

**PENERAPAN ALGORITMA *BACKTRACKING* DAN *MULTIPLICATIVE*
CRNG UNTUK MEMBANGKITKAN DAN MENYELESAIKAN
PERMAINAN SUDOKU HIJAIYAH**

SKRIPSI

Oleh:

MISBAKHUL MUSTOFIN

NIM: 09650218



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK IBRAHIM
MALANG
2013**

HALAMAN PENGAJUAN
PENERAPAN ALGORITMA *BACKTRACKING* DAN *MULTIPLICATIVE*
***CRNG* UNTUK MEMBANGKITKAN DAN MENYELESAIKAN**
PERMAINAN SUDOKU HIJAIYAH

SKRIPSI

Diajukan kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
MISBAKHUL MUSTOFIN
NIM: 09650218

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK IBRAHIM
MALANG
2013

HALAMAN PERSETUJUAN

PENERAPAN ALGORITMA *BACKTRACKING* DAN *MULTIPLICATIVE*
***CRNG* UNTUK MEMBANGKITKAN DAN MENYELESAIKAN**
PERMAINAN SUDOKU HIJAIYAH

SKRIPSI

Oleh :

Nama : Misbakhul Mustofin
NIM : 09650218
Jurusan : Teknik Informatika
Fakultas : Sains Dan Teknologi

Telah Disetujui, 12 Januari 2013

Dosen Pembimbing I

Dosen Pembimbing II

Hani Nurhayati, M.T
NIP. 19780625 200801 2 006

Fatchurrochman, M.Kom
NIP. 19700731 200501 1 002

Mengetahui,
Ketua Jurusan Teknik Informatika

Ririen Kusumawati, S.Si, M.Kom
NIP. 197203092005012002

HALAMAN PENGESAHAN

PENERAPAN ALGORITMA *BACKTRACKING* DAN *MULTIPLICATIVE* *CRNG* UNTUK MEMBANGKITKAN DAN MENYELESAIKAN PERMAINAN SUDOKU HIJAIYAH

SKRIPSI

Oleh :

Misbakhul Mustofin
NIM. 09650218

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal 18 Januari 2013

Susunan Dewan Penguji:

Tanda Tangan

- | | | | |
|--------------------|--------------------------------|---|---|
| 1. Penguji Utama | : <u>M. Faisal, M.T</u> | (|) |
| | NIP. 19740510 200501 1 007 | | |
| 2. Ketua Penguji | : <u>Totok Chamidy, M.Kom</u> | (|) |
| | NIP. 19691222 200604 1 001 | | |
| 3. Sekretaris | : <u>Hani Nurhayati, M.T</u> | (|) |
| | NIP. 19780625 200801 2 006 | | |
| 4. Anggota Penguji | : <u>Fatchurrochman, M.Kom</u> | (|) |
| | NIP. 19700731 200501 1 002 | | |

Mengetahui,
Ketua Jurusan Teknik Informatika

Ririen Kusumawati, S.Si, M.Kom
NIP. 197203092005012002

HALAMAN PERNYATAAN ORISINALITAS PENELITIAN

Saya yang bertandatangan di bawah ini:

Nama : Misbakhul Mustofin
NIM : 09650218
Fakultas/Jurusan : Sains Dan Teknologi / Teknik Informatika
Judul Penelitian : Penerapan Algoritma *Backtracking* dan
Multiplicative CRNG untuk Membangkitkan dan
Menyelesaikan Permainan Sudoku Hijaiyah

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut

Malang, 12 Januari 2013
Yang Membuat Pernyataan,

Misbakhul Mustofin
09650218

HALAMAN MOTTO

إِنَّ اللَّهَ لَا يُغَيِّرُ مَا بِقَوْمٍ حَتَّى يُغَيِّرُوا مَا
بِأَنْفُسِهِمْ

*“Sesungguhnya Allah tidak akan merubah nasib satu
kaum sehingga mereka sendiri merubahnya.”*

[QS. ar-Ra'd (13) : 11]

**“Yang Terpenting Bukanlah Seberapa Berat
Masalah, Tetapi Bagaimana Menyikapi Masalah Itu
Sendiri.”**

HALAMAN PERSEMBAHAN

Dengan rasa syukur seraya mengharap ridho Ilahi

Kupersembahkan karya ini kepada :

Ayahanda dan Ibunda tercinta

Imam Sukhairi dan Jamilah

Atas Segalanya.

Semoga Allah SWT melindungi

Dan menyayangi keduanya . . .

KATA PENGANTAR



Segala puji bagi Allah SWT yang telah melimpahkan rahmat serta karuniaNya kepada penulis sehingga bisa menyelesaikan skripsi dengan judul “Penerapan Algoritma *Backtracking* dan *Multiplicative CRNG* untuk Membangkitkan dan Menyelesaikan Permainan *Sudoku Hijaiyah*” dengan baik.

Shalawat serta salam semoga tercurah kepada Nabi Agung Muhammad SAW yang telah membimbing umatnya dari gelapnya kekufuran menuju cahaya Islam yang terang benderang.

Penulis menyadari keterbatasan pengetahuan yang penulis miliki, karena itu tanpa keterlibatan dan sumbangsih dari berbagai pihak, sulit bagi penulis untuk menyelesaikan skripsi ini. Maka dari itu dengan segenap kerendahan hati patutlah penulis ucapkan terima kasih kepada:

1. Hani Nurhayati, M.T, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan dalam pengerjaan skripsi ini.
2. Fatchurrohman, M.Kom, selaku dosen pembimbing II, yang selalu memberikan masukan, nasehat serta petunjuk dalam penyusunan laporan skripsi ini.
3. Ririen Kusumawati, M.Kom, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang mendukung dan mengarahkan dalam pengerjaan skripsi ini.

4. Segenap Dosen Teknik informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
5. Semua pihak yang tidak mungkin penulis sebutkan satu-persatu, atas segala yang telah diberikan kepada penulis dan dapat menjadi pelajaran.

Sebagai penutup, penulis menyadari dalam skripsi ini masih banyak kekurangan dan jauh dari sempurna. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya. Apa yang menjadi harapan penulis, semoga karya ini bermanfaat bagi kita semua. Amin.

Malang, 12 Januari 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	Error! Bookmark not defined.
HALAMAN PENGAJUAN.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN	v
HALAMAN MOTTO	vi
HALAMAN PERSEMBAHAN.....	vii
KATA PENGANTAR	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
ABSTRAK	xvi
BAB I PENDAHULUAN.....	Error! Bookmark not defined.
1.1 Latar Belakang.....	Error! Bookmark not defined.
1.2 Rumusan Masalah.....	Error! Bookmark not defined.
1.3 Batasan Masalah	Error! Bookmark not defined.
1.4 Tujuan Penelitian	Error! Bookmark not defined.
1.5 Manfaat Penelitian	Error! Bookmark not defined.
1.6 Sistematika Penulisan	Error! Bookmark not defined.
1.7 Metode Penelitian	Error! Bookmark not defined.
BAB II TINJAUAN PUSTAKA.....	Error! Bookmark not defined.

2.1	Huruf Hijaiyah	Error! Bookmark not defined.
2.2	Permainan Sudoku	Error! Bookmark not defined.
2.3	<i>Backtracking</i>	Error! Bookmark not defined.
2.4	<i>Multiplicative CRNG (Congruential Random Number Generator)</i> Error! Bookmark not defined.	
2.5	<i>Platform Android</i>	Error! Bookmark not defined.
BAB III ANALISA DAN PERANCANGAN		Error! Bookmark not defined.
3.1	Analisa dan Perancangan Sistem	Error! Bookmark not defined.
3.1.1	Keterangan Umum	Error! Bookmark not defined.
3.1.2	Rancangan Level Permainan	Error! Bookmark not defined.
3.2	Rancangan Algoritma <i>Backtracking</i> dan <i>Multiplicative CRNG</i> Error! Bookmark not defined.	
3.2.1	Pembangkitan <i>Puzzle</i>	Error! Bookmark not defined.
3.2.2	Penyelesaian <i>Puzzle</i>	Error! Bookmark not defined.
3.3	Perancangan Aplikasi	Error! Bookmark not defined.
3.3.1	Antarmuka Permainan	Error! Bookmark not defined.
3.3.2	Kebutuhan Sistem	Error! Bookmark not defined.
BAB IV HASIL DAN PEMBAHASAN		Error! Bookmark not defined.
4.1	Implementasi <i>Backtracking</i> dan <i>Multiplicative CRNG</i>	Error! Bookmark not defined.
4.1.1	<i>Multiplicative CRNG</i>	Error! Bookmark not defined.
4.1.2	<i>Backtracking</i>	Error! Bookmark not defined.
4.1.3	Pembangkitan <i>Puzzle</i>	Error! Bookmark not defined.

4.1.4	Penyelesaian <i>Puzzle</i>	Error! Bookmark not defined.
4.2	Impelementasi Aplikasi	Error! Bookmark not defined.
4.3	Uji Coba.....	Error! Bookmark not defined.
4.3.1	Perbandingan antara Algoritma <i>Backtracking</i> dan <i>Multiplicative CRNG</i> dengan Algoritma <i>Harmony Search</i> dalam Menyelesaikan <i>Puzzle</i> Sudoku Hijaiyah	Error! Bookmark not defined.
4.3.2	Uji Coba Pengguna	Error! Bookmark not defined.
4.4	Integrasi <i>Game</i> Sudoku Hijaiyah dengan Islam	Error! Bookmark not defined.
BAB V PENUTUP.....		Error! Bookmark not defined.
5.1	Kesimpulan	Error! Bookmark not defined.
5.2	Saran	Error! Bookmark not defined.
DAFTAR PUSTAKA		Error! Bookmark not defined.
LAMPIRAN.....		Error! Bookmark not defined.

DAFTAR GAMBAR

Gambar 1.1 Pengguna Smartphone Indonesia	Error! Bookmark not defined.
Gambar 2.1 Daftar Hijaiyah	Error! Bookmark not defined.
Gambar 2.2 Teka-Teki Sudoku dan Solusi Penyelesaiannya	Error! Bookmark not defined.
Gambar 2.3 Arsitektur Android	Error! Bookmark not defined.
Gambar 3.1 Flowchart Pembangkitan Puzzle	Error! Bookmark not defined.
Gambar 3.2 Flowchart Penyelesaian Puzzle	Error! Bookmark not defined.
Gambar 3.3 Rancangan Halaman Utama	Error! Bookmark not defined.
Gambar 3.4 Rancangan Halaman Pilih Level	Error! Bookmark not defined.
Gambar 3.5 Rancangan Halaman Pilih Huruf.....	Error! Bookmark not defined.
Gambar 3.6 Rancangan Halaman Puzzle	Error! Bookmark not defined.
Gambar 3.7 Rancangan Menu Puzzle	Error! Bookmark not defined.
Gambar 3.8 Flowchart Permainan Sudoku Hijaiyah.....	Error! Bookmark not defined.
Gambar 3.9 Class Diagram Sudoku Hijaiyah	Error! Bookmark not defined.
Gambar 4.1 Halaman Utama.....	Error! Bookmark not defined.
Gambar 4.2 Halaman Bantuan	Error! Bookmark not defined.
Gambar 4.3 Halaman Pilih Level.....	Error! Bookmark not defined.
Gambar 4.4 Halaman Pilih Huruf	Error! Bookmark not defined.
Gambar 4.5 Halaman Puzzle	Error! Bookmark not defined.
Gambar 4.6 Input Hijaiyah.....	Error! Bookmark not defined.
Gambar 4.7 Menu Puzzle	Error! Bookmark not defined.
Gambar 4.8 Hasil Penyelesaian.....	Error! Bookmark not defined.
Gambar 4.9 Aplikasi AlBaTa Sudoku dan Sudoku Hijaiyah pada Perangkat Keras	Error! Bookmark not defined.

Gambar 4.10 Pengujian Pertama Puzzle Sudoku Hijaiyah Beserta Solusinya**Error! Bookmark not defined**

Gambar 4.11 Pengujian Pertama Puzzle Sudoku AlBaTa Beserta Solusinya**Error! Bookmark not defined**

Gambar 4.12 Pengujian Ke Dua Puzzle Sudoku Hijaiyah Beserta Solusinya**Error! Bookmark not defined**

Gambar 4.13 Pengujian Ke Dua Puzzle Sudoku AlBaTa Beserta Solusinya**Error! Bookmark not defined**

Gambar 4.14 Pengujian Ke Tiga Puzzle Sudoku Hijaiyah Beserta Solusinya**Error! Bookmark not defined**

Gambar 4.15 Pengujian Ke Tiga Puzzle Sudoku AlBaTa Beserta Solusinya**Error! Bookmark not defined**



DAFTAR TABEL

Tabel 3.1 Jumlah huruf yang harus diisi setiap level **Error! Bookmark not defined.**

Tabel 3.2 Pembagian huruf hijaiyah **Error! Bookmark not defined.**

Tabel 4.1 Percobaan Pertama pada Sudoku Hijaiyah dan Sudoku AlBaTa... **Error! Bookmark not defined.**

Tabel 4.2 Percobaan Ke Dua pada Sudoku Hijaiyah dan Sudoku AlBaTa ... **Error! Bookmark not defined.**

Tabel 4.3 Percobaan Ke Tiga pada Sudoku Hijaiyah dan Sudoku AlBaTa... **Error! Bookmark not defined.**

Tabel 4.4 Rekapitulasi Hasil Uji Coba..... **Error! Bookmark not defined.**

Tabel 4.5 Detail Kebosanan Pengguna **Error! Bookmark not defined.**

Tabel 4.6 Detail Kemenarikan Game Sudoku Hijaiyah **Error! Bookmark not defined.**

Tabel 4.7 Detail Keinginan Bermain Lagi Game Sudoku Hijaiyah..... **Error! Bookmark not defined.**

Tabel 4.8 Rata-Rata Prosentase Tiap Kelas (dalam %) **Error! Bookmark not defined.**

ABSTRAK

Mustofin, Misbakhul. 2013. **Penerapan Algoritma *Backtracking* dan *Multiplicative CRNG* untuk Membangkitkan dan Menyelesaikan Permainan Sudoku Hijaiyah**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Hani Nurhayati, M.T (II) Fatchurrohman, M.Kom

Kata Kunci: *Sudoku, Huruf Hijaiyah, Backtracking, Multiplicative CRNG, Pembangkit, Penyelesai*

Sudoku merupakan salah satu permainan teka-teki angka. Secara umum permainan ini berbentuk table berukuran 9x9 yang di dalamnya terdapat sembilan blok/kelompok berukuran 3x3. Tujuan dari permainan ini adalah mengisi setiap kotak (sel tabel) yang kosong dengan angka yang terdiri dari 1-9 sedemikian hingga tidak terdapat pengulangan angka dalam satu blok, satu baris maupun satu kolom. Salah satu pengembangan dari *sudoku* adalah dalam bentuk huruf hijaiyah menggunakan algoritma *Harmony Search* sebagai pembangkit dan penyelesai permainan *Sudoku* yang diteliti oleh Riyadli Abrar pada tahun 2012. Namun dalam penelitian tersebut penyelesaian *puzzle sudoku* memerlukan waktu cukup lama dan dalam beberapa parameter algoritma tersebut tidak dapat memecahkan *puzzle sudoku*. Dalam penelitian ini, algoritma *Backtracking* yang merupakan pengembangan dari algoritma DFS dikombinasikan dengan algoritma pengacakan *Multiplicative CRNG* untuk menggantikan algoritma *Harmony Search* sebagai pembangkit dan penyelesai permainan *sudoku* Hijaiyah untuk menghasilkan proses yang lebih efektif dan efisien. Pengujian dilakukan pada perangkat *mobile* yang menggunakan *platform* Android.

ABSTRACT

Mustofin, Misbakhul. 2013. **Implementation of Backtracking and Multiplicative CRNG Algorithms for Generating and Solving Arabic Sudoku Game**. Thesis. Informatics Department of Faculty of Science and Technology. Maulana Malik Ibrahim State Islamic University, Malang.

Adviser: (I) Hani Nurhayati, M.T (II) Fatchurrohman, M.Kom

Keywords: *Sudoku, Arabic Letters, Backtracking, Multiplicative CRNG, Generator, Solver*

Sudoku is a number puzzle game. In general, the game is shaped 9x9 sized table in which there are nine blocks / group size of 3x3. The purpose of the game is to fill each square (table cells) are empty with numbers consisting of 1-9 so that there is no repetition of numbers in one block, one row and one column. One of the development is in the form of Arabic letters sudoku using Harmony Search algorithm for generating and solving are researched by Abrar Riyadli in 2012. However, in these studies requires the completion of a sudoku puzzle a long time and in some parameters of the algorithm can not solve the sudoku puzzle. In this study, a backtracking algorithm is an enhancement of the DFS algorithm combined with a randomization algorithm Multiplicative CRNG to replace the Harmony Search algorithm as a generator and solver a Arabic letters sudoku game to produce a more effective and efficient. Tests performed on mobile devices using the Android platform.



BAB I

PENDAHULUAN

1.1 Latar Belakang

Al-Qur'an adalah firman Allah yang diwahyukan kepada Nabi Muhammad SAW yang menggunakan lafazh Arab. Hal ini diperjelas dengan firman Allah SWT dalam Surah Yusuf yang berbunyi :

إِنَّا أَنْزَلْنَاهُ قُرْآنًا عَرَبِيًّا لَعَلَّكُمْ تَعْقِلُونَ (يوسف : ٢)

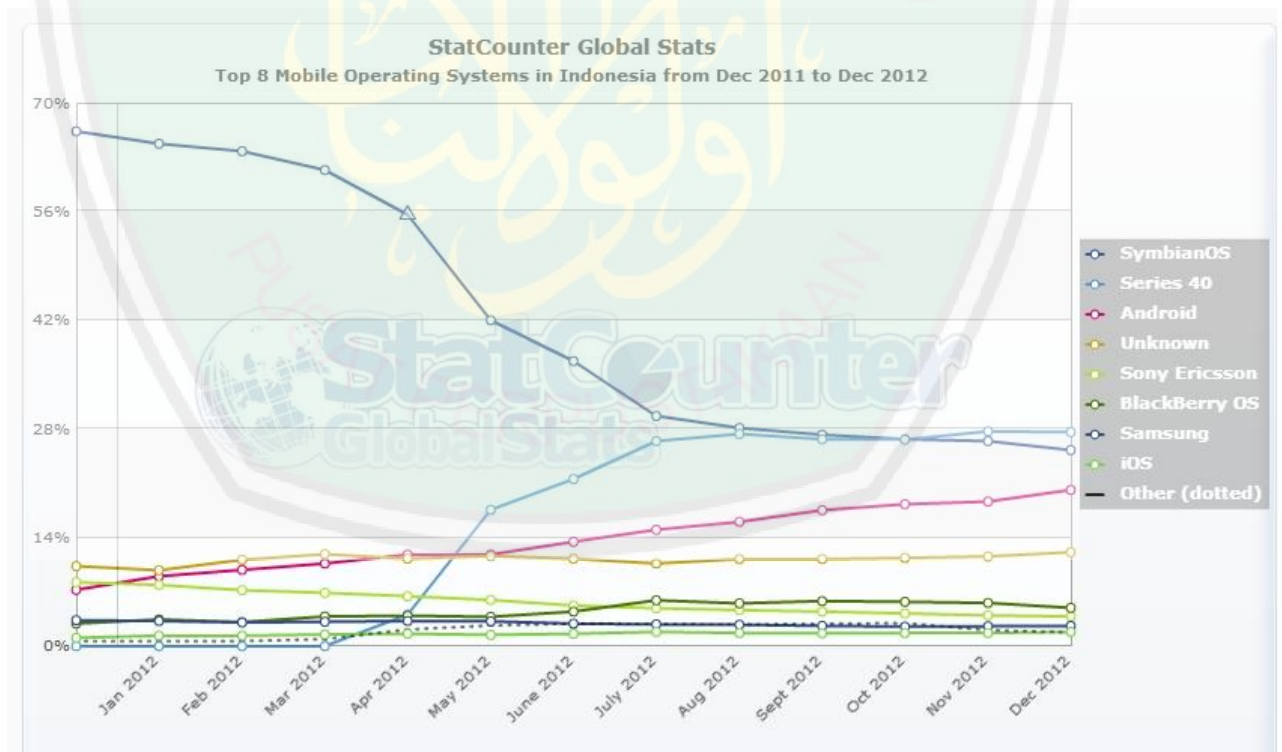
Artinya:

“Sesungguhnya Kami menurunkannya berupa Al-Qur'an dengan berbahasa Arab, agar kamu memahaminya.” (QS Yusuf : 2) (Al-Jumanatul 'Ali, 2005:236).

Al-Qur'an yang diturunkan menggunakan bahasa arab mengharuskan kita untuk mempelajari bahasa arab, karena dengan mengetahui bahasa arab maka kita juga dapat memahami isi dari Al-Qur'an. Dalam bahasa arab dikenal huruf hijaiyah. Huruf hijaiyah merupakan huruf yang digunakan bahasa arab. Terdapat berbagai macam cara pembelajaran, salah satunya yaitu melalui permainan.

Berdasarkan berita pada situs teknojurnal.com yang ditulis oleh Firman Nugraha tanggal 18 Januari 2012, pertumbuhan pelanggan seluler bertolak belakang dengan pertumbuhan pengguna komputer *desktop*. Pada berita tersebut juga dituliskan *“Data terbaru dari Asosiasi Telekomunikasi Seluler Indonesia (ATSI) menunjukkan bahwa jumlah pelanggan seluler di Indonesia per tahun*

2011 telah mencapai lebih dari 240 juta pelanggan pada akhir tahun 2011 lalu, naik 60 juta pelanggan dibanding tahun 2010. Angka ini mendekati jumlah penduduk Indonesia yang berjumlah 258 juta penduduk pada Desember 2010. Perkembangan jumlah pelanggan seluler di Indonesia bisa dibilang cukup fantastis”. Oleh karena itu permainan saat ini juga sudah mulai beralih ke arah perangkat *mobile*, yang memungkinkan pemain untuk memainkan permainan dimana saja dan kapan saja. Salah satu sistem operasi perangkat *mobile* yang sedang berkembang saat ini adalah sistem operasi Android. Perkembangan sistem operasi android yang cukup baik di Indonesia pada periode 2011-2012 ditunjukkan oleh grafik yang berasal dari GlobalStat Statcounter di bawah ini:



Gambar 1.1 Pengguna Smartphone Indonesia (Sumber: gs.statcounter.com, 2012)

Sistem operasi Android bersifat terbuka sehingga memberi kesempatan bagi para pengembang untuk menciptakan aplikasi mereka sendiri, terutama

aplikasi permainan. Salah satu permainan yang dapat dikembangkan di sistem operasi Android adalah Sudoku.

Sudoku adalah sebuah permainan atau logika yang berbasis kombinasi penempatan angka. Tujuan dari permainan ini adalah mengisi kotak-kotak dengan angka sehingga setiap kolom, baris dan area (kotak kecil yang terbentuk dari susunan kotak seperti kotak 3x3 pada *Sudoku* 9x9) berisi semua antara angka 1 sampai 9 (Riyadli Abrar, 2012:11).

Dalam menyelesaikan permainan sudoku terdapat beberapa algoritma, salah satunya adalah algoritma *backtracking* (runut-balik). Algoritma *backtracking* berbasis pada DFS (Depth First Search). Pada algoritma *backtracking*, pencarian solusi lebih difokuskan pada pencarian yang mengarah ke solusi saja sehingga mempersingkat proses pencarian. Selain algoritma *backtracking* terdapat algoritma *Multiplicative CRNG* yang berfungsi sebagai pembangkit angka secara acak. Algoritma pembangkit angka secara acak ini cukup mudah dipahami dan diaplikasikan karena memiliki batasan yang bisa disesuaikan dengan kebutuhan.

Penelitian tentang sudoku menggunakan bahasa Arab telah dilakukan oleh Riyadli Abrar (Abrar, 2012). Pada penelitian tersebut, angka 1 hingga 9 pada *puzzle* sudoku diganti dengan huruf hijaiyah berjumlah 9 dari 30 huruf hijaiyah yang bisa dipilih oleh pemain sendiri atau *default* dari *game* tersebut. Pembangkitan dan penyelesaian permainan pada penelitian tersebut menggunakan algoritma *Harmony Search*. Penggunaan *Harmony Search* pada penelitian tersebut dapat menyelesaikan *puzzle* Sudoku dengan benar dalam waktu tercepat

antara 1 – 15 detik dengan jumlah kotak kosong yang terdapat dalam *puzzle* antara 35 – 40, 2 – 12 detik dengan jumlah kotak kosong yang terdapat dalam *puzzle* antara 40 – 45, dan 5 – 35 detik dengan jumlah kotak kosong yang terdapat dalam *puzzle* antara 45 – 50. Solusi tidak ditemukan pada parameter tertentu yaitu dengan parameter HMS = 10, HMCR = 0.9, dan PAR = 0.1, 0.5, setelah melakukan 1000 improvisasi dalam waktu 49 – 55 detik.

Pada penelitian lain yang membahas penyelesaian permainan sudoku menggunakan algoritma *Backtracking* dan *Elimination* oleh Agustinus Tri Gunawan (Gunawan, 2010) menghasilkan waktu penyelesaian yang lebih cepat. Pada penelitian tersebut dapat menyelesaikan permainan sudoku dalam waktu 0.1 detik dengan jumlah kotak kosong yang terdapat dalam *puzzle* berjumlah 10, 29 dan 58.

Fungsi acak pada penelitian Agustinus Tri Gunawan adalah fungsi acak bawaan dari bahasa pemrograman. Penulis mencoba membangkitkan bilangan bulat acak sebanyak 10 kali menggunakan fungsi acak bawaan dari bahasa pemrograman yang akan digunakan yaitu java dengan source code sebagai berikut:

```
int acak = (int)(Math.random()*10);
```

menghasilkan output : 9, 9, 8, 1, 5, 0, 0, 5, 0 dan 6. Penulis juga melakukan uji coba pada algoritma pengacakan *Multiplicative CRNG* untuk membangkitkan bilangan bulat acak sebanyak 10 kali dengan parameter $a=2$, $m=11$, $Z_0=7$ menghasilkan output: 3, 6, 1, 2, 4, 8, 5, 10, 9 dan 7.

Dari percobaan tersebut terlihat bahwa bilangan acak yang dibangkitkan menggunakan fungsi bawaan bahasa pemrograman masih kurang maksimal

karena beberapa kali mengulang bilangan yang sama secara berturut-turut. Pada percobaan menggunakan algoritma *Multiplicative CRNG* tidak menghasilkan bilangan yang sama secara berturut-turut. Untuk itu pada penelitian ini tidak menggunakan fungsi bilangan acak bawaan bahasa pemrograman melainkan menggunakan algoritma *Multiplicative CRNG* sebagai pembangkit bilangan acak.

Pada penelitian ini menggunakan perpaduan antara algoritma *Backtracking* dan *Multiplicative CRNG* sebagai Generator dan Solver dari permainan sudoku yang akan dibangun dengan harapan dapat menyelesaikan permainan sudoku dengan waktu lebih cepat dan 100% solusi ditemukan. Generator digunakan untuk membangkitkan *puzzle* sudoku berdasarkan level yang dipilih, sedangkan Solver digunakan untuk menyelesaikan permainan jika pemain menyerah dalam menyelesaikan permainan secara manual. Penelitian ini mencoba membangun aplikasi permainan sudoku huruf hijaiyah dengan desain lebih menarik dari aplikasi sudoku huruf hijaiyah pada penelitian Riyadli Abrar (2012) serta memberi efek suara huruf hijaiyah pada permainan agar dapat diterima oleh anak-anak sebagai sasaran utama dari penelitian ini. Penelitian ini akan mengimplementasikan aplikasi permainan sudoku huruf hijaiyah pada perangkat *mobile* yang menggunakan sistem operasi Android dengan harapan dapat diaplikasikan dimana saja dan kapan saja.

1.2 Rumusan Masalah

Bagaimana membuat aplikasi permainan Sudoku huruf hijaiyah menggunakan algoritma *Backtracking* dan *Multiplicative CRNG* sebagai pembangkit dan penyelesai permainan serta membandingkan hasilnya dengan

hasil pembangkitan dan penyelesaian permainan menggunakan algoritma *Harmony Search*?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini sebagai berikut:

- a. Aplikasi permainan ini berbasis mobile.
- b. Permainan ini dimainkan single player.
- c. Huruf hijaiyah digunakan untuk menggantikan angka.
- d. Huruf hijaiyah yang digunakan berjumlah 30 huruf.
- e. Dimensi kotak *game* Sudoku adalah 9 x 9.
- f. *Backtracking* dan *Multiplicative CRNG* digunakan sebagai pembangkit dan penyelesai permainan Sudoku.
- g. Permainan Sudoku diimplementasikan pada Android OS mobile.
- h. Huruf yang diinputkan dengan benar oleh user pada permainan Sudoku huruf hijaiyah akan mengeluarkan suara sesuai huruf yang diinput.

1.4 Tujuan Penelitian

Membuat aplikasi permainan Sudoku huruf hijaiyah menggunakan algoritma *Backtracking* dan *Multiplicative CRNG* sebagai pembangkit dan penyelesai permainan serta membandingkan hasilnya dengan hasil pembangkitan dan penyelesaian permainan menggunakan algoritma *Harmony Search*.

1.5 Manfaat Penelitian

Manfaat pembuatan aplikasi permainan ini adalah mengenalkan huruf hijaiyah melalui permainan Sudoku huruf hijaiyah sebagai salah satu media

pembelajaran bahasa Arab sekaligus mengasah logika sehingga meningkatkan kemampuan otak kiri.

1.6 Sistematika Penulisan

Penulisan skripsi ini tersusun dalam lima bab dengan sistematika penulisan sebagai berikut :

BAB I Pendahuluan

Pendahuluan, membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penyusunan tugas akhir, metodologi, dan sistematika penyusunan tugas akhir.

BAB II Landasan Teori

Landasan teori berisikan beberapa teori yang mendasari dalam penyusunan tugas akhir ini. Adapun yang dibahas dalam bab ini adalah dasar teori yang berkaitan dengan pembahasan tentang Huruf Hijaiyah, Permainan Sudoku, *Backtracking*, *Multiplicative CRNG (Congruential Random Number Generator)*, dan Platform Android.

BAB III Analisa dan Perancangan

Menganalisa kebutuhan sistem untuk membuat *game* meliputi spesifikasi kebutuhan software dan langkah-langkah pembuatan Sudoku Hijaiyah.

BAB IV Hasil dan Pembahasan

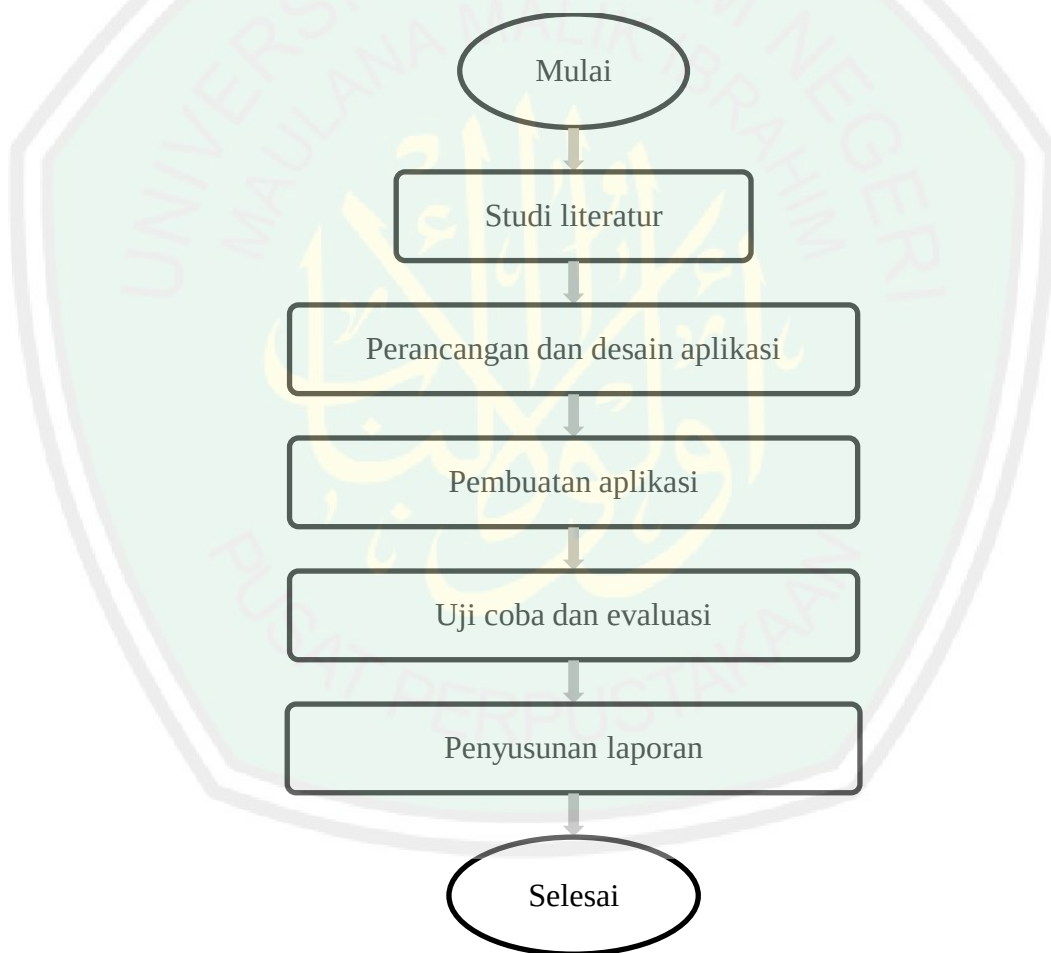
Menjelaskan tentang pengujian Sudoku Hijaiyah yang telah diterapkan dalam pembuatan *game*.

BAB V Penutup

Berisi kesimpulan dan saran.

1.7 Metode Penelitian

Berdasarkan jenis dan analisis data, penulisan skripsi ini menggunakan metode penelitian kualitatif dengan jenis penelitian studi dokumen dan bentuk analisis berupa analisis isi (content analysis). Pengumpulan data juga dilakukan dengan cara melakukan uji coba aplikasi pada pengguna yang disertai dengan angket. Berikut ini tahapan penelitian:



BAB II

TINJAUAN PUSTAKA

2.1 Huruf Hijaiyah

Huruf hijaiyah atau juga disebut Abjad Arab adalah huruf yang digunakan dalam bahasa Arab. Abjad Arab berasal dari aksara Aramaik (dari bahasa Syria dan Nabatea), dimana abjad Arab terlihat kemiripannya dengan abjad Koptik dan Yunani (Riyadli Abrar, 2012:8).

Bahasa Arab yang merupakan salah satu bahasa yang banyak digunakan di Timur Tengah. Bahasa Arab juga merupakan bahasa yang digunakan Al-Qur'an. Hal ini diperjelas dalam firman Allah SWT sebagai berikut:

وَكَذَٰلِكَ أَنْزَلْنَاهُ حُكْمًا وَعَرَبِيًّا وَلَئِنْ أَتْبَعْتَ أَهْوَاءَهُمْ بَعْدَ مَا جَاءَكَ مِنَ الْعِلْمِ مَا لَكَ مِنَ اللَّهِ مِنْ وَلِيٍّ وَلَا وَاقٍ (الرعد : ٣٧)

Artinya:

“Dan demikianlah, Kami telah menurunkan Al Quran itu sebagai peraturan (yang benar) dalam bahasa Arab. Dan seandainya kamu mengikuti hawa nafsu mereka setelah datang pengetahuan kepadamu, maka sekali-kali tidak ada pelindung dan pemelihara bagimu terhadap (siksa) Allah.” (Terjemah QS Ar Ra’d:37) (Al-Jumanatul ‘Ali, 2005:255)

Penggunaan bahasa Arab dalam Al-Qur'an karena keistimewaan bahasa Arab. Diantara keistimewaan bahasa Arab adalah sejak dahulu kala hingga

sekarang bahasa Arab itu merupakan bahasa yang hidup. Bahasa Arab adalah bahasa yang lengkap dan luas untuk menjelaskan tentang ketuhanan dan keakhiratan. Dan bentuk-bentuk kata dalam bahasa Arab mempunyai tasrif (konjugasi) yang amat luas sehingga dapat mencapai 3000 bentuk perubahan (Al-Jumanatul ‘Ali, 2005:255-256).

Terdapat perbedaan penulisan huruf-huruf hijaiyah antar Maghribi dan Timur Tengah. Di antaranya adalah penulisan huruf qaf dan fa. Di Maghribi, huruf qaf dan fa dituliskan dengan memiliki titik di bawah dan satu titik di atasnya (Riyadli Abrar, 2012:9).

Berikut ini daftar hijaiyah yang bersumber dari buku karya Muhammad Izzuddin (Izzudin, 2009 : 44):

No	Huruf (Cara Baca)	No	Huruf (Cara Baca)	No	Huruf (Cara Baca)
1	ا (الْأَيْفُ)	11	ز (الْزَايُ)	21	ق (الْقَافُ)
2	ب (الْبَاءُ)	12	س (السَّيْنُ)	22	ك (الْكَافُ)
3	ت (التَّاءُ)	13	ش (الشَّيْنُ)	23	ل (الْلَامُ)
4	ث (الثَّاءُ)	14	ص (الصَّادُ)	24	م (الْمِيمُ)
5	ج (الْجِيمُ)	15	ض (الضَّادُ)	25	ن (النُّونُ)
6	ح (الْحَاءُ)	16	ط (الطَّاءُ)	26	و (الْوَاوُ)
7	خ (الْخَاءُ)	17	ظ (الظَّاءُ)	27	هـ (الْهَاءُ)
8	د (الدَّالُ)	18	ع (الْعَيْنُ)	28	لا (لَامُ الْأَيْفُ)
9	ذ (الذَّالُ)	19	غ (الْغَيْنُ)	29	ء (الْهَمْزَةُ)
10	ر (الرَّاءُ)	20	ف (الْفَاءُ)	30	ي (الْيَاءُ)

Gambar 2.1 Daftar Hijaiyah (Sumber: Izzudin, 2009 : 44)

Dalam mempelajari huruf hijaiyah, perlu juga mempelajari makhorijul huruf atau tempat keluarnya huruf. Dengan mengetahui makhroj huruf, kita dapat mengucapkan huruf hijaiyah dengan baik dan benar. Secara umum, ulama’

qira'ah membagi tempat keluarnya huruf dalam 5 tempat, yaitu: Rongga Mulut, Tenggorokan, Lidah, Dua Bibir, dan Rongga Hidung. (Izzudin, 2009 : 45).

2.2 Permainan Sudoku

“Su” berarti angka dalam bahasa Jepang, dan “Doku” mengacu pada satu tempat di papan teka-teki yang setiap angka dapat masuk ke dalamnya. Walaupun namanya adalah Jepang, akan tetapi sebenarnya berasal dari Eropa dan Amerika (Agustinus Tri Gunawan, 2010:4).

Permainan ini pertama kali diterbitkan di sebuah surat kabar Perancis pada 1895. Versi modern permainan ini dimulai di Indianapolis pada 1979. Kemudian menjadi terkenal kembali di Jepang pada tahun 1986, ketika penerbit Nikoli menemukan teka-teki ini yang diciptakan Howard Garns, seorang mantan arsitek yang meninggal pada tahun 1989. Kemudian Nikoli membawa permainan ini ke Jepang dan menerbitkannya di sebuah media cetak khusus *puzzle* miliknya “*Monthly Nikolist*”. Mereka menamakannya “*Suuji Wa Dokushin Ni Kagiru*”, disingkat *Sudoku* (artinya “angka-angkanya harus tetap tunggal”) dan mematenkan kata ini. Media lain pun kemudian menerbitkan permainan ini dengan nama aslinya, *Number Place*. Mulai saat itulah permainan ini mewabah di Jepang. Lebih dari 600.000 majalah tentang *Sudoku* terjual di Jepang setiap bulannya (Rina Dewi Indah Sari, 2007:2).

Sudoku adalah sebuah permainan atau logika yang berbasis kombinasi penempatan angka. Tujuan dari permainan ini adalah mengisi kotak-kotak dengan angka sehingga setiap kolom, baris dan area (kotak kecil yang terbentuk dari

susunan kotak seperti kotak 3x3 pada *Sudoku* 9x9) berisi semua antara angka 1 sampai 9 (Riyadli Abrar, 2012:11).

Sudoku bisa memiliki beberapa bentuk, antara lain *Sudoku* gambar, angka dan huruf. Yang paling populer diantaranya adalah *Sudoku* angka. Walaupun *Sudoku* dapat berbentuk angka, *Sudoku* tidak berhubungan dengan operasi matematika. Logikalah yang menentukan dimana angka harus diletakkan dan di sinilah letak keasyikannya. Setiap kali kita bisa menyelesaikannya, kita pasti ingin mengerjakannya terus-menerus (Riyadli Abrar, 2012:11).

5	3			7						5	3	4	6	7	8	9	1	2
6			1	9	5					6	7	2	1	9	5	3	4	8
	9	8						6		1	9	8	3	4	2	5	6	7
8				6					3	8	5	9	7	6	1	4	2	3
4			8		3				1	4	2	6	8	5	3	7	9	1
7				2					6	7	1	3	9	2	4	8	5	6
	6						2	8		9	6	1	5	3	7	2	8	4
			4	1	9				5	2	8	7	4	1	9	6	3	5
				8				7	9	3	4	5	2	8	6	1	7	9

Gambar 2.2 Teka-Teki *Sudoku* dan Solusi Penyelesaiannya (Sumber: Riyadli Abrar, 2012:12)

Selain memiliki beberapa bentuk, *Sudoku* juga memiliki beragam tipe berdasarkan banyaknya kotak *Sudoku*, mulai dari yang paling mudah yaitu *Sudoku* 2x2, *Sudoku* 4x4, *Sudoku* 6x6 hingga *Sudoku* 9x9 (Riyadli Abrar, 2012:12).

Berikut ini peraturan permainan *Sudoku*:

- 1) Setiap angka/gambar/ huruf hanya dapat muncul sekali dalam setiap baris.
- 2) Setiap angka/ gambar/ huruf hanya dapat muncul sekali dalam setiap kolom.

- 3) Setiap angka/ gambar/ huruf hanya dapat muncul sekali dalam setiap area (kotak 3x3 pada *Sudoku* 9x9).

2.3 *Backtracking*

Algoritma *backtracking* pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950 menyajikan uraian umum tentang *backtracking* dan penerapannya dalam berbagai persoalan dan aplikasi. Algoritma *backtracking* (runut balik) merupakan salah satu metode pemecahan masalah yang termasuk dalam strategi yang berbasis pencarian pada ruang status. Algoritma *backtracking* bekerja secara rekursif dan melakukan pencarian solusi persoalan secara sistematis pada semua kemungkinan solusi yang ada. Oleh karena algoritma ini berbasis pada algoritma *Depth-First Search (DFS)* untuk mencari solusi persoalan secara lebih mangkus, maka pencarian solusi dilakukan dengan menelusuri suatu struktur berbentuk pohon berakar (Rina Dewi Indah Sari, 2007:3-4).

Backtracking adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih baik. *Backtracking* yang merupakan perbaikan dari algoritma *bruteforce*, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Dengan metode ini, pencarian tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang selalu dipertimbangkan. Hal ini menyebabkan waktu pencarian dapat dihemat. *Backtracking* lebih alami dinyatakan dalam algoritma rekursif. *Backtracking* sering disebut sebagai bentuk tipikal dari algoritma rekursif (Morenvino M, Ray A I, Anton R S, 2006).

Prinsip pencarian algoritma *backtracking* didasarkan pada pencarian solusi pada pohon ruang status yang dibangun secara dinamis. Langkah-langkah pencarian solusi adalah sebagai berikut (Rina Dewi Indah Sari, 2007:5):

- a. Solusi dicari dengan membentuk lintasan dari akar ke daun. Aturan pembentukan yang dipakai adalah mengikuti metode pencarian mendalam (DFS). Simpul-simpul yang sudah dilahirkan dinamakan simpul hidup (*live node*). Simpul hidup yang sedang diperluas dinamakan simpul-E (*expand node*). Simpul dinomori dari atas ke bawah sesuai dengan urutan kelahirannya.
- b. Tiap kali simpul-E diperluas, lintasan yang dibangun olehnya bertambah panjang. Jika lintasan yang sedang dibentuk tidak mengarah ke solusi, maka simpul-E tersebut dibunuh sehingga menjadi simpul mati (*dead node*). Fungsi yang digunakan untuk membunuh simpul-E adalah dengan menerapkan fungsi pembatas (*bounding function*). Simpul yang sudah mati tidak akan pernah diperluas lagi.
- c. Jika pembentukan lintasan berakhir dengan simpul mati, maka proses pencarian diteruskan dengan membangkitkan simpul anak yang lainnya. Bila tidak ada lagi simpul anak yang dapat dibangkitkan, maka pencarian solusi dilanjutkan dengan melakukan runut balik ke simpul hidup terdekat (simpul orang tua). Selanjutnya, simpul ini menjadi simpul-E yang baru. Lintasan baru dibangun kembali sampai lintasan tersebut membentuk solusi.
- d. Pencarian dihentikan bila telah menemukan solusi atau tidak ada lagi simpul hidup untuk *backtracking*.

2.4 Multiplicative CRNG (Congruential Random Number Generator)

Menurut Riani L. (2010:3), bilangan acak adalah bilangan yang tidak dapat diprediksi kemunculannya. Pada zaman dahulu bilangan acak diperoleh dengan cara melempar dadu atau mengocok kartu. Pada zaman modern bilangan acak diperoleh dengan cara membentuk bilangan acak secara numerik/ aritmatik (menggunakan komputer), disebut “*Pseudo Random Number*” (bilangan pseudo acak).

Pembangkit bilangan acak harus (Riani L, 2010:2):

- Berdistribusi uniform (0,1) dan tidak berkorelasi antar bilangan
- Membangkitkan secara cepat dan *storage* tidak besar
- Dapat di-“*reproduce*”
- Periode besar, karena kemungkinan bilangan acak dibangkitkan berulang

Menurut Riani L. (2010:3), tidak ada komputasi yang benar-benar menghasilkan deret bilangan acak secara sempurna. Bilangan acak yang dibangkitkan oleh computer adalah bilangan acak semu (*Pseudo Random Number*), karena menggunakan rumus-rumus matematika. Banyak algoritma atau metode yang dapat digunakan untuk membangkitkan bilangan acak, dan salah satu yang terkenal adalah *Linear Congruential Generators*. Algoritma ini dikemukakan oleh D. H. Lehmer pada tahun 1949. Algoritma *Linear Congruential Generator* ditentukan oleh 4 bilangan bulat antara lain (Charles N. Zeeb, dkk, 1984:2):

$$m \quad \text{modulus} \quad m > 0$$

a	faktor pengali	$0 \leq a < m$
c	<i>increment</i>	$0 \leq c < m$
Z_0	angka permulaan	$0 \leq Z_0 < m$

Pseudo RNG, berbentuk:

$$Z_i = (aZ_{i-1} + c) \bmod m$$

Dimana:

Z_i = bilangan acak ke- i dari deretnya

Z_{i-1} = bilangan acak sebelumnya

a = factor pengali

c = *increment*

m = *modulus*

Kunci pembangkit adalah Z_0 yang disebut **umpan** (*seed*) (Riani L, 2010:6).

Menurut Riani L. (2010:21), apabila nilai *increment* (c) adalah 0, maka disebut algoritma tersebut *Multiplicative Congruential Generator*. Maka bentuk *Pseudo RNG* dari *Multiplicative Congruential Generator* sebagai berikut:

$$Z_i = (aZ_{i-1}) \bmod m$$

2.5 Platform Android

Android merupakan salah satu sistem operasi yang terkenal dikalangan perangkat mobile yang merupakan pesaing dari sistem operasi perangkat *mobile* lainnya seperti *Windows Phone*, *iOS*, *BlackBerry*, *MeeGo*, *Bada* dan *Symbian*.

Namun berbeda dengan sistem operasi *mobile* lainnya, karena Android bersifat *Open Source* yang memungkinkan untuk dikembangkan lebih lanjut oleh pihak ketiga.

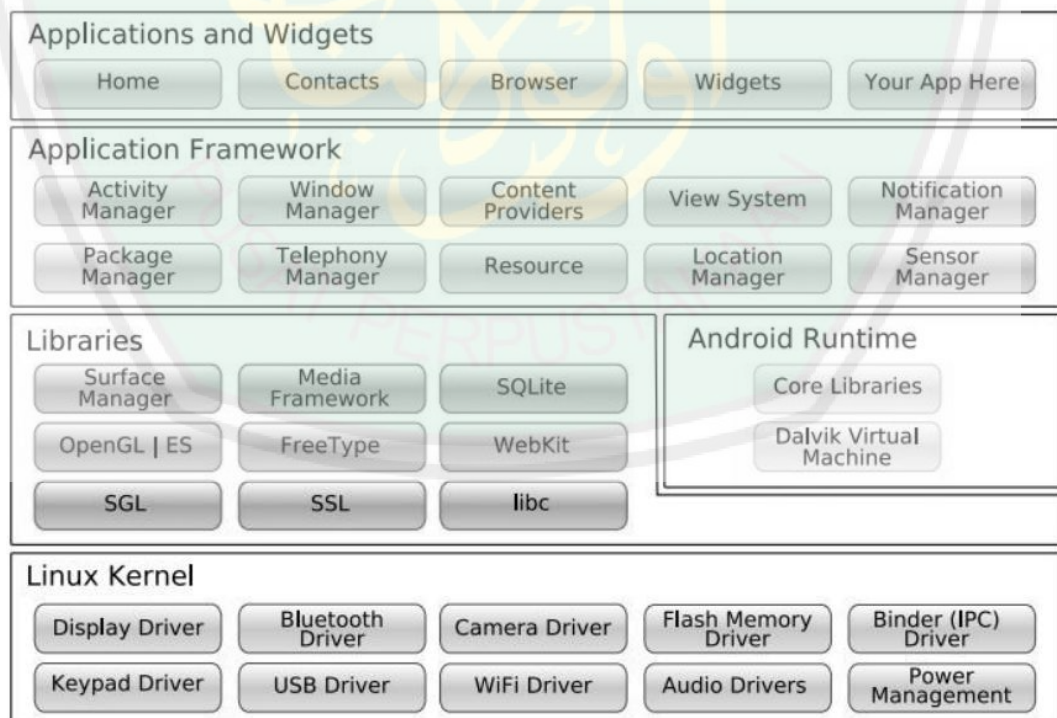
Menurut Safaat (2001:1), *android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Android diakusisi oleh *Google* pada Juli 2005, dan baru dirilis perdana pada 5 November 2007. Android berlisensi di bawah *GNU, General Public Lisensi Versi 2 (GPLv2)*, yang memperbolehkan pihak ketiga untuk mengembangkannya dengan menyertakan term yang sama. Pendistribusiannya di bawah Lisensi *Apache Software (ASL/Apache2)*, yang memungkinkan untuk distribusi kedua dan seterusnya.

Android dirancang dengan arsitektur sebagai berikut (Safaat, 2001:6-9):

- 1) *Application* dan *Widgets*, merupakan layer dimana kita berhubungan dengan aplikasi saja, seperti aplikasi untuk browsing. Selain itu, fungsi-fungsi seperti telepon dan sms juga terdapat pada layer ini.
- 2) *Application Frameworks*, merupakan layer dimana para pembuat aplikasi melakukan pengembangan/ pembuatan aplikasi yang akan dijalankan di sistem operasi *Android*. Beberapa komponen yang terdapat pada layer ini adalah, *Views, Content Provider, Resource Manager, Notification Manager* dan *Activity Manager*.
- 3) *Libraries*, merupakan layer dimana fitur-fitur *Android* berada yang dapat digunakan untuk menjalankan aplikasi. *Library* yang disertakan seperti

library untuk pemutaran audio dan video, tampilan, grafik, *SQLite*, *SSL* dan *Webkit*, dan 3D.

- 4) *Android Run Time*, merupakan layer yang berisi *Core Libraries* dan *Dalvik Virtual Machine (DVK)*. *Core libraries* berfungsi untuk menerjemahkan bahasa Java/C. Sedangkan DVK merupakan sebuah virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien.
- 5) *Linux Kernel*, merupakan layer yang berfungsi sebagai *abstraction/* pemisah antara *hardware* dan *software*. *Linux kernel* inilah yang merupakan inti sistem operasi dari *Android* yang berfungsi untuk mengatur sistem proses, *memory*, *resouce*, dan *driver*. *Linux kernel* yang digunakan *Android* adalah *linux kernel* release 2.6.



Gambar 2.3 Arsitektur Android (Sumber: Safaat, 2001:9)

Perkembangan *Android* yang cepat telah merilis beberapa versi. *Android* versi 4.0 (ICS: *Ice Cream Sandwich*) merupakan versi terbaru dari *Android* saat ini yang diumumkan pada tanggal 19 Oktober 2011. Versi terbaru ini membawa fitur versi sebelumnya yaitu *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 NFC (Riyadli Abrar, 2012:20-24).

Beberapa keunggulan *Platform Android* adalah sebagai berikut (Safaat, 2001:3):

- 1) Lengkap (*Complete Platform*). Para desainer dapat melakukan pendekatan yang komprehensif ketika sedang mengembangkan *platform Android*. *Android* menyediakan banyak *tools* dalam membangun *software* dan merupakan sistem operasi yang aman.
- 2) Terbuka (*Open Source Platform*). *Platform Android* disediakan melalui lisensi *open source*.
- 3) Bebas (*Free Platform*). *Android* merupakan *platform* atau aplikasi yang bebas untuk dikembangkan. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *platform Android*.

BAB III

ANALISA DAN PERANCANGAN

3.1 Analisa dan Perancangan Sistem

3.1.1 Keterangan Umum

Permainan Sudoku Hijaiyah ini dikembangkan pada perangkat bergerak (*mobile device*) khususnya pada sistem operasi *Android* yang bertujuan agar permainan ini dapat dimainkan tanpa dibatasi ruang dan waktu. Seperti teka-teki pada umumnya, permainan Sudoku Hijaiyah ini bermanfaat untuk mengasah logika. Selain itu dalam permainan ini terdapat unsur pendidikan, yaitu pendidikan agama Islam. Pemain akan dikenalkan dengan 30 huruf hijaiyah yang nantinya hanya diambil 9 diantaranya untuk dimainkan sebagai pengganti angka 1 hingga 9.

Pada permulaan permainan diawali dengan pemilihan level dan pemilihan huruf. Permainan terdiri dari 3 level, yaitu: mudah, sedang dan sulit. Semakin sulit level, maka akan semakin banyak jumlah kotak kosong yang harus diisi. Pada pemilihan huruf, user hanya diperkenankan memilih 9 huruf dari 30 huruf hijaiyah untuk mengisi permainan sudoku. Jika user tidak memilih huruf, maka sistem akan memilih 9 huruf hijaiyah secara *default* yang akan dijelaskan pada sub-Bab setelah ini.

Setiap pengisian huruf hijaiyah pada kotak-kotak kecil akan mengeluarkan suara sesuai dengan huruf yang diinputkan oleh user agar

permainan lebih interaktif sehingga manfaat dari penelitian ini dapat tercapai dengan baik yaitu sebagai media pengenalan huruf hijaiyah.

Objek dalam penelitian ini, yaitu algoritma *backtracking* dan *Multiplicative CRNG* akan diterapkan dalam pembangkitan dan penyelesaian *puzzle*. Setiap permainan akan dibangkitkan secara acak menggunakan *backtracking* dan *Multiplicative CRNG* agar pemain tidak merasa bosan.

3.1.2 Rancangan Level Permainan

Permainan terdiri dari 3 level, yaitu: mudah, sedang dan sulit. Semakin sulit level, maka akan semakin banyak jumlah kotak kosong yang harus diisi. Berikut ini level dalam permainan ini:

Tabel 3.1 Jumlah huruf yang harus diisi setiap level

Mudah	Sedang	Sulit
40 Huruf	45 Huruf	48 Huruf

Pada level mudah jumlah huruf dalam *puzzle* yang harus diisi adalah 40 huruf. Sedangkan pada level sedang huruf yang harus diisi adalah 45 huruf. Untuk level sulit, huruf yang harus diisi adalah 50 huruf.

Selain jumlah huruf yang harus diisi berbeda pada setiap level, huruf-huruf yang digunakan dalam setiap level juga berbeda. Huruf hijaiyah yang berjumlah 30 huruf akan dibagi menjadi tiga bagian sebagai *default* sebagaimana jumlah level yang ada. Untuk level mudah akan menggunakan huruf hijaiyah pada bagian 1 (ا, ب, ت, ث, ج, ح, خ, د, ذ). Pada level sedang digunakan huruf hijaiyah pada bagian 2 (ر, ز, س, ش, ص, ض, ط, ظ, ع).

Sedangkan untuk level sulit digunakan huruf hijaiyah pada bagian 3 (ق, ف, غ, و, هـ, ن, م, ل, ك).

Berikut ini pembagiannya:

Tabel 3.2 Pembagian huruf hijaiyah

Bagian 1	Bagian 2	Bagian 3
ا	ر	غ
ب	ز	ف
ت	س	ق
ث	ش	ك
ج	ص	ل
ح	ض	م
خ	ط	ن
د	ظ	هـ
ذ	ع	و

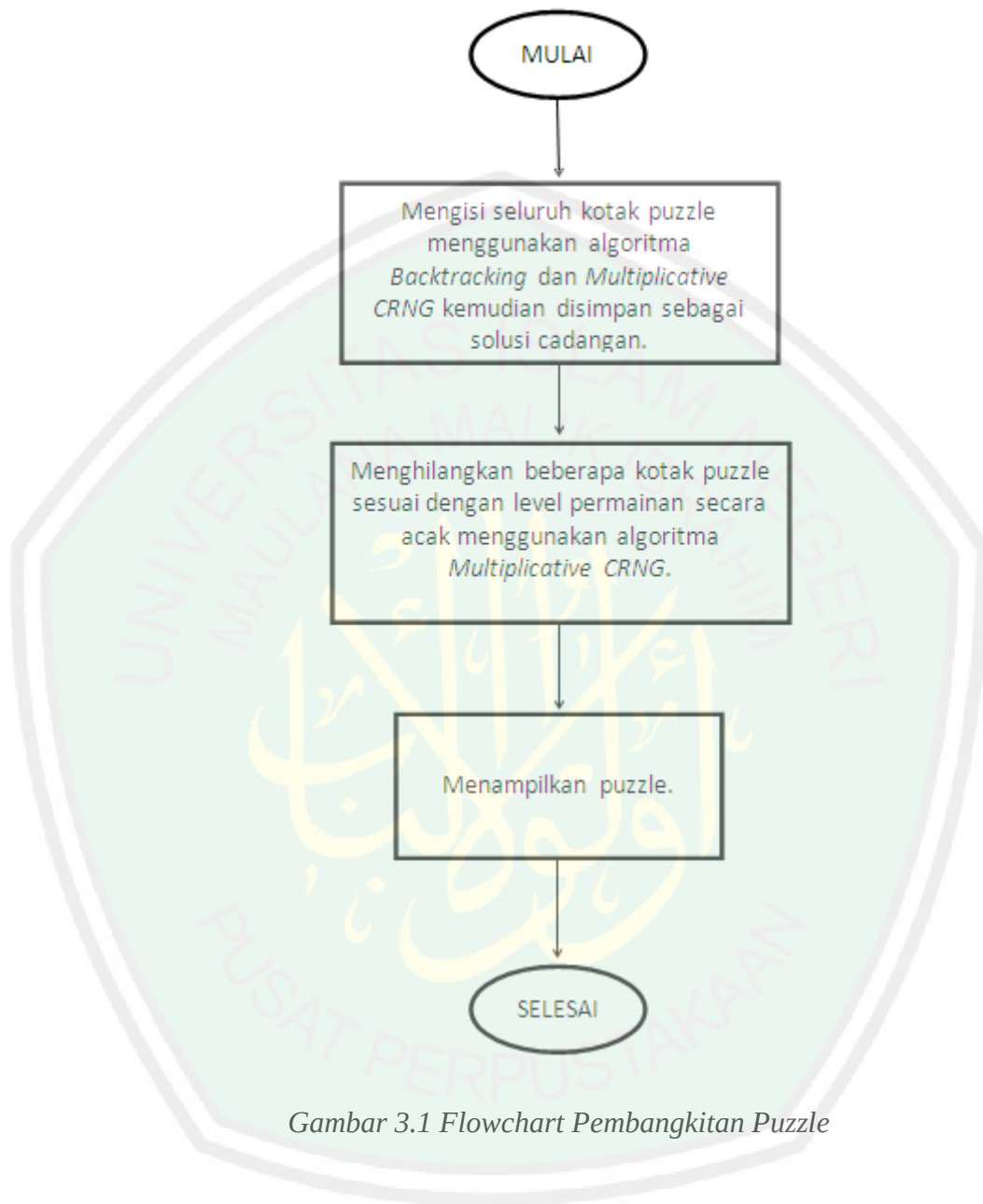
Selain pembagian huruf hijaiyah menjadi 3 bagian, pemain dapat memilih huruf-huruf hijaiyah yang berjumlah 30 huruf sesuai dengan keinginan pemain dengan level yang telah ditentukan. Sehingga pemain dapat bermain Sudoku berbasis huruf hijaiyah ini dengan menggunakan huruf hijaiyah yang diinginkan.

3.2 Rancangan Algoritma *Backtracking* dan *Multiplicative CRNG*

Algoritma *Backtracking* dan *Multiplicative CRNG* dalam permainan Sudoku Hijaiyah ini akan diimplementasikan sebagai pembangkit dan penyelesai *puzzle* permainan.

3.2.1 Pembangkitan *Puzzle*

- a) Mengisi seluruh kotak *puzzle* menggunakan algoritma *Backtracking* dan *Multiplicative CRNG* kemudian disimpan sebagai solusi cadangan.
- b) Menghilangkan beberapa kotak *puzzle* sesuai dengan level permainan secara acak menggunakan algoritma *Multiplicative CRNG*.
- c) Menampilkan *puzzle*.

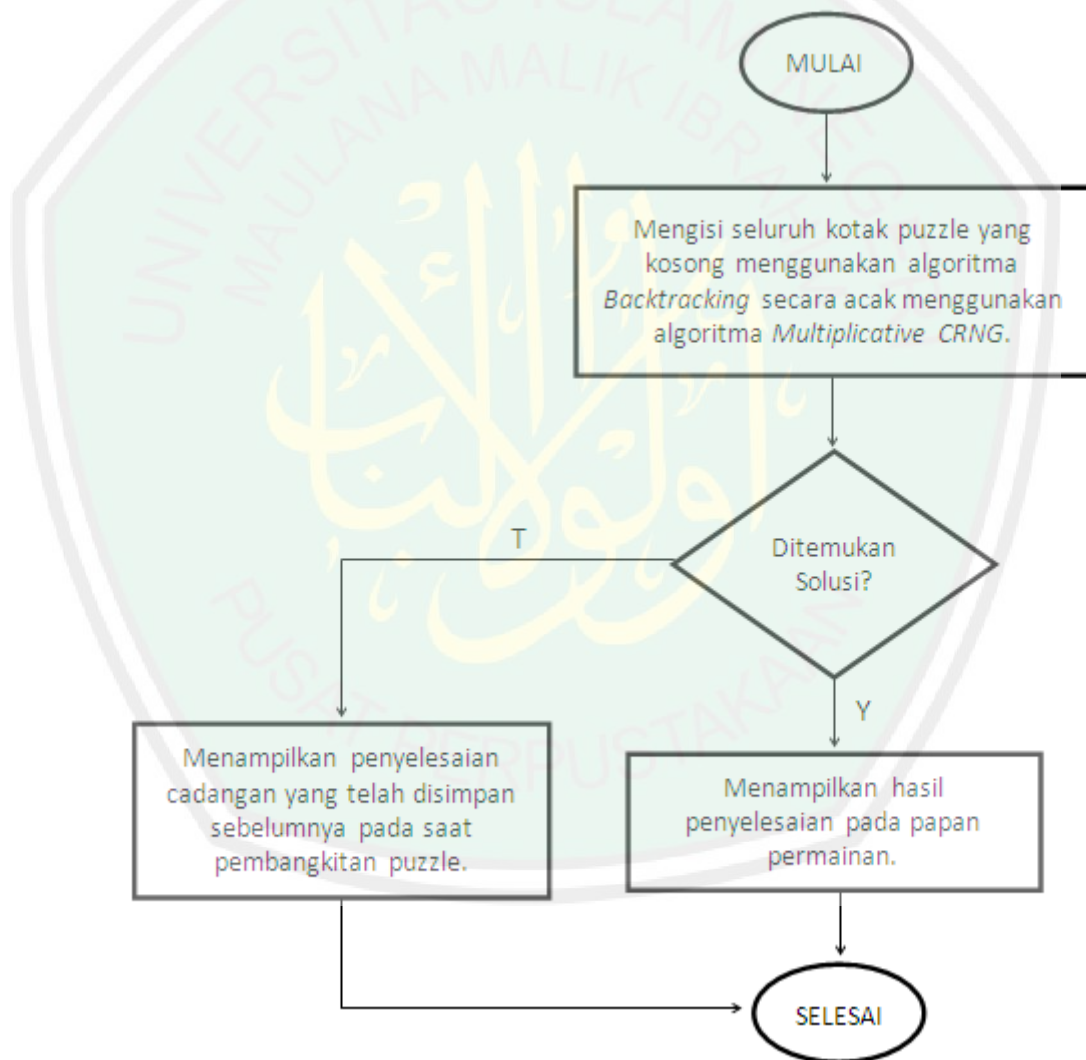


Gambar 3.1 Flowchart Pembangkitan Puzzle

3.2.2 Penyelesaian Puzzle

- a) Mengisi seluruh kotak *puzzle* yang kosong menggunakan algoritma *Backtracking* secara acak menggunakan algoritma *Multiplicative CRNG*.

- b) Jika ditemukan solusi dari inputan user, maka menampilkan hasil penyelesaian pada papan permainan.
- c) Jika tidak ditemukan solusi dari inputan user, maka menampilkan penyelesaian cadangan yang telah disimpan sebelumnya pada saat pembangkitan *puzzle*.



Gambar 3.2 Flowchart Penyelesaian Puzzle

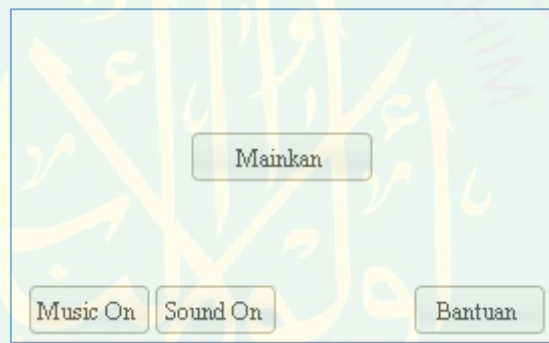
3.3 Perancangan Aplikasi

Untuk mempermudah pengguna mengingat aplikasi ini, aplikasi permainan Sudoku berbasis huruf hijaiyah ini diberi nama aplikasi **Sudoku Hijaiyah**.

3.3.1 Antarmuka Permainan

Dalam permainan Sudoku Hijaiyah terdapat beberapa halaman sebagai berikut:

a) Halaman Utama

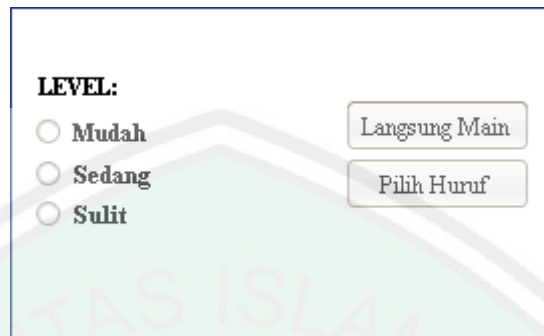


Gambar 3.3 Rancangan Halaman Utama

Dalam halaman utama terdapat beberapa tombol antara lain:

- **Mainkan**, tombol untuk menuju permainan *puzzle* Sudoku Hijaiyah.
- **Music**, tombol untuk mengaktifkan atau menon-aktifkan musik.
- **Sound**, tombol untuk mengaktifkan atau menon-aktifkan suara ketika menekan tombol maupun suara huruf hijaiyah ketika memainkan *puzzle*.
- **Bantuan**, tombol untuk menuju petunjuk permainan.

b) Halaman Pilih Level



Gambar 3.4 Rancangan Halaman Pilih Level

Halaman ini muncul setelah menekan tombol “**Mainkan**” pada halaman utama. Pada halaman ini pengguna diharuskan memilih level permainan kemudian langsung memainkan *puzzle* dengan menekan tombol “**Langsung Main**” atau memilih huruf hijaiyah yang akan dimainkan terlebih dahulu dengan menekan tombol “**Pilih Huruf**”.

c) Halaman Pilih Huruf



Gambar 3.5 Rancangan Halaman Pilih Huruf

Halaman ini muncul setelah menekan tombol “**Pilih Huruf**” pada halaman pilih level. Pada halaman ini pengguna diharuskan memilih 9 dari 30 huruf hijaiyah kemudian memainkannya dengan menekan tombol “**MAINKAN**”. Jika pengguna memilih huruf kurang atau lebih dari 9, maka tombol “**MAINKAN**” akan dinonaktifkan.

d) Halaman *Puzzle*

هـ			ح					ا
	خ			ت		ذ		
ج					ث		ب	
	ب			ا			ح	
ح					خ			د
ط		ن	ت			ا		
		ت		ب				
	ح						ا	ف
ا	د			خ	ج			ز

Gambar 3.6 Rancangan Halaman *Puzzle*

Halaman ini adalah halaman dimainkannya *puzzle* Sudoku Hijaiyah. Pengguna diharuskan mengisi semua kotak kosong dengan cara menekan secara langsung kotak yang hendak diisi. Ketika kotak kosong ditekan, maka akan muncul pilihan huruf yang akan diinputkan ke dalam kotak kosong

tersebut. Huruf yang diisi tidak boleh ada kesamaan dalam satu baris, satu kolom, maupun satu area (kotak 3x3). Ketika huruf yang diinputkan benar dan *Sound* Sudoku Hijaiyah dalam kondisi hidup, maka akan muncul suara huruf sesuai dengan huruf yang diinputkan.

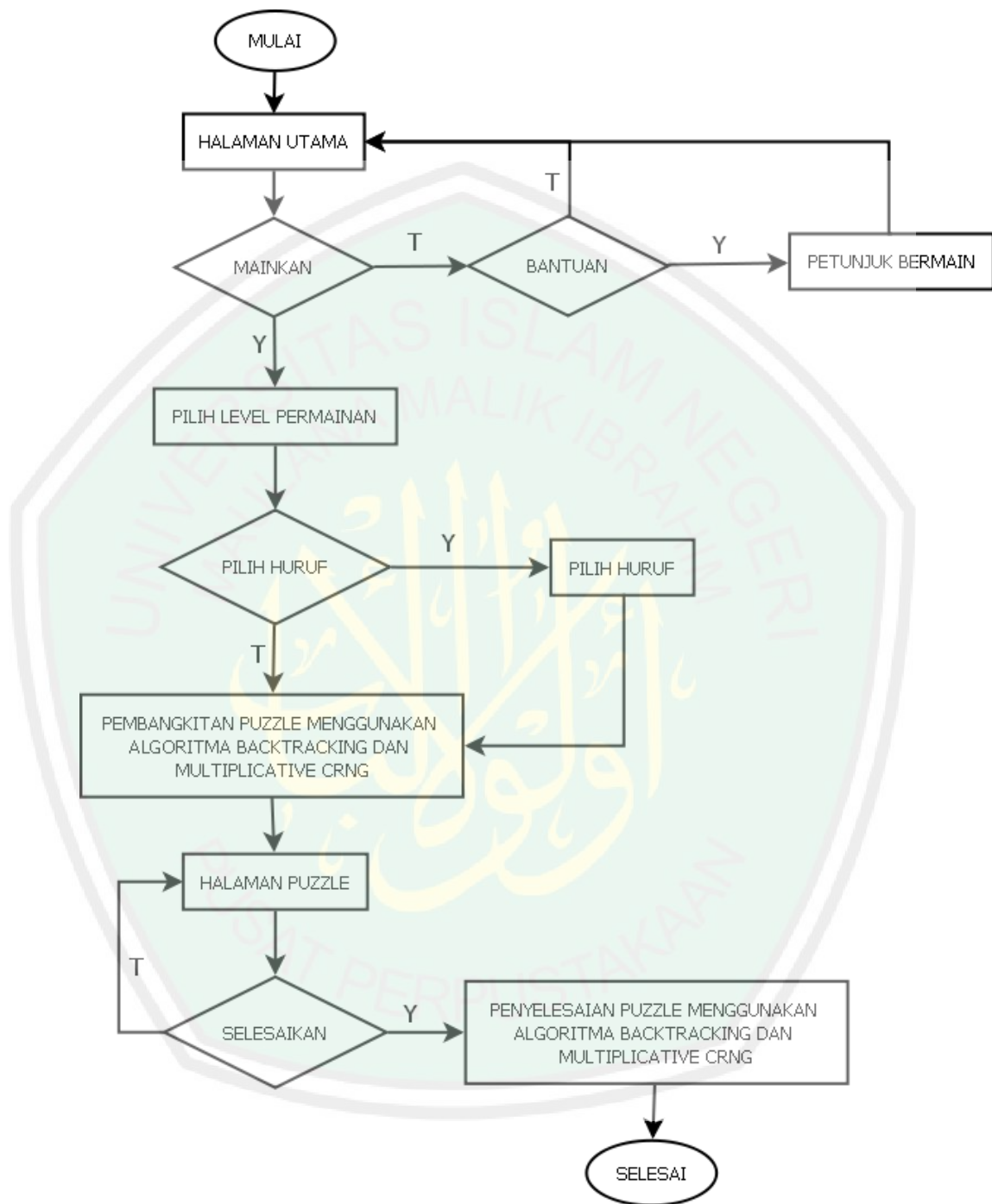
Dalam halaman *puzzle* ini pengguna tidak diperbolehkan menekan tombol “**Back**” pada perangkat *mobile*-nya. Ketika pengguna menekan tombol “**Menu**” maka akan muncul pilihan “**Selesaikan**” atau “**Keluar**”.



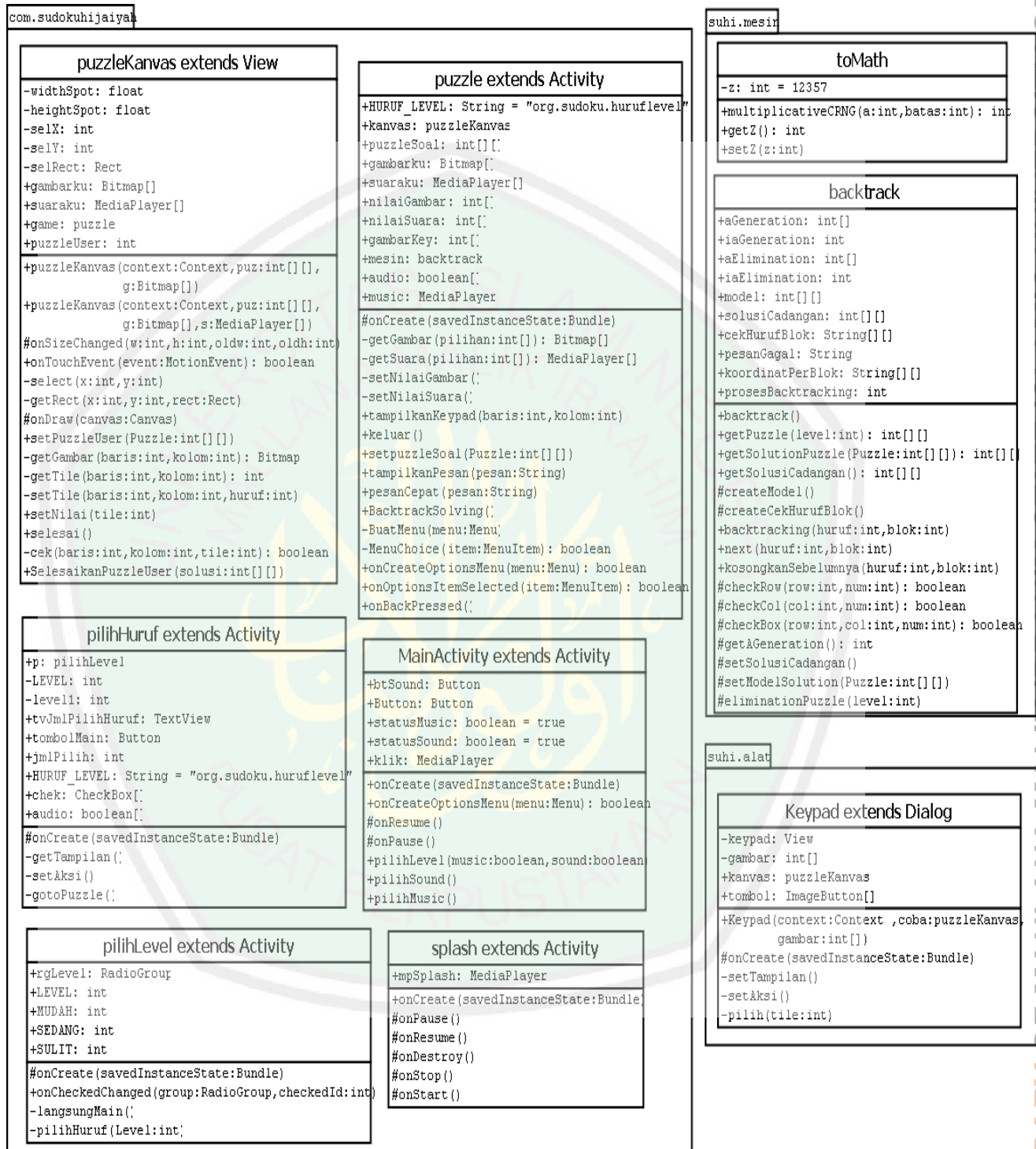
Gambar 3.7 Rancangan Menu Puzzle

Ketika pengguna menekan tombol “**Selesaikan**”, maka *puzzle* akan diselesaikan secara otomatis menggunakan algoritma *Backtracking* dan *Multiplicative CRNG*. Ketika pengguna menekan tombol “**Keluar**”, maka pengguna akan keluar dari halaman *puzzle* dan kembali ke halaman utama.

Berikut ini alur permainan Sudoku Hijaiyah:



Gambar 3.8 Flowchart Permainan Sudoku Hijaiyah



Gambar 3.9 Class Diagram Sudoku Hijaiyah

3.3.2 Kebutuhan Sistem

Berikut ini beberapa perangkat keras maupun lunak yang dibutuhkan untuk mendukung pembuatan dan uji coba aplikasi Sudoku Hijaiyah.

a) Perangkat Keras (*Hardware*)

- PC / Laptop dengan spesifikasi minimal : *Processor* Intel(R) Pentium(R) Dual CPU T2390 @ 1.86GHz (2 CPUs) dan *Memory* 1014MB RAM, digunakan untuk pembuatan aplikasi.
- HandPhone / Perangkat Mobile yang berbasis Android, dibutuhkan untuk melakukan uji coba aplikasi.

b) Perangkat Lunak (*Software*)

- Java, digunakan untuk dapat melakukan kompilasi aplikasi Android. Versi yang digunakan Sun Java SE versi 1.7 atau versi di atasnya.
- Software Eclipse. Merupakan software yang dibutuhkan untuk melakukan coding aplikasi Android. Eclipse yang digunakan adalah versi 4.2.1 (Eclipse Juno) yang support dengan Android Development Tools (ADT).
- ADT (Android Development Tools), plugin tambahan untuk Eclipse yang dibutuhkan untuk membuat aplikasi Android. ADT yang digunakan adalah versi 20.0.3.
- Android SDK (Software Development Kit), yang diperlukan sebagai alat bantu dan API dalam mengembangkan aplikasi Android menggunakan bahasa java. Android SDK yang digunakan adalah versi 20.0.3.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang implementasi dari perancangan yang telah dibuat sebelumnya.

4.1 Impelementasi *Backtracking* dan *Multiplicative CRNG*

4.1.1 *Multiplicative CRNG*

Algoritma *Multiplicative CRNG* diimplementasikan menjadi sebuah fungsi random dalam sebuah *class* java yang diberi nama “**toMath**”. Fungsi random tersebut diberi nama “**multiplicativeCRNG**” yang bersifat *static* dan memiliki parameter **a** dan **batas** yang bernilai integer. *Class* ini memiliki atribut **z** yang diinisialiasi dengan nilai 12357. Berikut ini kode program yang dimaksud:

```
private static int z = 12357;

public static int multiplicativeCRNG(int a, int batas){
    z = (a * z) % (batas + 1);
    return z;
}

public static int getZ() {
    return z;
}

public static void setZ(int z) {
    toMath.z = z;
}
```

4.1.2 *Backtracking*

Proses *backtracking* pada penelitian ini berbeda dengan proses *backtracking* pada umumnya. Proses pengisian huruf pada kotak kosong menggunakan backtrack pada penelitian ini dilakukan per blok atau per area,

bukan per kotak dalam baris dan kolom secara berurutan. Setiap blok atau area memiliki kotak berdimensi 3x3. Koordinat setiap kotak dalam blok disimpan dalam *array* 2 dimensi bertipe *String*, dengan kode program sebagai berikut:

```
static String koordinatPerBlok[][] = {
    { "0,0", "0,1", "0,2", "1,0", "1,1", "1,2", "2,0",
      "2,1", "2,2" }, // DATA BLOK 1
    { "0,3", "0,4", "0,5", "1,3", "1,4", "1,5", "2,3",
      "2,4", "2,5" }, // DATA BLOK 2
    { "0,6", "0,7", "0,8", "1,6", "1,7", "1,8", "2,6",
      "2,7", "2,8" }, // DATA BLOK 3

    { "3,0", "3,1", "3,2", "4,0", "4,1", "4,2", "5,0",
      "5,1", "5,2" }, // DATA BLOK 4
    { "3,3", "3,4", "3,5", "4,3", "4,4", "4,5", "5,3",
      "5,4", "5,5" }, // DATA BLOK 5
    { "3,6", "3,7", "3,8", "4,6", "4,7", "4,8", "5,6",
      "5,7", "5,8" }, // DATA BLOK 6

    { "6,0", "6,1", "6,2", "7,0", "7,1", "7,2", "8,0",
      "8,1", "8,2" }, // DATA BLOK 7
    { "6,3", "6,4", "6,5", "7,3", "7,4", "7,5", "8,3",
      "8,4", "8,5" }, // DATA BLOK 8
    { "6,6", "6,7", "6,8", "7,6", "7,7", "7,8", "8,6",
      "8,7", "8,8" } // DATA BLOK 9
};
```

Algoritma *backtracking* diimplementasikan dalam sebuah fungsi bernama “**backtracking**” dengan parameter **huruf** dan **blok** yang bertipe integer. Kode program *backtracking* sebagai berikut:

```
public void backtracking(int huruf, int blok) throws
Exception {
    if(false == stop){
        if (((3 < prosesBacktracking ) && (1 > huruf)) || (200 <
        prosesBacktracking)) {
            toMath.setZ(12357);
            stop = true;
            prosesBacktracking = 0;
        }
        int hijaiyah = huruf + 1;
        if (huruf > 8) {
            // SOLUSI DITEMUKAN
            pesanGagal = "";
        }
    }
}
```

```

    } else if ((huruf == 0 && blok > 0) || (huruf == 0 &&
prosesBacktracking > 0)) {
        // TIDAK DITEMUKAN SOLUSI
        pesanGagal = "TIDAK DITEMUKAN SOLUSI";
    }
    // Jika Huruf tertentu pada blok tertentu sudah terisi
    if (!cekHurufBlok[huruf][blok].equals("0") ||
cekHurufBlok[huruf][blok].equals("-")) {
        next(huruf, blok);
    } else {
        int xd = 0;
        int yd = 0;
        for (int iblok = 0; iblok <
koordinatPerBlok[blok].length; iblok++)
        {
            int kolomrandom = toMath.multiplicativeCRNG(
getAGeneration(),10);
            if (kolomrandom >= 10) {
                kolomrandom = toMath.multiplicativeCRNG(
getAGeneration(),10);
            }
            if (kolomrandom <= 0) {
                toMath.setZ(12357);
                kolomrandom = toMath.multiplicativeCRNG(
getAGeneration(),10);
            }
            String koordinat =
koordinatPerBlok[blok][kolomrandom - 1];
            String kor[] = koordinat.split(",");
            xd = Integer.parseInt(kor[0]);
            yd = Integer.parseInt(kor[1]);
            // Jika baris xd dan kolom yd sudah terisi
            if (model[xd][yd] != 0) {
                // JIKA DALAM POSISI Mencari solusi dari puzzle
                if (model[xd][yd] == hijaiyah) {
                    cekHurufBlok[huruf][blok] = xd + "," + yd;
                    next(huruf, blok);
                } else {
                    continue;
                }
            }
        }
        if (checkRow(xd, hijaiyah) && checkCol(yd,
hijaiyah)) {
            model[xd][yd] = hijaiyah;
            cekHurufBlok[huruf][blok] = xd + "," + yd;
            Thread.sleep(1);
            next(huruf, blok);
        }
    }
}
// JIKA TIDAK DITEMUKAN HINGGA SELURUH SPOT DALAM BLOK,

```

```
// MAKA LAKUKAN BACKTRACKING KE BLOK SEBELUMNYA
kosongkanSebelumnya(huruf, blok);
cekHurufBlok[huruf][blok] = "0";
prosesBacktracking++;
}}}
```

Fungsi backtracking di atas di desain agar dapat digunakan sebagai pembangkit sekaligus penyelesai *puzzle* sudoku. Dalam fungsi di atas juga menerapkan fungsi random multiplicativeCRNG yang sudah dibuat sebelumnya. Dalam fungsi backtracking di atas terdapat array **cekHurufBlok** bertipe String yang berfungsi untuk menyimpan koordinat setiap huruf yang telah diisi pada setiap kotak. Jika kotak kosong, maka nilai array **cekHurufBlok** pada indeks tertentu bernilai String 0.

Di dalam kode di atas terdapat variable `iaGeneration` dan `iaElimination`. `iaGeneration` merupakan indeks dari array variable **a** yang diberi nama `aGeneration` sebagai parameter fungsi random multiplicativeCRNG untuk proses pembangkit angka secara keseluruhan. Sedangkan `iaElimination` merupakan indeks dari array variable **a** yang diberi nama `aElimination` sebagai parameter fungsi random multiplicativeCRNG untuk proses eliminasi beberapa angka sesuai dengan level permainan dan hasilnya berupa soal *puzzle* yang siap dimainkan. Untuk mengimplementasikan indeks `iaGeneration` di atas digunakan fungsi `getAGeneration()`. Sedangkan `iaElimination` diimplementasikan pada fungsi `eliminationPuzzle()`. Array `aGeneration` memiliki beberapa nilai **a** terbaik hasil uji coba berulang kali untuk membangkitkan seluruh angka *puzzle* Sudoku, nilai-nilai tersebut sebagai berikut: 2, 6, 7, 8. Sedangkan

array `aElimination` memiliki beberapa nilai **a** terbaik hasil uji coba berulang kali untuk eliminasi beberapa angka sesuai level dan hasilnya berupa soal *puzzle* Sudoku yang siap dimainkan, nilai-nilai **a** tersebut sebagai berikut: 3, 23, 13, 37, 39, 40, 45, 112, 71, 81, 103, 129, 135, 139, 147, 157.

Berikut kode program `getAGeneration()`:

```
protected int getAGeneration() {
    if (iaGeneration >= aGeneration.length) {
        iaGeneration = 0;
    }
    return aGeneration[iaGeneration];
}
```

4.1.3 Pembangkitan *Puzzle*

Pembangkitan *puzzle* diimplementasikan dalam fungsi yang diberi nama `getPuzzle` dengan parameter level bertipe integer. Fungsi tersebut akan mengembalikan sebuah array 2 dimensi dengan tipe integer sebagai soal *puzzle* yang siap dimainkan. Berikut kode programnya:

```
public int[][] getPuzzle(int level) {
    createModel();
    try {
        stop = false;
        backtracking(0, 0);
    } catch (Exception e) {
        e.printStackTrace();
    }
    // SOLUSI CADANGAN
    setSolusiCadangan();
    iaGeneration++; // UBAH PENGACAKAN
    eliminationPuzzle(level); // Eliminasi Spot Sesuai Level
    return model;
}
```

Dalam fungsi di atas terdapat fungsi `createModel` yang berfungsi untuk inisialisasi array 2 dimensi *puzzle* yang diberi nama **model**. Setelah

inisialisasi *puzzle*, dilanjutkan mengisi semua kotak *puzzle* dalam fungsi *backtracking*. *Puzzle* yang telah diisi disimpan sebagai solusi cadangan dalam fungsi *setSolusiCadangan*. Setelah solusi cadangan telah didapatkan, dilanjutkan mengubah posisi indeks *a* untuk proses pembangkit dan menghilangkan beberapa angka pada kotak sesuai dengan level permainan sehingga membentuk soal *puzzle* yang siap dimainkan yaitu dalam fungsi *eliminationPuzzle*.

4.1.4 Penyelesaian Puzzle

Penyelesaian *puzzle* diimplementasikan dalam fungsi yang diberi nama *getSolutionPuzzle* dengan parameter soal *puzzle* berbentuk array 2 dimensi bertipe integer. Fungsi tersebut akan mengembalikan sebuah array 2 dimensi dengan tipe integer sebagai jawaban *puzzle* yang semua kotak sudah terisi. Berikut kode programnya:

```
public int[][] getSolutionPuzzle(int Puzzle[][]) {
    createCekHurufBlok();
    setModelSolution(Puzzle);
    try {
        stop = false;
        backtracking(0, 0);
    } catch (Exception e) {
        e.printStackTrace();
    }
    if (!pesanGagal.equals("")) {
        //Jika Tidak Ditemukan Solusi, Tampilkan Solusi Cadangan
        return solusiCadangan;
    } else {
        return model;
    }
}
```


Dalam fungsi di atas terdapat fungsi `createCekHurufBlok` yang berfungsi untuk inisialisasi array **cekHurufBlok**, sedangkan fungsi `setModelSolution` berfungsi untuk inisialisasi *puzzle* yang akan diselesaikan menggunakan backtracking dan multiplicativeCRNG. Jika ditemukan solusi, maka fungsi `getSolutionPuzzle` akan mengembalikan *puzzle* berupa array 2 dimensi bertipe integer. Jika tidak ditemukan solusi, maka fungsi `getSolutionPuzzle` akan mengembalikan solusi cadangan berupa array 2 dimensi bertipe integer yang sebelumnya telah disimpan saat pembangkitan *puzzle*.

4.2 Implementasi Aplikasi

Berikut hasil implementasi aplikasi Sudoku Hijaiyah:



Gambar 4.1 Halaman Utama

Dalam halaman utama terdapat beberapa tombol seperti pada perancangan sebelumnya, antara lain: **Mainkan**, **Musik**, **Sound**, **Bantuan** dan **Exit**. Apabila tombol **Bantuan** (Tanda Tanya) ditekan, maka akan menuju halaman bantuan cara memainkan *puzzle* Sudoku hijaiyah sebagai berikut:



Gambar 4.2 Halaman Bantuan

Untuk memainkan permainan sudoku, maka pengguna diharuskan menekan tombol **Mainkan**. Halaman berikutnya akan ditampilkan halaman pilih level, sebagai berikut:



Gambar 4.3 Halaman Pilih Level

Pada halaman pilih huruf pengguna diharuskan memilih level diantara level mudah, sedang atau sulit. Pengguna diberi pilihan untuk langsung memainkan *puzzle* menggunakan *default* huruf dari permainan atau memilih huruf yang akan dimainkan terlebih dahulu. Berikut halaman pilih huruf:



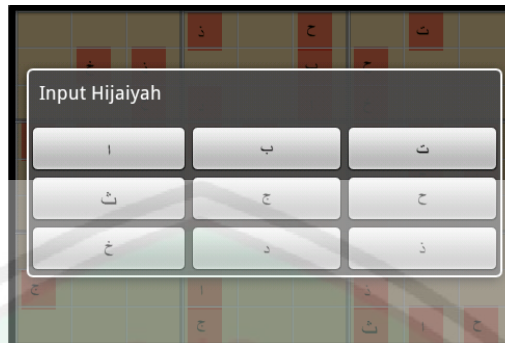
Gambar 4.4 Halaman Pilih Huruf

Pada halaman pilih huruf pengguna diharuskan memilih 9 huruf. Jika pengguna memilih kurang dari 9 maupun lebih dari 9 huruf, maka tombol **mainkan** tidak akan berfungsi. Setelah memilih 9 huruf dan menekan tombol **mainkan**, maka Sudoku Hijaiyah akan membangkitkan *puzzle* menggunakan algoritma *Backtracking* dan *Multiplicative CRNG*. Berikut halaman *puzzle* hasil pembangkitan:



Gambar 4.5 Halaman Puzzle

Untuk mengisi kotak kosong pada *puzzle*, pengguna hanya perlu mengklik kotak kosong tersebut kemudian akan muncul pilihan huruf yang akan diinput, berikut tampilan pilihan input huruf hijaiyah:



Gambar 4.6 Input Hijaiyah

Permainan Sudoku Hijaiyah akan memeriksa huruf yang diinput oleh pengguna. Jika huruf yang diinput pengguna benar maka huruf tersebut akan ditampilkan serta huruf tersebut akan dibunyikan jika **Sound** dalam posisi **On**, jika huruf yang diinputkan salah maka huruf tersebut tidak akan ditampilkan pada papan permainan. Dalam permainan *puzzle*, pengguna tidak diperkenankan menekan tombol **Back**. Dalam permainan *puzzle*, pengguna hanya diperkenankan menekan tombol **Home** dan tombol **Menu**. Berikut tampilan ketika tombol **Menu** ditekan:



Gambar 4.7 Menu Puzzle

Dalam menu terdapat 2 pilihan, yaitu **Selesaikan** dan **Keluar**. Tombol **Selesaikan** berfungsi untuk menyelesaikan *puzzle* menggunakan algoritma *Backtracking* dan *Multiplicative CRNG*. Tombol **Keluar** berfungsi untuk keluar dari permainan *puzzle* dan kembali ke halaman utama. Berikut tampilan *puzzle* setelah diselesaikan menggunakan algoritma *Backtracking* dan *Multiplicative CRNG* ketika menekan tombol **Selesaikan**:

غ	م	ق	و	هـ	ك	ن	ل	ف
هـ	ف	ك	ن	ل	م	ق	غ	و
ن	و	ل	ف	ق	غ	هـ	ك	م
Puzzle Telah Diselesaikan Menggunakan Algoritma Backtracking dan Multiplicative CRNG dengan Parameter: $Z0 = 12357$ $a = 7$ $m = 11$, Selama 0.921 detik								
ق	ن	غ	م	و	ف	ل	هـ	ك
ف	هـ	و	غ	ك	ل	م	ق	ن
ك	ل	م	هـ	ن	ق	ف	و	غ

Gambar 4.8 Hasil Penyelesaian

4.3 Uji Coba

4.3.1 Perbandingan antara Algoritma *Backtracking* dan *Multiplicative CRNG* dengan Algoritma *Harmony Search* dalam Menyelesaikan *Puzzle Sudoku Hijaiyah*

Pengujian perbandingan antara penerapan Algoritma *Harmony Search* pada aplikasi *Sudoku AlBaTa* dengan Algoritma *Backtracking* dan *Multiplicative CRNG* pada *Sudoku Hijaiyah* dalam menyelesaikan *puzzle*

Sudoku Hijaiyah dilakukan menggunakan perangkat keras Sony Ericsson XPERIA X8 dengan spesifikasi sebagai berikut:

Prosesor	: 600 MHz ARM 11
Memory Internal	: 128 MB
Memory Eksternal	: 2 Gb
Resolusi	: 320 x 480 Px (HVGA)
OS	: Android 2.3.7 (Gingerbread)



Gambar 4.9 Aplikasi AlBaTa Sudoku dan Sudoku Hijaiyah pada Perangkat Keras

Pengujian dilakukan terhadap penyelesaian *puzzle* Sudoku Hijaiyah sebanyak 3 kali. Pengujian pertama pada level mudah, pengujian ke dua pada level sedang, dan pengujian ke tiga pada level sulit. Perhitungan waktu menggunakan kode program yang sudah tertanam pada masing-masing aplikasi. Pada aplikasi Sudoku AlBaTa menggunakan parameter yang

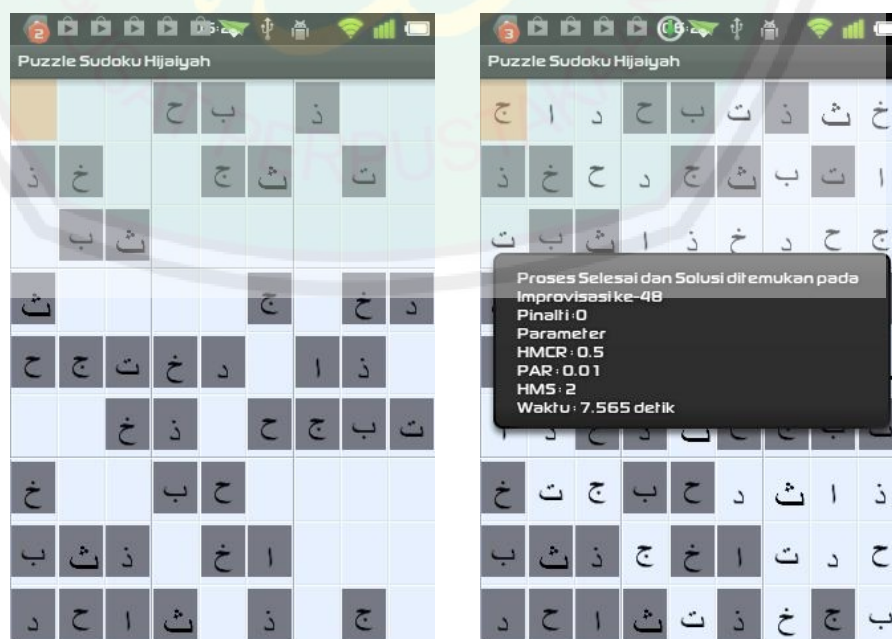
memberi hasil yang optimal pada penelitian sebelumnya. Berikut ini hasil pengujian yang telah dilakukan:

a. Pengujian Pertama

Pada pengujian pertama ini jumlah kotak kosong baik yang terdapat dalam aplikasi Sudoku Hijaiyah maupun dalam aplikasi Sudoku AlBaTa berjumlah 40. Berikut ini merupakan *puzzle* yang diuji:



Gambar 4.10 Pengujian Pertama Puzzle Sudoku Hijaiyah Beserta Solusinya



Gambar 4.11 Pengujian Pertama Puzzle Sudoku AlBaTa Beserta Solusinya

Berikut ini tabel hasil pengujian pertama:

Tabel 4.1 Percobaan Pertama pada Sudoku Hijaiyah dan Sudoku AlBaTa

Aplikasi	Sudoku Hijaiyah	Sudoku AlBaTa
Algoritma	<i>Backtracking</i> dan <i>Multiplicative CRNG</i>	<i>Harmony Search</i>
Parameter	Parameter <i>Multiplicative CRNG</i> : $Z_0 = 12357$, $a = 6$, $m = 11$	HMCR = 0.5, PAR = 0.01, HMS = 2
Level	Mudah	Mudah
Kotak Kosong	40	40
Waktu	0,278 detik	7,565 detik

Berdasarkan tabel 4.1 di atas, penyelesaian *puzzle* sudoku menggunakan algoritma *Backtracking* dan *Multiplicative CRNG* yang memiliki parameter *Multiplicative CRNG*: $Z_0 = 12357$, $a = 6$ dan $m = 11$ dengan jumlah kotak kosong 40 dapat menyelesaikan *puzzle* dalam waktu 0,278 detik. Algoritma *Harmony Search* yang memiliki parameter: HMCR = 0.5, PAR = 0.01 dan HMS = 2 dengan jumlah kotak kosong 40 dapat menyelesaikan *puzzle* dalam waktu 7,565 detik.

b. Pengujian Ke Dua

Pada pengujian ke dua, jumlah kotak kosong baik yang terdapat dalam aplikasi Sudoku Hijaiyah maupun dalam aplikasi Sudoku AlBaTa berjumlah 45. Berikut ini merupakan *puzzle* yang diuji:



Gambar 4.12 Pengujian Ke Dua Puzzle Sudoku Hijaiyah Beserta Solusinya



Gambar 4.13 Pengujian Ke Dua Puzzle Sudoku AlBaTa Beserta Solusinya

Berikut ini tabel hasil pengujian ke dua:

Tabel 4.2 Percobaan Ke Dua pada Sudoku Hijaiyah dan Sudoku AlBaTa

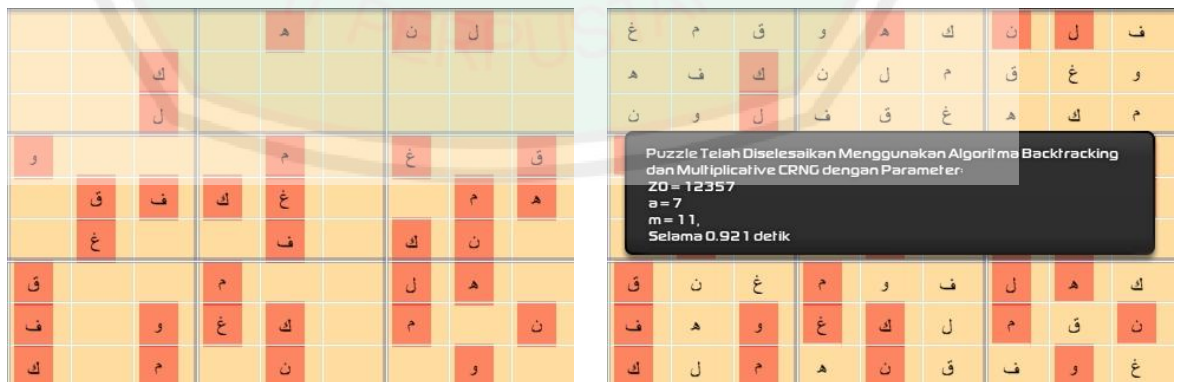
Aplikasi	Sudoku Hijaiyah	Sudoku AlBaTa
Algoritma	Backtracking dan Multiplicative CRNG	Harmony Search
Parameter	Parameter Multiplicative CRNG:	HMCR = 0.9, PAR = 0.1, HMS = 10

	$Z_0 = 12357, a = 7, m = 11$	
Level	Sedang	Sedang
Kotak Kosong	45	45
Waktu	0,036 detik	28,883 detik

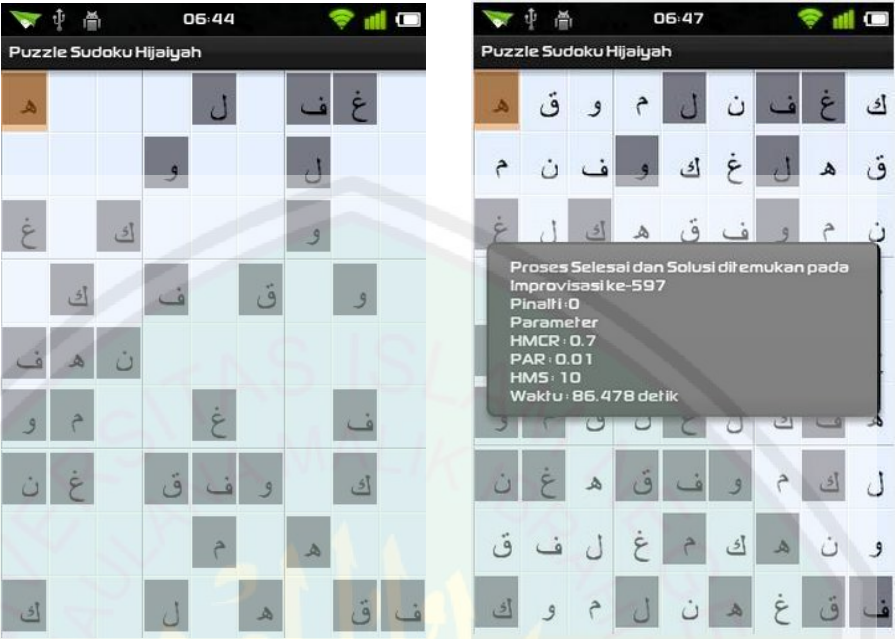
Berdasarkan tabel 4.2 di atas, penyelesaian *puzzle* sudoku menggunakan algoritma *Backtracking* dan *Multiplicative CRNG* yang memiliki parameter *Multiplicative CRNG*: $Z_0 = 12357$, $a = 7$ dan $m = 11$ dengan jumlah kotak kosong 45 dapat menyelesaikan *puzzle* dalam waktu 0,036 detik. Algoritma *Harmony Search* yang memiliki parameter: $HMCR = 0.9$, $PAR = 0.1$ dan $HMS = 10$ dengan jumlah kotak kosong 45 dapat menyelesaikan *puzzle* dalam waktu 28,883 detik.

c. Pengujian Ke Tiga

Pada pengujian ke Tiga, jumlah kotak kosong baik yang terdapat dalam aplikasi Sudoku Hijaiyah maupun dalam aplikasi Sudoku AlBaTa berjumlah 48. Berikut ini merupakan *puzzle* yang diuji:



Gambar 4.14 Pengujian Ke Tiga Puzzle Sudoku Hijaiyah Beserta Solusinya



Gambar 4.15 Pengujian Ke Tiga Puzzle Sudoku AlBaTa Beserta Solusinya

Berikut ini tabel hasil pengujian ke tiga:

Tabel 4.3 Percobaan Ke Tiga pada Sudoku Hijaiyah dan Sudoku AlBaTa

Aplikasi	Sudoku Hijaiyah	Sudoku AlBaTa
Algoritma	<i>Backtracking</i> dan <i>Multiplicative CRNG</i>	<i>Harmony Search</i>
Parameter	Parameter <i>Multiplicative CRNG</i> : $Z_0 = 12357$, $a = 7$, $m = 11$	HMCR = 0.7, PAR = 0.01, HMS = 10
Level	Sulit	Sulit
Kotak Kosong	48	48
Waktu	0,921 detik	86,478 detik

Berdasarkan tabel 4.3 di atas, penyelesaian *puzzle* sudoku menggunakan algoritma *Backtracking* dan *Multiplicative CRNG* yang memiliki parameter *Multiplicative CRNG*: $Z_0 = 12357$, $a = 7$ dan $m = 11$ dengan jumlah kotak kosong 48 dapat menyelesaikan *puzzle* dalam waktu 0,921 detik. Algoritma *Harmony Search* yang memiliki parameter: HMCR =

0.7, $PAR = 0.01$ dan $HMS = 10$ dengan jumlah kotak kosong 48 dapat menyelesaikan *puzzle* dalam waktu 86,478 detik.

Dari beberapa percobaan di atas dapat disimpulkan secara umum bahwa pada penyelesaian *puzzle*, algoritma *Backtracking* dan *Multiplicative CRNG* dapat menyelesaikan *puzzle* dengan jauh lebih cepat dibandingkan dengan algoritma *Harmony Search*. Pada penyelesaian *puzzle* level mudah dengan jumlah kotak kosong 40 algoritma *Backtracking* dan *Multiplicative CRNG* membutuhkan waktu 0,278 detik, sedangkan algoritma *Harmony Search* membutuhkan waktu 7,565 detik. Pada penyelesaian *puzzle* level sedang dengan jumlah kotak kosong 45 algoritma *Backtracking* dan *Multiplicative CRNG* membutuhkan waktu 0,036 detik, sedangkan algoritma *Harmony Search* membutuhkan waktu 28,883 detik. Pada penyelesaian *puzzle* level sulit dengan jumlah kotak kosong 48 algoritma *Backtracking* dan *Multiplicative CRNG* membutuhkan waktu 0,921 detik, sedangkan algoritma *Harmony Search* membutuhkan waktu 86,478 detik.

4.3.2 Uji Coba Pengguna

Pengujian terhadap pengguna dilakukan menggunakan metode pengambilan sampel acak stratifikasi terhadap responden yang berasal dari SDN Pagentan III Kecamatan Singosari Kabupaten Malang pada tanggal 9 Januari 2013 pukul 10.00 WIB. Pengambilan sampel dilakukan dengan membagi populasi menjadi beberapa strata, dimana setiap strata adalah

homogen, sedangkan antar-strata terdapat sifat yang berbeda. Penentuan strata didasarkan pada tingkat kelas siswa yang terdiri dari kelas I hingga kelas VI. Pengambilan sampel pada setiap strata dilakukan dengan proporsi yang sama yaitu 5 anak, sehingga metode ini disebut *Proportionate Stratified Simple Random Sampling*. Dalam pengumpulan data menggunakan teknik wawancara berdasarkan kuisioner yang sudah disiapkan sebelumnya.

Berikut ini adalah rekapitulasi hasil uji coba responden setelah memainkan *game* Sudoku Hijaiyah:



Tabel 4.4 Rekapitulasi Hasil Uji Coba

NO	URAIAN	KELAS I (5 Anak)		KELAS II (5 Anak)		KELAS III (5 Anak)		KELAS IV (5 Anak)		KELAS V (5 Anak)		KELAS VI (5 Anak)		JUMLAH (30 Anak)	
		YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK
1	Pemain mengetahui fungsi setiap menu pada halaman utama	0%	100%	0%	100%	20%	80%	20%	80%	100%	0%	80%	20%	37%	63%
2	Pemain memahami bantuan cara mengisi <i>puzzle</i> pada menu bantuan	0%	100%	0%	100%	40%	60%	20%	80%	0%	100%	40%	60%	17%	83%
3	Pemain mengetahui cara memilih level	100%	0%	80%	20%	100%	0%	100%	0%	100%	0%	100%	0%	97%	3%
4	Pemain mengetahui cara memilih huruf	80%	20%	100%	0%	100%	0%	100%	0%	100%	0%	100%	0%	97%	3%
5	Pemain mengetahui	100%	0%	60%	40%	80%	20%	100%	0%	100%	0%	100%	0%	90%	10%

NO	URAIAN	KELAS I (5 Anak)		KELAS II (5 Anak)		KELAS III (5 Anak)		KELAS IV (5 Anak)		KELAS V (5 Anak)		KELAS VI (5 Anak)		JUMLAH (30 Anak)	
		YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK
	cara mengisi <i>puzzle</i>														
6	Pemain mengetahui cara mencari bantuan mengisi <i>puzzle</i> jika tidak mampu menyelesaikan	100%	0%	100%	0%	100%	0%	80%	20%	100%	0%	100%	0%	97%	3%
7	Pemain merasa bosan memainkan <i>game</i> Sudoku Hijaiyah	0%	100%	40%	60%	80%	20%	0%	100%	20%	80%	80%	20%	37%	63%
8	<i>Game</i> Sudoku Hijaiyah menarik dan menyenangkan	100%	0%	100%	0%	100%	0%	100%	0%	80%	20%	100%	0%	97%	3%
9	Pemain lebih senang belajar huruf Hijaiyah dengan <i>game</i> Sudoku Hijaiyah	100%	0%	100%	0%	100%	0%	100%	0%	40%	60%	0%	100%	73%	27%

NO	URAIAN	KELAS I (5 Anak)		KELAS II (5 Anak)		KELAS III (5 Anak)		KELAS IV (5 Anak)		KELAS V (5 Anak)		KELAS VI (5 Anak)		JUMLAH (30 Anak)	
		YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK	YA	TIDAK
10	Pemain lebih bisa menghafal bentuk huruf Hijaiyah dengan <i>game</i> Sudoku Hijaiyah	100%	0%	100%	0%	100%	0%	100%	0%	80%	20%	60%	40%	90%	10%
11	Pemain lebih bisa menghafal suara bacaan huruf Hijaiyah dengan <i>game</i> Sudoku Hijaiyah	100%	0%	100%	0%	100%	0%	100%	0%	80%	20%	60%	40%	90%	10%
12	Pemain ingin memainkan <i>game</i> Sudoku Hijaiyah lagi	40	60%	0%	100%	0%	100%	20%	80%	0%	100%	0%	100%	10	90

Berdasarkan tabel 4.7 dari 30 responden, mengetahui fungsi setiap menu pada halaman utama sebanyak 37%, dan sebanyak 63% tidak mengetahui. Pemahaman responden mengenai bantuan cara mengisi *puzzle* pada menu bantuan sebanyak 17% memahami dan sebanyak 83% tidak memahami. Pengetahuan responden tentang cara memilih level sebanyak 97% mengetahui, dan sebanyak 3% tidak mengetahui. Responden mengetahui cara memilih huruf sebanyak 97%, dan sebanyak 3% tidak mengetahui. Responden mengetahui cara mengisi *puzzle* sebanyak 90%, dan sebanyak 10% tidak mengetahui. Responden mengetahui cara mencari bantuan mengisi *puzzle* jika tidak mampu menyelesaikan sebanyak 97%, dan sebanyak 3% tidak mengetahui. Responden merasa bosan memainkan *game* Sudoku Hijaiyah sebanyak 37%, dan sebanyak 63% tidak merasa bosan. Responden merasa *game* Sudoku Hijaiyah menarik dan menyenangkan sebanyak 97%, dan sebanyak 3% merasa tidak menarik dan tidak menyenangkan. Responden lebih senang belajar huruf Hijaiyah dengan *game* Sudoku Hijaiyah 73%, dan sebanyak 27% lebih senang belajar huruf Hijaiyah tanpa *game* Sudoku Hijaiyah. Responden lebih bisa menghafal bentuk huruf Hijaiyah dengan *game* Sudoku Hijaiyah sebanyak 90%, dan sebanyak 10% tidak lebih bisa. Responden lebih bisa menghafal suara bacaan huruf Hijaiyah dengan *game* Sudoku Hijaiyah sebanyak 90%, dan sebanyak 10% tidak lebih bisa. Responden ingin memainkan *game* Sudoku Hijaiyah lagi sebanyak 10%, dan 90% tidak ingin.

Tabel 4.5 Detail Kebosanan Pengguna

NO	URAIAN	ALASAN	PROSENTASE TIAP KELAS						PROSENTASE JUMLAH
			KELAS I	KELAS II	KELAS III	KELAS IV	KELAS V	KELAS VI	
1	Bosan	<i>Puzzle</i> Sulit	0%	40%	80%	0%	20%	80%	37%
2	Tidak Bosan	<i>Game</i> Mengasikkan	100%	20%	0%	100%	60%	0%	47%
3	Tidak Bosan	Gambar Menarik	0%	20%	20%	0%	0%	0%	7%
4	Tidak Bosan	Lagu dan Suara Menarik	0%	20%	0%	0%	20%	0%	7%
5	Tidak Bosan	Semua Bagus	0%	0%	0%	0%	0%	20%	3%

Berdasarkan tabel 4.8 dari 30 responden mengenai kebosanan terhadap *game* Sudoku Hijaiyah, responden merasa bosan memainkan *game* Sudoku Hijaiyah karena *puzzle* sulit sebanyak 37%, sebanyak 47% tidak merasa bosan karena *game* mengasikkan, sebanyak 7% tidak merasa bosan karena gambar menarik, sebanyak 7% tidak merasa bosan karena lagu dan suara yang menarik, dan sebanyak 3% tidak merasa bosan karena *game* Sudoku Hijaiyah bagus dari semua segi baik permainan, gambar maupun lagu dan suara.

Jika diklasifikasikan berdasarkan kelas dengan jumlah 5 anak tiap kelas, maka kelas I sebanyak 100% tidak bosan memainkan *game* Sudoku Hijaiyah karena *game* mengasikkan. Kelas II sebanyak 40% merasa bosan karena *puzzle* sulit, sebanyak 20% tidak merasa bosan karena *game* mengasikkan, sebanyak 20% tidak merasa bosan karena gambar menarik dan sebanyak 20% tidak bosan karena lagu dan suara yang menarik. Kelas III sebanyak 80% merasa bosan karena *puzzle* sulit, dan sebanyak 20% tidak bosan karena gambar menarik. Kelas IV sebanyak 100% tidak merasa bosan

karena *game* mengasikkan. Kelas V sebanyak 20% merasa bosan karena *puzzle* sulit, 60% tidak merasa bosan karena *game* mengasikkan dan 20% tidak merasa bosan karena lagu dan suara yang menarik. Kelas VI sebanyak 80% merasa bosan karena *puzzle* sulit, dan sebanyak 20% tidak merasa bosan karena *game* Sudoku Hijaiyah bagus dari semua segi baik permainan, gambar maupun lagu dan suara.

Tabel 4.6 Detail Kemenarikan *Game* Sudoku Hijaiyah

NO	URAIAN	ALASAN	PROSENTASE TIAP KELAS						PROSENTASE JUMLAH
			KELAS I	KELAS II	KELAS III	KELAS IV	KELAS V	KELAS VI	
1	Tidak Menarik	<i>Puzzle</i> Sulit	0%	0%	0%	0%	20%	0%	3%
2	Menarik	<i>Game</i> Mengasikkan	100%	20%	20%	100%	60%	0%	50%
3	Menarik	Gambar Menarik	0%	40%	80%	0%	0%	0%	20%
4	Menarik	Lagu dan Suara Menarik	0%	40%	0%	0%	20%	80%	24%
5	Menarik	Semua Bagus	0%	0%	0%	0%	0%	20%	3%

Berdasarkan tabel 4.9 dari 30 responden mengenai kemenarikan terhadap *game* Sudoku Hijaiyah, responden merasa *game* Sudoku Hijaiyah tidak menarik karena *puzzle* sulit sebanyak 3%, sebanyak 50% merasa *game* menarik karena *game* mengasikkan, sebanyak 20% merasa *game* menarik karena gambar menarik, sebanyak 24% merasa *game* menarik karena lagu dan suara yang menarik, dan sebanyak 3% merasa *game* menarik karena *game* Sudoku Hijaiyah bagus dari semua segi baik permainan, gambar maupun lagu dan suara.

Jika diklasifikasikan berdasarkan kelas dengan jumlah 5 anak tiap kelas, maka kelas I sebanyak 100% merasa *game* Sudoku Hijaiyah menarik karena *game* mengasikkan. Kelas II sebanyak 20% merasa *game* menarik karena *game* mengasikkan, sebanyak 40% merasa *game* menarik karena gambar menarik, dan sebanyak 40% merasa *game* menarik karena lagu dan suara yang menarik. Kelas III sebanyak 20% merasa *game* menarik karena *game* mengasikkan, dan sebanyak 80% merasa *game* menarik karena gambar menarik. Kelas IV sebanyak 100% merasa *game* menarik karena *game* mengasikkan. Kelas V sebanyak 20% merasa *game* tidak menarik karena *puzzle* sulit, 60% merasa *game* menarik karena *game* mengasikkan dan 20% merasa *game* menarik karena lagu dan suara yang menarik. Kelas VI sebanyak 80% merasa *game* menarik karena lagu dan suara yang menarik, dan sebanyak 20% merasa *game* menarik karena *game* Sudoku Hijaiyah bagus dari semua segi baik permainan, gambar maupun lagu dan suara.

Tabel 4.7 Detail Keinginan Bermain Lagi *Game* Sudoku Hijaiyah

NO	URAIAN	ALASAN	PROSENTASE TIAP KELAS						PROSENTASE JUMLAH
			KELAS I	KELAS II	KELAS III	KELAS IV	KELAS V	KELAS VI	
1	Ingin Main Lagi		40%	0%	0%	20%	0%	0%	10%
2	Tidak Ingin Main Lagi	Capek	60%	100%	60%	20%	0%	20%	43%
3	Tidak Ingin Main Lagi	<i>Puzzle</i> Sulit	0%	0%	40%	60%	100%	80%	47%

Berdasarkan tabel 4.10 dari 30 responden mengenai keinginan untuk memainkan *game* Sudoku Hijaiyah lagi, responden ingin memainkan *game* Sudoku Hijaiyah lagi sebanyak 10%, sebanyak 43% tidak ingin memainkan *game* lagi karena capek, dan sebanyak 47% tidak ingin memainkan *game* lagi karena *puzzle* sulit.

Jika diklasifikasikan berdasarkan kelas dengan jumlah 5 anak tiap kelas, maka kelas I sebanyak 40% ingin memainkan *game* Sudoku Hijaiyah lagi, dan sebanyak 60% tidak ingin memainkan lagi karena capek. Kelas II sebanyak 100% tidak ingin memainkan *game* lagi karena capek. Kelas III sebanyak 60% tidak ingin memainkan *game* lagi karena capek, dan sebanyak 40% tidak ingin memainkan *game* lagi karena *puzzle* sulit. Kelas IV sebanyak 20% ingin memainkan *game* lagi, sebanyak 20% tidak ingin memainkan *game* lagi karena capek, dan sebanyak 60% tidak ingin memainkan *game* lagi karena *puzzle* sulit. Kelas V sebanyak 100% tidak ingin memainkan *game* lagi karena *puzzle* sulit. Kelas VI sebanyak 20% tidak ingin memainkan *game* lagi karena capek, dan sebanyak 80% tidak ingin memainkan *game* lagi karena *puzzle* sulit.

Tabel 4.8 Rata-Rata Prosentase Tiap Kelas (dalam %)

Kelas	Halaman Utama	Bantuan	Pilih Level	Pilih Huruf	Isi Puzzle	Solver	Pemahaman Game	Tidak Bosan	Tertarik	Ingin Main Lagi	Minat Game	Senang Belajar	Hafal Bentuk	Hafal Suara	Minat Belajar	RATA2
I	0	0	100	80	100	100	63.33	100	100	40	80.00	100	100	100	100.00	81.11
II	0	0	80	100	60	100	56.67	60	100	0	53.33	100	100	100	100.00	70.00
III	20	40	100	100	80	100	73.33	20	100	0	40.00	100	100	100	100.00	71.11
IV	20	20	100	100	100	80	70.00	100	100	20	73.33	100	100	100	100.00	81.11
V	100	0	100	100	100	100	83.33	80	80	0	53.33	40	80	80	66.67	67.78
VI	80	40	100	100	100	100	86.67	20	100	0	40.00	0	60	60	40.00	55.56

Berdasarkan tabel 4.11 mengenai rata-rata prosentase penilaian baik responden terhadap *game* Sudoku Hijaiyah yang diklasifikasikan berdasarkan kelas, kelas I memiliki rata-rata prosentase penilaian baik terhadap *game* Sudoku Hijaiyah sebanyak 81,11% yang terdiri dari pemahaman *game* Sudoku Hijaiyah sebanyak 63,33%, minat terhadap *game* Sudoku Hijaiyah sebanyak 80%, dan minat belajar huruf Hijaiyah melalui *game* Sudoku Hijaiyah sebanyak 100%. Kelas II memiliki rata-rata prosentase sebanyak 70% yang terdiri dari pemahaman *game* Sudoku Hijaiyah sebanyak 65,67%, minat terhadap *game* Sudoku Hijaiyah sebanyak 53,33%, dan minat belajar huruf Hijaiyah melalui *game* Sudoku Hijaiyah sebanyak 100%. Kelas III memiliki rata-rata prosentase sebanyak 71,11% yang terdiri dari pemahaman *game* Sudoku Hijaiyah sebanyak 73,33%, minat terhadap *game* Sudoku Hijaiyah sebanyak 40%, dan minat belajar huruf Hijaiyah melalui *game* Sudoku Hijaiyah sebanyak 100%. Kelas IV memiliki rata-rata prosentase sebanyak 81,11% yang terdiri dari pemahaman *game* Sudoku Hijaiyah sebanyak 70%, minat terhadap *game* Sudoku Hijaiyah sebanyak 73,33%, dan minat belajar huruf Hijaiyah melalui *game* Sudoku Hijaiyah sebanyak 100%. Kelas V memiliki rata-rata prosentase sebanyak 67,78% yang terdiri dari pemahaman *game* Sudoku Hijaiyah sebanyak 83,33%, minat terhadap *game* Sudoku Hijaiyah sebanyak 53,33%, dan minat belajar huruf Hijaiyah melalui *game* Sudoku Hijaiyah sebanyak 66,67%. Kelas VI memiliki rata-rata prosentase sebanyak 55,56% yang terdiri dari pemahaman *game* Sudoku

Hijaiyah sebanyak 86,67%, minat terhadap *game* Sudoku Hijaiyah sebanyak 40%, dan minat belajar huruf Hijaiyah melalui *game* Sudoku Hijaiyah sebanyak 40%.

Dari data hasil pengujian *game* di atas dapat disimpulkan secara umum bahwa *game* Sudoku Hijaiyah berhasil diterima oleh siswa sekolah dasar, dibuktikan dengan hasil positif mendominasi prosentase terbesar dalam uji coba. Sistem navigasi *game* Sudoku Hijaiyah dapat dipahami oleh pemain dengan prosentase 97% pemain mengetahui cara memilih level, 97% pemain mengetahui cara memilih huruf, 90% pemain mengetahui cara mengisi *puzzle*, 97% pemain mengetahui cara mencari bantuan mengisi *puzzle* jika tidak mampu menyelesaikan, 37% pemain mengetahui fungsi setiap menu pada halaman utama dan 17% pemain memahami bantuan cara mengisi *puzzle* pada menu bantuan. Sebanyak 63% pemain tidak merasa bosan memainkan *game* Sudoku Hijaiyah dan sebanyak 97% pemain merasa *game* Sudoku Hijaiyah menarik dan menyenangkan.

Pembelajaran huruf Hijaiyah dalam *game* Sudoku Hijaiyah cukup berhasil dengan sebanyak 73% pemain lebih senang belajar huruf Hijaiyah menggunakan *game* Sudoku Hijaiyah, 90% pemain lebih bisa menghafal bentuk huruf Hijaiyah menggunakan *game* Sudoku Hijaiyah dan 90% pemain lebih bisa menghafal suara bacaan huruf Hijaiyah menggunakan *game* Sudoku Hijaiyah.

Game Sudoku Hijaiyah tidak membosankan bagi siswa sekolah dasar karena *game* mengasikkan, dibuktikan pada tabel 4.8 di atas dengan jumlah

tidak bosan memainkan *game* Sudoku Hijaiyah dengan alasan *game* mengasikkan mendominasi prosentase terbesar dalam uji coba yaitu sebanyak 47%. Selain itu juga alasan *game* yang mengasikkan mendominasi alasan kemenarikan *game* Sudoku Hijaiyah, dibuktikan dengan jumlah prosentasenya sebanyak 50% pada tabel 4.9 di atas.

Karena waktu uji coba berdekatan dengan jam pulang sekolah SDN Pagentan III Kecamatan Singosari Kabupaten, maka keinginan responden untuk memainkan *game* Sudoku Hijaiyah lagi hanya sebanyak 10% seperti pada tabel 4.10 di atas. Sebanyak 43% responden tidak mau bermain lagi karena capek dan sisanya 47% pemain tidak mau bermain lagi karena merasa *puzzle* sulit. *Game* Sudoku Hijaiyah lebih tepat dimainkan oleh siswa sekolah dasar yang duduk di kelas antara kelas I hingga kelas IV, dibuktikan pada tabel 4.11 dengan rata-rata prosentase penilaian baik yang dimiliki kelas I hingga kelas IV terhadap *game* Sudoku Hijaiyah di atas 70%.

4.4 Integrasi *Game* Sudoku Hijaiyah dengan Islam

Al-Qur'an merupakan kitab suci Agama Islam yang diturunkan menggunakan bahasa Arab. Hal ini diperjelas dalam firman Allah SWT sebagai berikut:

إِنَّا أَنْزَلْنَاهُ قُرْآنًا عَرَبِيًّا لَعَلَّكُمْ تَعْقِلُونَ (يوسف : ٢)

Artinya:

“Sesungguhnya Kami menurunkannya berupa Al-Qur’an dengan berbahasa Arab, agar kamu memahaminya.” (QS Yusuf : 2) (Al-Jumanatul ‘Ali, 2005:236).

Dalam sebuah hadits shahih riwayat Al-Bukhari dijelaskan pentingnya mempelajari dan mengajarkan Al-Qur’an. Berikut hadits yang dimaksud:

عَنْ عُثْمَانَ بْنِ عَمَّانَ (رَضِيَ) قَالَ قَالَ رَسُولُ اللَّهِ (ص)
خَيْرُكُمْ مَنْ تَعَلَّمَ الْقُرْآنَ وَعَلَّمَهُ .

Artinya:

Dari Usman bin ‘Affan ra, ia berkata, Rasulullah SAW bersabda, “Orang yang paling baik di antara kalian adalah yang mempelajari Al-Qur’an dan mengajarkannya”. (Ali Mustafa Yaqub, 1990:17)

Oleh karena itu sebagai umat muslim seharusnya mempelajari Al-Qur’an dan mengajarkannya. Dalam mempelajari Al-Qur’an diperlukan pengetahuan mengenai bahasa Arab karena bahasa yang digunakan dalam Al-Qur’an adalah bahasa Arab. Dalam mempelajari bahasa Arab diharuskan mengetahui huruf-huruf bahasa Arab terlebih dahulu. Huruf bahasa Arab biasa dikenal sebagai huruf hijaiyah .

Terdapat berbagai macam cara pembelajaran, salah satunya yaitu melalui permainan. Salah satu contoh aplikasi permainan adalah *game* Sudoku Hijaiyah. *Game* Sudoku Hijaiyah ini dimainkan pada perangkat *mobile* yang

menggunakan sistem operasi Android. Seperti isi kandungan hadits yang sudah disebutkan di atas, *game* Sudoku Hijaiyah berfungsi sebagai media pembelajaran huruf Hijaiyah yang merupakan salah satu bagian dari kegiatan mempelajari dan mengajarkan Al-Qur'an, yaitu dalam pengenalan huruf Hijaiyah. Bagi peneliti, *game* Sudoku Hijaiyah ini berfungsi sebagai media untuk mengajarkan huruf Hijaiyah, sedangkan bagi pemain *game* ini merupakan media untuk mempelajari huruf Hijaiyah.



BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil implementasi dan uji coba yang telah peneliti lakukan dapat disimpulkan bahwa Algoritma *Backtracking* dan *Multiplicative CRNG* merupakan dua algoritma yang baik sebagai pembangkit dan penyelesai *puzzle* sudoku, dibuktikan oleh hasil uji coba yang menunjukkan *game* Sudoku Hijaiyah yang mampu membangkitkan dan menyelesaikan *puzzle* sudoku baik pada level mudah, sedang maupun sulit. Aplikasi *game* Sudoku Hijaiyah menggunakan algoritma *Backtracking* dan *Multiplicative CRNG* sebagai pembangkit dan penyelesai permainan merupakan perbaikan dari penelitian Riyadli Abrar mengenai *game* Sudoku AlBaTa pada tahun 2012 yang menggunakan algoritma *Harmony Search* sebagai pembangkit dan penyelesai permainan, dibuktikan oleh hasil uji coba yang menunjukkan *game* Sudoku Hijaiyah yang mampu menyelesaikan *puzzle* sudoku lebih cepat dibandingkan dengan penyelesaian *puzzle* sudoku pada *game* Sudoku AlBaTa. *Game* Sudoku Hijaiyah ini lebih tepat jika dimainkan oleh siswa sekolah dasar yang duduk di kelas antara kelas I hingga kelas IV, hal ini dapat dilihat dari hasil uji coba pengguna yang menunjukkan rata-rata prosentase dari pemahaman *game*, minat memainkan *game* dan minat belajar huruf Hijaiyah melalui *game* Sudoku Hijaiyah di atas 70%.

5.2 Saran

Tentunya masih banyak kekurangan dalam penelitian aplikasi permainan Sudoku Hijaiyah ini. Oleh karena itu penulis menyarankan beberapa hal untuk bahan pengembangan selanjutnya, diantaranya:

1. Mengembangkan aplikasi Sudoku Hijaiyah dengan lebih menarik, baik dari segi tampilan maupun alur permainan agar lebih disukai oleh semua anak-anak.
2. Mengembangkan aplikasi Sudoku Hijaiyah untuk sistem operasi perangkat *mobile* yang lain seperti Windows Phone, BlackBerry, iOS dan yang lainnya.

DAFTAR PUSTAKA

- Abrar, Riyadli. 2012. *Aplikasi Permainan Arabic Sudoku Menggunakan Metode Harmony Search sebagai Pembangkit dan Penyelesaian Permainan*. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim. Malang.
- Gunawan, Agustinus Tri. 2010. *Penerapan Algoritma Backtracking dan Elimination untuk Membangun Generator dan Solver dalam Menyelesaikan Permainan Sudoku*. Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM. Yogyakarta.
- Izzuddin, Muhammad. 2009. *Memperbaiki bacaan Al-Qur'an (Metode Tartil 12 Jam)*. Assalam Publishing. Solo.
- L, Riani. 2010. *Pembangkit Bilangan Acak. Mata Kuliah Pemodelan & Simulasi*. Jurusan Teknik Informatika Universitas Komputer Indonesia. Bandung.
- Morenvino, M. Ray, I. dan Anton, S. 2006. *Penerapan Algoritma Runut-Balik Untuk Penyelesaian Teka-Teki Sudoku*. Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika Institut Teknologi Bandung. Bandung.
- Safaat H, Nazruddin. 2011. *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika. Bandung.
- Sari, Rina Dewi Indah. 2007. *Analisis Penyelesaian Puzzle Sudoku dengan Menerapkan Algoritma Backtracking Memanfaatkan Bahasa*

Pemrograman Visual Basic 6.0 dan Database Microsoft Access 2003.

Jurusan Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) “ASIA”. Malang.

Yaqub, Ali Mustafa. 1990. *Nasihat Nabi kepada Pembaca dan Penghafal QUR'AN*. Jakarta: GEMA INSANI.

Zeeb, Charles N. dan Patrick J. Burns. 1984. *Random Number Generator Recommendation*. Department of Mechanical Engineering Colorado State University. Colorado.

Al-Qur'an dan Terjemahannya. Al-Jumanatul 'Ali. Bandung.

http://en.wikipedia.org/wiki/Sudoku_algorithms, diakses pada tanggal 31 Mei 2012 pukul 14.04

<http://www.teknojurnal.com/2012/01/18/jumlah-pelanggan-seluler-di-indonesia-hampir-mendekati-jumlah-penduduk-indonesia/>, diakses pada tanggal 14 Januari 2013 pukul 11.12

http://gs.statcounter.com/#mobile_os-ID-monthly-201112-201212, diakses pada tanggal 19 Januari 2013 pukul 08.48

LAMPIRAN

Data Responden Uji Coba

NO	NAMA	L/P	KELAS	SEKOLAH
1	Dimas Ali Maftuh	L	I	SDN Pagentan III Singosari
2	M. Hardiansyah	L	I	SDN Pagentan III Singosari
3	Faris Hidayah	L	I	SDN Pagentan III Singosari
4	Najwa Regina Putri	P	I	SDN Pagentan III Singosari
5	Zahra Rahmania A.	P	I	SDN Pagentan III Singosari
6	Iqbal AhmadFaizin	L	II	SDN Pagentan III Singosari
7	Lailatur Rohmah	P	II	SDN Pagentan III Singosari
8	Arsidita Galuh P.	P	II	SDN Pagentan III Singosari
9	M. Hafid Akbar R.	L	II	SDN Pagentan III Singosari
10	Afinas Rizal P.	L	II	SDN Pagentan III Singosari
11	Vini Cahya D.	P	III	SDN Pagentan III Singosari
12	Rizky Dwi Stefani	P	III	SDN Pagentan III Singosari
13	Syahdana Arya W.	L	III	SDN Pagentan III Singosari
14	Septiano Erza A.P.	L	III	SDN Pagentan III Singosari
15	Randyto Ilham K.	L	III	SDN Pagentan III Singosari
16	Tanaya Dewanti B.	P	IV	SDN Pagentan III Singosari
17	Ade Bagus P.	L	IV	SDN Pagentan III Singosari
18	Tiara Nur B.	P	IV	SDN Pagentan III Singosari
19	M. Al-Vito R.	L	IV	SDN Pagentan III Singosari
20	M. Ariel	L	IV	SDN Pagentan III Singosari
21	M. Hafinudin	L	V	SDN Pagentan III Singosari
22	Inayah Widya N.	P	V	SDN Pagentan III Singosari
23	Cicik Berkah N.	P	V	SDN Pagentan III Singosari
24	Wahidah Tsamara P.Y.	P	V	SDN Pagentan III Singosari
25	Destria Audi N	P	V	SDN Pagentan III Singosari
26	Amalia Naura	P	VI	SDN Pagentan III Singosari
27	Hendri Idawan	L	VI	SDN Pagentan III Singosari
28	Syifa M. Fatah	L	VI	SDN Pagentan III Singosari
29	Haidar Ali A.	L	VI	SDN Pagentan III Singosari
30	Mareta H	P	VI	SDN Pagentan III Singosari

Foto Uji Coba Pengguna di SDN Pagentan III Singosari

