

**RANCANG BANGUN *GAME PUSH BOX* MENGGUNAKAN
ECLIPSE BERBASIS ANDROID**

SKRIPSI



Disusun oleh :
ABDI KURNIAWAN RADJA
NIM. 07.12.645

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012**

LEMBAR PERSETUJUAN

**RANCANG BANGUN GAME PUSH BOX MENGGUNAKAN
ECLIPSE BERBASIS ANDROID**

SKRIPSI

*Disusun Dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik*

Disusun oleh:

ABDI KURNIAWAN RADJA

07.12.645

Mengetahui

Ketua Program Studi Teknik Elektro S-1



Ir. Yusuf Ismail Nakhoda, MT

NIP.Y.101880089

Diperiksa dan Disetujui

Dosen Pembimbing I

M. Ibrahim Ashari, ST, MT
NIP.P.1030000365

Dosen Pembimbing II

Sandy Natali Mantja, SKom
NIP.P.1030800418

**PROGRAM STUDI TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012**

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : Abdi K. Radja

NIM : 07.12.645

Program Studi : Teknik Elektro S-1

Konsentrasi : Teknik Komputer & Informatika

Dengan 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.

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

Malang, Juli 2012

Yang membuat Pernyataan,


METERAI
TEMPEL
KEMENTERIAN KEHUKUMAN DAN
HAM
A853408250201111
6000 DUP

Abdi K. Radja
NIM : 0712645

KATA PENGANTAR

Puji Syukur ke hadirat Tuhan Yang Maha Kuasa atas segala limpahan berkat dan rahmat-Nya sehingga penelitian berjudul RANCANG BANGUN GAME PUSH BOX MENGGUNAKAN ECLIPSE BERBASIS ANDROID dapat terselesaikan. Penelitian ini dibuat untuk memenuhi salah satu syarat dalam memperoleh gelar sarjana teknik. Ucapan terima kasih yang sebesar-besarnya saya sampaikan pada:

1. Bapak Ir. Soeparno Djiwo, MT selaku Rektor Institut Teknologi Nasional Malang
2. Bapak Ir. Sidik Noertjahjono, MT selaku Dekan Fakultas Teknologi Industri ITN Malang
3. Bapak Ir. Yusuf Ismail Nakhoda, MT selaku Ketua Jurusan Elektro
4. Bapak Dr. Eng. Aryuanto Soetedjo, ST, Mt selaku Sekretaris Jurusan Elektro
5. Bapak M.Ibrahim Ashari, ST, MT selaku Dosen Pembimbing I
6. Ibu Sandy Nataly Mantja, SKom selaku Dosen Pembimbing II
7. Semua Pihak yang telah membantu dalam penulisan dan penyusunan penelitian ini.

Penulis menyadari bahwa penelitian ini masih jauh dari sempurna, untuk itu kritik dan saran dari pembaca sangat penulis harapkan untuk perbaikan penelitian ini

Malang, Agustus 2012

Penulis

RANCANG BANGUN GAME PUSH BOX MENGGUNAKAN ECLIPSE BERBASIS ANDROID

ABDI K. RADJA
(07.12.645)

Email : Labeneamata13@gmail.com
Jurusan Teknik Elektro S-1,
Konsentrasi Teknik Komputer dan Informatika
Fakultas Teknologi Industri
Institut Teknologi Nasional Malang

Dosen Pembimbing : I. **M.Ibrahim Ashari, ST, MT**
II. **Sandy Natali Mantja, SKom**

ABSTRAK

Dalam sebuah handphone pada saat ini terasa tidak lengkap rasanya bila tidak terdapat sebuah permainan. Sebuah permainan menjadi salah satu pelengkap di dalam sebuah handphone, baik permainan yang bersifat menghibur maupun permainan yang membutuhkan pikiran untuk memainkannya, jenis permainan ini mempunyai level yang singkat, kontrol yang mudah, serta tingkat kesulitan yang bertambah dengan waktu yang singkat. Daya tarik permainan ini adalah kesederhanaannya, tidak ada penjumlahan atau hitungan apapun, pemain diharuskan meletakkan setiap kotak ke tempatnya masing-masing.

Pembuatan aplikasi game ini ditujukan untuk handphone khususnya yang berbasis Android. Aplikasi dibuat dengan menggunakan bantuan beberapa software seperti JDK, SDK Android, dan IDE Eclipse Galileo.

Dengan adanya aplikasi game Push Box yang dapat digunakan pada handphone manapun dengan resolusi yang berbeda-beda yang mendukung sistem operasi Android. dapat dimainkan pemain untuk mengisi waktu dan bernostalgia memainkan game klasik puzzle.

Kata kunci : game, push box

DAFTAR ISI

LEMBAR PERSETUJUAN	i
SURAT PERNYATAAN ORISINALITAS.....	ii
KATA PENGANTAR	iii
ABSTRAK	iv
DAFTAR ISI	v
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
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 LANDASAN TEORI	5
2.1. Perkembangan Mobile Game	5
2.2. Android	6
2.2.1. Arsitektur Android	8
2.2.2. Komponen Aplikasi	11
2.2.2.1. Activity	12
2.2.2.2. Services	13
2.2.2.3. Intent.....	13
2.2.2.4.Broadcast Recivers.....	14

2.2.2.5. Content Providers.....	14
2.2.2.6. Google Map API.....	14
2.2.2.7. XML.....	14
2.2.3. Tipe Aplikasi Android	15
2.2.4. Komponen Aplikasi	16
2.2.4.1. Active Processes.....	17
2.2.4.2. Visible Processes.....	17
2.2.4.3. Started Service Processes.....	17
2.2.4.4. Background Processes.....	17
2.2.4.5. Empty Processes.....	17
2.2.5. Kelebihan Android	17
2.3. Kecerdasan buatan dalam Game	19
2.4. Push Box	14
BAB III PERANCANGAN DAN DESAIN SISTEM	22
3.1. Pembahasan	22
3.2. Perancangan	23
3.3. Konsep Game	27
3.4. Aturan Permainan	27
3.5. Desain Sistem	29
3.6. Flowchart	30
BAB IV IMPLEMENTASI DAN PENGUJIAN	31
4.1. Implementasi Sistem	31
4.1.1. Kebutuhan Hardware	31
4.1.2. Kebutuhan Software	32
4.2. Pengujian Sistem	32
4.2.1. Pengujian Menu Utama	32
4.2.2. Pengujian Menu Setting	33

4.2.3. Pengujian Menu HighScore	33
4.2.4. Pengujian Menu Permainan	34
4.3. Pengujian Aplikasi	34
4.4. Pengujian Terhadap User	36
BAB V PENUTUP	37
5.1. Kesimpulan	37
5.2. Saran	37
DAFTAR PUSTAKA	38

DAFTAR GAMBAR

BAB II LANDASAN TEORI

Gambar 2.1. Detail Arsitektur Android	8
Gambar 2.2. Urutan Prioritas pada aplikasi Android	16
Gambar 2.3. Game Push Box pertama pada Apple IIe	20
Gambar 2.4. Gameplay Push Box.....	21

BAB III ANALISA DAN PERANCANGAN SISTEM

Gambar 3.1. Rancangan Tampilan Menu Utama	23
Gambar 3.2. Rancangan Tampilan Menu Setting.....	23
Gambar 3.3. Rancangan Tampilan Menu About	24
Gambar 3.4. Rancangan Tampilan Menu Score.....	24
Gambar 3.5. Rancangan Tampilan Permainan.....	24
Gambar 3.6. Tampilan level 1	26
Gambar 3.7. Dead Square Deadlock	28
Gambar 3.8. Freeze Deadlock	28
Gambar 3.9. Deadlock due to Frozen Boxes	28
Gambar 3.10. Desain Sistem Game Push Box	29
Gambar 3.11. Flowchart Game Push Box	30

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Gambar 4.1. Tampilan Halaman Menu Utama	33
Gambar 4.2. Tampilan Halaman Setting	33
Gambar 4.3. Tampilan Halaman Highscore	34
Gambar 4.4. Tampilan Halaman Permainan	34
Gambar 4.5. Tampilan Permainan pada Samsung Galaxy mini S5570	35
Gambar 4.6. Tampilan Permainan pada Samsung Galaxy Y S5360	35
Gambar 4.4. Tampilan Permainan pada Samsung Galaxy S II	35

DAFTAR TABEL

BAB III ANALISA DAN PERANCANGAN SISTEM

Tabel 3.1. Tabel Item	25
Tabel 3.2. Tabel Simbol	26
Tabel 3.3. Konsep Game.....	27

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Tabel 4.1. Spesifikasi Perangkat Keras	31
Tabel 4.2. Nilai Uji Coba Aplikasi	36
Tabel 4.3. Nilai Uji Coba Aplikasi terhadap User.....	36

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Pada tahun 1965, game pertama kali diciptakan sebagai simulasi bagi tentara untuk belajar strategi perang di tengah situasi perang dingin. Kemudian pada tahun 1970, dikembangkan menjadi video game komersial pertama, yaitu mesin Magnavox sebagai objek hiburan atau entertainment yang terjual lebih dari 100.000 unit dengan harga US\$ 100 per unit. Sejak itu, game menjadi objek hiburan yang menarik. Seiring dengan perkembangan teknologi baik dari segi hardware maupun software, muncul berbagai peralatan home game console (Nintendo, SEGA, Sony PlayStation, xBox dan lain-lain). Selain itu, komputer yang pada dasarnya digunakan sebagai perangkat untuk membantu dalam menyelesaikan tugas-tugas kantor, kini juga merupakan perangkat yang digunakan untuk bermain game.

Dengan adanya kemajuan teknologi saat ini yang perkembangannya sangat cepat, permainan tidak hanya dapat dilakukan di rumah atau di depan komputer, permainan dapat juga dilakukan saat kita berada di luar ruangan. Melakukan permainan atau bermain *game* merupakan hal yang menarik karena dapat dimanfaatkan untuk melepas lelah, mengisi waktu luang, dan juga dapat digunakan untuk mengasah kemampuan logika. Permainan teka-teki merupakan salah satu permainan yang dapat mengasah kemampuan logika, yang salah satunya adalah permainan Push Box. Daya tarik permainan ini adalah kesederhanaannya, tidak ada penjumlahan atau hitungan apapun, pemain diharuskan meletakkan setiap kotak ke tempatnya masing-masing dengan cara mendorongnya. Secara umum permainan Push Box sebenarnya sangat sederhana, yaitu meletakkan kotak ke tempatnya masing-masing dengan cara mendorongnya. Meskipun aturannya sangat sederhana, namun penyelesaiannya tidak semudah aturannya, tentunya tingkat kesulitannya dapat bervariasi. Berdasarkan latar belakang tersebut, maka penulis mencoba untuk membuat Penulisan Ilmiah dengan judul "RANCANG BANGUN *GAME* PUSH BOX MENGGUNAKAN ECLIPSE BERBASIS ANDROID".

1.2. Rumusan Masalah

Permasalahan yang diangkat pada skripsi ini adalah bagaimana membangun perangkat lunak program aplikasi *game* Push Box?

1.3. Tujuan

Tujuan penulisan ini untuk menghasilkan suatu permainan untuk mengasah otak kita, dimana dalam memainkan permainan tersebut kita harus menyusun strategi agar permainan tersebut dapat kita selesaikan. Permainan ini juga dapat untuk mengisi waktu luang kita bila kita sedang jenuh, karena permainan ini dapat kita akses dimana saja dan kapan saja.

1.4. Batasan Masalah

Permasalahan yang akan dibahas dalam penulisan ini dibatasi pada:

1. Program aplikasi ini hanya berfungsi pada Sistem Operasi Android.
2. Permainan ini bersifat *single player*.
3. Permainan menyediakan 20 tingkatan permainan.

1.5. Metodologi Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut:

1. Studi literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

2. Analisa kebutuhan aplikasi

Data dan informasi yang telah diperoleh akan dianalisa agar dihasilkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan aplikasi dimana nantinya akan digunakan sebagai acuan perancangan aplikasi.

3. Perancangan dan implementasi

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun aplikasi ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari aplikasi yang akan dibuat dan diimplementasikan ke dalam aplikasi.

4. Eksperimen dan evaluasi

Pada tahap ini, aplikasi yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

1.6. Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut:

Bab I : Pendahuluan

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Batasan Masalah, Metode Penelitian, dan Sistematika Penulisan.

Bab II : Tinjauan Pustaka

Berisi tentang Landasan Teori mengenai permasalahan yang berhubungan dengan perancangan aplikasi yang dilakukan.

Bab III : Perancangan dan Analisa Aplikasi

Dalam bab ini berisi mengenai analisa kebutuhan aplikasi, baik *software* maupun *hardware* yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang dibuat.

Bab IV : Pembuatan dan Pengujian Sistem

Berisi tentang implementasi dari perancangan aplikasi yang telah buat serta pengujian terhadap aplikasi tersebut.

Bab V : Penutup

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II LANDASAN TEORI

Sesuai dengan bahasan skripsi, perlu dibahas teori yang berkaitan sebagai dasar penyelesaian masalah. Penulisan ini yang mengacu kepada pembuatan aplikasi mobile game PUSH BOX, sehingga penting untuk dibahas mengenai perkembangan game khususnya pada mobile device, sebagai berikut.

2.1. Perkembangan Mobile Game

Perkembangan game dari masa ke masa semakin menunjukkan eksistensinya. Jika dulu belum mengenal video game, yang dimainkan hanya permainan di dunia nyata seperti permainan tradisional. Tetapi seiring berkembangnya teknologi, kini permainan bisa dilakukan di rumah sendiri dengan video game dan bukan tidak mungkin untuk kedepannya bukan hanya video game yang bisa di mainkan tapi virtual game seperti dalam film *The Matrix*, kita yang masuk kedalam game itu dan bermain di dunia virtual.

Salah satu perkembangan teknologi game yaitu pada mobile device. Eksistensi game untuk handphone seringkali dipandang sebelah mata oleh kalangan gamer. Game di handphone seringkali dianggap sebagai pengisi waktu luang saja, dan bukan merupakan hiburan utama.

Seiring waktu, perkembangan game handphone justru menunjukkan bahwa makin banyak game berbasis java (J2ME) yang dirilis ke pasaran, Ini dikarenakan sifat bahasa pemrograman java yang cross-platform, yang artinya bisa dijalankan di berbagai macam tipe handphone, entah yang menggunakan operating system maupun tidak. Hal inilah yang membuat developer game lebih tertarik merilis game berbasis java. Gameloft, Digital Chocolate, Glu, dan FishLabs merupakan developer game handphone ternama yang kerap merilis game Java bermutu.

Seiring berkembangnya zaman mempengaruhi perkembangan teknologi game pada mobile. Android merupakan salah satu sistem operasi yang mendukung program aplikasi game pada mobile. Sistem operasi android memang semakin terkenal dan bahkan menjadi sistem operasi handphone yang menarik bagi masyarakat. Contoh Handphone dengan sistem operasi Android adalah Motorola

Milestone, Sony Ericsson xPeria X10, Samsung Galaxy S, LG Optimus, sampai Samsung Galaxy S Mini. Salah satu daya tarik dari Android adalah banyaknya dukungan dan game terbaru yang tersedia untuk Android. Game-game terbaru seperti Angry Birds Rio ataupun Angry Birds Season sampai Need For Speed, Real Football 2011 ataupun PES 2011 tersedia secara gratis untuk Android.

2.2. Android

Android adalah sistem operasi yang digunakan di smartphone dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry OS. Android tidak terikat ke satu merek Handphone saja, beberapa vendor terkenal yang sudah memakai Android antara lain Samsung , Sony Ericsson, HTC, Nexus, Motorola, dan lain-lain. Android pertama kali dikembangkan oleh perusahaan bernama Android Inc., dan pada tahun 2005 di akuisisi oleh Google. Android dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap release-nya diberi kode nama berdasarkan nama hidangan makanan. Keunggulan utama Android adalah gratis dan *open source*, yang membuat smartphone Android dijual lebih murah dibandingkan dengan Blackberry atau iPhone meski fitur (*hardware*) yang ditawarkan Android lebih baik. Berikut Beberapa tingkatan Android sejak dimulainya pengembangan pada tahun 2008 antara lain :

1. Android versi 1.1

Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah 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 kontak) teknologi *text*

to change speech (tidak tersedia pada semua ponsel) pengadaan resolusi VWGA.

3. Android versi 2.1 (Eclair)

Android dengan versi 2.0/2.1 (Eclair) diluncurkan pada 3 Desember 2009, perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan *User Interface* dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

4. Android versi 2.2 (Froyo)

Android versi 2.2 (Froyo) diluncurkan pada 20 Mei 2010. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 *JavaScript engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam *SD Card*, kemampuan WiFi Hotspot portabel, dan kemampuan *auto update* dalam aplikasi Android Market.

5. Android versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan pada 6 Desember 2010. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu.

6. Android versi 3.0 (Honeycomb)

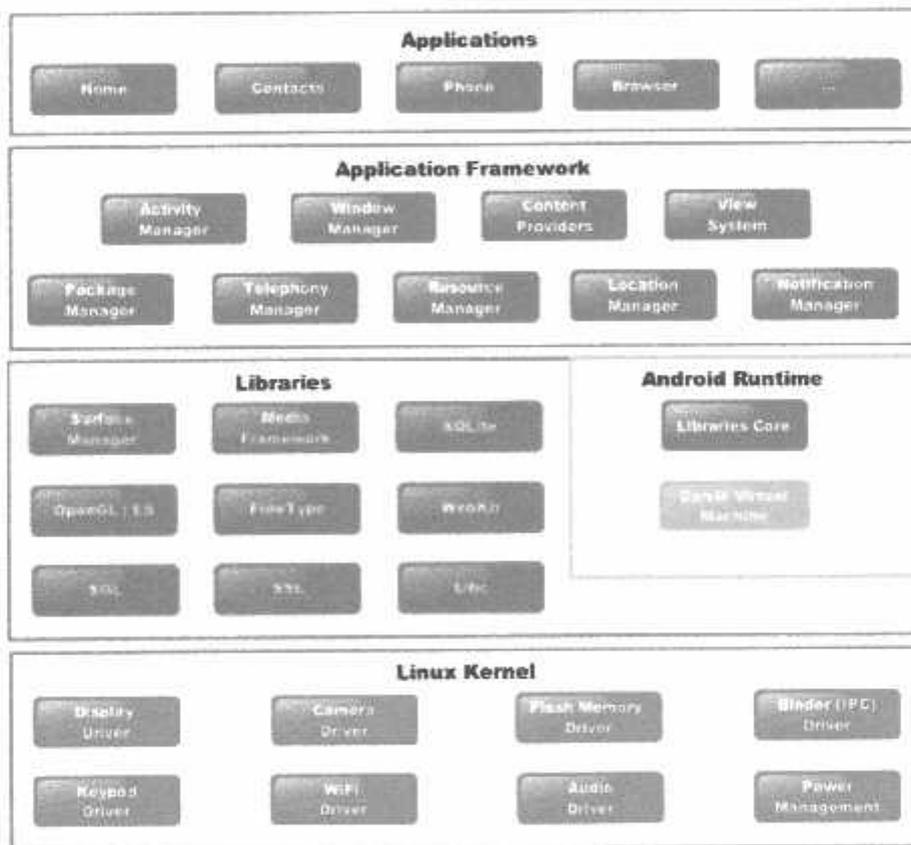
Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis.

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

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

2.2.1. Arsitektur Android

Dalam paket sistem operasi android terdiri dari beberapa unsur seperti tampak pada gambar 2.1 berikut :



Gambar 2.1 Detail Arsitektur Android

Sesuai dengan gambar 2.1 di atas arsitektur android terdiri dari beberapa lapisan, jika dilihat dari lapisan terbawah hingga lapisan teratas terdiri dari :

1. Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan android bukanlah linux, karena dalam android tidak terdapat paket standar yang dimiliki oleh linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel linux menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

2. Android Runtime

Pada android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. *Android Runtime* merupakan mesin virtual yang membuat aplikasi android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam Android Runtime terdapat 2 bagian utama, diantaranya:

- a. Pustaka Inti, android dikembangkan melalui bahasa pemrograman Java, tapi Android Runtime bukanlah mesin virtual Java. Pustaka inti android menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus android.
- b. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable (*.dex)*. Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

3. Libraries

Bertempat di level yang sama dengan Android Runtime. Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel Linux. Beberapa pustaka diantaranya:

- a. *Media Library* untuk memutar dan merekam berbagai macam format audio dan video.
- b. *Surface Manager* untuk mengatur hak akses layer dari berbagai aplikasi.
- c. *Graphic Library* termasuk didalamnya *SGL* dan *OpenGL*, untuk tampilan 2D dan 3D.
- d. *SQLite* untuk mengatur relasi database yang digunakan pada aplikasi.
- e. *SSI* dan *WebKit* untuk browser dan keamanan internet.

Pustaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan pustaka sendiri menggunakan *Native Development Toolkit* (NDK).

4. Application Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi android. Selain itu, juga menyediakan abstraksi generik untuk mengakses perangkat, serta mengatur tampilan *user interface* dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi android adalah sebagai berikut :

1. *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan menjaga keadaan "*Backstack*" untuk navigasi penggunaan.
2. *Content Providers*, berfungsi untuk merangkum data yang memungkinkan digunakan oleh aplikasi lainnya, seperti daftar nama.

3. *Resource Manager*, untuk mengatur sumber daya yang ada dalam program. Serta menyediakan akses sumber daya diluar kode program, seperti karakter, grafik, dan file layout.
4. *Location Manager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat android berada.
5. *Notification Manager*, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada *status bar*.

5. Applications

Puncak dari diagram arsitektur android adalah lapisan aplikasi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program. Pengguna hanya akan melihat program ketika digunakan tanpa mengetahui proses yang terjadi dibalik lapisan aplikasi. Lapisan ini berjalan dalam *Android runtime* dengan menggunakan kelas dan service yang tersedia pada *framework* aplikasi.

Lapisan aplikasi android sangat berbeda dibandingkan dengan sistem operasi lainnya. Pada android semua aplikasi, baik aplikasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunakan pustaka API (*Application Programming Interface*) yang sama.

2.2.2. Komponen Aplikasi

Fitur penting android adalah bahwa satu aplikasi dapat menggunakan elemen dari aplikasi lain (untuk aplikasi yang memungkinkan). Sebagai contoh, sebuah aplikasi memerlukan fitur *scroller* dan aplikasi lain telah mengembangkan fitur *scroller* yang baik dan memungkinkan aplikasi lain menggunakannya. Maka pengembang tidak perlu lagi mengembangkan hal serupa untuk aplikasinya, cukup menggunakan *scroller* yang telah ada.

Agar fitur tersebut dapat bekerja, sistem harus dapat menjalankan aplikasi ketika setiap bagian aplikasi itu dibutuhkan, dan pemanggilan objek java untuk bagian itu. Oleh karenanya android berbeda dari sistem-sistem lain, Android tidak

memiliki satu tampilan utama program seperti fungsi `main()` pada aplikasi lain. Sebaliknya, aplikasi memiliki komponen penting yang memungkinkan sistem untuk memanggil dan menjalankan ketika dibutuhkan.

2.2.2.1 Activity

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *Activity* menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap *Event*. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan.

Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack*. Keadaan suatu *activity* ditentukan oleh posisinya dalam tumpukan *activity*, LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back*, atau menutup *activity* yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

Ketika *activity* diambil dan disimpan dalam tumpukkan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi :

- 1) **Active**, setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.
- 2) **Paused**, dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused*. Keadaan ini terjadi jika *activity*

transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat *active* namun tidak dapat menerima masukan dari pengguna. Dalam kasus ekstrim, android akan menghentikan *activity* dalam keadaan *paused* ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.

- 3) **Stopped**, ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped*. *Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu *activity* dalam kondisi *stopped* maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi *inactive*.
- 4) **Inactive**, kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah diiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali.

Kondisi transisi ini sepenuhnya ditangani oleh manajer memori android. Android akan memulai menutup aplikasi yang mengandung *activity inactive*, kemudian *stopped activity*, dan dalam kasus luar biasa *paused activity* juga akan di tutup.

2.2.2.2 Services

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

2.2.2.3 Intent

Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain sebagainya. *Intents* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi suatu hal, misal pesan masuk. *Intents* dapat eksplisit atau implisit, contohnya jika suatu aplikasi ingin

menampilkan *URL*, sistem akan menentukan komponen apa yang dibutuhkan oleh *Intents* tersebut.

2.2.2.4 Broadcast Recivers

Broadcast Receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *Broadcast* berasal dari sistem misalnya, Baterai sudah hampir habis, informasi zona waktu telah berubah, atau pengguna telah merubah bahasa *default* pada perangkat. Sama halnya dengan *service*, *Broadcast Receivers* tidak menampilkan antarmuka pengguna. Namun, *Broadcast Receivers* dapat menggunakan *Notification Manager* untuk memberitahukan sesuatu kepada pengguna.

2.2.2.5 Content Providers

Content Providers digunakan untuk mengelola dan berbagi database. Data dapat disimpan dalam file sistem, dalam database *SQLite*, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *Content Provider* memungkinkan antar aplikasi untuk saling berbagi data. Komponen ini sangat berguna ketika sebuah aplikasi membutuhkan data dari aplikasi lain, sehingga mudah dalam penerapannya.

2.2.2.6 Google Map API

Google Map API merupakan aplikasi interface yang dapat diakses lewat java agar Google Map dapat ditampilkan pada platform android. Untuk dapat mengakses Google Map, Kita harus melakukan pendaftaran Api Key terlebih dahulu ke Google Maps service dan menyetujui ketentuan layanan yang berlaku.

2.2.2.7 XML

XML (*Extensible Markup Language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML didesain untuk mampu menyimpan data secara ringkas dan mudah diatur. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen.

Dalam Android XML digunakan untuk mendisain *User interface* dengan pengguna. Berikut adalah beberapa *form widget* dan *layout* pada android:

1. **Relative Layout**

RelativeLayout adalah sebuah layout dimana posisi dari sebuah komponen (simbol, text, dsb) bisa diatur letaknya terhadap komponen lainnya.

2. **Linear Layout**

LinearLayout adalah layout yang menampilkan elemen-elemen view dalam arah linear, vertikal atau pun horizontal. Ini adalah layout paling sederhana di Android.

3. **Text View**

TextViw digunakan untuk menampilkan teks pada platform android

2.2.3 Tipe Aplikasi Android

Terdapat tiga kategori aplikasi pada android yaitu :

1. **Foreground Activity**

Aplikasi yang hanya dapat dijalankan jika tampil pada layar dan tetap efektif walaupun tidak terlihat. Aplikasi dengan tipe ini pasti mempertimbangkan siklus hidup *activity*, sehingga perpindahan antar *activity* dapat berlangsung dengan lancar.

2. **Background Service**

Aplikasi yang memiliki interaksi terbatas dengan user, selain dari pengaturan konfigurasi, semua dari prosesnya tidak tampak pada layar. Contohnya aplikasi penyaringan panggilan atau sms auto respon.

3. **Intermittent Activity**

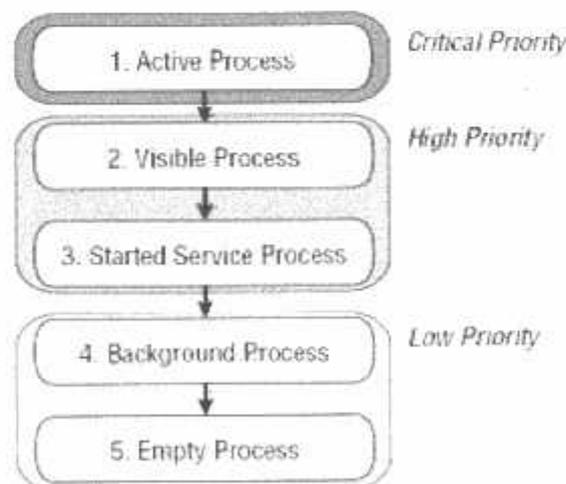
Aplikasi yang masih membutuhkan beberapa masukan dari pengguna, namun sebagian sangat efektif jika dijalankan di *background* dan jika diperlukan akan memberi tahu pengguna tentang kondisi tertentu. Contohnya pemutar musik.

Untuk aplikasi yang kompleks akan sulit untuk menentukan kategori aplikasi tersebut apalagi aplikasi memiliki ciri-ciri dari semua kategori. Oleh karenanya perlu pertimbangan bagaimana aplikasi tersebut digunakan dan menentukan kategori aplikasi yang sesuai.

2.2.4 Siklus Hidup Aplikasi Android

Siklus hidup aplikasi android dikelola oleh sistem, berdasarkan kebutuhan pengguna, sumberdaya yang tersedia, dan sebagainya. Misalnya Pengguna ingin menjalankan browser web, pada akhirnya sistem yang akan menentukan menjalankan aplikasi. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika pengguna ketika itu sedang menjalankan sebuah *Activity*, maka sistem akan memberikan prioritas utama untuk aplikasi yang tersebut. Sebaliknya, jika suatu *Activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *Activity* yang prioritas rendah akan ditutup.

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin virtual pengolah sendiri, dengan ini melindungi penggunaan memori pada aplikasi. Selain itu juga android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karenanya android sangat sensitive dengan siklus hidup aplikasi dan komponen-komponennya. Perlu adanya penanganan terhadap setiap kondisi agar aplikasi menjadi stabil. Untuk lebih jelasnya urutan prioritas dapat dilihat pada gambar 2.2 berikut.



Gambar 2.2 Urutan Prioritas pada Aplikasi Android

2.2.4.1 Active Processes

Active Process berjalan pada *foreground* yang tampak pada layar yang dapat berintraksi langsung dengan pengguna.

2.2.4.2 Visible Processes

Proses terlihat, tetapi tidak di latar depan atau menanggapi perintah user. Hal ini terjadi ketika sebuah aktivitas hanya sebagian yang tertutup. Dan akan di berhentikan secara paksa bila di perlukan supaya Active Processes dapat terus berjalan.

2.2.4.3 Started Service Processes

Service mendukung proses secara berkelanjutan yang harus meneruskan tanpa antarmuka. Karena *Service* tidak berinteraksi langsung dengan user, tetapi dia menerima prioritas yang sedikit lebih rendah daripada Visible Processes. Proses ini masih dianggap proses latar depan(*foreground*) dan tidak akan diberhentikan kecuali sumberdaya tersebut dibutuhkan oleh Active Processes atau Visible Processes.

2.2.4.4 Background Processes

Proses ini masih dianggap sebagai proses *foreground* dan tidak akan dianggap proses *background* meskipun sudah mulai dieksekusi. Pada umumnya ada sejumlah besar proses *background* akan diberhentikan oleh Android untuk dapat memberikan sumberdaya bagi *Foreground Processes*.

2.2.4.5 Empty Processes

Untuk meningkatkan kinerja sistem secara keseluruhan, Android sering mempertahankan aplikasi dalam memori, setelah mencapai akhir eksekusi dari proses. Android mempertahankan *cache* untuk meningkatkan waktu *start-up* aplikasi ketika kembali diluncurkan. Proses *Round-tinely* ini dieksekusi sesuai kebutuhan.

2.2.5 Kelebihan Android

Sudah banyak *platform* untuk perangkat selular saat ini, termasuk didalamnya Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile

Edition, Linux Mobile (LiM0), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut :

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasis Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar *royalty*. Sementara pengembang software menyukai karena android dapat digunakan diperangkat manapun dan tanpa terikat oleh vendor manapun.
2. Arsitektur komponen dasar android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan service, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu khawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suara terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan

Keyboard, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.3. Kecerdasan Buatan Dalam Game

Salah satu unsur yang berperan penting dalam sebuah game adalah kecerdasan buatan. Dengan kecerdasan buatan, elemen-elemen dalam game dapat berperilaku sealami mungkin layaknya manusia.

Game AI adalah aplikasi untuk memodelkan karakter yang terlibat dalam permainan baik sebagai lawan, ataupun karakter pendukung yang merupakan bagian dari permainan tetapi tidak ikut bermain (NPC = Non Playable Character). Peranan kecerdasan buatan dalam hal interaksi pemain dengan permainan adalah pada penggunaan interaksi yang bersifat alami yaitu yang biasa digunakan manusia untuk berinteraksi dengan sesama manusia.

Contoh media interaksi ialah:

- a. Penglihatan (vision)
- b. Suara (voice), ucapan (speech)
- c. Gerakan anggota badan (gesture)

Untuk pembentukan Artificial Intelligence pada game ternyata digunakan pula algoritma, yaitu jenis pohon n-ary untuk suatu struktur. Implementasi pohon (tree) ini biasa disebut game tree. Berdasarkan game tree inilah sebuah game disusun algoritma kecerdasan buatannya. Artificial intelligence yang disematkan dalam sebuah game yang membentuk analisis game tree biasanya merepresentasikan kondisi atau posisi permainan dari game sebagai suatu node, dan merepresentasikan langkah yang mungkin dilakukan sebagai sisi berarah yang menghubungkan node kondisi tersebut ke anak (child) sebagaimana representasi suatu pohon (tree).

Namun, biasanya representasi langsung tersebut mempunyai kelemahan, yaitu representasi data pohon akan menjadi sangat lebar dan banyak. Mungkin bagi sebuah mesin komputer mampu melakukan kalkulasi sebanyak apapun masalah, namun game tree yang lebar dan besar memberikan beberapa masalah,

antara lain konsumsi proses memori, kapasitas penyimpanan yang cukup besar dan kinerja yang kurang pada konsol game berspesifikasi rendah. Karena itu dibentuklah beberapa algoritma dan penyederhanaan bagi sebuah game tree.

Pada salah satu contoh game klasik, yaitu tic tac toe, penyederhanaan dapat dilakukan dengan berbagai metode. Salah satu diantaranya adalah minimax. Metode ini berhasil diterapkan dan memberikan nilai reduksi yang cukup signifikan. Dan tidak hanya bisa digunakan secara monoton, minimax juga bisa digunakan untuk game-game yang lebih rumit seperti catur, tentunya dengan algoritma dan representasi berbeda.

Minimax yang merupakan salah satu metode penerapan (implementasi) pohon n-ary pada suatu game, menandakan bahwa implementasi struktur (pohon khususnya) sangatlah diperlukan pada pembuatan dan penerapan Artificial Intelligence, dan tidak menutup kemungkinan ilmu dan metode baru yang lebih canggih akan ditemukan di masa depan.

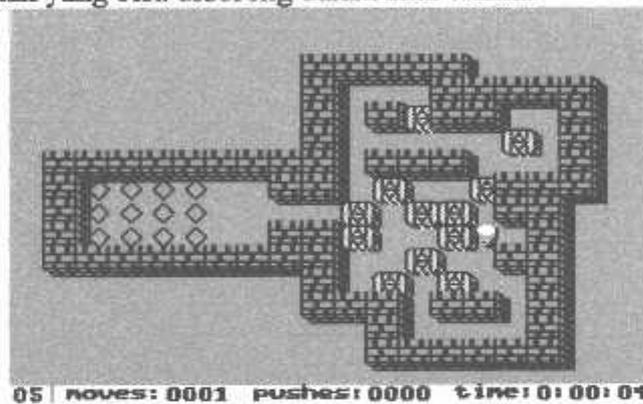
2.4. PUSH BOX

Push Box pertama kali dikenalkan pada tahun 1982 oleh THINKING RABBIT, sebuah perusahaan permainan komputer di kota Takarazuka, Jepang. Push Box adalah game teka-teki klasik, Game yang aslinya bernama Sokoban ini dibuat oleh Hiroyuki Imabayashi. Belakangan ini Push Box adalah salah satu game teka-teki yang paling populer untuk dimainkan.



Gambar 2.3 Game Push Box pertama pada Apple IIe

Nama Sokoban adalah dari Jepang yang berarti "Penjaga Gudang". Sesimpel namanya begitu juga dengan ide game ini : setiap level menampilkan sebuah ruangan, dimana kotak-kotak muncul secara acak. Pemain bertugas untuk membantu Penjaga Gudang mendorong kota-kotak dalam gudang yang ruwet sehingga pada akhirnya semua kotak berada pada tempat yang sudah ditentukan. Peraturannya adalah pemain hanya bisa mendorong kotak, tidak bisa menarik, dan hanya satu kotak yang bisa didorong dalam satu waktu.



Gambar 2.4 Gameplay Push Box

Setiap level mempunyai perbedaan struktur gudang/ruangan, yang membutuhkan solusi yang berbeda untuk tiap levelnya. Kelebihan dari game ini adalah pada kotaknya, kotak yang sudah ditempatkan ditempatnya, bisa jadi akan menghalang langkah berikutnya. Kemudahan peraturan permainannya, kombinasi dari tingkat kesulitan tiap levelnya, telah membuat Push Box menjadi game klasik dan selalu menyenangkan untuk dimainkan berulang-ulang.

BAB III PERANCANGAN DAN DESAIN SISTEM

3.1. Pembahasan

Game Push Box ini adalah suatu game berbasis android yang dibangun untuk memenuhi kebutuhan hiburan bagi pengguna ponsel disela-sela waktu luang untuk menghindari kebosanan dalam aktivitas sehari-hari. Game Push Box merupakan salah satu game bertemakan logika sehingga seorang pemain tidak membutuhkan kecekatan tangan tetapi lebih kepada kecekatan berpikir.

Pemain game Push Box bertugas memindahkan paket-paket yang disediakan menuju lokasi-lokasi tertentu yang ditetapkan dengan menggerakkan pusher untuk mendorong paket menuju goal. List di bawah ini akan menjelaskan kegiatan yang dapat dilakukan pemain.

1. memulai permainan
2. memainkan level yang telah tersedia
3. main pada level berikutnya
4. melihat score permainan
5. keluar dari permainan
6. mendorong box ke kanan, kiri, atas dan bawah tapi tidak diagonal
7. bergerak ke kanan, kiri, atas dan bawah tapi tidak diagonal
8. kembali pada gerakan sebelumnya
9. pemain dapat menggerakkan box jika tidak ada box yang lain atau dinding yang menghalangi
10. pemain dapat bergerak selama tidak ada dinding penghalang.

3.2 Perancangan

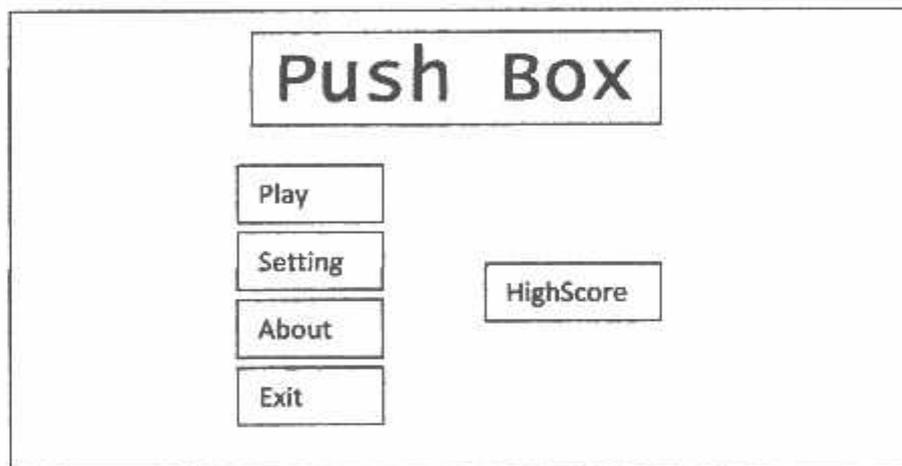
Aplikasi permainan Push Box ini dirancang dengan menggunakan bahasa pemrograman android.

A. Rancangan Tampilan

Berikut beberapa rancangan tampilan dari aplikasi ini :

1. Tampilan Menu Utama

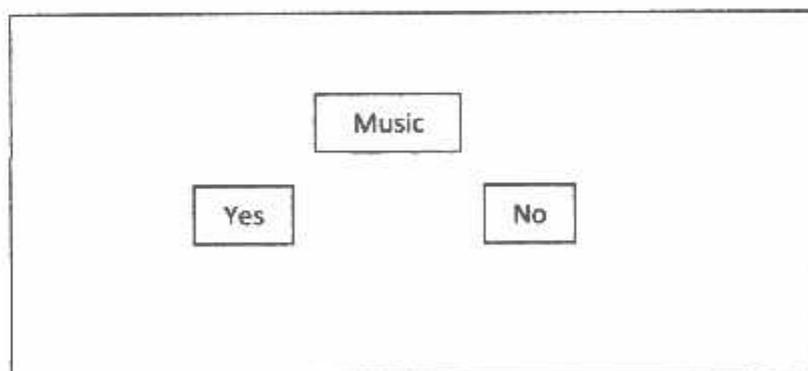
Halaman ini dirancang diisi dengan beberapa menu seperti Play, Setting, About, dan HighScore. Tampilan menu utama dapat dilihat pada gambar 3.1.



Gambar 3.1 Rancangan Tampilan Menu Utama

2. Tampilan Setting

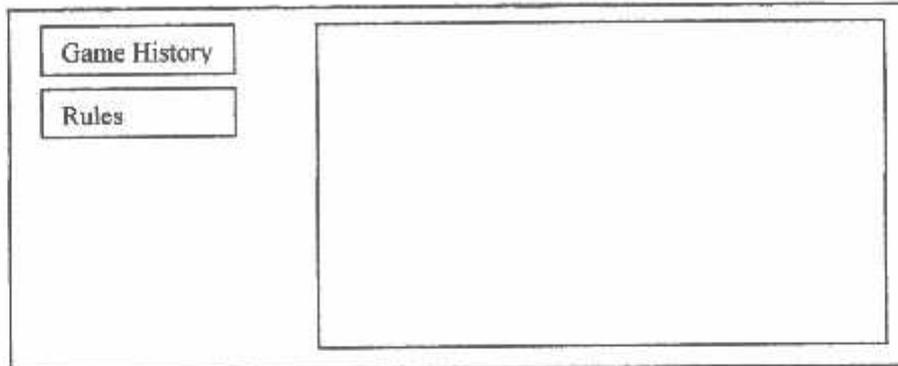
Halaman ini berisi setting suara, untuk dipakai atau tidak selama permainan berlangsung. Tampilan menu options dapat dilihat pada gambar 3.2.



Gambar 3.2 Rancangan tampilan Menu Setting

3. Tampilan About

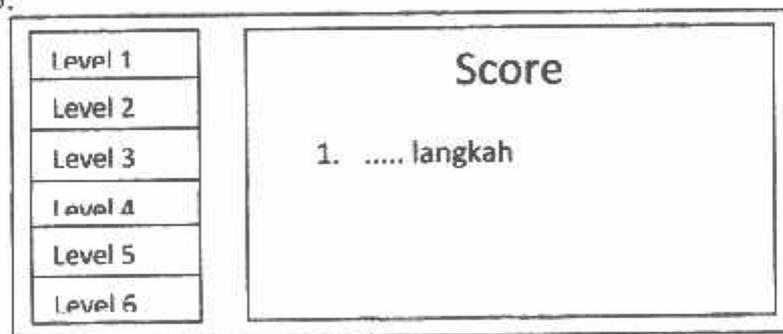
Halaman ini berisi tentang sejarah permainan, dan peraturan serta cara bermain. Tampilan Help dapat dilihat pada gambar 3.4.



Gambar 3.3 Rancangan tampilan menu About

4. Tampilan HighScore

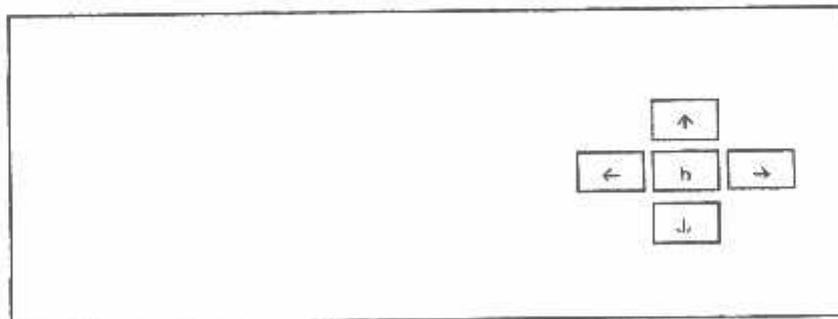
Halaman ini berisi skor tertinggi yang pernah diperoleh dari permainan untuk tiap level. Tampilan Score dapat dilihat pada gambar 3.5.



Gambar 3.4 Rancangan tampilan Menu Score

5. Tampilan Permainan

Halaman ini berisi *display* permainan yang sedang berlangsung. Tampilan permainan dapat dilihat pada gambar 3.6.



Gambar 3.5 Rancangan tampilan permainan

B. Pergerakan Pemain.

Dalam menentukan pergerakan pemain, penanganan *event* merupakan hal yang penting, sehingga aplikasi yang dibuat dapat menangani kejadian yang dilakukan *player*.

EventListener adalah sebuah antarmuka dikelas *view* yang berisi *method* pemanggilan tunggal. *Method* ini akan dipanggil oleh kerangka kerja android ketika *event* yang terdaftar dipicu oleh interaksi pengguna/player. *onTouch* () adalah *event* yang digunakan untuk menggerakkan pemain dalam game ini, *event* ini terjadi bila pengguna melakukan tindakan sentuhan ke layar yang dianggap tindakan sentuhan adalah seperti menekan, melepas, atau melakukan gerakan dilayar pada batas-batas item. Berikut adalah item yang digunakan pada aplikasi ini :

Tabel 3.1 Tabel Item

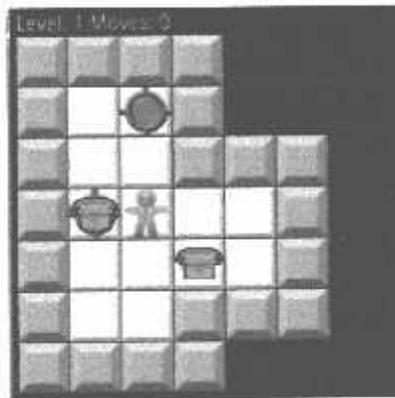
No.	Item	Fungsi	Keterangan
1		Menggerakkan player kearah bawah	Bisa disentuh/tombol
2		Menggerakkan player kearah kiri	Bisa disentuh/tombol
3		Menggerakkan player kearah kanan	Bisa disentuh/tombol
4		Menggerakkan pemain kearah atas	Bisa disentuh/tombol
5		Membatalkan langkah yang telah diambil	Bisa disentuh/tombol
6		Player/Pusher	Tidak bisa disentuh
7		Box	Tidak bisa disentuh
8		Tujuan	Tidak bisa disentuh
9		Lantai	Tidak bisa disentuh
10		Tembok/Pembatas	Tidak bisa disentuh

C. Pembuatan Level

Dalam game ini disediakan 20 level, dengan tingkat kesulitan berbeda. Penentuan tingkatan level berdasarkan :

1. Besar atau luasnya ruangan.
2. Jumlah box.

Namun dalam pembuatannya sebuah level seharusnya dibuat tidak melebihi ukuran 9 x 12 simbol, agar dapat menyesuaikan dengan besarnya layar handphone. Sebagai perbandingan level 1 memiliki ukuran 6 x 7 simbol.



Gambar 3.6 Tampilan level 1

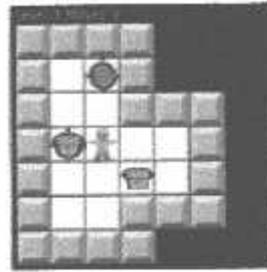
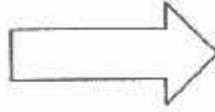
Pada gambar 3.6 untuk mempermudah pembuatan level setiap item seperti , , ,  (keterangan item pada tabel 3.1), masing-masing item diwakili oleh simbol seperti @, \$, ., spasi, #.

Tabel 3.2 Tabel Simbol

No.	Simbol	Item
1	@ (at)	
2	\$ (dollar)	
3	. (titik)	
4	(spasi)	
5	# (pagar)	

Perancangan bentuk level dapat dibuat dalam notepad dan kemudian disimpan dalam bentuk .txt, sehingga isi dari file level-1.txt adalah sebagai berikut.

```
####
# .#
# ###
#*@ #
# $ #
# ###
####
```



3.3. Konsep Game

Langkah awal yang dilakukan dalam pembuatan game adalah konsep atau *storyboard* yang dirumuskan dalam bentuk table. *Storyboard* yang disusun merupakan gambaran umum aplikasi game yang akan dihasilkan melalui tahapan perancangan yang meliputi judul, jenis game, sistem kendali yang digunakan, penentuan target, penentuan *background* dan gambar sistem permainan. Tabel konsep atau *Storyboard* dapat di lihat pada Tabel 3.2.

Tabel 3.3. Konsep Game

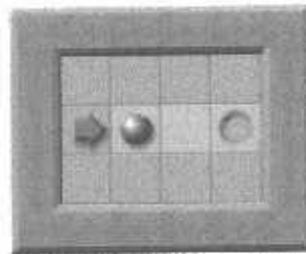
Judul Game	Push Box
Jenis Game	Puzzle
Jenis Rating	Semua Umur
Sistem Kendali	Aplikasi game ini menggunakan layar <i>touchscreen</i> maka cukup menyentuh layar
Target	Setiap box menuju ketempat yang ditentukan
Backgroud	<i>Static view</i> dengan ukuran stage 240 x 320
Sistem Permainan	<ul style="list-style-type: none"> - Game ini mempunyai 20 level dimana setiap level tingkat kesulitan berbeda-beda - Pemain mendorong box menuju tempat yang sudah ditentukan

3.4 Aturan permainan

Peraturan bermain sokoban sangat sederhana, pemain hanya mengontrol pusher untuk melangkah dan mendorong semua box sehingga menempati tempat yang sudah ditentukan dalam suatu maze. Pemain dapat bergerak ke kanan, ke kiri, ke atas, ke bawah, dan mengulang langkah namun pemain juga harus berhati-hati untuk menghindari kondisi *deadlock*, kondisi *deadlock* terjadi dimana tidak ada gerakan lagi yang bisa dilakukan untuk menggerakkan box.

Berikut beberapa contoh kondisi *deadlock* :

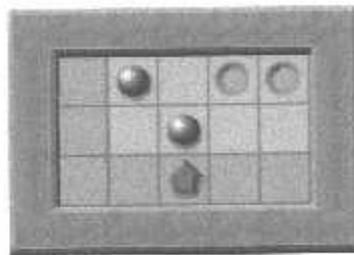
1. *Dead square deadlocks*



Gambar 3.7 *Dead Square Deadlock*

Pemain bisa mendorong Box menuju semua arah. Tapi dengan mendorong Box ke arah tempat yang berwarna gelap akan menyebabkan kondisi *deadlock*.

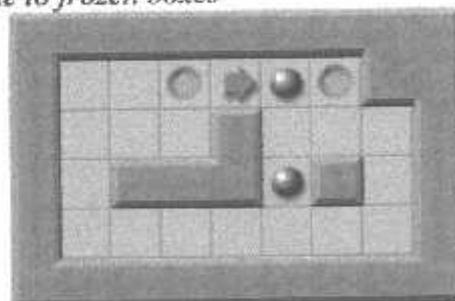
2. *Freeze deadlocks*



Gambar 3.8 *Freeze Deadlock*

Jika pemain mendorong bola ke atas akan menyebabkan *deadlock*. Box menjadi tidak bisa digerakan lagi dan tidak satu box pun yang bisa menuju tujuannya.

3. *Deadlocks due to frozen boxes*



Gambar 3.9 *Deadlocks due to frozen boxes*

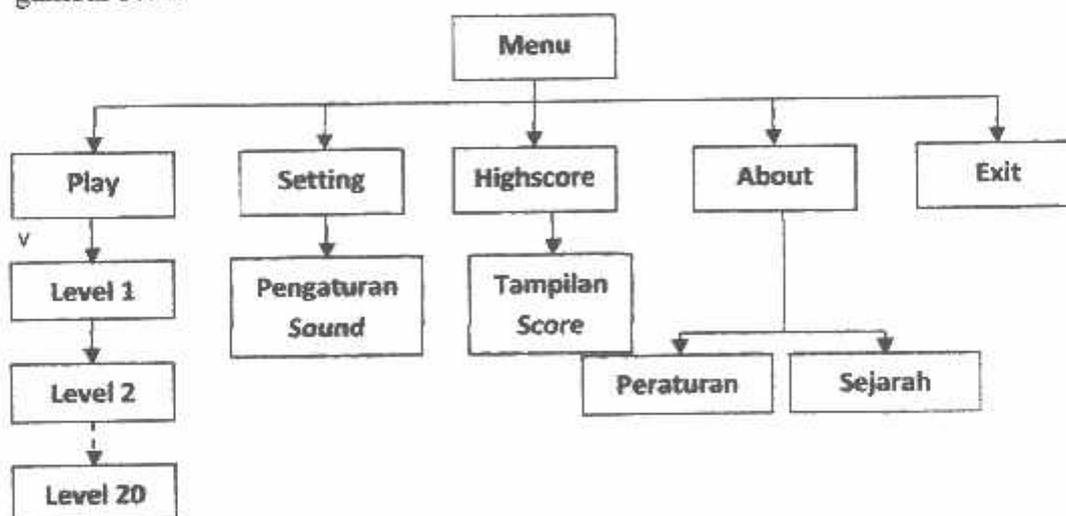
Jika Player mendorong box menuju tujuan (kanan) tidak akan menyebabkan kondisi *Freeze deadlock* karena box sudah pada tujuan. Meskipun begitu akan menyebabkan box lainnya mengalami *deadlock*, sehingga tidak bisa didorong ke tujuan lagi.

Pemain dapat melanjutkan permainan ke level selanjutnya jika semua box sudah berada pada tujuannya masing-masing. Berikut adalah perintah yang akan dijalankan untuk maju ke level selanjutnya.

```
public void advanceLevel()
{
    playSound(mClapSound);
    scoreActivity.setNewScore(m_level+" "+Moves);
    setLevel(m_level + 1);
}
```

3.5 Desain Sistem

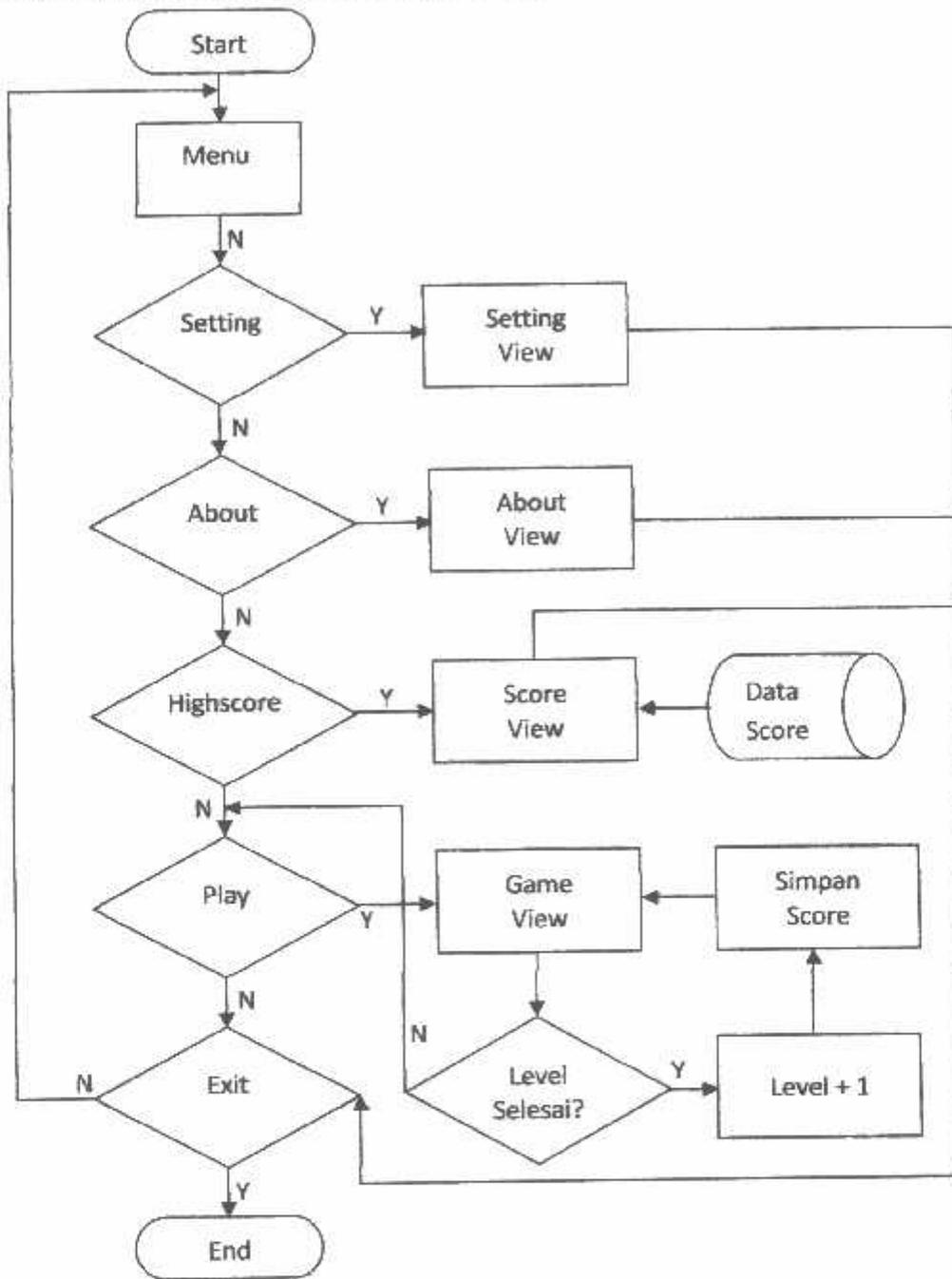
Secara garis besar game ini mempunyai desain sistem seperti ditunjukkan pada gambar 3.10.



Gambar 3.10 Desain sistem Game Push Box

3.6 Flowchart

Gambar 3.11 adalah alur permainan game Push box :



Gambar 3.11 Flowchart Game Push Box

BAB IV IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap implementasi pengembangan perangkat lunak merupakan proses pengubahan spesifikasi sistem menjadi sistem yang dapat dijalankan. Tahap ini merupakan lanjutan dari proses perancangan, yaitu proses pemrograman perangkat lunak sesuai dengan spesifikasi dan desain sistem.

Pembuatan aplikasi game Push Box ini menggunakan Java, Android SDK, ADT/Plugins Eclipse, dan Eclipse. Android adalah aplikasi yang dikembangkan dengan berbasis java, sehingga sebelum melakukan *coding* aplikasi berbasis android, komputer/laptop harus sudah terinstal program java.

ADT (*software development kit*) ini diperlukan sebagai alat bantu dan API dalam mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman java. Eclipse berfungsi sebagai IDE (*Integrated Development Environment*) menggunakan bahasa pemrograman java kemudian dikompilasi bersama data file resource yang dibutuhkan oleh aplikasi, dimana prosesnya di-*package* oleh tool yang dinamakan "apt tools" ke dalam paket android sehingga menghasilkan file dengan ekstensi .apk.

4.1.1 Kebutuhan Hardware

Perangkat keras (*hardware*) adalah handphone yang sudah berbasis android. Rincian Perangkat keras (*Hardware*) pendukung seperti dalam Tabel 4.1

Tabel 4.1 Spesifikasi Perangkat Keras.

Perlengkapan	Spesifikasi	Keterangan
Handphone	Processor	600 MHz ARMv6
	Memori	384 RAM
	Resolusi	240 x 320

4.1.2 Kebutuhan Software

Spesifikasi yang digunakan untuk membuat aplikasi ini *minimum requirement* untuk menjalankan aplikasi adalah android dengan sistem operasi android 2.2 (Froyo).

4.2 Pengujian Sistem

Perangkat lunak ini didesain pada sistem operasi android 2.2 (Froyo) dengan resolusi monitor 240 x 320 pixel. Oleh karena itu, akan diuji dengan menggunakan beberapa handphone berbeda-beda dengan resolusi monitor yang berbeda – beda juga.

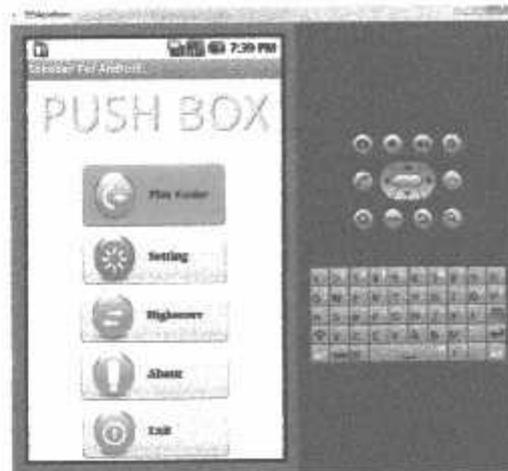
Pengujian pertama dilakukan dengan spesifikasi perangkat keras dan perangkat lunak sebagai berikut :

- | | |
|-------------------|-----------------------------|
| 1) Handphone | : Samsung Galaxy Mini S5570 |
| 2) Processor | : single core 600 Mhz |
| 3) Memory | : 2Gb, 384 RAM |
| 4) Resolusi | : 240 x 320 pixel |
| 5) Sistem Operasi | : Froyo |
| 6) Versi Android | : 2.2 |

4.2.1 Pengujian Menu Utama

Bagian ini adalah bagian yang pertama kali keluar saat game di nyalakan, setelah game Push Box dinyalakan seorang pemain dapat melihat tombol *Play*, *Setting*, *Highscore*, *About* dan *Exit*. Tombol *Play* berfungsi untuk menuju ke halaman level, tombol *Setting* berfungsi untuk menuju ke halaman *Setting* dimana di halaman tersebut terdapat pengaturan bunyi, tombol *Highscore* berfungsi untuk menuju halaman *Highscore*, dan tombol *About* berfungsi untuk menuju ke halaman *About* yang menceritakan tentang aplikasi game ini.

Halaman *menu utama* di buat dengan sangat menarik karena fungsi dari halaman ini adalah untuk dilihat pengunjung secara umum. Tampilan menu utama



Gambar 4.1 Tampilan Halaman Menu Utama.

4.2.2 Pengujian menu *Setting*

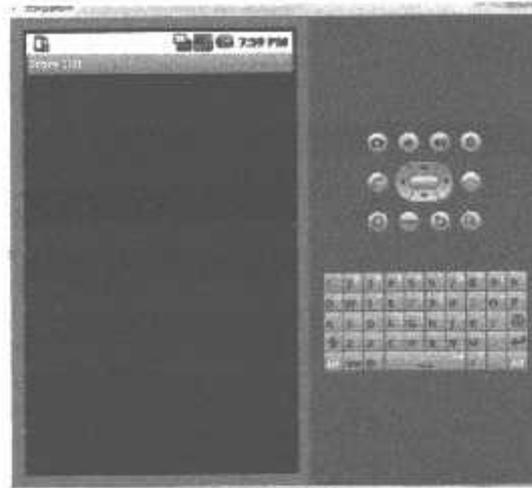
Bagian ini adalah bagian untuk merubah settingan didalam aplikasi game Push Box. Didalam halaman *Setting* pemain dapat merubah settingan suara diaktifkan atau di nonaktifkan, apabila suara diaktifkan maka akan muncul suara pada saat game berjalan begitu juga sebaliknya apabila suara dinonaktifkan maka tidak muncul suara selama game berjalan. Tampilan menu utama ini dapat dilihat pada gambar 4.2.



Gambar 4.2 Tampilan Halaman Setting.

4.2.3. Pengujian menu *HighScore*

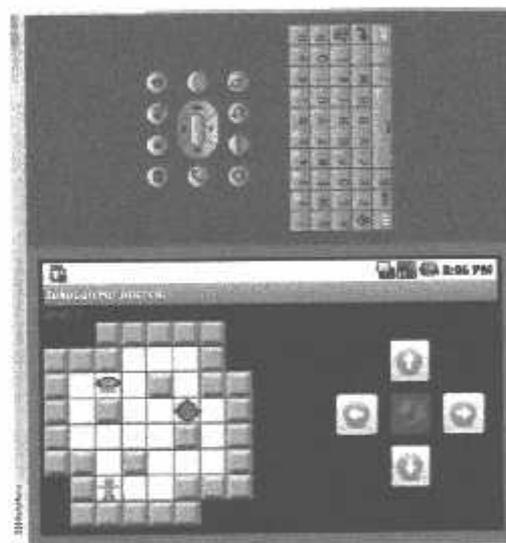
Bagian ini adalah bagian yang menunjukkan nilai terbaik pada game Push Box, didalam halaman ini pemain akan ditunjukkan 5 nilai terbaik untuk setiap level yang telah diperoleh dari permainan sebelumnya. Tampilan *HighScore* ini dapat dilihat pada gambar 4.3.



Gambar 4.3 Tampilan Halaman HighScore.

4.2.4 Pengujian Permainan

Bagian ini adalah bagian permainan yang sedang berlangsung, didalam halaman ini pemain dapat bermain dengan cara mendorong box menuju tempat tujuan yang sudah ditentukan. Tampilan *Level* ini dapat dilihat pada gambar 4.4.

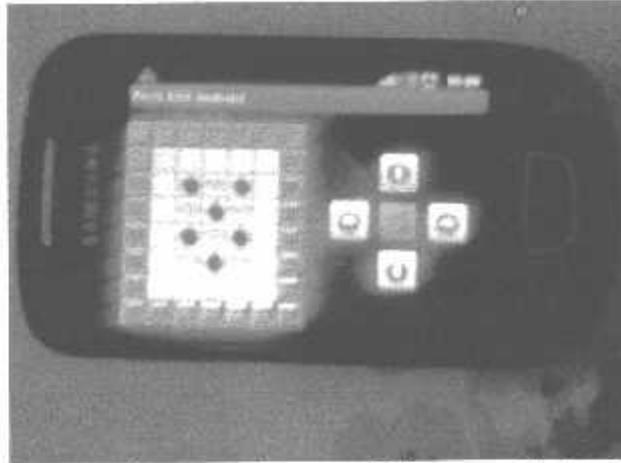


Gambar 4.4 Tampilan Halaman Permainan.

4.3 Pengujian Aplikasi

Pada uji coba aplikasi game ini telah dilakukan pada handphone Samsung Galaxy Mini S5570 dengan system operasi android 2.2 (Froyo) , Samsung Galaxy Y S5360 dengan sistem operasi android 2.3.5 (gingerbread), Samsung 19100 Galaxy S II.

1). Samsung Galaxy Mini S5570.



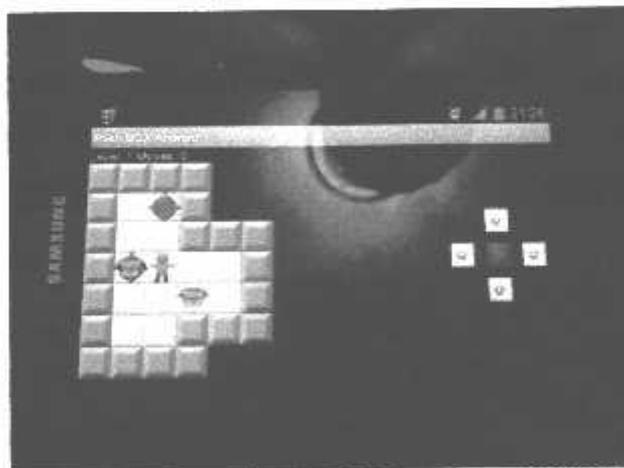
Gambar 4.5 Tampilan permainan pada Samsung Galaxy Mini S5570

2). Samsung Galaxy Y S5360



Gambar 4.6 Tampilan permainan pada Samsung Galaxy Y S5360

3). Samsung 19100 Galaxy S II



Gambar 4.7 Tampilan permainan pada Samsung 19100 Galaxy S II

BAB V PENUTUP

5.1 Kesimpulan

1. Aplikasi game Push Box ini dapat digunakan pada handphone manapun dengan resolusi yang berbeda-beda yang mendukung sistem operasi Android.
2. Kelebihan dari aplikasi game Push Box yang dibuat oleh penulis ini adalah level yang dimainkan sederhana namun cukup rumit dengan 20 level yang harus diselesaikan dan penentuan *highscore* yang dihitung dari jumlah langkah yang dilakukan untuk menyelesaikan satu level.

5.2 Saran

1. Perlu adanya peningkatan tampilan kualitas grafik dari aplikasi game push box ini masih perlu dikembangkan kearah yang lebih baik lagi seperti pencitraan 3 Dimensi untuk tampilan permainan.
2. Jumlah level yang lebih banyak dan rumit agar pemain/user tidak mudah bosan.

DAFTAR PUSTAKA

1. Indrajani dan Martin. 2007. *Pemrograman Berbasis Objek dengan Bahasa Java*. Jakarta : PT. Elex Media Komputindo.
2. Siregar, Ivan Michael dkk. 2010. *Mengembangkan Aplikasi Enterprise Berbasis Android*. Yogyakarta : Gaya Media.
3. S. Mulyadi. 2007. *Membuat Aplikasi untuk Android*. Bandung : Informatika.
4. Winarno ST, M.Eng. 2011. *Membuat Sendiri Aplikasi Android untuk Pemula*. Jakarta : PT. Elex Media Komputindo.
5. Priyanta, F. *Pemrograman Android untuk Pemula*. Cerdas Pustaka Publisher
6. Hermawan, Stephanus. 2011. *Mudah Membuat Aplikasi Android*. Yogyakarta : Penerbit Andi.
7. <http://www.android.com>, 19 April 2012
8. <http://www.oracle.com>, 19 April 2012
9. http://sokobano.de/wiki/index.php?title=Main_Page, 13 April 2012

LAMPIRAN

MenuActivity.java

```
package com.xcmzcm.androidstuff.sokoban;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

public class menuActivity extends Activity {
    Button btnPlay, btnSetting, btnAbout, btnScore, btnExit;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.mcnu);

        btnPlay = (Button) findViewById(R.id.btnPlay);
        btnSetting = (Button) findViewById(R.id.btnSetting);
        btnAbout = (Button) findViewById(R.id.btnAbout);
        btnScore = (Button) findViewById(R.id.btnScore);
        btnExit = (Button) findViewById(R.id.btnExit);
        btnExit.setOnClickListener(new OnClickListener() {

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

```
        // TODO Auto-generated method stub
        quit();
    }
});

btnPlay.setOnClickListener(new OnClickListener() {

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

btnAbout.setOnClickListener(new OnClickListener() {

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

btnSetting.setOnClickListener(new OnClickListener() {

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

```

        }
    });
    btnScore.setOnClickListener(new OnClickListener() {

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

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

protected void showScore() {

    startActivity(new Intent( this,scoreActivity.class));
}

protected void showAbout() {
    startActivity(new Intent( this,AboutActivity.class));
}

public void quit() {
    AlertDialog.Builder builder = new AlertDialog.Builder( this);
    builder.setMessage("Are you sure to end Application?")
        .setCancelable(false)
        .setPositiveButton("Yes", new DialogInterface.OnClickListener()

```

```
        public void onClick(DialogInterface dialog, int id) {  
            finish();  
        }  
    }  
    .setNegativeButton("No", new DialogInterface.OnClickListener()  
{  
        public void onClick(DialogInterface dialog, int id) {  
            dialog.cancel();  
        }  
    });  
    AlertDialog alert = builder.create();  
    alert.show();  
}
```

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

Move.java

```
package com.xomzom.androidstuff.sokoban;

/**
 * A class representing a single game move.
 */
public class Move
{
    /**
     * Constants - Directions.
     */

    /**
     * The 'Up' direction.
     */
    public final static int DIR_UP = 0;

    /**
     * The 'Down' direction.
     */
    public final static int DIR_DOWN = 1;

    /**
     * The 'Left' direction.
     */
    public final static int DIR_LEFT = 2;

    /**
     * The 'Right' direction.
     */
    public final static int DIR_RIGHT = 3;

    /**
     * Members.
     */

    /**
     * The move direction.
     */
    private int a_dir;

    /**
     * A flag to indicate that this move moved a block. Used for undoing a
     * move.
     */
    private boolean a_isMoving = false;

    /**
     * Operation.
     */

    /**
     * Create a move object representing a move in the given direction.
     */
    public Move(int dir)
    {
        a_dir = dir;
    }
}
```

```

...
 * Get the change in the 'x' coordinate when performing this move.
 */
public int getXDelta()
{
    int xDelta;
    switch (m_dir)
    {
        case DIR_LEFT:
            xDelta = -1;
            break;
        case DIR_RIGHT:
            xDelta = 1;
            break;
        default:
            xDelta = 0;
            break;
    }
    return xDelta;
}

...
 * Get the change in the 'y' coordinate when performing this move.
 */
public int getYDelta()
{
    int yDelta;
    switch (m_dir)
    {
        case DIR_UP:
            yDelta = -1;
            break;
        case DIR_DOWN:
            yDelta = 1;
            break;
        default:
            yDelta = 0;
            break;
    }
    return yDelta;
}

...
 * Get the move direction.
 * Return the move direction.
 */
public int getDir()
{
    return m_dir;
}

...
 * Get the 'isMoving' flag, which indicates that this move is moving a
 * block.
 * Return the value of the 'isMoving' flag.
 */
public boolean isMoving()
{
    return m_isMoving;
}

...

```

```

    * Set the "isMoving" flag, to indicate that this move is moving a
block.
    * Return isMoving true if the move is moving a block.
    */
public void setMoving(boolean isMoving)
{
    m_isMoving = isMoving;
}
}

```

AboutActivity.java

```
package com.xomzom.androidstuff.sokoban;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

public class AboutActivity extends Activity {

    Button btnSejarah,btnRule ;
    TextView txData;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.about);

        btnSejarah =(Button) findViewById(R.id.btnSejarah);
        btnRule =(Button) findViewById(R.id.btnRule);
        txData =(TextView) findViewById(R.id.txtData);
        txData.setText("Sejarah & Peraturan Permainan");
        btnSejarah.setOnClickListener(new OnClickListener() {

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

SettingActivity.java

```
package com.xonzon.androidstuff.sokoban;

import android.content.Context;
import android.os.Bundle;
import android.preference.PreferenceActivity;
import android.preference.PreferenceManager;

public class settingActivity extends PreferenceActivity{
    private static final String PIL_SOUND="pilihan_setting_suara";
    private static final String PIL_SOUND_DEF="0";

    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.layout.pengaturan);
    }

    public static String getSoundSetting(Context context){
        return PreferenceManager.getDefaultSharedPreferences(context)
            .getString(PIL_SOUND, PIL_SOUND_DEF);
    }
}
```

ScoreActivity.java

```
package com.xomzom.androidstuff.sokoban;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Iterator;

import android.R.integer;
import android.app.ListActivity;
import android.os.Bundle;
import android.os.Environment;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.AdapterView;

public class scoreActivity extends ListActivity {
    private static final int MENU_RESET = 0;
    private static final int MENU_BACK = 1;
    static String fileScore =
Environment.getExternalStorageDirectory()+"/my.score".db";

    public static scoreActivity innstace ;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);

        setListAdapter(new ArrayAdapter<String>(this, R.layout.score,
R.id.player_score, readFile()));
    }
}
```

```
}
```

```
@Override
```

```
public boolean onPrepareOptionsMenu(Menu menu) {  
    menu.clear();  
  
    menu.add(0, MENU_RESET, 0, "Reset Score").setIcon(  
        android.R.drawable.ic_menu_revert);  
    menu.add(0, MENU_BACK, 0, "Back").setIcon(  
        android.R.drawable.ic_menu_more);  
  
    return true;  
}
```

```
@Override
```

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch(item.getItemId()) {  
        case MENU_RESET: {  
  
            resetScore();  
            this.finish();  
  
            break;  
        }  
        case MENU_BACK: {  
            this.finish();  
  
            break;  
        }  
    }  
}
```

```

        return true;
    }

    public static void setNewScore(String Scores){
        String[] scrList = readFile();
        for (int i = 0; i < scrList.length; i++) {
            String tmp=scrList[i];

            tmp = tmp.replace("Level ", "").trim();
            tmp = tmp.replace(" :", ":").trim();
            tmp = tmp.replace(" Moves", "").trim();

            scrList[i]=tmp;
        }
        String[] tmp = Scores.trim().split(":");
        scrList[Integer.parseInt(tmp[0])-1] = tmp[0]+":"+tmp[1] ;
        resetScore();

        FileWriter f;

        try {
            f = new FileWriter(fileScore);
            for (int i = 0; i < scrList.length; i++) {
                f.write(scrList[i]+"\n");
            }

            f.flush();
            f.close();
        }catch (Exception e) {
            System.out.println("Errorot :" + e.toString());
        }
    }

```

```
}  
  
public static void resetScore() {  
    FileWriter f;  
    try {  
        f = new FileWriter(fileScore);  
        f.write("");  
        f.flush();  
        f.close();  
    } catch (Exception e) {  
        System.out.println("Error : " + e.toString());  
    }  
}
```

```
public static String[] readFile() {  
  
    Integer [] valScore = new Integer[21];  
    for (int i = 1; i <= 20; i++) {  
        valScore[i] = 0;  
    }  
  
    StringBuffer sb = new StringBuffer();  
    try {  
        FileReader f = new FileReader(fileScore);  
        BufferedReader br = new BufferedReader(f);  
        String line;  
  
        while ((line = br.readLine()) != null) {
```

```

        sb.append(line);

        sb.append("\n");

        int tmp = Integer.parseInt(line.trim().split(":")[0]);

        valScore[tmp] =
Integer.valueOf(line.trim().split(":")[1]);

    }

    sb.toString().trim();

    br.close();

    f.close();

}

catch ( Exception e) {

    System.out.println("Error : " + e.toString());

}

sb = new StringBuffer();

for (int i = 1; i <= 20; i++) {

    sb.append("Level "+i+" :"+String.valueOf(valScore[i])+

Moves");

    sb.append("=");

}

if (sb.length()!=0){

    return sb.toString().split("=");

}else{

    return new String[] {};

}

}

}

```



PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

Nama : Abdi k. Radja
 N I M : 07.12.649
 Semester :
 Fakultas : Teknologi Industri
 Jurusan : Teknik Elektro S-1
 Konsentrasi : TEKNIK ELEKTRONIKA
TEKNIK ENERGI LISTRIK
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 tersebut, bersama 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 mata kuliah ≥ 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 kebenaran data tersebut diatas
Recording Teknik Elektro

(.....)

Malang,201

Pemohon

(.....)

Disetujui
Ketua Jurusan Teknik Elektro

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

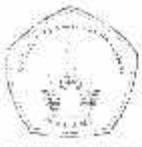
Mengetahui
Dosen Wali

(.....)

Catatan :

Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. IP 426 / 138 = 3.09
2.
3. - prakt Das. Telekomunikasi



Lampiran : 1 (satu) berkas
Pembimbing Skripsi

kepada : Yth. Bapak/Ibu **M. Ibrahim Ashari, ST, MT**
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan dibawah

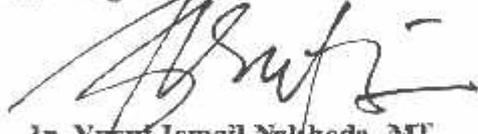
Nama : **ABDI KRADJA**
Nim : **0712645**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer & Informatika**

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

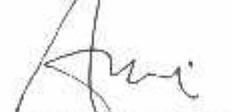
"RANCANG BANGUN GAME PUSH BOX MENGGUNAKAN ECLIPSE BERBASIS ANDROID"

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

Mengetahui
Ketua Program Studi Teknik Elektro S-1


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

Hormat Kami


ABDI KRADJA
NIM. 0712645



PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i :

Nama : **ABDI K RADJA**
Nim : **0712645**
Semester : **X (Sepuluh)**
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 PUSH BOX MENGGUNAKAN ECLIPSE BERBASIS ANDROID"

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Hormat Kami

M. Ibrahim Ashari, ST, MT

NIP.P. 1030100358

*) Coret yang tidak perlu



jumlah : 1 (satu) berkas
Pembimbing Skripsi

kepada : Yth. Bapak/Ibu **Sandy Natali Mantja, SKom**
Dosen Teknik Elektro S-1
ITN Malang

Yang bertanda tangan di bawah

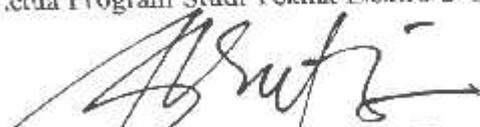
Nama : **ABDI K RADJA**
Nim : **0712645**
Jurusan : **Teknik Elektro S-1**
Konsentrasi : **Teknik Komputer & Informatika**

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

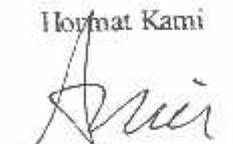
"RANCANG BANGUN GAME PUSH BOX MENGGUNAKAN ECLIPSE BERBASIS ANDROID"

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

Mengetahui
Ketua Program Studi Teknik Elektro S-1


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

Hormat Kami


ABDI K RADJA
NIM. 0712645



FORMULIR BIMBINGAN SKRIPSI

Nama : ABDI K. RADJA
NIM : 07.12.645
Masa Bimbingan :
Judul Skripsi : RANCANG BANGUN GAME PUSH BOX
MENGUNAKAN ECLIPSE BERBASIS
ANDROID

no.	Tanggal	Uraian	Paraf Bimbingan
1.	24 msi 12	Revisi bab I dan bab II	
2.	4 juni 12	Acc Bab I.	
3.	26 juni 12	Revisi Bab II dan Bab III	
4.			
5.			
6.			
7.			
8.			
9.			
10.			

Malang, 10 Mei 2012
Dosen Pembimbing,

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



FORMULIR BIMBINGAN SKRIPSI

Nama : ABDI K. RADJA
NIM : 07.12.645
Masa Bimbingan :
Judul Skripsi : RANCANG BANGUN GAME PUSH BOX
MENGUNAKAN ECLIPSE BERBASIS
ANDROID

No.	Tanggal	Uraian	Paraf Bimbingan
1.	SELASA 12/5/12	REVISI URAIAN BAB I BAB II OK, CANTIT BAB III	
2.	SUMPAH 10/7/12	REVISI: BAB III (REVISI)	
3.	RABU 16/2/12	REVISI TAK ADA	
4.			
5.			
6.			
7.			
8.			
9.			
10.			

Malang,
Dosen Pembimbing,

Sandy Natali Mantja, SKom
NIP.P.1030800418



Formulir Perbaikan Ujian Skripsi

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

NAMA : Herf Abdi K. Raji
NIM : 0712045
Perbaikan meliputi :

- hal - 21 print ulang
- abstrak ganti
- flow chart Hg pembuatan germa. → sesuai
buku
- pembuatan Hg magang. → disesuaikan buku

Matang,

Juwana



**BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI**

NAMA : Abdi K. Radja
NIM : 07.12.645
PROGRAM STUDI : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika S-1
MASA BIMBINGAN : Semester Genap Tahun Akademik 2011 - 2012
JUDUL : RANCANG BANGUN GAME PUSH BOX
MENGUNAKAN ECLIPSE BERBASIS
ANDROID

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

Hari : Rabu
Tanggal : 8 Agustus 2012
Dengan Nilai : 81,5 (A) γ

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji,

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

Sekretaris Majelis Penguji,

Dr. Eng. Aryanto S, ST, MT
NIP. Y. 1030800417

ANGGOTA PENGUJI

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT
NIP. P. 1030000365

Dosen Penguji II

Ahmad Faisol, ST
NIP. P. 1031000431



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 NAGA MALANG

Kampus I : J. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
 Kampus II : J. Raya Karanglo, Km 2 Telp. (0341) 417536 Fax. (0341) 417534 Malang

FORMULIR PERBAIKAN SKRIPSI

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

NAMA : ABDI K. RADJA
 NIM : 07.12.645
 JURUSAN : Teknik Elektro S-1
 KONSENTRASI : Teknik Komputer dan Informatika
 MASA BIMBINGAN : Semester Genap Tahun Akademik 2011-2012
 JUDUL : RANCANG BANGUN GAME PUSH BOX MENGGUNAKAN ECLIPSE BERBASIS ANDROID

Tanggal	Uraian	Paraf
Penguji I 08-08-2012	<ul style="list-style-type: none"> Hal 21 print ulang Abstrak ganti Flowchart tentang pembuatan game (score/level) Pembuatan tentang mapping (dimasukan di bab III) 	
Penguji II 08-08-2012	<ul style="list-style-type: none"> Metode pembuatan level dimasukan di bab III Perbaiki flowchart Tambah pengujian di perangkat lain dan masukan hasilnya ke tabel. Perbaiki kesimpulan dan saran no. 2 	

Disetujui,

Dosen Penguji I

Irmalia Suryani Faradisa, ST, MT
 NIP.P.1030000365

Dosen Penguji II

Ahmad Faisal, ST
 NIP.P.1031000431

Mengetahui,

Dosen Pembimbing I

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

Dosen Pembimbing II

Sandy Natali Mantia, SKom
 NIP.P.1030800418

Kuesioner Aplikasi Game Push Box Android Mobile

Nama : JANUARLO QUZFI

Umur : 22

Berilah penilaian tentang program aplikasi *Aplikasi Game Push Box Android Mobile* dengan memberikan tanda silang (X) pada kotak nilai yang sudah disediakan.

No	Pertanyaan	Nilai			
		1	2	3	4
1	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini mudah dimainkan? 1.Sangat Mudah 2. Mudah 3 Cukup Mudah 4. Tidak Mudah	X			
2	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menarik? 1. Sangat Menarik 2. Menarik 3. Cukup Menarik 4. Tidak Menarik			X	
3	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menghibur ? 1.Sangat menghibur 2. menghibur 3.Cukup menghibur 4 Tidak menghibur	X			
4	Bagaimana tampilan grafik <i>Aplikasi Game Push Box Android Mobile</i> secara keseluruhan ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik	X			
5	Bagaimana kualitas suara <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat jernih 2. jernih 3. Cukup jernih 4. Tidak jernih			X	
6	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini berjalan dengan baik (tanpa error)? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik	X			
7	Bagaimana Penilaian secara keseluruhan anda tentang <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik			X	

Atas kesediaan waktu anda untuk mencoba dan menilai aplikasinya, saya mengucapkan terima kasih.

Kuesioner Aplikasi Game Push Box Android Mobile

Nama : RENNY ELGA DEBA

Umur : 27

Berilah penilaian tentang program aplikasi *Aplikasi Game Push Box Android Mobile* dengan memberikan tanda silang (X) pada kotak nilai yang sudah disediakan.

No	Pertanyaan	Nilai			
		1	2	3	4
1	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini mudah dimainkan? 1.Sangat Mudah 2. Mudah 3 Cukup Mudah 4. Tidak Mudah	X			
2	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menarik? 1. Sangat Menarik 2. Menarik 3. Cukup Menarik 4. Tidak Menarik		X		
3	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menghibur ? 1.Sangat menghibur 2. menghibur 3.Cukup menghibur 4 Tidak menghibur	X			
4	Bagaimana tampilan grafik <i>Aplikasi Game Push Box Android Mobile</i> secara keseluruhan ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik	X			
5	Bagaimana kualitas suara <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat jernih 2. jernih 3. Cukup jernih 4. Tidak jernih		X		
6	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini berjalan dengan baik (tanpa error)? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik		X		
7	Bagaimana Penilaian secara keseluruhan anda tentang <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik				X

Atas kesediaan waktu anda untuk mencoba dan menilai aplikasinya, saya mengucapkan terima kasih.

Kuesioner Aplikasi Game Push Box Android Mobile

Nama : Ichak Samuel Sada

Umur : 26

Berilah penilaian tentang program aplikasi *Aplikasi Game Push Box Android Mobile* dengan memberikan tanda silang (x) pada kotak nilai yang sudah disediakan.

No	Pertanyaan	Nilai			
		1	2	3	4
1	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini mudah dimainkan? 1.Sangat Mudah 2. Mudah 3 Cukup Mudah 4. Tidak Mudah			X	
2	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menarik? 1. Sangat Menarik 2. Menarik 3. Cukup Menarik 4. Tidak Menarik		X		
3	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menghibur ? 1.Sangat menghibur 2. menghibur 3.Cukup menghibur 4 Tidak menghibur			X	
4	Bagaimana tampilan grafik <i>Aplikasi Game Push Box Android Mobile</i> secara keseluruhan ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik		X		
5	Bagaimana kualitas suara <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat jernih 2. jernih 3. Cukup jernih 4. Tidak jernih	X			
6	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini berjalan dengan baik (tanpa error)? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik	X			
7	Bagaimana Penilaian secara keseluruhan anda tentang <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik		X		

Atas kesediaan waktu anda untuk mencoba dan menilai aplikasinya, saya mengucapkan terima kasih.

Time

4

Game Push Box
Game Push Box

Kuesioner Aplikasi Game Push Box Android Mobile

Nama : ASWIN ANTHON J K

Umur : 25 TAHUN

Berilah penilaian tentang program aplikasi *Aplikasi Game Push Box Android Mobile* dengan memberikan tanda silang (x) pada kotak nilai yang sudah disediakan.

No	Pertanyaan	Nilai			
		1	2	3	4
1	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini mudah dimainkan? 1.Sangat Mudah 2. Mudah 3 Cukup Mudah 4. Tidak Mudah		X		
2	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menarik? 1. Sangat Menarik 2. Menarik 3. Cukup Menarik 4. Tidak Menarik		X		
3	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menghibur ? 1.Sangat menghibur 2. menghibur 3.Cukup menghibur 4 Tidak menghibur		X		
4	Bagaimana tampilan grafik <i>Aplikasi Game Push Box Android Mobile</i> secara keseluruhan ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik		X		
5	Bagaimana kualitas suara <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat jernih 2. jernih 3. Cukup jernih 4. Tidak jernih		X		
6	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini berjalan dengan baik (tanpa error)? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik		X		
7	Bagaimana Penilaian secara keseluruhan anda tentang <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik	X			

Atas kesediaan waktu anda untuk mencoba dan menilai aplikasinya, saya mengucapkan terima kasih.

Kuesioner Aplikasi Game Push Box Android Mobile

Nama : Sylvia

Umur : 24 thn

Berilah penilaian tentang program aplikasi *Aplikasi Game Push Box Android Mobile* dengan memberikan tanda silang (x) pada kotak nilai yang sudah disediakan.

No	Pertanyaan	Nilai			
		1	2	3	4
1	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini mudah dimainkan? 1.Sangat Mudah 2. Mudah 3 Cukup Mudah 4. Tidak Mudah		x		
2	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menarik? 1. Sangat Menarik 2. Menarik 3. Cukup Menarik 4. Tidak Menarik		x		
3	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menghibur ? 1.Sangat menghibur 2. menghibur 3.Cukup menghibur 4 Tidak menghibur		x		
4	Bagaimana tampilan grafik <i>Aplikasi Game Push Box Android Mobile</i> secara keseluruhan ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik			x	
5	Bagaimana kualitas suara <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat jernih 2. jernih 3. Cukup jernih 4. Tidak jernih		x		
6	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini berjalan dengan baik (tanpa error)? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik		x		
7	Bagaimana Penilaian secara keseluruhan anda tentang <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik		x		

Atas kesediaan waktu anda untuk mencoba dan menilai aplikasinya, saya mengucapkan terima kasih.

Kuesioner Aplikasi Game Push Box Android Mobile

Nama : PAULLO XIMENES MAGNO

Umur : 22

Berilah penilaian tentang program aplikasi *Aplikasi Game Push Box Android Mobile* dengan memberikan tanda silang (x) pada kotak nilai yang sudah disediakan.

No	Pertanyaan	Nilai			
		1	2	3	4
1	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini mudah dimainkan? 1.Sangat Mudah 2. Mudah 3 Cukup Mudah 4. Tidak Mudah	X			
2	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menarik? 1. Sangat Menarik 2. Menarik 3. Cukup Menarik 4. Tidak Menarik	X			
3	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini menghibur ? 1.Sangat menghibur 2. menghibur 3.Cukup menghibur 4 Tidak menghibur	X			
4	Bagaimana tampilan grafik <i>Aplikasi Game Push Box Android Mobile</i> secara keseluruhan ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik		X		
5	Bagaimana kualitas suara <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat jernih 2. jernih 3. Cukup jernih 4. Tidak jernih		X		
6	Apakah program <i>Aplikasi Game Push Box Android Mobile</i> ini berjalan dengan baik (tanpa error)? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik		X		
7	Bagaimana Penilaian secara keseluruhan anda tentang <i>Aplikasi Game Push Box Android Mobile</i> ? 1. Sangat Baik 2. Baik 3. Cukup Baik 4. Tidak Baik	X			

Atas kesediaan waktu anda untuk mencoba dan menilai aplikasinya, saya mengucapkan terima kasih.



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 Karangjo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 Juni 2011

Nomor : ITN-268/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI
Kepada : Yth. Sdr./i . **AHMAD FAISOL, ST**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama : NANDA PRIMA. A
Nim : 0712638
Fakultas : Teknologi Industri
Jurusan : Teknik Elektro S-1
Konsentrasi : Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya
kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai
tanggal :

24 Mei 2011 s/d 24 November 2011

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima
kasih



Ketua Jurusan
Teknik Elektro S-1

(Signature)
Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018200189

Tembusan Kepada Yth :

1. Mahasiswa Yang Bersangkutan
2. Arsip
3. Corat yang tidak perlu

Form. S 4a