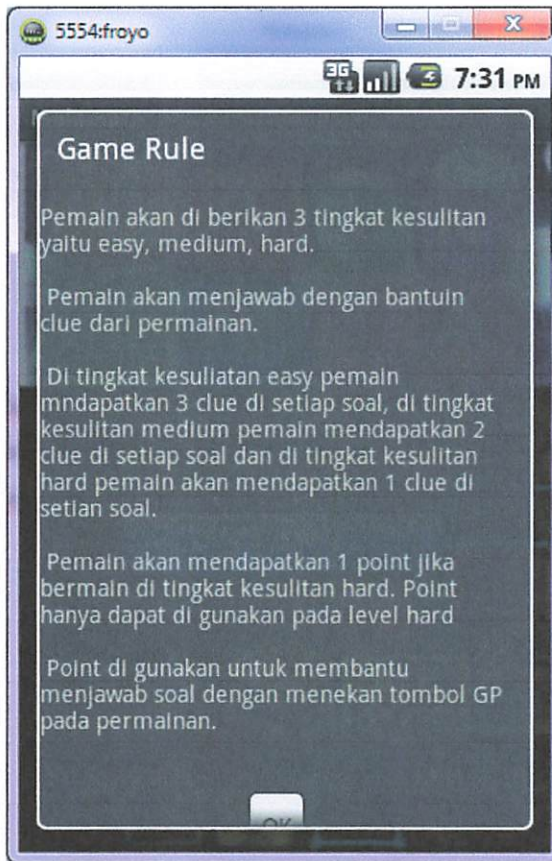
Antarmuka dari game main kata ini menggunakan *Eclipse .exe*. File ini bias ditemui pada *adt-bundle-windows-x86-20130522* dan *folder Eclipse*, Antarmuka menu utama merupakan tampilan menu utama pada saat pertama kali pengguna membuka aplikasi *game*. Implementasi antarmuka untuk *game main menu* dapat dilihat dari gambar 4.1



Gambar 4.1 Tampilan Main Menu

Di dalam *game* ini di bagi menjadi 2 menu utama yaitu *Start* dan *Rules*. *Menu Start* berfungsi untuk memulai permainan sedangkan menu *Rules* berfungsi untuk melihat aturan main di dalam game tersebut. Di dalam menu start terdapat sub menu tingkat kesulitan yang terdiri dari *Easy*, *Medium*, *Hard*. Sedangkan di dalam menu tingkat kesulitan terdapat menu kategori. *Menu Rules*

dapat di lihat pada gambar 4.2. Menu tingkat kesulitan dapat di lihat pada gambar 4.3. Tampilan menu kategori dapat di lihat pada gambar 4.4



Gambar 4.2 Tampilan Menu Rules



Gambar 4.3 Tampilan Menu Tingkat Kesulitan



Gambar 4.4 Tampilan Menu Kategori

4.3 Pengujian

Pengujian merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak. Adapun metode pengujian yang digunakan pada perangkat lunak ini adalah metode pengujian *black box* dan *whitebox*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak, sedangkan *white box* pengujian dengan melihat kedalam modul untuk

meneliti kode - kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak.

4.3.1 Skenario Pengujian Game

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada *game Main Kata*. Skenario pengujian dari *game Main Kata* dapat dilihat pada tabel 4.3 :

Tabel 4.3 Skenario Pengujian

NO	Proses	Jenis Pengujian
1	Main Menu	Black Box
2	Rules	Black Box
3	Start	Black Box
4	Tingkat Kesulitan	Black Box
5	Jenis Kategori	Black Box
6	Tampil Petunjuk	White Box

4.3.2 Kasus dan Hasil Pengujian

Kasus dan hasil uji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box* berdasarkan pengamatan yang dilakukan dan menghasilkan kesimpulan dari pengujian tersebut yang dijelaskan diantaranya sebagai berikut.

1. Pengujian *Main Menu*

Pengujian *main menu* dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada table 4.4

Table 4.4 Tabel pengujian menu utama

Kasus dan Hasil Uji			
Masukan	Yang di Harapkan	Pengamatan	Kesimpulan
Memilih Menu Start	Menampilkan Halaman Start	Berhasil menampilkan halaman start	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
	Memilih Menu Rules	Menampilkan halaman rules	Berhasil menampilkan [<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak

		halaman rules	Berhasil
Memilih menu tingkat kesulitan	Menampilkan halaman tingkat kesulitan	Berhasil menampilkan halaman tingkat kesulitan	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
Memilih menu jenis kategori	Menampilkan halaman jenis kategori	Berhasil menampilkan halaman jenis kategori	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
Memilih menu exit	Menampilkan halaman exit	Berhasil menampilkan halaman exit	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil

2. Pengujian Penyajian Petunjuk di Tiap Tingkat Kesulitan

Pengujian penyajian Petunjuk di Tiap Tingkat Kesulitan dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.5

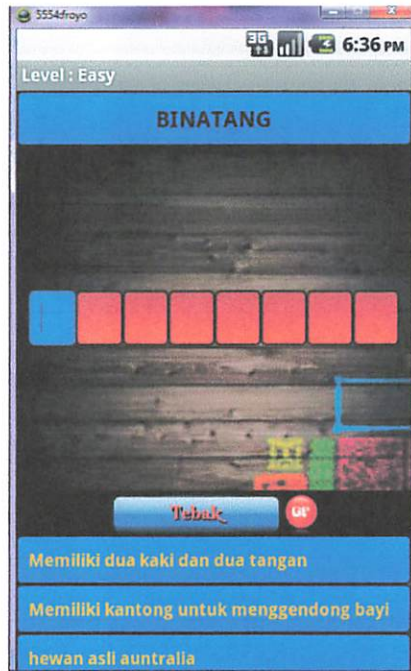
Tabel 4.5 Pengujian Petunjuk di Tiap Tingkat Kesulitan

Kasus dan Hasil Uji			
Masukan	Yang di harapkan	Pengamatan	Kesimpulan
Petunjuk pada tingkat kesulitan easy	Menampilkan petunjuk pada tingkat kesulitan easy	Berhasil menampilkan petunjuk pada tingkat kesulitan easy	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
Petunjuk pada tingkat kesulitan medium	Menampilkan petunjuk pada tingkat kesulitan medium	Berhasil menampilkan petunjuk pada tingkat kesulitan medium	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil

Petunjuk pada tingkat kesulitan hard	Menampilkan petunjuk pada tingkat kesulitan hard	Berhasil menampilkan petunjuk pada tingkat kesulitan hard	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
--------------------------------------	--	---	---



Gambar 4.5 Papan permainan di tingkat kesulitan *Hard* dan *Medium*



Gambar 4.6 Papan permainan pada tingkat kesulitan *Easy*

3. Pengujian Menu Jenis Kategori

Pengujian Menu Jenis Kategori dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.6

Table 4.6 Pengujian Menu Jenis Kategori

Kasus dan Hasil Uji			
Masukan	Yang di harapkan	Pengamatan	Kesimpulan
Menekan jenis kategori binatang	Menampilkan jenis kategori	Berhasil	[<input checked="" type="checkbox"/>] Berhasil
	binatang pada permainan	Menampilkan jenis kategori binatang pada permainan	[<input type="checkbox"/>] Tidak Berhasil
Menekan jenis kategori tokoh	Menampilkan kategori tokoh	Berhasil	[<input checked="" type="checkbox"/>] Berhasil
	pada permainan	menampilkan kategori tokoh	[<input type="checkbox"/>] Tidak Berhasil

		pada permainan	
Menekan jenis kategori negara	Menampilkan kategori Negara pada permainan	Berhasil Menampilkan kategori Negara pada permainan	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
Menekan jenis kategori anime	Menampilkan kategori anime pada permainan	Berhasil Menampilkan kategori anime pada permainan	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
Menekan tombol tebak	Menggunakan tombol tebak pada permainan	Berhasil Menggunakan tombol tebak pada permainan	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil

4. Pengujian Penyajian sistem point

Pengujian penyajian sistem point dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.7

Tabel 4.7 Pengujian sistem point

Kasus dan Hasil Uji			
Masukan	Yang di harapkan	Pengamatan	Kesimpulan
Menggunakan point	Menggunakan dan mendapatkan point pada level hard	Berhasil Menggunakan dan mendapatkan point pada level hard	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil

5. Pengujian Memutar *Sound*

Pengujian Memutar *Sound* dengan kasus dan hasil uji, kemudian diamati dan menghasilkan kesimpulan dapat dilihat pada tabel 4.8

Tabel 4.8 Pengujian Sound

Kasus dan Hasil Uji			
Masukan	Yang di harapkan	Pengamatan	Kesimpulan
Backsound pada permainan	Memutar backsound pada pemilihan menu, tingkat kesulitan dan ketegori	Berhasil Memutar backsound pada pemilihan menu, tingkat kesulitan dan ketegori	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
Sound klik	Memutar sound klik di setiap menekan tombol	Berhasil Memutar sound klik di setiap menekan tombol	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
Sound sukses menjawab	Memutar sound sukses saat berhasil menjawab soal	Berhasil Memutar sound sukses saat berhasil menjawab soal	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil
Sound salah menjawab	Memutar sound gagal saat salah menjawab soal	Berhasil Memutar sound gagal saat salah menjawab soal	[<input checked="" type="checkbox"/>] Berhasil [<input type="checkbox"/>] Tidak Berhasil

4.3.3 Kasus dan Hasil Pengujian Permainan Pada Sistem Operasi Android

Berikut ini adalah hasil dari pengujian *game main kata* pada beberapa sistem operasi *android* dari yang paling rendah *froyo 2.2* sampai yang tertinggi

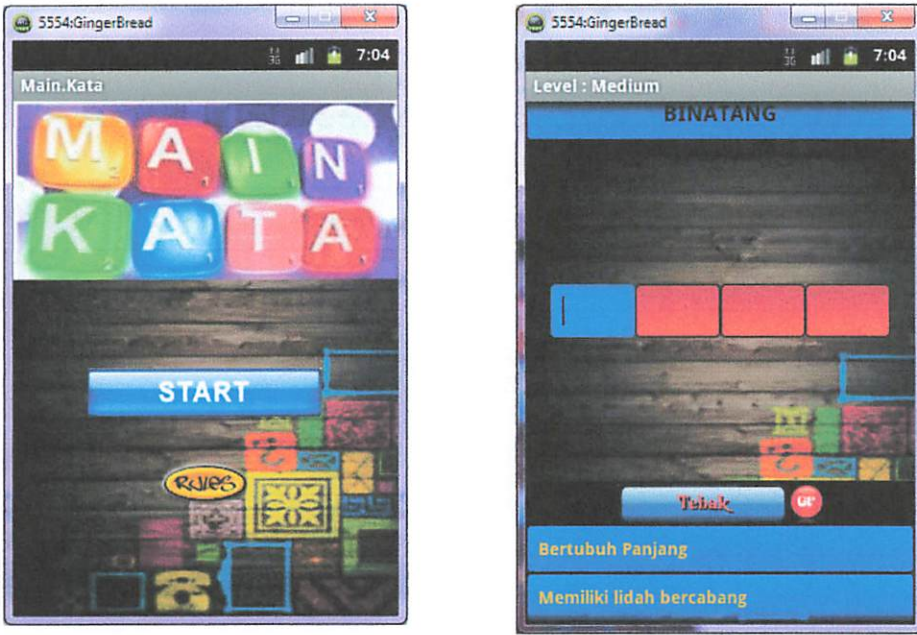
jelly bean 4.2. berikut adalah sistem operasi yang akan di coba pada simulasi AVD (*Android Virtual Device*).

1. Froyo 2.2



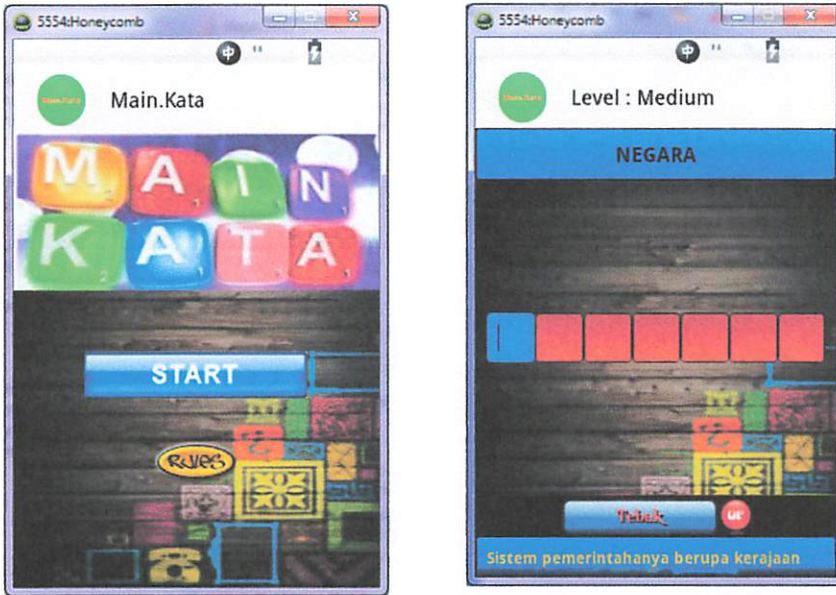
Gambar 4.7 Permainan pada sistem operasi Froyo 2.2

2. GingerBread 2.3



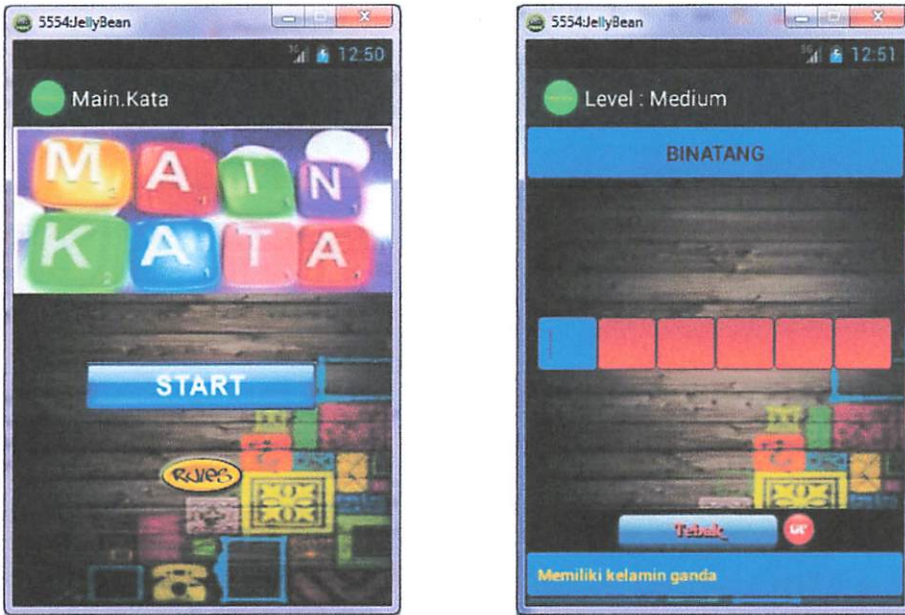
Gambar 4.8 Permainan pada sistem operasi *Gingerbread 2.3.3*

3. Honeycomb



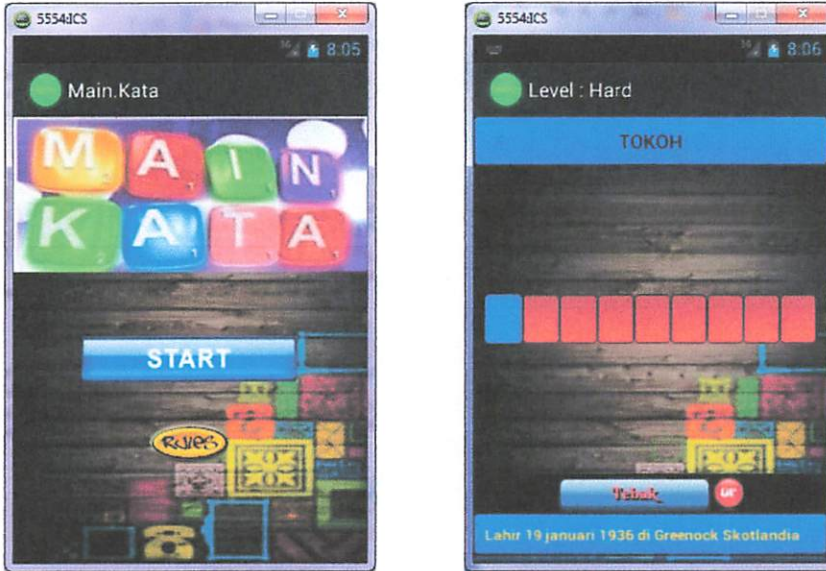
Gambar 4.9 Permainan pada system operasi Honeycomb 3.2

4. Jelly Bean 4.2.2



Gambar 4.10 Permainan pada sistem operasi Jelly Bean 4.2.2

5. Ice Cream Sandwich



Gambar 4.11 Permainan pada sistem operasi ICS 4.0

4.3.4 Pengujian Pada Perangkat *SmartPhone* Berbasis Android

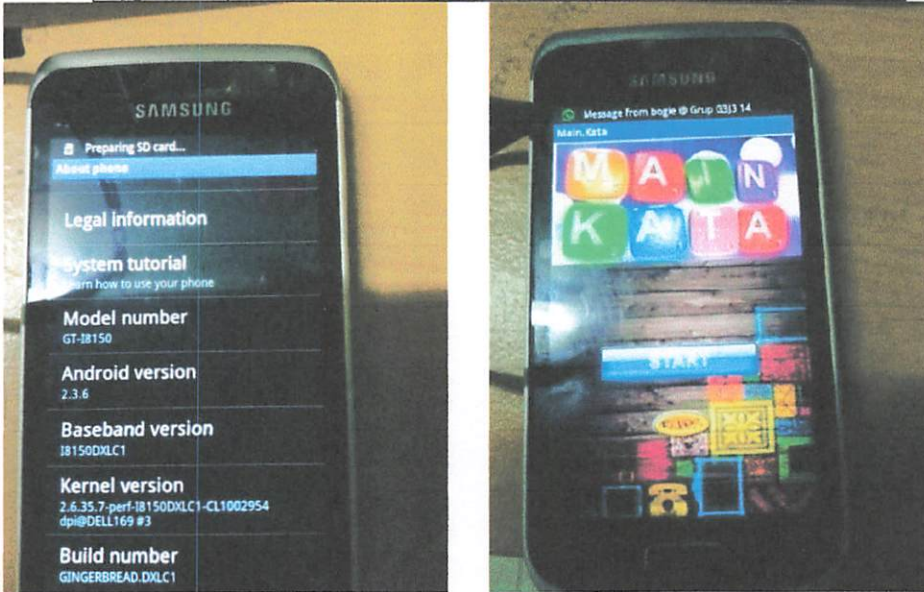
Pengujian berikut akan di lakukan pada perangkat keras atau *Hardware Mobile Phone* berbasis *android* yang akan di lakukan secara sampling.

1. Pengujian pertama akan di lakukan pada *Smart Phone Android Samsung Galaxy Wonder GT- I8150*.

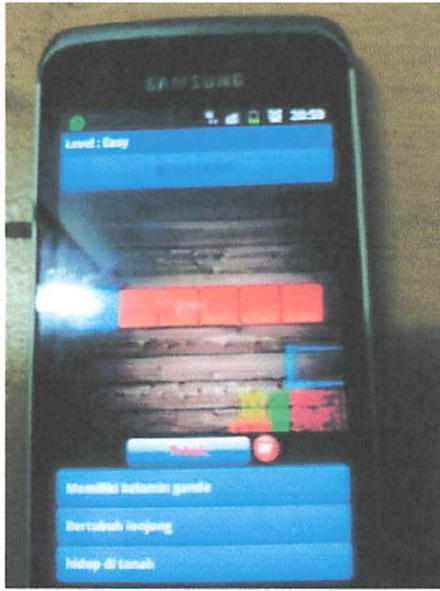
Spesifikasi *Galaxy Wonder*

Tabel 4.9 Spesifikasi *Galaxy Wonder*

OS	Android OS, v2.3.6 (Gingerbread)
Chipset	Qualcomm MSM8255T Snapdragon
CPU	1.4 GHz Scorpion
GPU	Adreno 205
Size	480 x 800 pixels, 3.7 inches (~252 ppi pixel density)
Type	TFT capacitive touchscreen, 16M colors



Gambar 4.12 Tampilan Samsung GTI8150 dan Main Menu Pada Permainan



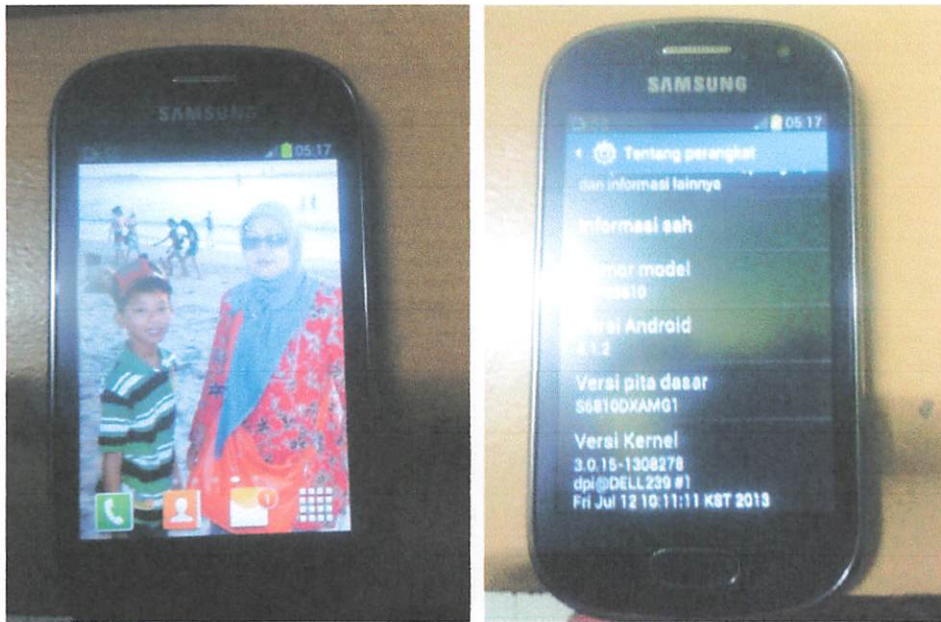
Gambar 4.13 Tampilan papan permainan pada *Samsung Galaxy GT I8150*

2. Pengujian ke dua akan di lakukan pada *Smart Phone Android Samsung Galaxy Fame S6810*.

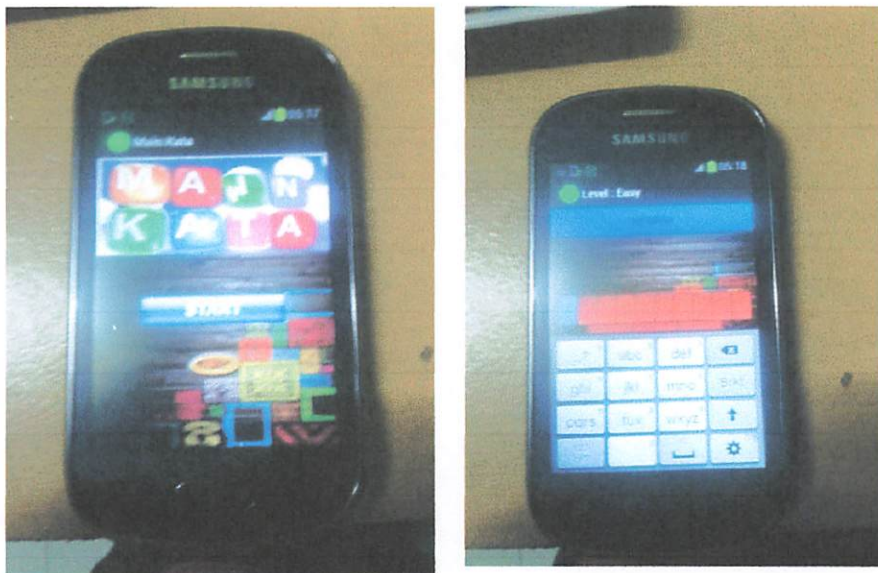
Spesifikasi *Samsung Galaxy Fame S6810*

Tabel 4.10 Spesifikasi *Samsung Galaxy Fame*

<u>OS</u>	<i>Android OS, v4.1.2 (Jelly Bean)</i>
<u>CPU</u>	<i>1 GHz Cortex-A9</i>
<u>GPU</u>	<i>Broadcom VideoCore IV</i>
<u>Type</u>	<i>TFT capacitive touchscreen, 256K colors</i>
<u>Size</u>	<i>320 x 480 pixels, 3.5 inches (~165 ppi pixel density)</i>



Gambar 4.14 Tampilan *Samsung Galaxy S6810*



Gambar 4.15 Tampilan Permainan Pada *Samsung Galaxy S6810*

3. Pengujian ke tiga di lakukan pada *Smart Phone Android Samsung Galaxy Core I8262*

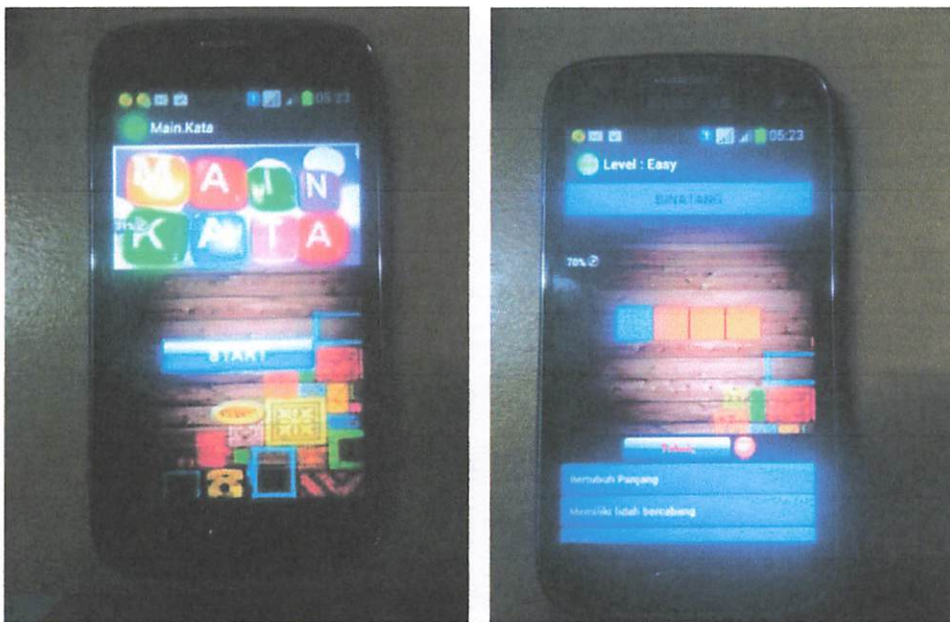
Spesifikasi *Samsung Galaxy Core*

Tabel 4.11 Spesifikasi *Samsung Galaxy Core*

<i>OS</i>	<i>Android OS, v4.1.2 (Jelly Bean)</i>
<i>Chipset</i>	<i>Snapdragon MSM8225 S4 Play</i>
<i>CPU</i>	<i>Dual-core 1.2 GHz Cortex-A5</i>
<i>GPU</i>	<i>Adreno 203</i>
<i>Type</i>	<i>TFT capacitive touchscreen, 16M colors</i>
<i>Size</i>	<i>480 x 800 pixels, 4.3 inches (~217 ppi pixel density)</i>



Gambar 4.16 Tampilan *Samsung Galaxy Core*



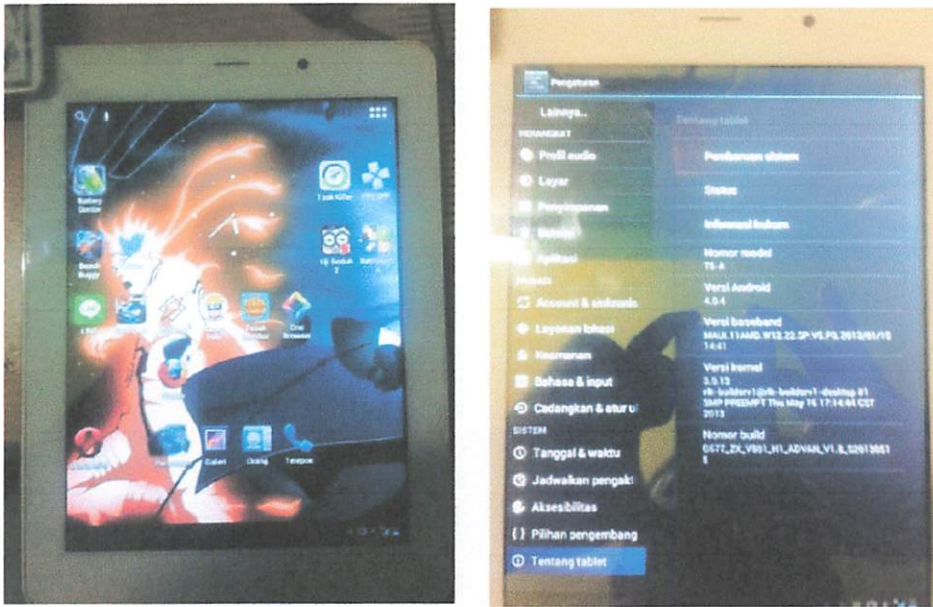
Gambar 4.17 Tampilan permainan pada *Samsung Galaxy Core*

4. Pengujian ke empat akan di lakukan pada *Tablet Android Advan Vandroid T5-A*

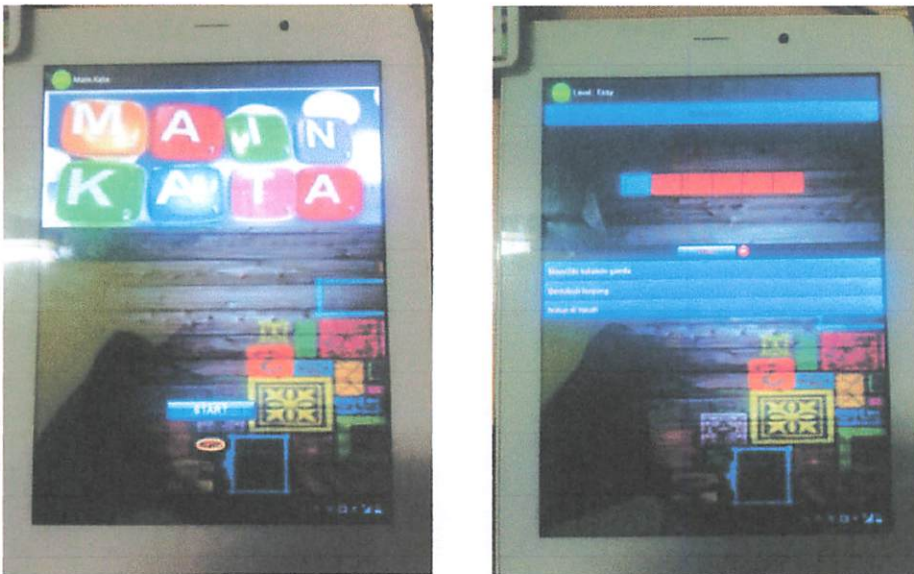
Spesifikasi *Advan Vandroid T5-A*

Tabel 4.12 Spesifikasi Advan Vandroid T5-A

<i>OS</i>	<i>Android 4.0 Ice Cream Sandwich</i>
<i>Chipset</i>	<i>Qualcomm Snapdragon S4</i>
<i>CPU</i>	<i>Dual Core 1 GH</i>
<i>GPU</i>	<i>Broadcom VideoCore IV</i>
<i>Type</i>	<i>TFT capacitive touchscreen, 16M colors</i>
<i>Size</i>	<i>1024x768 8Inches</i>
















Gambar 4.18 Tampilan Advan Vandroid T5-A



Gambar 4.19 Tampilan permainan pada Advan Vandroid T5-A

4.4 Tutorial Memainkan Game Pada *Smart Phone Android*

Game main kata yang di kembangkan oleh penulis ini memiliki gaya permainan yang simple dan mudah di pahami. Berikut adalah beberapa cara untuk memainkan permainan main kata.

- Tekan *touch screen* untuk mengoprasikan permainan dan untuk melakukan akses di dalam permainan.
- Tekan tombol  untuk kembali / *back* atau keluar dari permainan.
- Tekan tombol *home*  untuk me *minimize game* dan tekan ikon *game main kata* untuk kembali ke permainan.
- Tekan tombol    untuk memilih tingkat kesulitan pada permainan.
- Tekan tombol     untuk memilih jenis kategori.
- Tekan tombol  untuk menjawab soal pada permainan.
- Tekan tombol  untuk menggunakan point.
- Tekan tombol  untuk memulai permainan.
- Tekan tombol  untuk mengetahui aturan permainan.

4.4.1 Kuisisioner

Kuisisioner merupakan pengujian yang di lakukan secara objektif dimana pengujian dilakukan secara langsung kepada responden dengan membuat kuisisioner mengenai kepuasan pengguna dengan kandungan point syarat. Berdasarkan hasil pengujian, dicari persentase masing-masing jawaban dengan menggunakan rumus:

$$Y=P/Q*100\%$$

Keterangan:

P = jumlah jawaban responden tiap soal

Q = jumlah responden

Y = nilai persentase

Point penilaian : - Sangat Setuju [5]

- Setuju [4]

- Ragu-ragu [3]

- Tidak Setuju [2]

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus di atas. Kuesioner ini telah diujikan kepada 10 orang. Hasil perhitungan persentase masing-masing jawaban dapat dilihat pada tabel 4.11 – 4.16.

1. Untuk pertanyaan pertama

“ Apakah design tampilan *game main kata* ini udah cukup menarik ”

Tabel 4.13 Kuisisioner pertama

Kategori Jawaban	Frekuensi Jawaban (responden)	Jumlah keseluruhan Responden	Jumlah Persentase
Sangat Setuju	4	10	40%
Setuju	6	10	60%
Ragu-ragu	0	10	0%
Tidak setuju	0	10	0%

2. Untuk pertanyaan kedua

“ Apakah anda tertantang untuk mencoba di semua tingkat kesulitan “

Tabel 4.14 Kuisisioner kedua

Kategori Jawaban	Frekuensi Jawaban (responden)	Jumlah keseluruhan Responden	Jumlah Persentase
Sangat Setuju	3	10	30%
Setuju	5	10	50%

Ragu-ragu	2	10	20%
Tidak setuju	0	10	0%

3. Untuk pertanyaan ketiga

“ Apakah *audio* sudah sesuai dengan *game main kata* ”

Tabel 4.15 Kuisisioner ketiga

Kategori Jawaban	Frekuensi Jawaban (responden)	Jumlah keseluruhan Responden	Jumlah Persentase
Sangat Setuju	0	10	0%
Setuju	8	10	80%
Ragu-ragu	2	10	20%
Tidak setuju	0	10	0%

4. Untuk Pertanyaan keempat

“ Apakah pengoprasian *game main kata* sudah cukup mudah untuk di mainkan “

Tabel 4.16 Kusioner keempat

Kategori Jawaban	Frekuensi Jawaban (responden)	Jumlah keseluruhan Responden	Jumlah Persentase
Sangat Setuju	9	10	90%
Setuju	1	10	10%
Ragu-ragu	0	10	0%
Tidak setuju	0	10	0%

5. Untuk Pertanyaan kelima

“ Apakah *game main kata* dapat di mainkan oleh semua kalangan “

Tabel 4.17 Kuisisioner kelima

Kategori Jawaban	Frekuensi Jawaban (responden)	Jumlah keseluruhan Responden	Jumlah Persentase
Sangat Setuju	7	10	70%

Setuju	3	10	30%
Ragu-ragu	0	10	0%
Tidak setuju	0	10	0%

6. Untuk pertanyaan keenam

“ Apakah *game main kata* cukup menarik untuk di mainkan “

Tabel 4.18 Kuisisioner keenam

Kategori Jawaban	Frekuensi Jawaban (responden)	Jumlah keseluruhan Responden	Jumlah Persentase
Sangat Setuju	3	10	30%
Setuju	7	10	70%
Ragu-ragu	0	10	0%
Tidak setuju	0	10	0%

Saran dari responden dapat disimpulkan sebagai berikut :

1. Tolong tambahkan lagi jenis kategori yang lebih banyak agak semakin menarik.
2. Tolong segera di *upload* di *play store* agar dapat di mainkan oleh banyak pengguna *android*.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil uji coba yang telah dilakukan oleh penulis pada akhir perancangan aplikasi tersebut dapat dirumuskan beberapa kesimpulan, sebagai berikut:

1. Aplikasi *game main kata* ini dapat di sistem operasi android *Froyo 2.3, Gingerbread 2.3, Honeycomb 3.0, ICS 4.0 dan Jelly Bean 4.2*
2. *Game main kata* ini cukup menarik untuk di mainkan dan dapat menambah wawasan dalam menmainkan kata.
3. Dari beberapa responden, aplikasi game main kata ini memiliki beberapa kekurangan yaitu kurang nya jenis kategori yang tersedia dan masih belum bisa di upload pada *play store* sehingga belum bisa di mainkan oleh semua pengguna *android* yang ada.
4. Aplikasi game main kata dapat terintegrasi dan berfungsi pada beberapa tipe *handphone* yang berbeda dan di sistem operasi yang sama.

5.2 Saran

Untuk mengembangkan aplikasi ini menjadi lebih baik lagi, penulis menyarankan untuk:

1. Mengembangkan aplikasi *game android* ini supaya dapat di jalakan pada sistem operasi *android* yang jebih tinggi lagi atau di atas *jelly bean 4.3* supaya dapat di mainkan oleh semua tipe *android*.
2. Menambahkannya fitur *connect to facebook* dan *share reward* lewat *facebook* supaya terjadi kompetisi secara tidak langsung antar pemain game main kata.

3. Di tambahkan nya jenis kategori lebih banyak lagi dan mempercantik tampilan game main kata agar lebih menarik lagi dan tidak cepat bosan untuk dimainkan.

DAFTAR PUSTAKA

- [1] Dawang Muchtar (2005). *Desain game*, Jakarta
- [2] Eueung Mulyana, *APP Inventor :Ciptakan Sendiri Aplikasi Android mu*, Andi 2012, Yogyakarta
- [3] Katie Salen & Eric Zimmerman, (2004).*Rules of Play*.Cambridge: Game Design Fundamental.
- [4] Nazruddin Safaat H, *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, IF 2012, Bandung
- [5] Priyanta F, *Pemrograman Java Untuk Pemula*, Cerdas Pustaka 2011, Jakarta
- [6] Red Bird, *Hewan Hewan Menakjubkan*, Gramedia Agustus 2012, Jakarta
- [7] Aisi Ariniro, *Buku Pintar Negara Negara di Dunia*, Diva April 2012, Jakarta
- [8] Biografi tokoh – tokoh dunia, (<http://kolom-biografi.blogspot.com/>, diakses pada tanggal 22 Januari 2014)
- [9] Definisi game dari beberapa tokoh
(<http://gamedesignconcepts.wordpress.com/2009/06/29/level-1-overview-what-is-a-game/>, diakses pada tanggal 21 April 2013)
- [10] Kumpulan Anime <http://www.alibabasub.net/>di akses pada tanggal 12 Maret 2014
- [11] <http://stackoverflow.com>di akses pada tanggal 10 Januari 2014



LAMPIRAN

LAMPIRAN

```
package me.main.kata;

import java.util.ArrayList;

import me.main.kata.R.id;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.app.Activity;
import android.content.Intent;

public class JenisActivity extends Activity {
    Utils u;
    TextView [] tvJenis;
    String LEVEL;

    private MediaPlayer mMediaPlayer;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.jenis);

        Intent i = getIntent();
        u = new Utils(getApplicationContext(),this );
        mMediaPlayer = u.playFromRaw(this, R.raw.main,true) ;

        LEVEL = i.getStringExtra("LEVEL");
        tvJenis = new TextView[4];

        ArrayList<String> strJawab =
u.files.toArray(ArrayList(SplashScreenActivity.fileSetting) );
        tvJenis[0] = (TextView) findViewById(id.txtJenis_1);
        tvJenis[0].setText(strJawab.get(0));

        tvJenis[1] = (TextView) findViewById(id.txtJenis_2);
```

```
tvJenis[1].setText(strJawab.get(1));
```

```
tvJenis[2] = (TextView) findViewById(id.txtJenis_3);  
tvJenis[2].setText(strJawab.get(2));
```

```
tvJenis[3] = (TextView) findViewById(id.txtJenis_4);  
tvJenis[3].setText(strJawab.get(3));
```

```
ImageButton btnJenis1 = (ImageButton)  
findViewById(id.btnJenis_1);  
btnJenis1.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        startJenis("0");  
    }  
});
```

```
ImageButton btnJenis2 = (ImageButton)  
findViewById(id.btnJenis_2);  
btnJenis2.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        startJenis("1");  
    }  
});
```

```
ImageButton btnJenis3 = (ImageButton)  
findViewById(id.btnJenis_3);  
btnJenis3.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View v) {  
        // TODO Auto-generated method stub  
        startJenis("2");  
    }  
});
```

```
ImageButton btnJenis4 = (ImageButton)  
findViewById(id.btnJenis_4);
```

```
btnJenis4.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View v) {
```

```
        // TODO Auto-generated method stub  
        startJenis("3");
```

```
    }
```

```
});
```

```
}
```

```
@Override
```

```
protected void onResume() {
```

```
    // TODO Auto-generated method stub
```

```
    mMediaPlayer.start();
```

```
    ArrayList<String> strJawab =
```

```
u.files.toArray( new ArrayList(SplashScreenActivity.fileSetting) );
```

```
    for (int j = 0; j < tvJenis.length; j++) {
```

```
        tvJenis[j].setText(strJawab.get(j) );
```

```
    }
```

```
    super.onResume();
```

```
}
```

```
@Override
```

```
protected void onPause() {
```

```
    // TODO Auto-generated method stub
```

```
    mMediaPlayer.pause();
```

```
    super.onPause();
```

```
}
```

```
protected void startJenis(String string) {
```

```
    // TODO Auto-generated method stub
```

```
    Intent intent = new Intent(this, PlayActivity.class);
```

```
    intent.putExtra("JENIS", string);
```

```
    intent.putExtra("LEVEL", LEVEL);
```

```
    startActivity(intent); }
```

```
}
```

```

package me.main.kata;

import java.util.ArrayList;

import me.main.kata.R.id;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.widget.ImageButton;

public class LevelActivity extends Activity {
    Utils u;
    private MediaPlayer mMediaPlayer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.level);
        u = new Utils(getApplicationContext(),this );

        mMediaPlayer = u.playFromRaw(this, R.raw.main,true) ;
        ImageButton btnEasy = (ImageButton) findViewById(id.btnEasy);
        btnEasy.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                startLevel("0");
            }
        });
        ImageButton btnMed = (ImageButton) findViewById(id.btnMed);
        btnMed.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                startLevel("1");
            }
        });
    }
}

```

```
ImageButton btnHard = (ImageButton) findViewById(id.btnHard);
btnHard.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        startLevel("2");
    }
});
```

```
}
```

```
protected void startLevel(String string) {
    // TODO Auto-generated method stub
    Intent intent = new Intent(this, JenisActivity.class);
    intent.putExtra("LEVEL", string);
    startActivity( intent);
}
```

```
@Override
protected void onResume() {
    // TODO Auto-generated method stub
    mMediaPlayer.start();
    super.onResume();
}
```

```
@Override
protected void onPause() {
    // TODO Auto-generated method stub
    mMediaPlayer.pause();
    super.onPause();
}
}
```

```
package me.main.kata;
```

```
import me.main.kata.R.id;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.media.SoundPool;
import android.os.Bundle;
import android.app.Activity;
```

```

import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;

public class MainActivity extends Activity {

    Utils u ;
    public boolean mSoundOn = true;
    private MediaPlayer mMediaPlayer;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        mSoundOn = true;
        u = new Utils(getApplicationContext(),this );

        mMediaPlayer = u.playFromRaw(this, R.raw.main,true) ;

        ImageButton btnStart = (ImageButton) findViewById(id.btnStart);
        btnStart.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                starGame();
            }
        });

        ImageButton btnRule = (ImageButton) findViewById(id.btnRule);
        btnRule.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {

```



```

        // TODO Auto-generated method stub
        showRule();
    }
});

//mSoundPool.setLoop(mMainSound,10);

}
@Override
protected void onStart() {
    // TODO Auto-generated method stub

    super.onStart();
}
@Override
protected void onResume() {
    // TODO Auto-generated method stub
    mMediaPlayer.start();
    super.onResume();
}
@Override
protected void onDestroy() {
    // TODO Auto-generated method stub

    super.onDestroy();
}
@Override
protected void onPause() {
    // TODO Auto-generated method stub
    mMediaPlayer.pause();
    super.onPause();
}

protected void showRule() {
    final Dialog dialog = new Dialog(this);
    dialog setContentView(R.layout.help);
}

```

```
        .setNegativeButton("Tidak", new  
DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int id) {  
                dialog.cancel();  
            }  
        });  
        AlertDialog alert = builder.create();  
        alert.show();
```

```
    }  
}  
package me.main.kata;
```

```
import java.io.File;  
import java.util.ArrayList;
```

```
import me.main.kata.R.id;  
import android.R.integer;  
import android.R.string;  
import android.hardware.input.InputManager;  
import android.media.AudioManager;  
import android.media.SoundPool;  
import android.os.Bundle;  
import android.app.Activity;  
import android.app.AlertDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.view.KeyEvent;  
import android.view.Menu;  
import android.view.MotionEvent;  
import android.view.View;  
import android.view.WindowManager;  
import android.view.inputmethod.InputMethodManager;  
import android.widget.EditText;  
import android.widget.ImageButton;  
import android.widget.LinearLayout;  
import android.widget.TextView;
```

```
public class PlayActivity extends Activity {
```

```
ImageButton btnTebak, btnGP;
```

```
TextView[] txtClue;  
LinearLayout[] layClue;  
TextView txtJenis,txtPoin;  
EditText[] edInput;
```

```
String JENIS;
```

```
Utils u = new Utils();
```

```
String LEVEL;
```

```
ArrayList<TSoal> SOAL = new ArrayList<TSoal>();
```

```
String [] arJenis = new String []  
{"BINATANG", "NEGARA", "TOKOH", "ANIME"};  
String [] arLevel = new String [] {"Easy", "Medium", "Hard"};
```

```
private int indexSOAL;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.play);
```

```
    layClue = new LinearLayout[3];  
    layClue[0] = (LinearLayout) findViewById(id.layC1);  
    layClue[1] = (LinearLayout) findViewById(id.layC2);  
    layClue[2] = (LinearLayout) findViewById(id.layC3);
```

```
    txtClue = new TextView [3];  
    txtClue[0] = (TextView) findViewById(id.txtClue1);  
    txtClue[1] = (TextView) findViewById(id.txtClue2);  
    txtClue[2] = (TextView) findViewById(id.txtClue3);
```

```
    Intent iN = getIntent();  
    JENIS = iN.getStringExtra("JENIS");  
    LEVEL = iN.getStringExtra("LEVEL");
```

```
txtJenis = (TextView) findViewById(id.txtJenis);
txtPoin = (TextView) findViewById(id.txtPoin);
```

```
txtJenis.setText(arJenis[u.i(JENIS)]);
setTitle("Level : " + arLevel[u.i(LEVEL)]);
for (int i = 0; i < txtClue.length; i++) {
    txtClue[i].setText("");
}
```

```
SOAL = new ArrayList<TSoal>();
ArrayList<String> strSoal = u.loadArrayFromAsset(
```

```
getApplicationContext(), "data");
```

```
for (int i = 0; i < strSoal.size(); i++) {
    String[] strTmp = strSoal.get(i).toString().split(":");
    if (strTmp[0].equals( JENIS)){
        TSoal tmp= new TSoal(i,
strTmp[1].toUpperCase().replace(" ", ""), new
String[] {strTmp[2],strTmp[3],strTmp[4]});
        SOAL.add(tmp);
    }
}
```

```
edInput = new EditText[30];
```

```
int idex = id.ed01;
```

```
int i ;
```

```
for ( i = 0; i < edInput.length; i++) {
```

```
    final int id =i;
```

```
    edInput[i] = (EditText) findViewById(idex+i);
```

```
    edInput[i].setOnTouchListener(new View.OnTouchListener()
```

```
{
```

```
        @Override
```

```
        public boolean onTouch(View v, MotionEvent event) {
            ((EditText) v).setText("");
```

```
            return false;
```

```
        }
```

```
    });
```

```
    edInput[i].setOnKeyListener(new View.OnKeyListener() {
```

```

        @Override
        public boolean onKeyDown(View v, int keyCode, KeyEvent
event) {
            // TODO Auto-generated method stub

            if
(event.getAction()==(KeyEvent.ACTION_UP))
            {
                String s=((EditText)
v).getText().toString().toUpperCase().replace("[^a-zA-Z]", "");
                ((EditText) v).setText(s);
                playSound(mClickSound);

                if (id+1<edInput.length)
edInput[id+1].requestFocus();

            }

            return false;
        }
    });

    edInput[i].setWidth(70);
}

btnTebak = (ImageButton) findViewById(id.btnTebak) ;
btnTebak.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        jawab();
    }
});
btnTebak.requestFocus();
btnGP = (ImageButton) findViewById(id.btnGP);

```

```

btnGP.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        useGP();
    }
});

restartGame();

```

```

getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_HIDDEN);

```

```

mSoundPool = new SoundPool(4, AudioManager.STREAM_MUSIC,
100);

```

```

mClickSound = mSoundPool.load(this, R.raw.klik, 1);
mCorrectSound = mSoundPool.load(this, R.raw.correct, 1);
mWrongSound = mSoundPool.load(this, R.raw.wrong, 1);

```

```

}
@Override
protected void onResume() {
    // TODO Auto-generated method stub
    for (int i = 0; i < edInput.length; i++) {
        edInput[i].clearFocus();
    }
    super.onResume();
}

```

```

boolean useGP =false;

```

```

protected void useGP() {
    // TODO Auto-generated method stub

```

```

    for (int id = 0; id < SOAL.get(indexSOAL).soal.length(); id++) {

```

```

        edInput[id].setText(String.valueOf(
SOAL.get(indexSOAL).soal.charAt(id)));

```

```

    }

```

```

        if (u.i(points.get(u.i(LEVEL)))<=0) {
            btnGP.setVisibility(View.GONE);
        }else {
            u.files.Reset(SplashScreenActivity.fileGp);
            int lastPoint = u.i( points.get(u.i(LEVEL))) - 1;

            points.set(u.i(LEVEL),u.s(lastPoint<=0 ? 0 : lastPoint) );
            for (int i = 0; i < points.size(); i++) {
                u.files.Append(SplashScreenActivity.fileGp,
points.get(i)+"\n");
            }
            if (u.i(points.get(u.i(LEVEL)))<=0) {
                btnGP.setVisibility(View.GONE);
            }
            txtPoin.setText("Point : " + points.get(u.i(LEVEL)));
            useGP = true;
        }

        //restartGame();
    }
    boolean mSoundOn = true;
    public void playSound(int sound) {
        if (!mSoundOn) {
            return;
        }
        AudioManager audioManager = (AudioManager)
getSystemService(Context.AUDIO_SERVICE);
        int streamVolume =
audioManager.getStreamVolume(AudioManager.STREAM_MUSIC);

        mSoundPool.play(sound, streamVolume, streamVolume, 1, 0, 1f);
    }
    protected void jawab() {
        // TODO Auto-generated method stub
        StringBuffer sbJawab = new StringBuffer();
        for (int i = 0; i < SOAL.get(indexSOAL).soal.length(); i++) {

            sbJawab.append(edInput[i].getText().toString().toUpperCase().trim());
        }
    }

```

```

        System.out.println("PlayActivity.jawab() : " + sbJawab.toString()+"-
>" + SOAL.get(indexSOAL).soal);
        if (sbJawab.toString().equals( SOAL.get(indexSOAL).soal )){
            playSound(mCorrectSound);
            if (!useGP ) {
                u.files.Reset(SplashScreenActivity.fileGp);
                int lastPoint = u.i( points.get(u.i(LEVEL))) + 1;

                points.set(u.i(LEVEL),u.s(lastPoint) );
                for (int i = 0; i < points.size(); i++) {
                    u.files.Append(SplashScreenActivity.fileGp,
points.get(i)+"\n");

                }
            }

            showBener();
        }else{
            playSound(mWrongSound);
            showSalah();
        }
        useGP = false;
    }
    private void showSalah() {
        // TODO Auto-generated method stub
        AlertDialog.Builder builder = new AlertDialog.Builder( this);
        builder.setMessage("Jawaban salah, ulangi lagi")
        .setCancelable(false)
        .setPositiveButton("Ya", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                AindexSoal.remove(AindexSoal.size()-1);

                restartGame();

            }
        })
        .setNegativeButton("Tidak", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                finish();
            }
        }
    }

```



```

    });
    AlertDialog alert = builder.create();
    alert.show();
}
ArrayList<Integer> AindexSoal = new ArrayList<Integer>();

```

```

private SoundPool mSoundPool;
public int mClickSound;
public int mCorrectSound;
public int mWrongSound;

```

```

ArrayList<String> points;
protected void restartGame() {
    // TODO Auto-generated method stub

    indexSOAL = u.randRange(0, SOAL.size()-1);
    if (AindexSoal.size() >= SOAL.size()){
        showComplete();
        return;
    }
    if (!u.FileExist( SplashScreenActivity.fileSetting)){
        u.files.Reset(SplashScreenActivity.fileSetting);
        for (int i = 0; i < 4; i++) {
            u.files.Append(SplashScreenActivity.fileSetting,
"0/10\n");
        }
    }else{
        ArrayList<String> strJawab =
u.files.toArrayList(SplashScreenActivity.fileSetting) ;
        u.files.Reset(SplashScreenActivity.fileSetting);
        strJawab.set(u.i(JENIS), u.s(AindexSoal.size()) +"/10");
        for (int i = 0; i < strJawab.size(); i++) {
            u.files.Append(SplashScreenActivity.fileSetting,
strJawab.get(i)+"\n");
        }
    }
    points = u.files.toArrayList(SplashScreenActivity.fileGp) ;
    txtPoin.setText("Point : " + points.get(u.i(LEVEL)));
}

```

```

        while (AindexSoal.contains(indexSOAL) && AindexSoal.size() <
SOAL.size()) {
            indexSOAL = u.randRange(0, SOAL.size()-1);
        }
        AindexSoal.add(indexSOAL);
        int first =R.id.ed02;
        for (int id = 0; id < edInput.length ; id++) {
            edInput[id].setVisibility(View.GONE);
            if (id<edInput.length-1 )
edInput[id].setNextFocusForwardId(first+id);
        }
        for (int id = 0; id < SOAL.get(indexSOAL).soal.length(); id++) {
            edInput[id].setVisibility(View.VISIBLE);
            edInput[id].setText("");
            //edInput[id].setText(String.valueOf(
SOAL.get(indexSOAL).soal.charAt(id)));
        }
        edInput[SOAL.get(indexSOAL).soal.length()-
1].setNextFocusForwardId(R.id.ed01);
        int idShow = u.randRange(0, SOAL.get(indexSOAL).soal.length()-1
);

        edInput[idShow].setText(String.valueOf(
SOAL.get(indexSOAL).soal.charAt(idShow)));
        for (int id = 0; id < SOAL.get(indexSOAL).clue.length; id++) {
            txtClue[id].setText(SOAL.get(indexSOAL).clue[id]);
            layClue[2].setVisibility(View.VISIBLE);
        }
        if (u.i(LEVEL)==1) {
            txtClue[2].setText("");
            layClue[2].setVisibility(View.GONE);
            btnGP.setVisibility(View.GONE);
            txtPoin.setVisibility(View.GONE);
        }
        if (u.i(LEVEL)==2) {
            txtPoin.setVisibility(View.VISIBLE);
            txtClue[1].setText("");
            txtClue[2].setText("");
            layClue[1].setVisibility(View.GONE);
            layClue[2].setVisibility(View.GONE);

```

```

        if (u.i(points.get(u.i(LEVEL)))<=0) {
            btnGP.setVisibility(View.GONE);
        }else{
            btnGP.setVisibility(View.VISIBLE);
        }
    }
    if (u.i(LEVEL)==0) {
        btnGP.setVisibility(View.GONE);
        txtPoin.setVisibility(View.GONE);
    }
}

private void showComplete() {

    // TODO Auto-generated method stub
    AlertDialog.Builder builder = new AlertDialog.Builder( this);
builder.setMessage("Pertanyaa dilevel ini sudah terjawab semua..")
    .setCancelable(false)
    .setNeutralButton("Kembali", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            finish();
        }
    });
AlertDialog alert = builder.create();
alert.show();

}

private void showBener() {
    // TODO Auto-generated method stub
    AlertDialog.Builder builder = new AlertDialog.Builder( this);
builder.setMessage("Jawaban Benar, Lanjut ke soal berikutnya ?")
    .setCancelable(false)
    .setPositiveButton("Ya", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            restartGame();
        }
    }
}
})

```

```

        .setNegativeButton("Tidak", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                finish();
            }
        });
        AlertDialog alert = builder.create();
        alert.show();
    }
}

package me.main.kata;

```

```
import java.io.File;
```

```
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
```

```
import android.widget.ProgressBar;
```

```
public class SplashScreenActivity extends Activity {
    ProgressBar pbMain;
    Utils u ;
    public static final String fileSetting = Utils.getSdPath() + "/tmp.dat";
    public static final String fileGp = Utils.getSdPath() + "/gp.dat";

```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.splash);
    setTitle("Selamat Datang..");

```

```
u = new Utils(getApplicationContext(),this );
```

```
if (!u.FileExist(fileSetting))
{
    u.files.Reset(fileSetting);
    for (int i = 0; i < 4; i++) {
        u.files.Append(fileSetting, "0/10\n");
    }
}

```

```

    }
}
u.files.Reset(fileGp);
for (int i = 0; i < 3; i++) {
    u.files.Append(fileGp, "0\n");
}
final int welcomeScreenDisplay =100;
pbMain = (ProgressBar) findViewById(R.id.progressBar1 );
pbMain.setMax(welcomeScreenDisplay);

```

```

Thread threads = new Thread() {

    int wait = 0;

    @Override
    public void run() {
        try {
            super.run();

            while (wait < welcomeScreenDisplay) {
                sleep(10);
                wait += 1;
                pbMain.setProgress(wait);
            }
        } catch (Exception e) {
            System.out.println("Error:" + e);
        } finally {

            newIntens();
        }
    }
};
threads.start();
}

```

```

protected void newIntens() {
    // TODO Auto-generated method stub
    startActivity(new Intent( this,MainActivity.class ));
}

```

```

        finish();
    }
}
package me.main.kata;

class TSoal{
    int ID;
    String soal;
    String[] clue;
    public TSoal(int id, String soals, String [] clues) {
        // TODO Auto-generated constructor stub
        this.ID = id;
        this.soal = soals;
        this.clue = clues;
    }
}
package me.main.kata;

import java.util.ArrayList;

import android.app.Activity;
import android.content.Context;
import android.media.MediaPlayer;

public class Utils extends android.base.org.sun.java.Utils {

    /* (non-Javadoc)
     * @see android.base.org.sun.java.Utils#LogArray(double[])
     */
    @Override
    public void LogArray(double[] data) {
        // TODO Auto-generated method stub
        super.LogArray(data);
    }

    /* (non-Javadoc)
     * @see android.base.org.sun.java.Utils#LogArray(java.lang.Integer[])
     */
    @Override

```

```

public void LogArray(Integer[] data) {
    // TODO Auto-generated method stub
    super.LogArray(data);
}

/* (non-Javadoc)
 * @see android.base.org.sun.java.Utills#LogArray(java.lang.String[])
 */
@Override
public void LogArray(String[] data) {
    // TODO Auto-generated method stub
    super.LogArray(data);
}

/* (non-Javadoc)
 * @see android.base.org.sun.java.Utills#Logs(int)
 */
@Override
public void Logs(int logs) {
    // TODO Auto-generated method stub
    super.Logs(logs);
}

/* (non-Javadoc)
 * @see android.base.org.sun.java.Utills#Logs(java.lang.Object)
 */
@Override
public void Logs(Object logs) {
    // TODO Auto-generated method stub
    super.Logs(logs);
}

/* (non-Javadoc)
 * @see android.base.org.sun.java.Utills#Logs(java.lang.String)
 */
@Override
public void Logs(String logs) {
    // TODO Auto-generated method stub
    super.Logs(logs);
}

```

```

    /* (non-Javadoc)
     * @see android.base.org.sun.java.Utills#Show(android.content.Context,
ava.lang.String)
     */
    @Override
    @Deprecated
    public void Show(Context context, String s) {
        // TODO Auto-generated method stub
        super.Show(context, s);
    }

    /* (non-Javadoc)
     * @see android.base.org.sun.java.Utills#d(java.lang.String)
     */
    @Override
    public Double d(String i) {
        // TODO Auto-generated method stub
        return super.d(i);
    }

    /* (non-Javadoc)
     * @see
android.base.org.sun.java.Utills#playFromRaw(android.content.Context, int, boolean)
     */
    @Override
    public MediaPlayer playFromRaw(Context c, int raws, boolean isLooped) {
        // TODO Auto-generated method stub
        return super.playFromRaw(c, raws, isLooped);
    }

    /* (non-Javadoc)
     * @see android.base.org.sun.java.Utills#s(int)
     */
    @Override
    public String s(int i) {
        // TODO Auto-generated method stub
        return super.s(i);
    }
}

```



```

@SuppressWarnings("deprecation")
public Utils() {
    // TODO Auto-generated constructor stub
}

public Utils(Context applicationContext, Activity jenisActivity) {
    // TODO Auto-generated constructor stub
    super(applicationContext, jenisActivity);
}

/* (non-Javadoc)
 * @see android.base.org.sun.java.Utils#FileExist(java.lang.String)
 */
@Override
public boolean FileExist(String fl) {
    // TODO Auto-generated method stub
    return super.FileExist(fl);
}

/* (non-Javadoc)
 * @see
android.base.org.sun.java.Utils#loadArrayFromAsset(android.content.Context,
ava.lang.String)
 */
@Override
public ArrayList<String> loadArrayFromAsset(Context c, String name) {
    // TODO Auto-generated method stub
    return super.loadArrayFromAsset(c, name);
}

```



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
 FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
 PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
 BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
 Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : WIM ADHIATMA
 NIM : 06.12.600
 JURUSAN : Teknik Elektro S-1
 KONSENTRASI : Teknik Komputer
 MASA BIMBINGAN: SEMESTER GANJIL '013/2014
 JUDUL : RANCANG BANGUN GAME MAIN KATA BERBASIS
 ANDROID

Tanggal	Uraian	Paraf
Pengujian I 04 - 08 - 2012		
Pengujian II 04 - 08 - 2012	1. Referensi di tambahkan 2. Skala Penilaian	

Disetujui,

Dosen Pengujian I

M. Ibrahim Ashari, ST, MT
 NIP.P.1030100358

Dosen Pengujian II

Yuli Wahyuni, ST, MT
 NIP.P.1031200456

Mengetahui,

Dosen Pembimbing I

Ir. Yusuf Ismail Nakhoda, MT
 NIP.Y.1018800189

Dosen Pembimbing II

Sotyohadi, ST
 NIP.P.1039700309



PERMOHONAN PERSETUJUAN SKRIPSI

Yang Bertanda Tangan Dibawah Ini:

Nama : Wun Abhiatma
 N I M : 0612600
 Semester : CRIPY (XIV)
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-I
 Konsentrasi : **TEKNIK ENERGI L'S'TRIK**
 TEKNIK ELEKTRONIKA
 TEKNIK KOMPUTER DAN INFORMATIKA
 TEKNIK KOMPUTER
 TEKNIK TELEKOMUNIKASI
 Alamat :

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat SKRIPSI Tingkat Sarjana. Untuk melengkapi permohonan tersenut, bersama ini kami lampirkan persyaratan-persyaratan yang harus dipenuhi.

Adapun persyaratan- persyaratan pengambilan SKRIPSI adalah sebagai berikut:

- | | |
|--|---------|
| 1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya | (.....) |
| 2. Telah lulus dan menyerahkan laporan Praktek Kerja | (.....) |
| 3. Telah lulus seluruh mata kuliah keahlian (MKB)sesuai konsentrasinya | (.....) |
| 4. Telah menempuh matakuliah> 134 sks dengan IPK > 2 dan tidak ada nilai E | (.....) |
| 5. Telah mengikuti secara aktif kegiatan seminar Skripsi yang diadakan Jurusan | (.....) |
| 6. Memenuhi persyaratan administrasi | (.....) |

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenarannya data tersebut diatas
 Recording Teknik Elektro S-I

Malang, 11-5-.....2013

[Signature]
 (.....)

Pemohon
[Signature]
 (.....)

Disetujui
Ketua Prodi Teknik Elektro S-I

Mengetahui
Dosen Wali

[Signature]

[Signature]
 (.....)

M. Ibrahim Ashari, ST, MT
NIP. P. 1030100358

Catatan:

Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Prodi T. elektro S-I

1. IP 395.5 / 2-91
2. 136
3. - Mc pend. kewirausahaan
- 6 praktikum



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-314/EL-FTI/2013
Tempor : -
Materi : BIMBINGAN SKRIPSI

Kepada : Yth. Bapak/Ibu **Ir. Yusuf Ismail Nakhoda, MT**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **WIM ADHIATMA**
Nim : **0612600**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Ganjil Tahun Akademik 2013-2014 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

Ketua Program Studi Teknik Elektro S-1

M Ibrahim Ashari, ST, MT

NIP.P. 1030100358



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Nomor Surat : ITN-314/EL-FTI/2013
Lampiran : -
Perihal : BIMBINGAN SKRIPS}

Kepada : Yth. Bapak/Ibu **Sotyohadi, ST**
Dosen Teknik Elektro S-1
ITN MALANG

Dengan Hormat

Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi untuk mahasiswa :

Nama : **WIM ADHIATMA**
Nim : **0612600**
Fakultas : **Teknologi Industri**
Program Studi : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu :

" Semester Ganjil Tahun Akademik 2013-2014 "

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.



Mengetahui

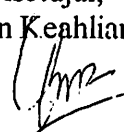
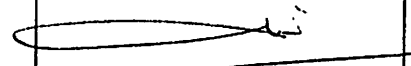
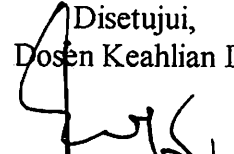

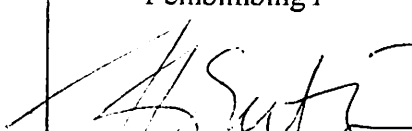
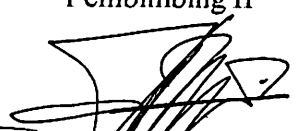
Ketua Program Studi Teknik Elektro S-1

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358



BERITA ACARA SEMINAR PROPOSAL SKRIPSI
PROGRAM STUDI TEKNIK ELEKTRO S-1
Konsentrasi : Teknik Komputer & Informatika

1.	Nim	: 0612600		
2.	Nama	: WIM ADHIATMA		
3.	Konsentrasi Jurusan	: Teknik Komputer & Informatika		
4.	Jadwal Pelaksanaan:	Waktu	Tempat	
	18 Mei 2013	09:00	III.1.4	
5.	Judul proposal yang diseminarkan Mahasiswa	RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID		
6.	Perubahan judul yang diusulkan oleh Kelompok Dosen Keahlian			
7.	Catatan :			
8.	Catatan :			
	Persetujuan judul Skripsi			
	Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II	Disetujui, Dosen Keahlian III	
	 (..... Birra)	 (Bambang Priu H.)	 (..... Eko N.)	
	Mengetahui, Ketua Program Studi Teknik Elektro S-1	Disetujui, Calon Dosen Pembimbing ybs		
 M. Ibrahim Ashari, ST, MT NIP. P 1030100358	Pembimbing I	Pembimbing II		
	 (.....)	 (..... SOTYO HADI)		



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

jumlah : 1 (satu) berkas
Pembimbing Skripsi

kepada : Yth. Bapak/Ibu **Ir. Yusuf Ismail Nakhoda, MT**
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan dibawah

Nama : **WIM ADHIATMA**
Nim : **0612600**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer & Informatika**

Dengan ini mengajukan permohonan, kiranya Bapak/Ibu bersedia menjadi Dosen Pembimbing untuk penyusunan Skripsi dengan judul :

"RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID"

Demikian permohonan kami buat dan atas kesediaan Bapak kami ucapkan terima kasih.

Mengetahui

Wakil Program Studi Teknik Elektro S-1

M. Ibrahim Ashari ST, MT

NIP.P. 1030100358

Hormat Kami

WIM ADHIATMA

NIM. 0612600



PROGRAM STUDI TEKNIK ELEKTRO S-1

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

Kampus II : Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

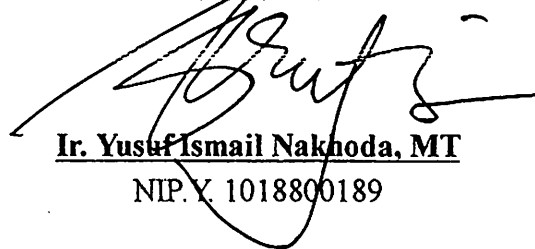
Nama : **WIM ADHIATMA**
Nim : **0612600**
Semester : **XIV (Empatbelas)**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer & Informatika**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami



Ir. Yusuf Ismail Nakhoda, MT
NIP. X. 1018800189

*) Coret yang tidak perlu



PROGRAM STUDI TEKNIK ELEKTRO S-1

**FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG**

Kampus II - Jl. Raya Karanglo Km. 2 Telp. (0341) 417636 Malang

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **WIM ADHIATMA**
Nim : **0612600**
Semester : **XIV (Empatbela)**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer & Informatika**

Dengan ini menyatakan bersedia/tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul :

" RANCANG BANGUN GAME MAIN KATA BERBASIS ANDROID"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

Sotyahadi, ST

NIP.Y. 1039700309

14/5 '13

Catatan :

Setelah disetujui agar formulir ini Diserahkan mahasiswa/i yang bersangkutan kepada jurusan untuk diproses lebih lanjut

*) Coret yang tidak perlu



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T. Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Wim Adhiatma
NIM : 06.12.600
Perbaikan meliputi :

① Referensi ditambahkan

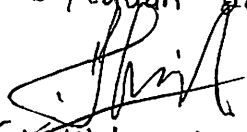
② Soal penalaran

- sangat baik

- baik

dan lain-lain.

Malang, 18 Februari 2014.


(Tuti Wahyuni, ST, MT)

SURAT PERNYATAAN ORISINALITAS

ng bertanda tangan di bawah ini :

ma : **Wim Adhiatma**
M : **06.12.600**
ogram Studi : **Teknik Elektro S-1**
nsentrasi : **Teknik Komputer S-1**

ngan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

mikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sanksinya.

Malang, Maret 2014

Yang membuat Pernyataan,



Wim Adhiatma
06.12.600