

**APLIKASI PERMAINAN SUDOKU SHARAF MENGGUNAKAN
LINEAR CONGRUENTIAL GENERATOR SEBAGAI PEMBANGKIT
DAN BRANCH AND BOUND SEBAGAI PENYELESAI PERMAINAN**

SKRIPSI

Oleh:

AHMAD BAIHAQI

NIM. 09650212



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

**APLIKASI PERMAINAN SUDOKU SHARAF MENGGUNAKAN
LINEAR CONGRUENTIAL GENERATOR SEBAGAI PEMBANGKIT
DAN BRANCH AND BOUND SEBAGAI PENYELESAI PERMAINAN**

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:
AHMAD BAIHAQI
NIM. 09650212**

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

**APLIKASI PERMAINAN SUDOKU SHARAF MENGGUNAKAN
LINEAR CONGRUENTIAL GENERATOR SEBAGAI PEMBANGKIT
DAN BRANCH AND BOUND SEBAGAI PENYELESAI PERMAINAN**

SKRIPSI

Oleh:

AHMAD BAIHAQI

NIM. 096502012

Telah Diperiksa dan Disetujui untuk Diuji

Tanggal : 4 Juli 2013

Dosen Pembimbing I

Dosen Pembimbing II

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

M. Imamuddin, Lc., M.A
NIP. 19740602 200901 1 010

Mengetahui
Ketua Jurusan Teknik Informatika

Ririen Kusumawati, M.Kom
NIP. 19720309 200501 2 002

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Ahmad Baihaqi

NIM : 09650212

Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika

Judul Penelitian : **APLIKASI PERMAINAN *SUDOKU SHARAF* MENGGUNAKAN *LINEAR CONGRUENTIAL GENERATOR* SEBAGAI PEMBANGKIT DAN *BRANCH AND BOUND* SEBAGAI PENYELESAI PERMAINAN**

Menyatakan dengan sebenarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat orang lain, kecuali yang secara tertulis dikutip dalam naskah ini atau disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur jiplakan maka saya bersedia untuk mempertanggung jawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 4 Juli 2013
Yang Membuat Pernyataan,

Ahmad Baihaqi
NIM. 09650212

MOTTO

-- Barangsiapa Bersungguh-sungguh Maka Dia Akan
Mendapatkannya--



PERSEMBAHAN



Buat:

*Mama, Ayah, Abang dan Kakakku.
yang selalu mendo'akan , Mendukungku dan menyayangiku.*

KATA PENGANTAR

Assalamu'alaikum Wr.Wb.

Syukur Alhamdulillah penulis haturkan kehadiran Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Ucapan terima kasih penulis sampaikan kepada:

1. Prof. Dr. H. Mudjia Raharjo, M.Si selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. drh. Hj. Bayyinatul Muchtaromah, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Ririen Kusumawati, M.Kom selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Hani Nurhayati, M.T selaku pembimbing dalam skripsi ini yang telah memberikan bimbingan dan pengarahan dalam proses penyelesaian skripsi ini.
5. M. Imamudin, Lc., MA selaku pembimbing dalam skripsi ini yang telah memberikan bimbingan dan pengarahan dalam proses penyelesaian skripsi ini.
6. A'la Syauqi, M.Kom selaku dosen wali memberikan bimbingan dan pengarahan dalam skripsi ini.

7. Seluruh Dosen Universitas Islam Negeri Maulana Malik Ibrahim Malang, khususnya dosen Teknik Informatika beserta seluruh staf yang telah memberikan ilmu dan membantu dalam penyelesaian skripsi ini.
8. Seluruh teman-teman Jurusan Teknik Informatika khususnya angkatan 2009.
9. Sahabat-sahabat penulis yang telah memotivasi dan membantu dalam proses penyelesaian skripsi ini.
10. Dan kepada seluruh pihak yang membantu penulisan skripsi ini, yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan. Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada pembaca dan khususnya bermanfaat bagi penulis secara pribadi.

Wassalamu'alaikum Wr. Wb.

Malang, 4 Juli 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
HALAMAN PERNYATAAN.....	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	4
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	4
1.5 Batasan Masalah	4
1.6 Metode Penelitian	5
1.7 Sistematika penyusunan.....	5
BAB II TINJAUAN PUSTAKA.....	7
2.1 Ilmu <i>Sharaf</i>	7
2.2 Permainan dan Jenisnya.....	9
2.3 Permainan <i>Sudoku</i>	13
2.4 <i>Sudoku Sharaf</i>	15
2.5 <i>Branch and Bound</i>	16
2.6 <i>Linear Congruential Generator</i>	17
2.7 Platform Android.....	19
BAB III ANALISA DAN PERANCANGAN SISTEM.....	22
3.1 Analisa dan Perancangan Sistem	22
3.1.1 Keterangan Umum	22
3.1.2 Rancangan Level Permainan.....	24
3.2 Rancangan Algoritma <i>Linear Congruential Generator</i> dan <i>Branch and Bound</i>	25
3.2.1 <i>Generator</i> Permainan	25
3.2.2 <i>Solver</i> Permainan	32
3.3 Perancangan Aplikasi.....	35
3.3.1 Antarmuka Permainan.....	35
3.3.2 Kebutuhan Sistem	42

BAB IV HASIL DAN PEMBAHASAN	43
4.1 Implementasi <i>Linear Congruential Generator</i> dan <i>Branch And Bound</i>	43
4.1.1 <i>Linear Congruential Generator</i>	43
4.1.2 <i>Branch And Bound</i>	44
4.1.3 Pembangkitan <i>Puzzle</i>	48
4.1.4 Penyelesaian <i>Puzzle</i>	52
4.2 Implementasi Aplikasi	53
4.3 Uji Coba	59
4.3.1 Uji Coba Algoritma <i>Linear Congruential Generator</i> dan <i>Branch And Bound</i> dalam Menyelesaikan <i>Puzzle Sudoku Sharaf</i>	59
4.3.2 Uji Coba Pengguna	67
4.4 Integrasi <i>Game Sudoku Sharaf</i> dengan Islam	71
BAB V KESIMPULAN DAN SARAN	74
5.1 Kesimpulan	74
5.2 Saran	74
DAFTAR PUSTAKA	
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

Gambar 2.1 Teki <i>Sudoku</i> dan Solusi Penyelesaiannya	14
Gambar 2.2 Contoh Pohon Status	17
Gambar 3.1 Perilaku <i>Player</i>	23
Gambar 3.2 <i>Finite State Machine</i> untuk <i>Player</i>	23
Gambar 3.3 Contoh <i>Puzzle</i> Setelah Terisi.....	28
Gambar 3.4 Contoh <i>Puzzle</i> <i>Sudoku</i> Hasil <i>Generate</i>	29
Gambar 3.5 <i>Flowchart</i> Pengisian <i>Puzzle</i> Kosong.....	30
Gambar 3.6 <i>Flowchart</i> Menentukan <i>Puzzle</i> Soal.....	31
Gambar 3.7 <i>Flowchart</i> Menentukan Banyaknya Angka Solusi.....	34
Gambar 3.8 <i>Flowchart</i> Algoritma <i>Branch And Bound</i>	34
Gambar 3.9 Rancangan Halaman Utama	35
Gambar 3.10 Rancangan Halaman Pilih Wazn.....	36
Gambar 3.11 Rancangan Halaman Pilih Fi'il	37
Gambar 3.12 Rancangan Halaman Pilih Shighah	37
Gambar 3.13 Rancangan Halaman Lihat Shighah.....	38
Gambar 3.14 Rancangan Halaman Pilih Level.....	38
Gambar 3.15 Rancangan Halaman <i>Puzzle</i>	39
Gambar 3.16 Rancangan Menu <i>Puzzle</i>	40
Gambar 3.17 <i>Flowchart</i> Permainan <i>Sudoku Sharaf</i>	41
Gambar 4.1 Halaman Utama.....	53
Gambar 4.2 Halaman <i>Sharaf</i>	54
Gambar 4.3 Halaman Petunjuk	54
Gambar 4.4 Halaman Pilih Wazn.....	55
Gambar 4.5 Halaman Pilih Fi'il	55
Gambar 4.6 Halaman Pilih Shighah.....	56
Gambar 4.7 Halaman Lihat Shighah.....	56
Gambar 4.8 Halaman Pilih Level.....	57
Gambar 4.9 Halaman <i>Puzzle</i>	57
Gambar 4.10 Halaman Shighah Pilihan	58
Gambar 4.11 Menu <i>Puzzle</i>	58
Gambar 4.12 Hasil Penyelesaian.....	59
Gambar 4.13 <i>Puzzle</i> pada level Mudah	60
Gambar 4.14 Solusi pada Level Mudah	61
Gambar 4.15 <i>Puzzle</i> pada Level Sedang.....	63
Gambar 4.16 Solusi pada Level Sedang.....	63
Gambar 4.17 <i>Puzzle</i> pada Level Sedang.....	65
Gambar 4.18 Solusi pada Level Sulit.....	65

DAFTAR TABEL

Tabel 2.1 <i>Tashrif 6 Wazn</i> dari <i>Tsulasi Mujarrad</i> Karya <i>Assyaikh Muhammad Ma'shum bin Ali</i>	9
Tabel 3.1 Jumlah Kotak Kosong yang Harus Diisi Pada Setiap Level.....	24
Tabel 3.2 <i>Tashrif 6 Wazn</i> dari <i>Tsulasi Mujarrad</i> Karya <i>Assyaikh Muhammad Ma'shum bin Ali</i>	25
Tabel 3.3 Jumlah Kombinasi.....	33
Tabel 4.1 Hasil Uji Coba <i>Puzzle</i> Level Mudah.....	61
Tabel 4.3 Hasil Uji Coba <i>Puzzle</i> Level Sedang.....	64
Tabel 4.3 Hasil Uji Coba <i>Puzzle</i> Level Sulit.....	66
Tabel 4.4 Rekapitulasi Hasil Uji Coba.....	68
Tabel 4.5 Prosentase Berdasarkan Parameter Penilaian.....	69

ABSTRAK

Baihaqi, Ahmad. 2013. **Aplikasi Permainan Sudoku Sharaf Menggunakan Linear Congruential Generator Sebagai Pembangkit dan Branch And Bound Sebagai Penyelesai Permainan**. Dosen Pembimbing: (1) Hani Nurhayati, MT. (2) M. Imamudin, Lc, MA.

Kata kunci: *Ilmu Sharaf, Linear Congruential Generator, Branch And Bound*

Al-Qur'an sebagai kitab suci umat islam diturunkan dengan bahasa arab. Tentunya sudah menjadi kewajiban bagi umat islam untuk mempelajari bahasa arab agar dapat memahami kandungan dalam Al-Qur'an. Dalam mempelajari bahasa arab tentunya banyak cabang ilmu yang harus dipelajari, salah satunya yaitu Ilmu *Sharaf*.

Berdasarkan latar belakang tersebut dilakukan penelitian untuk menciptakan aplikasi permainan Sudoku Sharaf dengan algoritma *Linear Congruential Generator* sebagai pembangkit dan *Branch And Bound* sebagai penyelesai permainan yang diharapkan dapat menjadi media pembelajaran Ilmu *Sharaf*.

Berdasarkan uji coba yang telah peneliti lakukan, hasil kuesioner dari 30 sample menunjukkan 84% responden menyatakan penilaian baik pada aplikasi ini dan membantu siswa dalam mempelajari Ilmu *Sharaf* khususnya pada *wazntsulasi mujarrad*.

ABSTRACT

Baihaqi, Ahmad. 2013. **An Application of Sharaf Sudoku Game Uses the Linear Congruential Generator for Generating and Branch And Bound for Solving the Game.** Advisor : (1) Hani Nurhayati, MT. (2) M. Imamudin, Lc, MA.

Al-Quran as a holy book of Islam revealed to the Arabic language. Surely it is obligatory for Muslims to learn Arabic in order to understand the content of the Qur'an. In studying the Arabic language course many disciplines that must be learned, one of which is Sharaf Science.

Based on this background, the research to create an application of Sharaf Sudoku Game uses the Linear Congruential Generator for generating and Branch And Bound for solving the game that is expected to be a medium of learning Sharaf science.

Based on research trials that have been done, the results of questionnaires of 30 samples showed 84% of respondents said good assessment on this application and assist students in studying Sharaf science especially in *wazn tsulasi mujarrad*.

Keywords : *Sharaf science, Linear Congruential Generator, Branch And Bound*

BAB I PENDAHULUAN

1.1 Latar Belakang

Allah SWT berfirman:

إِنَّا جَعَلْنَاهُ قُرْءَانًا عَرَبِيًّا لَعَلَّكُمْ تَعْقِلُونَ ﴿٣﴾

“*Sesungguhnya Kami menjadikan Al Quran dalam bahasa Arab supaya kamu memahaminya*” (QS. Az-Zukhruf: 3).

Ayat diatas menjelaskan secara jelas bahwa Al-Qur’an sebagai kitab suci umat Islam diturunkan menggunakan bahasa Arab agar manusia memahami isi Al-Qur’an. Konsekuensi dari penjelasan tersebut ialah kewajiban bagi umat Islam untuk mempelajari bahasa Arab agar dapat memahami kandungan dalam Al-Qur’an. Dalam mempelajari bahasa Arab tentunya banyak cabang ilmu yang harus dipelajari, salah satunya yaitu Ilmu *Sharaf*.

Sharaf secara bahasa memiliki arti perubahan kata. *Sharaf* secara istilah adalah perubahan bentuk kata dari bentuk yang satu ke yang lain. Jadi ilmu *sharaf* adalah ilmu yang mengetahui tentang berubahnya kalimat dari aslinya (Mufarakhah, 2011:16). Mempelajari ilmu *sharaf* akan membantu pembelajaran perubahan dalam kata bahasa Arab sehingga penggunaan bahasa Arab semakin beragam. Pembelajaran ilmu *sharaf* dapat dilakukan dengan berbagai cara, salah satunya pembelajaran melalui permainan.

Perkembangan teknologi yang melaju pesat saat ini telah menyentuh banyak kegiatan manusia, seperti pendidikan, pekerjaan dan tentunya permainan. Model permainan telah lama dikembangkan oleh para ahli komputer, karena dengan

sentuhan teknologi maka permainan tersebut dapat dimainkan dimanapun dan kapanpun. Banyak sekali permainan yang telah dikembangkan melalui proses digitalisasi komputer, semisal permainan kartu, catur, *game adventure* dan tak terkecuali permainan *Sudoku*.

Sudoku adalah sebuah permainan teka-teki angka berbasis logika yang pertama kali didesain oleh seorang arsitek berkebangsaan Amerika Serikat, Howard Garns, pada tahun 1979. Barulah kemudian permainan ini menjadi populer di Jepang. Kata *Sudoku* sendiri merupakan singkatan dari sebuah frase kalimat dalam bahasa Jepang, *Suji wa dokushin ni kagiru*, yang berarti setiap digit harus tetap satu jumlahnya (Syam, 2007).

Dalam definisi lain, *Sudoku* adalah salah satu permainan *puzzle* yang unik dan menggunakan angka acak yang telah diletakkan secara acak dalam sebuah kotak matrik. Kotak matrik dapat berupa matrik 4x4, 6x6, 9x9 dan seterusnya. Permainan *Sudoku* sendiri memiliki tingkat kerumitan dalam mengacak letak angka acak dalam kotak matrik serta penyelesaiannya permainan *Sudoku* yang cukup sulit (Abrar, 2012:3).

Penyelesaian permainan *Sudoku* dapat dilakukan dengan banyak algoritma, salah satunya yaitu *Branch And Bound* (percabangan dan pembatasan). Algoritma *Branch And Bound* pada umumnya hampir serupa dengan algoritma *Backtracking* (runut balik) yang telah banyak digunakan dalam penelitian untuk menyelesaikan permainan *Sudoku*. Perbedaan paling utamanya ada pada pembentukan pohon ruang status yang digunakan untuk mencari ruang solusi. Pada algoritma *Backtracking* ruang solusi dibangun secara dinamis berbasis Pencarian Mendalam

atau DFS (*Depth First Search*), sedangkan algoritma *Branch And Bound* membangun ruang solusi berdasarkan skema Pencarian Melebar atau BFS (*Breadth First Search*) (Syam, 2007).

Penelitian tentang penyelesaian *Sudoku* menggunakan algoritma *Branch And Bound* telah dilakukan oleh Simon H.S. dengan judul pencarian solusi *Sudoku* dengan algoritma berbasis *Branch And Bound* (Simon HS, 2007). Namun, pada penelitian tersebut tidak disebutkan berapa lama waktu yang diperoleh untuk menyelesaikan permainan *Sudoku* dan tidak menggunakan algoritma *Linear Congruential Generator* untuk pembangkit angka acaknya.

Untuk penelitian permainan *Sudoku* menggunakan bahasa Arab telah diteliti oleh Riyadlil Abrar (Abrar, 2012). Metode yang digunakan yaitu *Harmony Search* sebagai pembangkit dan penyelesai permainan. Dengan menggunakan metode *Harmony Search*, permainan *Sudoku* dengan jumlah kotak kosong 35-40 dapat diselesaikan dalam waktu tercepat antara 1-15 detik, dengan jumlah kotak kosong 40-45 dapat diselesaikan dalam waktu antara 2-12 detik, dan jumlah kotak kosong 45-50 dapat diselesaikan dalam waktu antara 5-35 detik. 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.

Untuk penelitian ini, penulis menggunakan algoritma *Linear Congruential Generator (LCG)* sebagai pembangkit dan *Branch and Bound* sebagai penyelesai permainan *Sudoku* yang akan difokuskan pada pembelajaran ilmu *sharaf*. *LCG* merupakan pembangkit angka acak yang sederhana, mudah dimengerti teorinya dan juga mudah untuk diimplementasikannya (Ramadhan, Tanpa Tahun:4). Hasil

dari proses pembangkitan angka acak akan ditampilkan dalam soal dan algoritma penyelesai (*solver*) digunakan ketika user tidak dapat menyelesaikan permainan.

1.2 Rumusan Masalah

Bagaimana menerapkan algoritma *Linear Congruential Generator* sebagai pembangkit dan *Branch and Bound* sebagai penyelesai permainan *Sudoku Sharaf* serta mengeluarkan suara sesuai dengan kata yang dipilih oleh *user*?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah membuktikan bahwa algoritma *Linear Congruential Generator* dan *Branch and Bound* mampu diterapkan sebagai pembangkit dan penyelesai permainan *Sudoku Sharaf* untuk pembelajaran *tashrif* berbasis *Android OS Mobile*.

1.4 Manfaat Penelitian

Manfaat pembuatan aplikasi permainan ini adalah membuat media pembelajaran tentang dasar-dasar ilmu *sharaf* yang lebih menarik dan menyenangkan sehingga lebih cepat dalam memahaminya.

1.5 Batasan Masalah

Batasan masalah pada penelitian ini sebagai berikut:

1. Aplikasi permainan ini berbasis *mobile*.
2. Permainan ini dimainkan *single player*.

3. Hanya pada pembelajaran bab I-VI dari *tsulasi mujarrad* pada Kitab *Al-Amsilatut Tashrifiyah*.
4. *Shighah* digunakan untuk menggantikan angka.
5. *Shighah* yang digunakan berjumlah 11.
6. Dimensi kotak *game Sudoku* adalah 9 x 9.
7. *Linear Congruential Generator* digunakan sebagai pembangkit dan *Branch and Bound* sebagai penyelesai permainan *Sudoku*.
8. Permainan *Sudoku* diimplementasikan pada *Android OS Mobile* dengan ukuran minimal *Display* 480x800 pixels, 4.3 inches.
9. Aplikasi ini dapat dimainkan oleh siswa-siswi *Madrasah Tsanawiyah* mulai kelas VIII.

1.6 Metode Penelitian

Untuk mencapai tujuan yang telah ditentukan sebelumnya, maka metodologi pengumpulan data yang dilakukan dalam penulisan skripsi ini adalah *library research* yaitu suatu cara penelitian dan pengumpulan data teoritis dari buku-buku, artikel, jurnal dan berbagai literatur yang mendukung penyusunan skripsi. Pengumpulan data juga dilakukan dengan melakukan uji coba aplikasi pada pengguna yang disertai dengan angket.

1.7 Sistematika Penyusunan

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

BAB I Pendahuluan

Bab ini membahas tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penyusunan.

Bab II Landasan Teori

Landasan teori menjelaskan beberapa teori yang menjadi dasar penyusunan tugas akhir ini. Teori-teori yang dibahas dalam bab ini adalah teori yang berkaitan dengan pembahasan tentang ilmu *Sharaf*, permainan dan jenisnya, permainan *Sudoku*, *Sudoku Sharaf*, *Branch And Bound*, *Linear Congruential Generator* dan *Android OS*.

BAB III Analisa dan Perancangan Sistem

Bab ini menjelaskan analisa dari kebutuhan sistem untuk membuat *game* yang meliputi spesifikasi kebutuhan software dan langkah-langkah pembuatan *Sudoku Sharaf*.

BAB IV Hasil dan Pembahasan

Bab ini menjelaskan hasil pengujian *Sudoku Sharaf* yang telah diterapkan dalam *game*.

BAB V Penutup

Bab ini berisi kesimpulan dan saran.

BAB II TINJAUAN PUSTAKA

2.1 Ilmu *Sharaf*

Sharaf asal artinya memalingkan tetapi yang terkenal didalam urusan bahasa Arab ialah satu ilmu yang menerangkan hal memalingkan satu kalimat kepada beberapa huruf dan sifat untuk menghasilkan beberapa makna seperti *nasara-yansuru-unsur* dan seterusnya. Dan ilmu ini ada juga yang menamakan dengan ilmu *tashrif* yakni ilmu yang memalingkan satu kalimat kepada beberapa huruf (Hasan, 2010:2). Menurut Muhammad Idris Djauhari (2012:1), *sharaf* secara bahasa bermakna mengulangi atau melindungi, sedangkan ilmu *sharaf* adalah ilmu yang membahas bentuk kalimat bahasa Arab dan keadaannya yang bukan termasuk *i'rab* dan *bina'*. Jadi ilmu *sharaf* dapat didefinisikan sebagai ilmu yang mempelajari perubahan dari satu *shighah* ke *shighah* lain. *Shighah* menurut bahasa adalah bentuk atau macam, sedangkan menurut istilah adalah perubahan bentuk kalimat bahasa Arab dalam bentuk khusus untuk makna yang diinginkan atau dimaksudkan (Idris, 2001:6).

Shighah yang terdapat dalam *Fi'il Tsulatsi Mujarrad* berjumlah 12 (Mufarakhah, 2011:17). Yaitu:

1. *Fi'il Mâdhi* (فعل ماضى) adalah *kalimah* yang menunjukkan arti kata kerja yang terikat oleh waktu lampau (Farida, 2008:1).
2. *Fi'il Mudhâri'* (فعل مضارع) adalah *kalimah* yang menunjukkan arti kata kata kerja yang terikat oleh waktu sedang atau akan dilaksanakannya pekerjaan itu (Farida, 2008:1).

3. *Isim Mashdar Ghairu Mîm* (اسم مصدر غير ميم) adalah tiap kalimat *isim* yang dibaca *nashab* dan tidak didahului huruf mim (Mufarakhah, 2011:17).
4. *Isim Mashdar Mîm* (اسم مصدر ميم) adalah adalah tiap kalimat *isim* yang dibaca *nashab* dan didahului huruf mim (Mufarakhah, 2011:17).
5. *Isim Dhamîr* (اسم ضمير) adalah kata ganti orang ke 1,2 atau 3 (Mufarakhah, 2011:17).
6. *Isim Fa'il* (اسم الفاعل) adalah kata yang menunjukkan orang yang melakukan pekerjaan.
7. *Isim Isyârah* (اسم اشارة) adalah kata yang mengisyaratkan sesuatu.
8. *Isim Maf'uâl* (اسم المفعول) adalah kata yang menunjukkan sesuatu yang menjadi objek pekerjaan.
9. *Fi'il Amar* (فعل الامر) adalah *kalimah* yang menunjukkan arti perintah dilaksanakannya suatu pekerjaan di waktu yang akan datang (Farida, 2008:2).
10. *Fi'il Nâhî* (فعل نهى) adalah kata *kalimah* yang menunjukkan arti larangan melakukan pekerjaan di waktu yang akan datang (Farida, 2008:2)..
11. *Isim Zamân Makân* (اسم الزما والمكان) adalah kata yang menunjukkan waktu dan tempat terjadinya perbuatan.
12. *Isim Âlat* (اسم الآلة) adalah kata yang menunjukkan alat untuk mengerjakan sesuatu.

Selain terdapat beberapa *shighah*, dalam ilmu *sharaf* juga terdapat *wazn* (وزن). *Wazn* ialah suatu bentuk kata yang menjadi patokan bagi kata-kata yang lain. *Wazn* yang terdapat pada *tsulasi mujarrad* ada 6 *wazn* yang setiap *wazn* mempunyai bentuk *tashrif* tersendiri, yaitu:

Tabel 2.1 *Tashrif 6 Wazn* dari *Tsulasi Mujarrad* Karya Assyaikh Muhammad Ma'shum bin Ali

التصريف الاصطلاحي	وزن	رقم
فَعْلٌ - يَفْعُلٌ - فَعْلًا - مَفْعَلًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعِلٌ - مَفْعَلٌ - مَفْعَلٌ - مَفْعَلٌ	فَعْلٌ - يَفْعُلٌ	١
فَعْلٌ - يَفْعِلٌ - فَعْلًا - مَفْعَلًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ	فَعْلٌ - يَفْعِلٌ	٢
فَعْلٌ - يَفْعِلٌ - فَعْلًا - مَفْعَلًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ	فَعْلٌ - يَفْعِلٌ	٣
فَعْلٌ - يَفْعِلٌ - فَعْلًا - مَفْعَلًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ	فَعْلٌ - يَفْعِلٌ	٤
فَعْلٌ - يَفْعِلٌ - فَعْلًا - مَفْعَلًا - فَعْلٌ - أَفْعَلٌ - لَاتَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ	فَعْلٌ - يَفْعِلٌ	٥
فَعْلٌ - يَفْعِلٌ - فَعْلَانًا - مَفْعَلًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ - مَفْعِلٌ	فَعْلٌ - يَفْعِلٌ	٦

2.2 Permainan dan Jenisnya

Dalam Al-Qur'an telah dijelaskan bahwa kehidupan dunia ini hanyalah permainan dan senda gurau. Manusia menjadi aktor dalam permainan tersebut. Hal tersebut dijelaskan Al-Qur'an surat Muhammad ayat 36 yang berbunyi:

إِنَّمَا الْحَيَاةُ الدُّنْيَا لَعِبٌ وَلَهْوٌ وَإِنْ تُؤْمِنُوا وَتَتَّقُوا يُؤْتِكُمْ أَجْرَكُمْ وَلَا

يَسْأَلُكُمْ أَمْوَالَكُمْ ﴿٣٦﴾

Artinya:

“Sesungguhnya kehidupan dunia hanyalah permainan dan senda gurau. Dan jika kamu beriman dan bertakwa, Allah akan memberikan pahala kepadamu dan Dia tidak akan meminta harta-hartamu.” (QS Muhammad:36).

Al-Imam Jalaluddin Muhammad Al-Mahalli dan Al-Imam Jalaluddin Abdurrahman As-Suyuthi dalam Tafsir Jalalain (2010: 250) menafsirkan Q.S. Muhammad ayat 36 dengan berkata, ” إِنَّمَا الْحَيَاةُ ”, yakni menyibukkan diri dengannya, لَعِبٌ وَ لَهُوَ وَإِنْ تُؤْمِنُوا وَتَتَّقُوا, Dan jika kamu beriman serta bertakwa, yakni kepada Allah SWT. Karena semua ini termasuk perkara Akhirat. يُؤْتِكُمْ أَجُورَكُمْ وَ لَا يَسْأَلْكُمْ أَمْوَالَكُمْ , yakni semuanya. Kecuali zakat yang diwajibkan kepada kalian dalam harta tersebut ”.

M. Quraish Shihab dalam Tafsir Al-Mishbah (2003:160) menafsirkan Q.S. Muhammad ayat 36 dengan berkata, ” إِنَّمَا الْحَيَاةُ الدُّنْيَا ”, bagi orang-orang yang lengah, لَعِبٌ, yakni kegiatan tanpa tujuan yang benar , وَ لَهُوَ, dan senda gurau yang mengantarkan kepada kelengahan sehingga mereka meninggalkan yang penting atau yang lebih penting, berbeda dengan kehidupan akhirat bagi yang sadar dan mengingat Allah”.

Berdasarkan penafsiran M. Quraish Shihab, penulis menyimpulkan bahwa kehidupan ini hanya akan menjadi permainan dan senda gurau belaka apabila manusia lengah karena menjalani kehidupan ini tanpa tujuan yang jelas. Sehingga permainan dunia ini akan menjadi bermanfaat bagi manusia apabila dijalani dengan adanya tujuan yang benar, dalam hal ini kehidupan akhirat.

Game merupakan kata dalam bahasa Inggris yang berarti permainan. Permainan adalah sesuatu yang dapat dimainkan dengan aturan tertentu sehingga ada yang menang dan ada yang kalah. Permainan merupakan bagian dari bermain dan bermain merupakan bagian dari permainan, keduanya saling

berhubungan. Sebuah permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan yang terjadi dalam permainan yang merupakan sebuah rekayasa. *Game* adalah program komputer sangat kompleks yang merangsang otak untuk melakukan serangkaian tugas kognitif dan menghasilkan tingkat pemikiran yang lebih tinggi (Beck dan Wade, 2006).

Game adalah media untuk melakukan aktifitas bermain. Aktifitas bermain merupakan suatu aktifitas yang meliputi pemecahan masalah yang menjadi tantangan dari *game* tersebut, dengan mengikuti suatu aturan tertentu. *Game* menjadi menarik karena tantangan dan aturan pada *game* dikemas dalam suatu skenario tertentu. Dari satu sisi, aktifitas bermain *game* dipandang sebagai suatu aktifitas yang tidak produktif. Namun, disisi lain bermain *game* dapat dipandang sebagai sebuah aktifitas belajar. Hal ini terjadi karena pemain dituntut untuk mempelajari cara-cara yang harus dilakukan untuk menaklukkan tantangan yang diberikan. Dengan demikian, dengan memasukkan konten pembelajaran didalamnya, *game* dapat digunakan sebagai sebuah sistem instruksional (Mahtarami, Ifansyah, 2010).

Menurut para pakar psikologi, permainan merupakan sebuah metode yang baik digunakan untuk belajar. Melalui permainan tercipta suatu suasana santai dan menyenangkan, sehingga anak dapat belajar lebih baik dan bersungguh-sungguh. Selain itu, sudah terbukti bahwa tingkah laku seseorang dalam permainan sama dengan tingkah lakunya dalam kehidupan sehari-hari. Misalnya, cara mengambil keputusan, memecahkan masalah, merencanakan sesuatu dan berkomunikasi (Zaman, 2009:6).

Pada dasarnya bermain memiliki dua pengertian yang harus dibedakan. Bermain menurut pengertian pertama dapat bermakna sebagai sebuah aktivitas bermain yang murni mencari kesenangan tanpa mencari “menang-kalah” (*play*). Sedangkan yang kedua disebut sebagai aktivitas bermain yang dilakukan dalam rangka mencari kesenangan dan kepuasan, namun ditandai dengan adanya pencarian “menang-kalah” (*games*) (Ismail, 2006:15).

Jenis-jenis *game* menurut Lindsay Grace dapat diklasifikasikan menjadi beberapa jenis, antara lain:

1. *Action Game*

Game jenis ini adalah *game* yang menawarkan aksi sebagai daya tarik utama. Respon merupakan keterampilan utama yang dibutuhkan dalam memainkan *game* ini. Contoh dari *game* ini adalah Metal Gear.

2. *Adventure Game*

Game jenis ini adalah *game* dengan mengambil konsep petualangan. Pemain berjalan menuju ke suatu tempat dan disepanjang perjalanan pemain menemukan hal-hal baru untuk dieksplorasi. Sebagai contohnya adalah *game* Crash Bandicoot.

3. *Puzzle Game*

Jenis *game* ini memberikan tantangan kepada pemainnya dengan cara memecahkan masalah yang terdapat pada *game* yang dimainkan. Teka-teki merupakan daya tarik utama dalam *game* ini. Contoh dari jenis *game* ini adalah jigsaw.

4. *Role Playing Game*

Game bertipe ini ini lebih mengutamakan cerita dan biasanya pemain diajak masuk ke dalam cerita tersebut untuk menyelesaikan sebuah misi. Dalam *game* ini pemain akan berperan menjadi sebuah karakter dengan berbagai atribut, seperti kesehatan, intelegensi, kekuatan dan keahlian. Salah satu *game* yang terkenal dengan RPG adalah Final Fantasy.

5. *Simulation Game*

Game ini merupakan *game* yang mengambil simulasi seperti keadaan nyata. Dalam beberapa jenis *game* ini pemain diajak untuk menciptakan lingkungan yang diinginkan, seperti membangun kota. Contoh *game* yang populer dari jenis ini adalah The Sims.

6. *Strategy Game*

Merupakan jenis *game* yang melibatkan strategi dan logika melalui pemecahan masalah. Contoh dari *game* ini adalah game Civilization.

Dari penjelasan jenis-jenis *game* berdasarkan ciri-cirinya diatas, maka permainan *Sudoku* termasuk pada jenis permainan *puzzle game*.

2.3 Permainan *Sudoku*

Sudoku atau *SU Doku* tidak lain *puzzle* atau teka-teki yang didasarkan pada konsep *Latin Square* dimana konsep ini diperkenalkan pada tahun 1783 oleh Leonhard Euler, seorang matematikawan asal Swiss. Namun *puzzle* ini memang

berbahasa Jepang, tetapi asalnya bukanlah dari sana. Permainan ini tadinya bernama *Number Place* dan pertama kali dipublikasikan pada tahun 1979 dalam sebuah majalah khusus *puzzle*, *Dell Puzzle Magazines*, di Manhattan, New York dan terus berkembang sampai sekarang. Adalah Howard Garns, seorang mantan arsitek, yang meninggal pada tahun 1989, dianggap sebagai orang yang menciptakan permainan ini ketika dia berusia 74 tahun (Omimura, 2009;1).

Dalam literatur lain disebutkan bahwa permainan *Sudoku* 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 (Sari, 2007:2).

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

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

Gambar 2.1 Teka-Teki *Sudoku* dan Solusi Penyelesaiannya (Sumber: 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 (Abrar, 2012:12). Aturan dalam permainan *Sudoku* cukup sederhana, yaitu isi penuh semua matriks, tetapi untuk setiap kolom, baris, maupun submatriks berukuran 3 x 3 ($n \times n$) hanya terisi 1x dengan angka 1 sampai 9.

2.4 *Sudoku Sharaf*

Sudoku Sharaf adalah permainan *Sudoku* yang dikembangkan dengan penambahan konsep ilmu *sharaf*. Konsep yang digunakan tidak lagi berupa angka yang digunakan untuk menjadi variable permainan, tetapi *shighah-shighah* dalam ilmu *sharaf* yang akan digunakan. Jadi, ketika pemain ingin mengisi kotak kosong dalam *puzzle Sudoku*, maka pemain akan memilih *shighah* yang sesuai untuk diisi di kotak kosong tersebut.

Mengenai aturan permainan, *Sudoku Sharaf* tetap menggunakan aturan dalam permainan *Sudoku* yaitu isi penuh semua matriks, tetapi untuk setiap kolom, baris, maupun submatriks berukuran 3 x 3 ($n \times n$) hanya terisi 1x dengan *shighah* yang sama. Oleh karena itu, penulis menyimpulkan bahwa permainan *Sudoku Sharaf* merupakan pengembangan dari permainan *Sudoku* karena dari segi peraturan yang merupakan inti permainan masih menggunakan aturan permainan *Sudoku*.

2.5 *Branch and Bound*

Branch and Bound untuk memecahkan pemrograman linier bulat (ILP) telah dikembangkan melalui Land and Doig (1960). Metode, yang secara langsung dihubungkan dengan simpleks metode untuk pemrograman linier(LP), kemudian dimodifikasi oleh Dakin (1965) dan telah dengan sukses menerapkan di dalam kitab undang-undang hukum dagang banyak orang untuk memecahkan permasalahan ILP (Sidabutar, 2008:29).

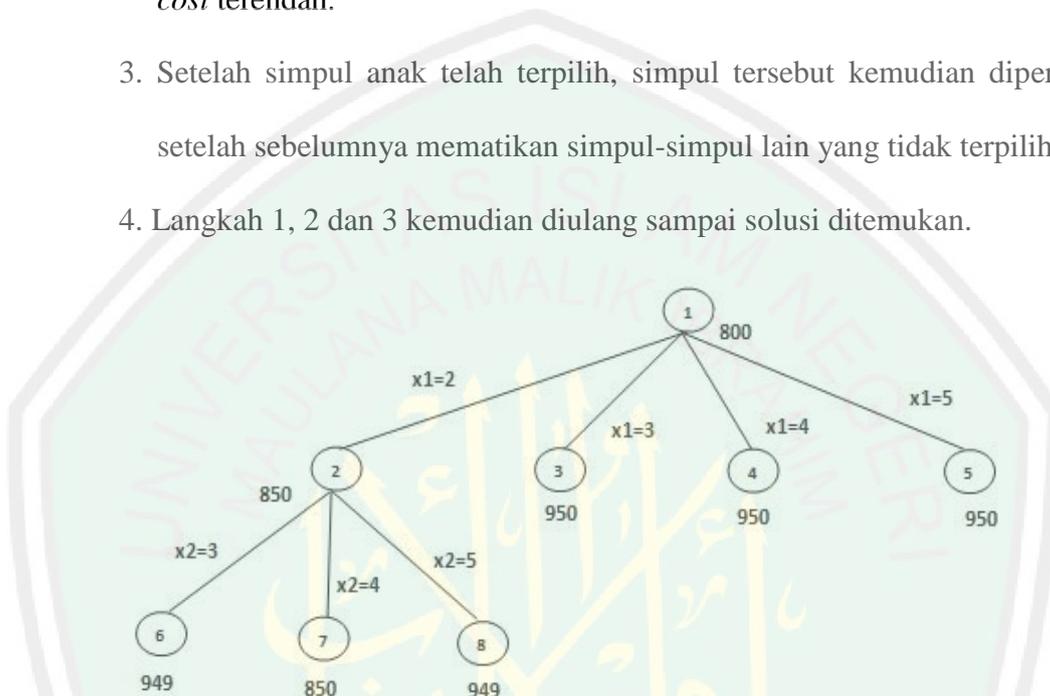
Dalam definisi lain, Algoritma *Branch and Bound* adalah metode algoritma umum untuk mencari solusi optimal dari berbagai permasalahan optimasi, terutama untuk optimasi diskrit dan kombinatorial. Sebagaimana pada algoritma runut-balik, algoritma *Branch and Bound* juga merupakan metode pencarian di dalam ruang solusi secara sistematis (Aprilia, 2007:1).

Pada algoritma *Branch and Bound*, proses pencarian ke simpul solusi bisa dipercepat dengan menentukan simpul yang hidup berdasarkan nilai *cost* (ongkos). Nilai *cost* setiap simpul yang hidup menyatakan nilai batas yang pada umumnya merupakan nilai perkiraan atau taksiran.

Adapun langkah-langkah pembangunan solusi algoritma *Branch And Bound* ini umumnya adalah sebagai berikut (Simon H S, 2007:2):

1. Solusi dicari dengan cara membangkitkan setiap simpul anak yang mungkin terjadi dengan aturan seperti algoritma *BFS* pada umumnya. Namun, pada saat pembangkitan simpul, *cost* untuk masing-masing simpul juga dihitung.

2. Dari tiap simpul anak yang telah dibangkitkan, akan dipilih 1 buah simpul yang dianggap simpul yang paling mendekati solusi, yaitu simpul dengan *cost* terendah.
3. Setelah simpul anak telah terpilih, simpul tersebut kemudian diperluas setelah sebelumnya mematikan simpul-simpul lain yang tidak terpilih.
4. Langkah 1, 2 dan 3 kemudian diulang sampai solusi ditemukan.



Gambar 2.2 Contoh Pohon Status (Sumber: Sumitro, 2007).

2.6 Linear Congruential Generator

Dalam permainan *Sudoku*, pembangkit bilangan secara acak sebagai soal yang akan ditampilkan sangatlah penting untuk menentukan tingkat kesulitan permainan tersebut. Oleh karena itu dibutuhkan sebuah *generator* (pembangkit) yang mampu membangkitkan bilangan acak dan dalam penelitian ini akan menggunakan *Linear Congruential 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).

Pseudo Number Random Generator (PNRG) atau dalam Bahasa Indonesia pembangkit bilangan acak semu adalah sebuah algoritma yang membangkitkan sebuah deret bilangan yang tidak benar-benar acak. Keluaran dari pembangkit bilangan acak semu hanya mendekati beberapa dari sifat-sifat yang dimiliki bilangan acak (Ramadhan, Tanpa Tahun:1).

Linear Congruential Generator atau jika diubah kedalam Bahasa Indonesia menjadi pembangkit bilangan acak kongruen-lanjar merupakan pembangkit bilangan acak yang sederhana, mudah dimengerti teorinya dan juga mudah untuk diimplementasikan (Ramadhan, Tanpa Tahun:4).

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

Pseudo RNG dari algoritma *Linear Congruential Generator*, 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).

2.7 Platform Android

Salah satu sistem operasi yang terkenal dikalangan pengguna *smartphone* adalah Android. Sistem operasi ini mempunyai nilai lebih yang paling mendasar dibandingkan dengan sistem operasi lainnya seperti Windows Phone, iOS, BlackBerry, MeeGo, Bada dan Symbian yaitu karena bersifat *open source*.

Menurut Safaat (2011:1), Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android diakuisisi oleh *Google* pada Juli 2005 dan baru dirilis perdana pada 5 November 2007. Android berlisensi di bawah *GNU, General Public Lisensi Versi2 (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 menjadikan kernel Linux 2.6 sebagai dasar dari platformnya, dengan beberapa alasan, antara lain (Tim EMS, 2012:2):

1. *Security, kernel* Linux ini dapat mengukur keamanan antara *system* dan aplikasi.

2. Manajemen memori, *kernel* Linux ini dapat mengatur manajemen memori sehingga lebih hemat ketika melakukan *develop* aplikasi.
3. Manajemen proses, *kernel* Linux ini dapat mengatur proses lebih baik, mengatur *resource* yang diperlukan sesuai dengan kebutuhan untuk menjalankan aplikasi.
4. *Network stack*, *kernel* Linux ini dapat mengatur komunikasi jaringan.
5. *Driver*, *kernel* Linux ini menjamin sesuatu dapat berjalan dengan baik. Berbagai manufaktur *hardware* akan dapat bekerja dengan *kernel* ini.

Perkembangan Android yang cepat telah merilis beberapa versi. Android versi 4.1 Jelly Bean merupakan versi terbaru dari Android saat ini yang dirilis pada tanggal 27 Juni 2012 bersamaan dengan dirilisnya sistem operasi *mobile* iOS versi 6. Tampilan versi terbaru ini terlihat lebih *smooth* dari Android versi sebelumnya serta fitur *multi-tasking* juga berjalan dengan lebih cepat dan lancar di Android Jelly Bean. Berikut ini versi Android yang telah dirilis:

1. Android 1.0, dirilis tanggal 23 September 2008
2. Android 1.1, dirilis tanggal 9 Februari 2009
3. Android 1.5, dirilis tanggal 30 April 2009 dengan nama Cupcake.
4. Android 1.6, dirilis tanggal 15 September 2009 dengan nama Donut.
5. Android 2.0, dirilis tanggal 26 Oktober 2009 dengan nama Eclair.
6. Android 2.0.1, dirilis tanggal 3 Desember 2009 dengan nama Eclair.
7. Android 2.1, dirilis tanggal 12 Januari 2010 dengan nama Eclair.
8. Android 2.2, dirilis tanggal 20 Mei 2010 dengan nama Froyo.

9. Android 2.3, dirilis tanggal 6 Desember 2010 dengan nama
Gingerbread.
10. Android 2.3.3, dirilis tanggal 22 Februari 2011 dengan nama
Gingerbread.
11. Android 3.0, dirilis tanggal 22 Februari 2011 dengan nama Honeycomb.
12. Android 3.1, dirilis tanggal 10 Mei 2011 dengan nama Honeycomb.
13. Android 3.2, dirilis tanggal 18 Juli 2011 dengan nama Honeycomb.
14. Android 4.0 dirilis tanggal 19 Oktober 2011 dengan nama Ice Cream
Sandwich.
15. Android 4.1 dirilis tanggal 27 Juni 2012 dengan nama Jelly Bean.

Beberapa keunggulan Platform Android adalah sebagai berikut (Safaat, 2011:3):

- 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 SISTEM

3.1 Analisa dan Perancangan Sistem

Penelitian ini akan membuat sebuah permainan *Sudoku* yang didasari dari pembelajaran ilmu *sharaf*. Permainan *Sudoku* merupakan permainan logika dalam penempatan angka sesuai dengan aturannya. Dalam penempatan angka di kotak-kotak *Sudoku*, pemain diharuskan mengisi semua kotak dengan syarat tidak ada angka yang sama pada setiap kolom, baris dan area yang ditentukan. Angka yang menjadi bahan permainan yaitu angka 1 sampai 9. Namun pada penelitian ini, angka-angka tersebut akan diganti dengan *shighah-shighah* dari *wazn tsulasi mujarrad* sebagai dasar dalam pembelajaran ilmu *sharaf*.

Permainan *Sudoku* yang akan dirancang mempunyai desain sebagai berikut:

3.1.1 Keterangan Umum

Permainan *Sudoku* ini akan dikembangkan dengan tambahan unsur pendidikan, terutama pendidikan dasar ilmu *sharaf*. Pemain akan dijelaskan tentang pembagian *wazn* pada *fi'il tsulasi mujarrad* yang akan menjadi bahan dalam permainan *Sudoku* ini. Jadi, pemain tidak hanya mengisi setiap kotak *Sudoku* dengan *shighah* berdasarkan *wazn*-nya, tetapi juga akan mengetahui teori dasar dari pembagian *wazn* dalam *fi'il tsulasi mujarrad*.

Untuk penelitian ini, skenario *game* akan dibuat untuk 1 karakter saja karena *game* yang akan dibuat merupakan *game Sudoku* yang hanya mempunyai karakter sebagai pemain (*player*). Pemain mempunyai perilaku menjawab sekaligus

variabel yang mempengaruhi perubahan perilaku seperti pada gambar 3.1 dibawah.

Skenario ini tentang respon *player* yang akan membaca soal *Sudoku* dan menjawabnya sampai semua soal terjawab dengan benar.

Karakter	Variabel Input Perilaku	Output Perilaku
Player	Soal dan Pilihan Jawaban	Membaca soal dan menjawab

Gambar 3.1 Perilaku Player

Untuk menampilkan desain FSM (*Finite State Machine*) untuk *player*, penulis buat seperti gambar 3.2 dibawah. *State* utama yang tersusun dalam FSM dapat dijelaskan sebagai berikut:

1. Mulai

Player mendapatkan soal *Sudoku*.

2. Membaca soal

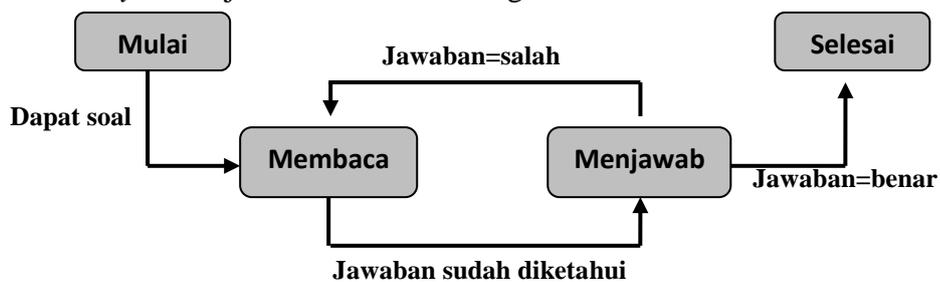
Player membaca soal *Sudoku* pada *puzzle*.

3. Menjawab soal

Player memilih 1 dari 9 kemungkinan jawaban.

4. Selesai

Player menjawab semua soal dengan benar.



Gambar 3.2 *Finite State Machine* untuk *Player*

Dalam permainan *Sudoku* ini, pemain juga dapat mendengarkan pelafalan *shighah* ketika memilih *shighah* yang akan ditempatkan pada kotak kosong dalam *puzzle*. Sehingga pemain diharapkan lebih tertarik dan terhibur dalam menyelesaikan *puzzle Sudoku*.

Permainan *Sudoku* ini akan dikembangkan pada platform Android yang tentunya berbasis *mobile* sehingga akan menambah ketertarikan dalam bermain *Sudoku*, karena untuk menyelesaikan permainan *Sudoku* terkadang membutuhkan waktu yang lama sehingga dibutuhkan perangkat *mobile* yang dapat dibawa kemanapun dan kapanpun.

Dalam penelitian ini, objek penelitiannya ialah penggunaan algoritma *Linear Generator Congruential* sebagai pembangkit permainan agar *puzzle* menjadi dinamis pada setiap level dan *Branch And Bound* sebagai penyelesaian permainan.

3.1.2 Rancangan Level Permainan

Permainan ini akan dibagi pada 3 level seperti pada tabel 3.1. Perbedaan pada setiap level didasari pada banyaknya kotak kosong yang harus diisi oleh pemain. Semakin tinggi levelnya, maka semakin banyak kotak kosong yang harus diisi.

Tabel 3.1 Jumlah Kotak Kosong yang Harus Diisi pada Setiap Level

No	Level	Jumlah kotak Kosong
1	Mudah	36 atau 4 per area 3x3
2	Sedang	45 atau 5 per area 3x3
3	Sulit	54 atau 6 per area 3x3

Untuk level mudah, kotak kosong yang harus diisi *shighah* ada 36. Pada level sedang ada 45 kotak kosong, sedangkan pada level sulit terdapat 54 kotak kosong yang harus diisi.

Selain pengaturan jumlah kotak kosong pada setiap level, permainan ini juga mengharuskan pemain untuk memilih *wazn* yang akan dimainkan. Dengan memilih *wazn* terlebih dahulu, maka *shighah* yang akan dipilih untuk mengisi kotak kosong juga akan disesuaikan berdasarkan *wazn*-nya.

Tabel 3.2 *Tashrif 6 Wazn* dari *Tsulasi Mujarrad Karya Assyaikh Muhammad Ma'shum bin Ali*

التصريف الاصطلاحي	وزن	رقم
فَعْلٌ - يَفْعُلُ - فَعْلًا - مَفْعَلًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعُلُ - مَفْعَلٌ - مَفْعَلٌ - مَفْعَلٌ	فَعْلٌ - يَفْعُلُ	١
فَعْلٌ - يَفْعُلُ - فَعْلًا - مَفْعَلًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعُلُ - مَفْعَلٌ - مَفْعَلٌ - مَفْعَلٌ	فَعْلٌ - يَفْعُلُ	٢
فَعْلٌ - يَفْعُلُ - فَعْلًا - مَفْعَلًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعُلُ - مَفْعَلٌ - مَفْعَلٌ - مَفْعَلٌ	فَعْلٌ - يَفْعُلُ	٣
فَعْلٌ - يَفْعُلُ - فَعْلًا - مَفْعَلًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعُلُ - مَفْعَلٌ - مَفْعَلٌ	فَعْلٌ - يَفْعُلُ	٤
فَعْلٌ - يَفْعُلُ - فَعْلًا - مَفْعَلًا - فَعْلٌ - أَفْعَلٌ - لَاتَفْعُلُ - مَفْعَلٌ - مَفْعَلٌ	فَعْلٌ - يَفْعُلُ	٥
فَعْلٌ - يَفْعُلُ - فَعْلَانًا - مَفْعَلَانًا - فَاعِلٌ - مَفْعُولٌ - أَفْعَلٌ - لَاتَفْعُلُ - مَفْعَلٌ - مَفْعَلٌ	فَعْلٌ - يَفْعُلُ	٦

3.2 Rancangan Algoritma *Linear Congruential Generator* dan *Branch and Bound*

Dalam permainan *Sudoku* ini, algoritma *Linear Congruential Generator* digunakan sebagai pembangkit permainan (*generator*), sedangkan algoritma *Branch And Bound* digunakan sebagai penyelesai permainan (*solver*).

3.2.1 *Generator* Permainan

- A. Mengisi semua kotak kosong dengan *shighah* menggunakan algoritma *Linear Congruential Generator*.

B. Menghilangkan beberapa kotak *puzzle* berdasarkan level menggunakan algoritma *Linear Congruential Generator*.

C. Menampilkan soal hasil *generate*.

Proses awal ketika mengisi semua kotak kosong pada *puzzle* dengan *shighah* ialah mengisi sebuah *arraylist* dengan angka-angka terlebih dahulu. Karena gambar *shighah* yang tampil diambil dari angka yang telah ada pada *arraylist* tersebut. Pengisian angka pada *arraylist* dilakukan menggunakan algoritma *Linear Congruential Generator*. Untuk menghasilkan output dari algoritma tersebut, maka pemberian nilai dari variabel a , c , m dan Z_0 yang merupakan bagian dari algoritma *Linear Congruential Generator* harus dilakukan secara *trial and error*. Penulis telah mencoba beberapa kali pemberian nilai pada variabel a , c , m , Z_0 dan akhirnya penulis mendapatkan angka-angka yang cocok. Berikut ini hasil dari uji coba:

Input:

$a=7, c=6, m=11$ dan $Z_0=1$

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

Output= 2

Jumlah output dari algoritma *Linear Congruential Generator* ditentukan sesuai keinginan. Untuk proses ini, penulis hanya membutuhkan output 1 angka karena hanya akan mengambil 1 dari beberapa pilihan kumpulan angka yang valid. Kumpulan angka tersebut disimpan pada *arraylist* yang dimulai dari indeks 1 sampai 80 sesuai dengan banyaknya kotak pada *puzzle*. Berikut struktur dari *arraylist* yang menyimpan kumpulan angka valid untuk *puzzle*:

Indeks	Angka
0	9
1	7
2	6
3	4
4	8
5	5
6	3
7	2
8	1
.....	...
80	3

Isi dari *arraylist* dipindahkan ke dalam *array* 4 dimensi untuk memudahkan dalam mengecek indeks dari setiap kotak kosong. Berikut struktur dari *array* 4 dimensinya:

Indeks	Angka
0,0,0,0	9
0,0,0,1	7
0,0,0,2	6
0,0,0,3	4
0,0,0,4	8
0,0,0,5	5
0,0,0,6	3
0,0,0,7	2
0,0,0,8	1
.....	...
2,2,2,2	3

Setelah *array* 4 dimensi telah terisi oleh kumpulan angka yang valid, maka *puzzle* telah terisi semua dengan *shighah* yang diambil berdasarkan angka pada *array* 4 dimensi tersebut.

فاعِل	فَعِلا	مفعِلا	مفعول	أفَعِل	لا تفعِل	مفعِل	فَعِل	يفَعِل
مفعول	أفَعِل	يفَعِل	فَعِل	مفعِل	فاعِل	فَعِلا	مفعِلا	لا تفعِل
فَعِل	مفعِل	لا تفعِل	فَعِلا	مفعِلا	يفَعِل	فاعِل	مفعول	أفَعِل
لا تفعِل	فاعِل	مفعِل	أفَعِل	مفعول	فَعِل	مفعِلا	يفَعِل	فَعِلا
مفعِلا	يفَعِل	مفعول	لا تفعِل	فاعِل	فَعِلا	أفَعِل	مفعِل	فَعِل
أفَعِل	فَعِل	فَعِلا	مفعِل	يفَعِل	مفعِل	لا تفعِل	فاعِل	مفعول
مفعِل	مفعول	فَعِل	فاعِل	فَعِلا	أفَعِل	يفَعِل	لا تفعِل	مفعِلا
يفَعِل	لا تفعِل	أفَعِل	مفعول	فَعِل	مفعِل	مفعول	فَعِلا	فاعِل
فَعِلا	مفعِلا	فاعِل	يفَعِل	لا تفعِل	مفعول	فَعِل	أفَعِل	مفعِل

Gambar 3.3 Contoh *Puzzle* setelah Terisi.

Semua kotak yang telah terisi *shighah* yang valid akan dihilangkan beberapa isinya dengan menggunakan algoritma *Linear Congruential Generator*. Inilah fungsi utama dari algoritma *Linear Congruential Generator* dalam penelitian ini. Penghilangan kotak yang menjadi soal dilakukan per area 3x3. Perbedaan proses penentuan kotak yang akan menjadi soal dengan proses diatas hanya pada jumlah output angka yang dibutuhkan. Jika pada proses pengisian semua kotak kosong dibutuhkan 1 output untuk memilih kumpulan angka yang akan diisi ke array 4 dimensi, maka pada proses penentuan kotak yang menjadi soal dibutuhkan angka yang jumlahnya berdasarkan level permainan. Seperti penjelasan sebelumnya, untuk level mudah kotak soal per area 3x3 sebanyak 4, level sulit 5 kotak dan level sulit 6 kotak soal.

Penulis akan memberikan contoh penggunaan algoritma *Linear Congruential Generator* untuk menentukan kotak yang menjadi soal pada level mudah untuk area 3x3 yang pertama:

Input:

$a=7, c=6, m=11$ dan $Z_0=1$

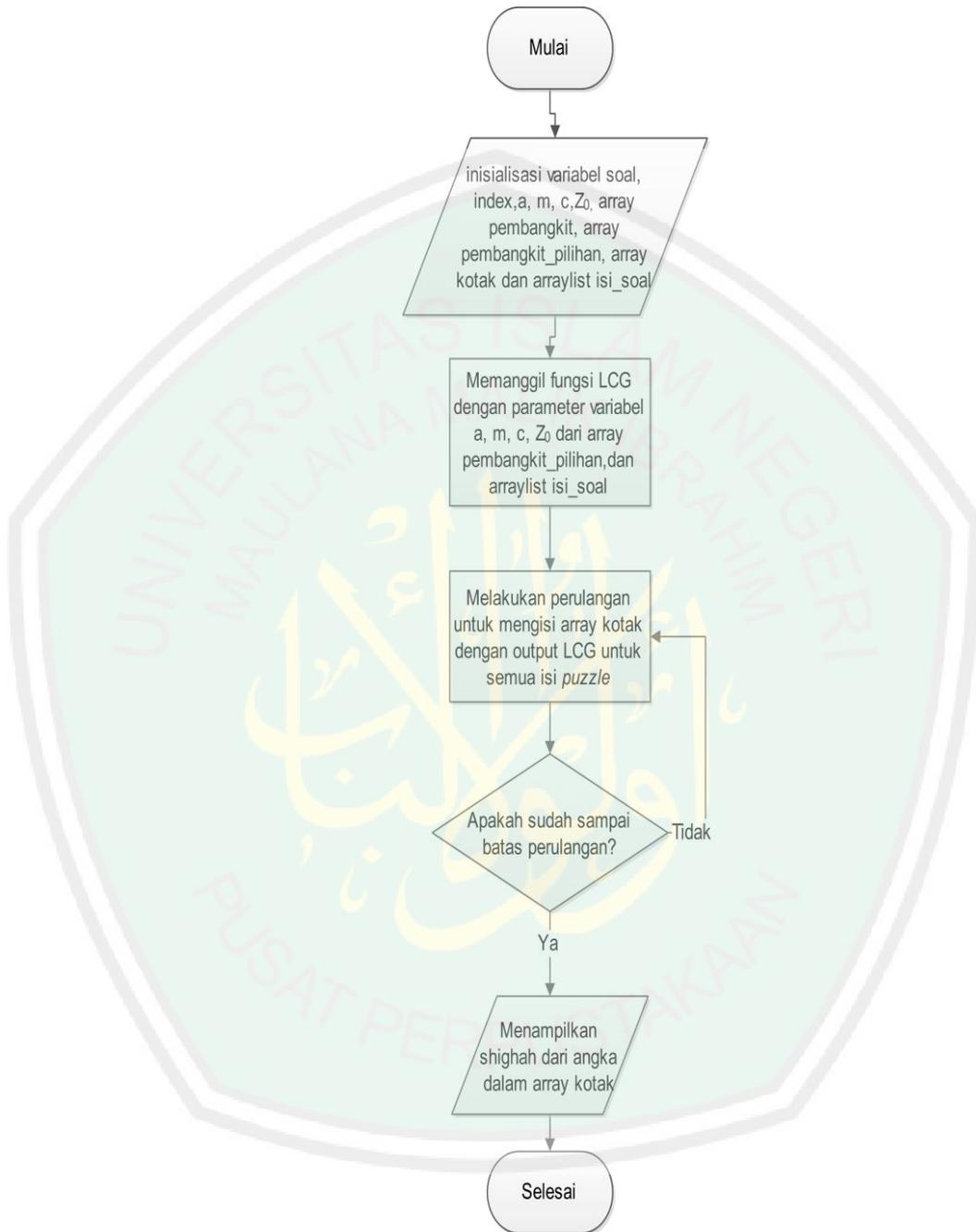
$$Z_i = (aZ_{i-1} + c) \bmod m$$

Output= 2,9,3,5

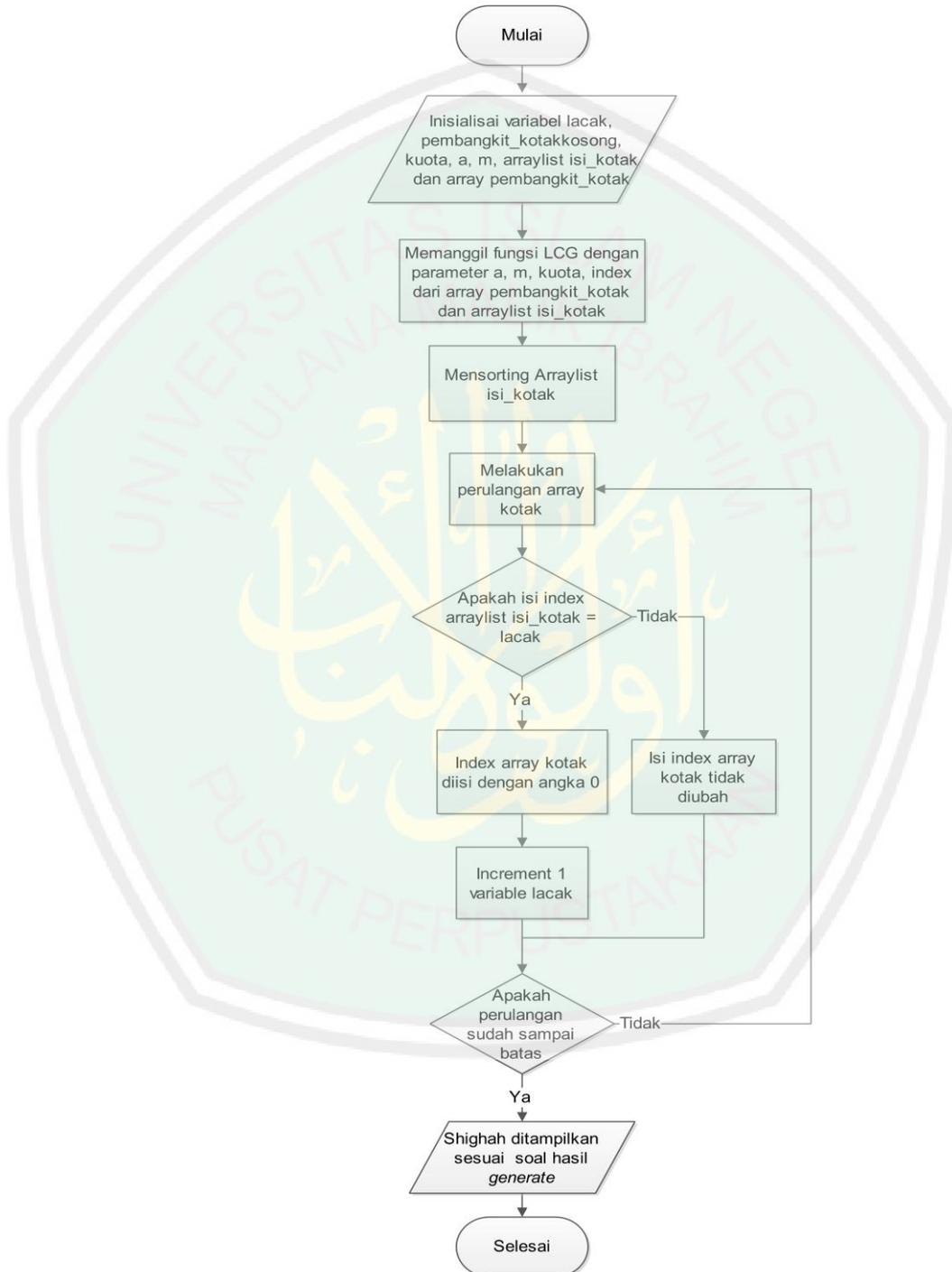
Output akan disimpan pada *arraylist* dan langsung disorting dari urutan yang terkecil untuk memudahkan proses penghilangan beberapa kotak yang menjadi soal. Jadi, sesuai contoh diatas, maka kotak yang dihilangkan pada area 3x3 pertama adalah kotak 2,3,5 dan 9. Angka yang disimpan pada *array* 4 dimensi yang indexnya sama dengan kotak yang menjadi soal akan diubah menjadi angka 0. Jadi isi dari *array* 4 dimensi untuk area 3x3 pertama yang awalnya 9,7,6,4,8,5,3,2,1 akan menjadi 9,0,0,4,0,5,3,2,0. Proses ini berlangsung dalam perulangan 9 kali karena jumlah area 3x3 dalam *puzzle* berjumlah 9. Penulis juga sudah menyiapkan beberapa angka yang akan menjadi nilai dari variabel Z_0 agar output yang dihasilkan tidak sama.

فَاعِلٌ			مَفْعُولٌ	أَفْعَلٌ		مَفْعَلٌ	فَعْلٌ	
مَفْعُولٌ		يَفْعَلُ		مَفْعَلٌ				لَا تَفْعَلُ
فَعْلٌ	مَفْعَلٌ		فَعْلًا	يَفْعَلُ		مَفْعُولٌ		أَفْعَلٌ
لَا تَفْعَلُ	فَاعِلٌ	مَفْعَلٌ		مَفْعُولٌ		مَفْعَلًا		فَعْلًا
		مَفْعُولٌ	لَا تَفْعَلُ		فَعْلًا	أَفْعَلٌ		فَعْلٌ
	فَعْلٌ		مَفْعَلٌ		مَفْعَلٌ			مَفْعُولٌ
	مَفْعُولٌ		فَاعِلٌ		أَفْعَلٌ	يَفْعَلُ		مَفْعَلًا
يَفْعَلُ		أَفْعَلٌ		فَعْلٌ			فَعْلًا	
فَعْلًا		فَاعِلٌ	يَفْعَلُ	لَا تَفْعَلُ		فَعْلٌ		مَفْعَلٌ

Gambar 3.4 Contoh *Puzzle* Sudoku Sharaf Hasil *Generate*.



Gambar 3.5 *Flowchart* Pengisian *Puzzle* Kosong



Gambar 3.6 *Flowchart* Menentukan *Puzzle* Soal

3.2.2 Solver Permainan

- A. Menentukan banyaknya angka yang menjadi solusi per area 3x3.
- B. Mengisi seluruh kotak *puzzle* yang kosong menggunakan algoritma

Branch And Bound.

Menentukan banyaknya angka yang menjadi solusi untuk mengisi semua kotak kosong per area 3x3 dengan melihat level permainan. Karena pada perancangan aplikasi ini, penulis menetapkan banyaknya kotak kosong per area 3x3 berdasarkan level permainan.

Setelah menentukan banyaknya angka yang mungkin menjadi solusi untuk semua kotak kosong per area 3x3, maka langkah selanjutnya adalah menentukan dan menyimpan angka yang dapat menjadi solusi. Tahapan ini merupakan salah satu ciri dari algoritma *Branch And Bound* yang hanya memperhitungkan kemungkinan angka. Sebagai contoh dari soal sudoku yang diatas, angka yang mengisi area 3x3 pertama adalah 9,0,0,4,0,5,3,2,0 dimana angka yang 0 berarti soal. Sehingga kemungkinan angka yang menjadi solusi pada area 3x3 pertama adalah 1,6,7 dan 8. Deretan angka ini akan dibentuk menjadi sebuah kombinasi solusi untuk area 3x3.

Kombinasi angka yang telah disimpan akan diperluas sebanyak mungkin sehingga terbentuk semua kemungkinan kombinasi dari deretan angka tersebut. Penulis menggunakan rumus faktorial untuk menghitung jumlah maksimal kombinasi dari setiap deretan angka yang menjadi solusi.

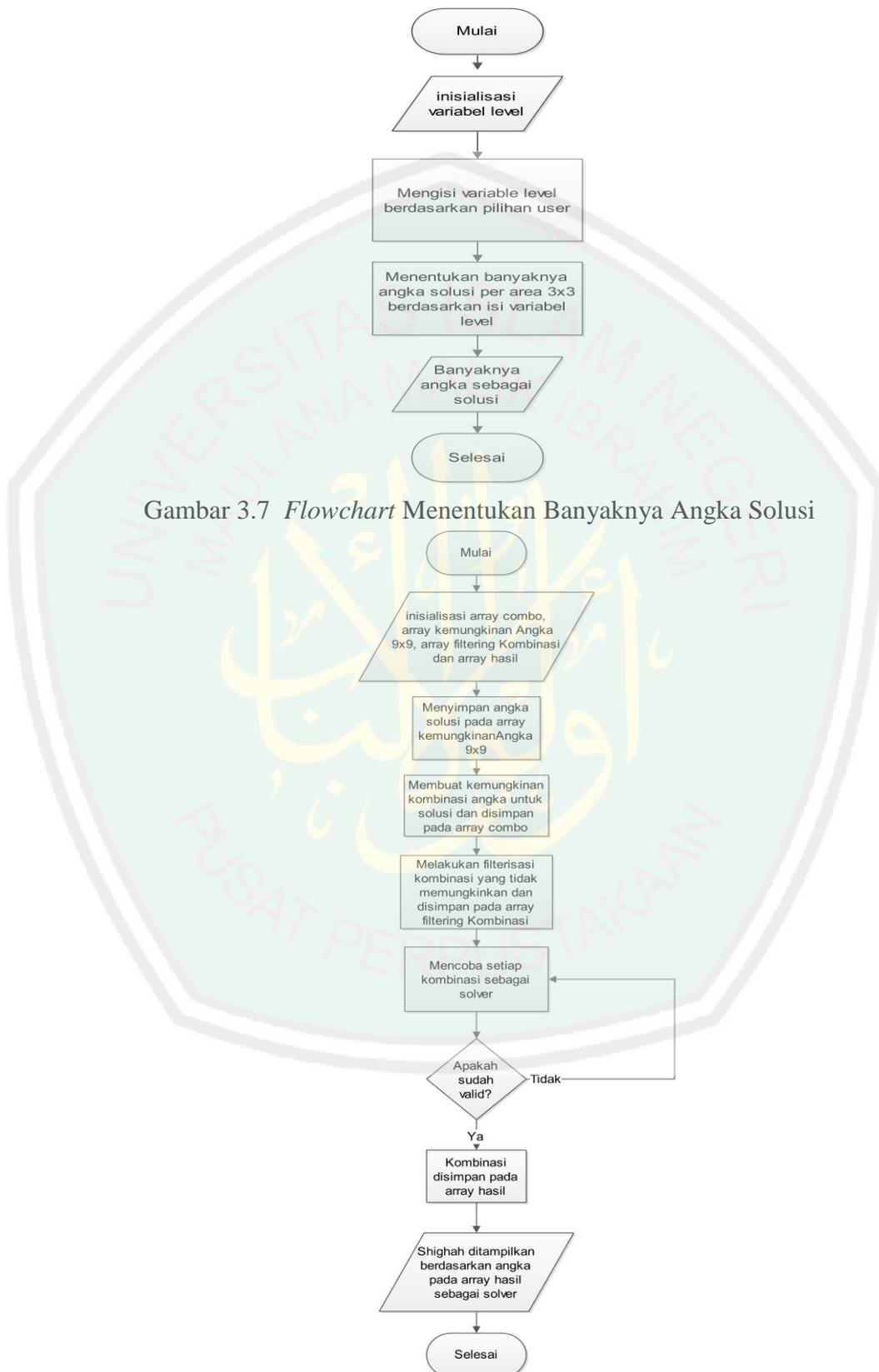
Tabel 3.3 Jumlah Kombinasi

Level	Jumlah Angka Solusi per Area 3x3	Banyaknya Kombinasi
Mudah	4	$4! = 4 \times 3 \times 2 \times 1 = 24$ Kombinasi
Sedang	5	$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ Kombinasi
Sulit	6	$6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 720$ kombinasi

Proses pembentukan kombinasi akan dilakukan sebanyak jumlah maksimalnya sekaligus mematikan kombinasi yang tidak memungkinkan untuk menjadi solusi dan setelah itu akan mencoba setiap kombinasinya. Jika telah ditemukan kombinasi angka yang valid, maka proses akan berhenti dan *shighah* akan ditampilkan berdasarkan angka yang menjadi *solver*.

Sebagai contoh jika deretan angka kemungkinannya adalah 1,6,7,8 seperti contoh soal diatas, maka akan terbentuk 24 kombinasi seperti 1678,1687,1768,1786 dst. Kombinasi angka 1678 akan dimasukkan ke dalam *array* 4 dimensi dan akan dilakukan pengecekan. Pengecekan dilakukan berdasarkan peraturan dalam permainan *Sudoku*. Jika tidak sesuai, maka kombinasi selanjutnya yaitu 1687 yang akan dicek. Proses ini berjalan per area 3x3 sampai pada kombinasi yang sesuai sebagai *solver*.

Ketika pada area 3x3 pertama telah selesai maka proses dilanjutkan sampai area 3x3 yang terakhir. Tapi perlu diingat bahwa salah satu ciri dari algoritma *Branch and Bound* ialah mampu melakukan *backtrack* ke area 3x3 sebelumnya jika pada area yang sedang dicek tidak ditemukan solusi padahal semua kombinasi telah dicoba.



Gambar 3.7 *Flowchart* Menentukan Banyaknya Angka Solusi

Gambar 3.8 *Flowchart* Algoritma Branch And Bound

3.3 Perancangan Aplikasi

Aplikasi permainan ini diberi nama aplikasi permainan **Sudoku Sharaf**. Nama tersebut sesuai dengan basis permainan yang didasari ilmu *sharaf*. Dengan adanya nama aplikasi ini, diharapkan pemain lebih mudah untuk mengingat aplikasi permainan ini. Berikut ini penjelasan tentang rancangan permainan *Sudoku Sharaf*.

3.3.1 Antarmuka Permainan

Aplikasi permainan *Sudoku Sharaf* memiliki beberapa halaman sebagai berikut:

a. Halaman Utama

Ketika permainan dijalankan, halaman yang pertama kali muncul adalah halaman utama. Halaman ini berisi pilihan menu-menu, yaitu menu mainkan, petunjuk, *sharaf*, keluar dan *audio*.



Gambar 3.9 Rancangan Halaman Utama

Dari menu-menu pada halaman utama, penjelasannya sebagai berikut:

1. *Mainkan*: Menu ini untuk melanjutkan ke permainan.
2. *Audio*: Menu ini untuk mengaktifkan suara pelafalan *shighah*.
3. *Petunjuk*: Menu ini untuk memberikan petunjuk permainan.

4. *Sharaf*: Menu ini untuk memberikan pengetahuan ilmu *sharaf*.
5. Keluar: Menu ini untuk keluar dari aplikasi.

b. Halaman Pilih *Wazn*

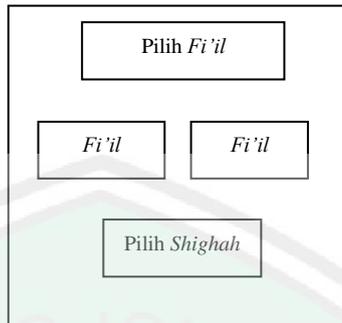


Gambar 3.10 Rancangan Halaman Pilih *Wazn*

Halaman ini mengharuskan pemain untuk memilih *wazn* yang akan dimainkan dalam permainan ini. Setiap *wazn* mempunyai urutan *shighah-shighah* yang berbeda. Setelah memilih *wazn*, pemain menekan tombol “**Pilih *Fi'il***” untuk memilih *fi'il* di tampilan selanjutnya.

c. Halaman Pilih *Fi'il*

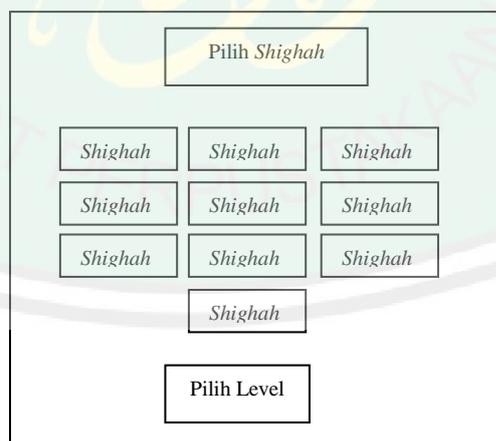
Pada Halaman ini, pemain akan memilih 1 *fi'il* dari *wazn* yang telah dipilih sebelumnya.



Gambar 3.11 Rancangan Halaman Pilih *Fi'il*

d. Halaman Pilih *Shighah*

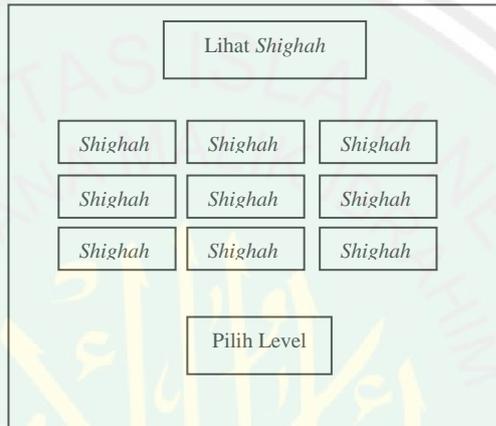
Pada Halaman ini, pemain akan memilih 9 *shighah* dari *fi'il* yang telah dipilih sebelumnya. 9 *shighah* ini yang akan menjadi pilihan untuk mengisi kotak kosong *puzzle*. Tapi perlu diketahui bahwa tidak semua *fi'il* memiliki *shighah* lebih dari 9. Oleh karena itu, halaman ini akan tampil ketika pemain memilih *wazn* yang memiliki lebih dari 9 *shighah*.



Gambar 3.12 Rancangan Halaman Pilih *Shighah*

e. Halaman Lihat *Shighah*

Halaman ini merupakan halaman lanjutan ketika pemain memilih *fi'il* yang hanya memiliki 9 *shighah*. Jadi pemain hanya melihat *shighah* yang akan dipilih untuk mengisi kotak kosong *puzzle*.



Gambar 3.13 Rancangan Halaman Lihat *Shighah*

f. Halaman Level



Gambar 3.14 Rancangan Halaman Pilih Level

Halaman ini mengharuskan pemain untuk memilih level yang akan dimainkan. Setiap level memiliki perbedaan pada banyaknya kotak kosong yang harus diisi, sehingga semakin tinggi levelnya maka semakin sulit untuk menyelesaikan permainan.

g. Halaman *Puzzle*

مَفْعَلٌ			أَفْعَلٌ					
	فَعْلٌ		مَفْعُولٌ				مَفْعَلًا	
								مَفْعَلٌ
	يَفْعَلُ		مَفْعَلٌ				أَفْعَلٌ	
		يَفْعَلُ				مَفْعُولٌ		
							فَعْلًا	
	لَا تَفْعَلُ		فَاعِلٌ					

Gambar 3.15 Rancangan Halaman *Puzzle*

Halaman *puzzle* merupakan halaman inti dari permainan ini. Pada halaman ini, pemain harus mengisi semua kotak kosong apabila ingin memenangkan permainan ini. Pemain harus menekan kotak kosong untuk melihat pilihan *shighah* yang akan dipilih untuk mengisi kotak kosong. Jika pada halaman utama pilihan *audio* dicentang, maka akan ada suara yang melafalkan *shighah* tersebut.

Ketika memilih *shighah*, tentunya pemain harus memenuhi syarat yang berlaku dalam permainan *Sudoku* pada umumnya. Untuk permainan *Sudoku Sharaf* ini, *shighah* yang dipilih untuk mengisi kotak kosong tidak boleh ada yang sama dengan *shighah* pada kotak pada baris, kolom dan satu area 3x3.

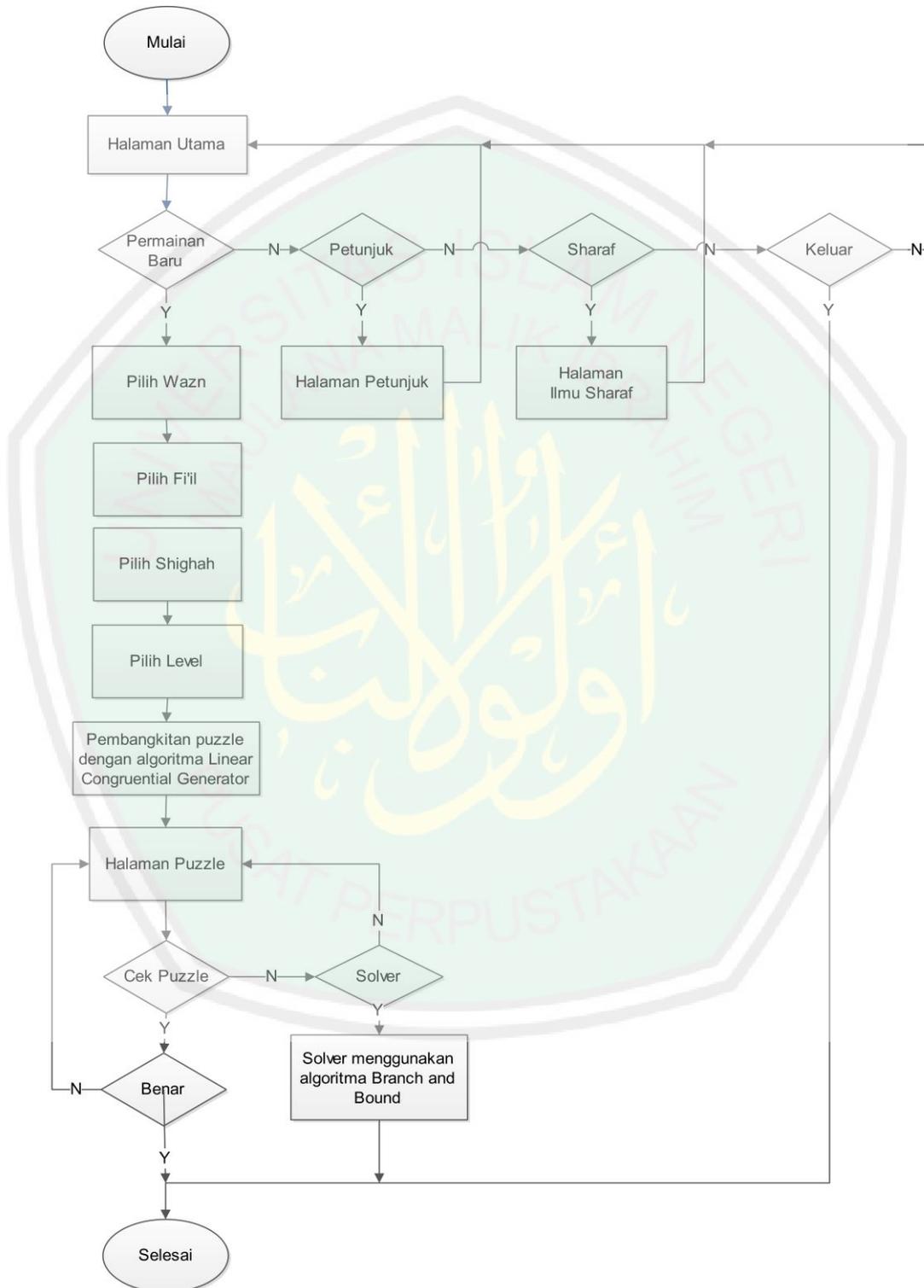
Pada halaman *puzzle*, pemain tidak bisa menekan tombol “**back**”. Pemain hanya bisa menekan tombol “**Menu**” untuk memilih tombol “**Cek *Puzzle***”, “**Selesaikan**” atau “**Keluar**”.

Cek <i>Puzzle</i>	Selesaikan	Keluar

3.16 Rancangan Menu *Puzzle*



Berikut ini *flowchart* dari permainan *Sudoku Sharaf*:



3.17 *Flowchart* Permainan *Sudoku Sharaf*

3.3.2 Kebutuhan Sistem

Dalam pembuatan aplikasi *Sudoku Sharaf*, penulis membutuhkan beberapa perangkat keras dan perangkat lunak.

1. Perangkat Keras
 - a. Penulis menggunakan Laptop dengan spesifikasi: *Processor* Intel(R) Core(TM)2 Duo CPU T6500 @ 2.10GHz dan RAM 2.00 GB, yang digunakan dalam pembuatan aplikasi.
 - b. Untuk melakukan ujicoba aplikasi, penulis menggunakan Perangkat *Tablet* yang berbasis Android dengan spesifikasi: *Processor* 1,2 GHz *multicore*, Memory 1GB DDR3 dan LCD 7", 1024×600 Pixels.
2. Perangkat Lunak
 - a. Software Eclipse, yang merupakan software untuk melakukan coding aplikasi android. Eclipse yang digunakan sudah support ADT (Android Development Tools).
 - b. ADT, plugin tambahan untuk software Eclipse agar dapat digunakan untuk pembuatan aplikasi berbasis android. ADT yang digunakan adalah ADT versi 21.0.1.
 - c. Android SDK (Software Development Kit), yang merupakan API (Application Programming Interface) dan alat bantu dalam mengembangkan aplikasi berbasis android yang menggunakan bahasa java. Android SDK yang digunakan adalah Android SDK versi 21.0.1.
 - d. Java, yang digunakan untuk melakukan kompilasi aplikasi berbasis android. Versi yang digunakan adalah Sun Java SE versi 1.7.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas tentang implementasi dari rancangan yang telah dibuat pada bab sebelumnya. Selain itu juga akan dijelaskan analisa dari uji coba aplikasi yang dibuat sehingga akan diketahui apakah aplikasi telah berjalan sesuai dengan yang diharapkan. Baik dari implementasi algoritma maupun kegunaan bagi penggunaannya.

4.1 Implementasi *Linear Congruential Generator* dan *Branch And Bound*

4.1.1 *Linear Congruential Generator*

Algoritma *Linear Congruential Generator* diimplementasikan menjadi sebuah fungsi random dalam sebuah class java yang diberi nama “**generator**”. Nama dari fungsi random tersebut adalah LCG dengan parameter berupa range, z_0 , a , m , c dan sebuah *arraylist* yang digunakan untuk menampung output dari fungsi tersebut. Semua parameter dari fungsi LCG bernilai integer.

4.1.2 *Branch And Bound*

Proses *Branch And Bound* pada penelitian kali ini dilakukan dengan mengisi kotak kosong per blok atau per area, bukan per kotak dalam baris dan kolom secara berurutan. Setiap blok atau area memiliki kotak berdimensi 3x3.











Seperti pada analisa algoritma pada bab sebelumnya, kode program diatas telah menpresentasikan konsep algoritma *Branch And Bound*. Angka solusi per area 3x3 dicari melalui method *CariKemungkinanAngka*, setelah itu angka-angka tersebut akan dibuatkan kombinasi melalui method *bikinSeluruhKombinasi*. Kombinasi angka akan difilter melalui fungsi komposisi dan pada akhirnya akan dicek dengan fungsi *deadLock*.

4.1.3 Pembangkitan *Puzzle*

Proses pembangkitan *puzzle* diimplementasikan dalam fungsi *rumus_posisi* dengan parameter *int* kuota dan sebuah array *level*. Awalnya, fungsi tersebut akan mengisi semua kotak kosong *puzzle* dan membuang sebagian kotak *puzzle* untuk dijadikan sebagai soal.







Dari fungsi diatas, dapat dilihat bahwa proses pembangkitan *puzzle* diawali dengan membangkitkan soal untuk mengisi semua kotak kosong berdasarkan hasil dari algoritma *Linear Congruential Generator*. Objek **I** merupakan objek dari kelas *generator* yang terdapat fungsi LCG. Setelah itu, array kotak diisi berdasarkan hasil *generator*.

Untuk membedakan jumlah kotak kosong sebagai soal, maka variabel kuota diisi berdasarkan level yang dipilih pengguna. Fungsi LCG kembali dipanggil untuk menghasilkan angka acak sesuai kuota. Outputnya ditampung dalam array *isi_kotak* dan kemudian dilakukan penyortiran secara *ascending*.

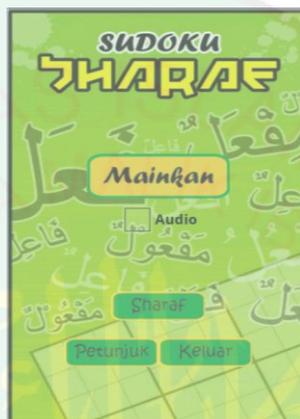
Setelah array `isi_kotak` disorting, maka langkah selanjutnya adalah melakukan perulangan *array* kotak dan setiap *indeks array* kotak akan diberikan nilai 0 apabila nilai variabel `lacak` sama dengan isi *indeks array* `isi_kotak` yang sedang melakukan perulangan juga. Variabel `acak` akan mengecek setiap area 3x3, apabila nilai dari *indeks array* kotak sama dengan nilai dari *indeks array* `isi_kotak`, maka nilai `variable` `lacak` di-increment 1.

4.1.4 Penyelesaian *Puzzle*

Penyelesaian dilakukan dengan memanggil *constructor* dari kelas `branchbound` untuk mendapatkan soal. Setelah itu memanggil metode `ConvertFilterComposisi` untuk membuat pohon ruang solusi dari soal.

4.2 Implementasi Aplikasi

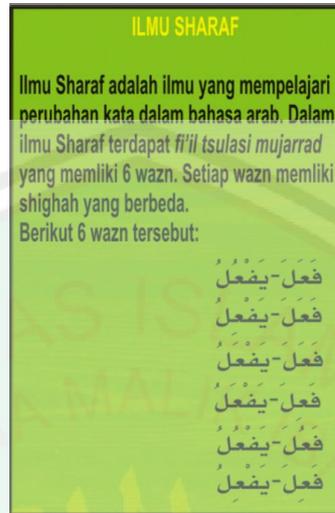
Berikut ini adalah implementasi rancangan ke dalam aplikasi *Sudoku Sharaf*:



Gambar 4.1 Halaman Utama

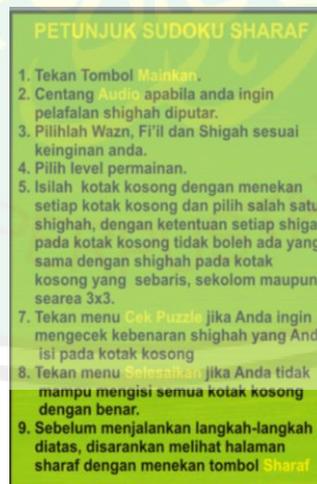
Pada halaman utama terdapat tombol mainkan, *Sharaf*, petunjuk, dan keluar seperti pada rancangan sebelumnya. Selain itu juga terdapat *checkbox* audio yang apabila dicentang maka ketika sedang memilih *shighah* untuk mengisi kotak kosong akan diikuti pelafalan *shighah* tersebut.

Untuk mengetahui sekilas tentang ilmu *Sharaf*, pengguna dapat menekan tombol *Sharaf*. Dalam halaman *Sharaf* akan dijelaskan definisi dari ilmu *Sharaf* dan pembagian *wazn* dari *fi'il tsulatsi mujarrad* yang terdiri dari 6 *wazn*. Berikut ini tampilan halaman *Sharaf*:



Gambar 4.2 Halaman *Sharaf*

Jika tombol petunjuk ditekan pada halaman utama, maka halaman petunjuk yang berisi petunjuk untuk memainkan *game Sudoku Sharaf* akan ditampilkan sebagai berikut:



Gambar 4.3 Halaman Petunjuk

Untuk memulai permainan *Sudoku Sharaf* yang menjadi inti dari aplikasi ini, pengguna dapat menekan tombol mainkan dan akan tampil halaman pilih

wazn. Pada halaman ini pengguna diharuskan memilih 1 *wazn* dari 6 *wazn* yang ditampilkan. Berikut tampilan dari halaman pilih *wazn*:



Gambar 4.4 Halaman Pilih Wazn

Setelah memilih *wazn*, pengguna akan memilih 1 *fi'il* dari 2 *fi'il* yang ditampilkan setelah sebelumnya menekan tombol pilih *fi'il*. Berikut tampilan dari halaman pilih *fi'il*:



Gambar 4.5 Halaman Pilih *Fi'il*

Ketika pengguna telah memilih *fi'il* yang memiliki lebih dari 9 *shighah* dan menekan tombol pilih *shighah*, maka halaman pilih *shighah* akan

ditampilkan. Pengguna akan memilih 9 *shighah* dari 10 *shighah* yang ditampilkan. Tetapi apabila pengguna memilih *fi'il* yang hanya memiliki 9 *shighah*, maka halaman lihat *shighah* yang ditampilkan. Berikut tampilan dari halaman pilih *shighah* dan lihat *shighah*:



Gambar 4.6 Halaman Pilih *Shighah*



Gambar 4.7 Halaman Lihat *Shighah*

Halaman terakhir yang ditampilkan sebelum ke halaman *puzzle* adalah halaman pilih level. Pada halaman ini, pengguna akan memilih 1 dari 3 level yang

Untuk mengisi kotak kosong, pengguna cukup menekan kotak kosong tersebut dan akan tampil pilihan 9 *shighah* yang akan dipilih untuk mengisi kotak kosong tersebut. Berikut tampilan pilihan *shighah*:



Gambar 4.10 Halaman *Shighah* Pilihan

Saat pengguna memilih *shighah* untuk mengisi kotak kosong, maka akan dibunyikan pelafalan dari *shighah* tersebut jika *Audio* dalam posisi *On*. Apabila pengguna ingin mengecek apakah *shighah* yang dipilih untuk mengisi kotak kosong benar atau tidak, maka pengguna dapat menekan tombol menu lalu pilih *Cek puzzle*. Berikut tampilan ketika menu ditekan:



Gambar 4.11 Menu *Puzzle*

Selain itu juga terdapat pilihan selesaikan jika pengguna merasa tidak mampu untuk mengisi semua kotak kosong dengan benar. Penyelesaian permainan *Sudoku Sharaf* ini menggunakan algoritma *Branch And Bound*. Berikut tampilan halaman *puzzle* setelah diselesaikan menggunakan metode *Branch And Bound*.



Gambar 4.12 Hasil Penyelesaian

4.3 Uji Coba

4.3.1 Uji Coba Algoritma *Linear Congruential Generator* dan *Branch And Bound* dalam Menyelesaikan *Puzzle Sudoku Sharaf*

Pengujian terhadap penerapan Algoritma *Linear Congruential Generator* dan *Branch And Bound* pada aplikasi *Sudoku Sharaf* dilakukan dengan perangkat keras Axioo Picopad 7 GGM dengan spesifikasi sebagai berikut:

1. Prosesor : 1,2 GHz *multicore*

2. Internal Memory : 1GB DDR3
3. Resolusi : 1024×600 Pixels
4. OS : Android ICS 4.0.3 (Ice Cream Sandwich)

Pengujian dilakukan dengan menyelesaikan *game Sudoku Sharaf*. Waktu yang digunakan telah terprogram pada aplikasi yang akan diuji. Pengujian dilakukan 10 kali setiap levelnya, yaitu: level mudah, sedang dan sulit. Perbedaan level terdapat pada banyaknya kotak kosong yang harus diisi.

a. Pengujian Level Mudah

Kotak kosong pada halaman *puzzle* berjumlah 36 kotak sesuai dengan level mudah dalam *game Sudoku Sharaf*. Berikut ini contoh *puzzle* yang diuji:

		أفعل		يفعل			فعل
	لاتنقل		فعل	فعل			يفعل
	مفعلا	يفعل	فعل	مفعلا	أفعل		فاعلا
		مفعلا		أفعل			فعل
	فعل			مفعلا	يفعل		أفعل
	يفعل	أفعل	لاتنقل	فعل	مفعلا		فاعلا
لاتنقل			فعل				مفعلا
أفعل		فعل	يفعل	لاتنقل			مفعلا
مفعلا	فعل		مفعلا	أفعل			فاعلا

Gambar 4.13 *Puzzle* pada level mudah



Gambar 4.14 Solusi dari *Puzzle* Mudah

Setelah melakukan pengujian 10 kali pada level mudah dengan parameter *Linear Congruential Generator* yaitu $a=7$, $c=6$ dan $m=11$, maka didapatkan hasil sebgar berikut:

Tabel 4.1 Hasil Uji Coba *Puzzle* Level Mudah

No	Percobaan	Posisi Kotak Soal	Waktu (detik)
1	Pertama	1,2,3,5,6,7,8,9,14,21,24,27,29,30,31,32,35,36,38,40,44,48,51,54,55,56,57,59,60,61,62,68,70,71,75,78.	0,659
2	Kedua	2,3,4,5,8,9,11,13,15,17,21,27,28,29,30,31,32,34,35,40,42,43,48,54,55,56,58,59,61,62,64,66,67,70,78,81.	0,564
3	Ketiga	1,2,3,5,6,7,8,9,14,21,24,27,28,29,30,31,32,34,35,40,42,43,45,48,55,56,57,58,59,60,61,62,63,75,78,81.	0,512
4	Keempat	1,2,3,4,5,7,8,13,16,21,24,27,28,29,30,31,32,34,35,40,42,43,45,48,55,56,57,59,60,61,62,63,68,75,78,81.	0,365
5	Kelima	1,2,3,4,5,6,7,8,9,21,24,27,28,29,31,32,33,35,36,37,39,44,51,54,56,57,58,59,61,62,65,67,70,75,78,81.	0,899
6	Keenam	1,2,4,5,6,7,8,10,16,21,24,27,28,29,31,32,35,36,3	0,566

No	Percobaan	Posisi Kotak Soal	Waktu (detik)
		7,40,44,48,51,54,56,57,58,59,62,63,65,67,69,71,75,81.	
7	Ketujuh	1,2,3,4,5,6,8,9,17,21,24,27,28,29,30,31,32,33,35,36,44,51,54,55,56,57,58,59,62,63,67,69,71,75,81.	0,401
8	Kedelapan	1,2,4,5,7,8,10,11,12,16,18,24,29,30,32,33,35,36,38,41,44,48,51,54,55,56,59,60,61,62,63,64,66,68,78,81.	0,439
9	Kesembilan	2,3,4,5,7,8,11,13,15,16,18,21,29,30,32,33,34,35,38,41,43,48,51,54,55,56,58,59,61,62,64,66,67,70,78,81.	0,365
10	Kesepuluh	1,2,3,4,5,6,8,9,17,21,24,27,28,29,32,33,35,36,37,41,44,48,51,54,55,56,59,60,61,62,64,68,70,72,75,78.	0,371

Berdasarkan data dari tabel 4.1, maka dapat diketahui bahwa waktu untuk mendapatkan solusi dengan jumlah kotak kosong sebanyak 36 adalah antara 0.365-0,899 detik.

b. Pengujian Level Sedang

Pada pengujian level ini, jumlah kotak kosong sebanyak 45 sesuai dengan level permainan sedang dalam *game Sudoku Sharaf*. Berikut contoh *puzzle* yang diuji:



Gambar 4.15 *Puzzle* pada level sedang



Gambar 4.16 *Puzzle* pada level sedang dan solusinya

Setelah melakukan pengujian 10 kali pada level sedang dengan parameter *Linear Congruential Generator* yaitu $a=7$, $c=6$ dan $m=11$, maka didapatkan hasil sebagai berikut:

Tabel 4.2 Hasil Uji Coba Puzzle Level Sengah

No	Percobaan	Posisi Kotak Soal	Waktu (detik)
1	Pertama	1,2,3,5,6,7,8,9,10,14,17,20,23,24,27,28,29,30,31,32,33,34,35,38,40,43,45,48,51,54,55,56,57,58,59,60,61,62,63,64,68,70,75,78,81.	2,193
2	Kedua	1,2,3,4,5,6,7,8,11,13,16,18,21,24,27,28,29,30,31,32,33,34,35,37,41,43,45,48,51,54,55,56,57,58,59,60,61,62,63,64,67,70,75,78,81.	4,346
3	Ketiga	2,3,4,5,6,7,8,11,14,16,18,20,21,24,27,28,29,31,32,34,35,36,37,39,40,42,44,48,51,54,56,57,58,59,60,61,62,65,68,70,72,74,75,78,81.	2,065
4	Keempat	1,2,3,4,5,6,7,8,10,13,16,18,21,24,27,29,30,31,32,34,35,36,38,40,42,44,47,48,51,54,55,56,57,58,59,61,62,64,66,68,69,72,75,78,81.	3,214
5	Kelima	1,2,3,4,5,6,7,8,9,10,14,17,21,24,27,28,29,31,32,33,35,36,37,39,41,44,48,51,54,56,57,58,59,60,61,62,63,64,65,66,68,70,72,75,78,81.	2,416
6	Keenam	2,3,5,6,7,8,9,11,14,17,20,21,23,24,27,28,29,30,31,32,34,35,37,39,40,42,45,48,51,54,55,56,57,58,59,60,61,62,63,65,68,71,75,78,81.	2,278
7	Ketujuh	1,2,3,4,5,7,8,9,10,13,15,17,21,24,27,28,29,31,32,34,35,36,37,39,40,42,44,48,51,54,55,56,57,58,59,60,61,62,64,67,70,72,75,78,81.	2,278
8	Kedelapan	1,2,3,4,5,6,8,9,11,14,17,20,24,26,27,28,29,30,31,32,33,34,35,36,38,41,43,48,51,54,56,57,58,59,60,62,63,65,67,71,74,75,77,80,81.	2,223
9	Kesembilan	2,3,4,5,6,7,8,9,11,13,16,20,21,24,27,28,29,30,31,32,33,34,35,36,37,40,43,48,51,54,55,56,57,58,59,60,61,62,63,64,67,69,70,75,78,81.	4,195
10	Kesepuluh	1,2,4,5,6,7,8,9,10,12,13,17,21,24,27,28,29,30,32,33,34,35,38,41,43,45,48,50,51,54,55,56,58,59,60,62,63,64,66,68,71,75,78,80,81.	5,367

Berdasarkan data dari tabel 4.2, maka dapat diketahui bahwa waktu untuk mendapatkan solusi dengan jumlah kotak kosong sebanyak 45 adalah antara 2,065-5,367 detik.

c. Pengujian Level Sulit

Pada pengujian ketiga ini, jumlah kotak kosong sebanyak 54 buah sesuai dengan level permainan sulit pada *game Sudoku Sharaf*. Berikut contoh *puzzle* yang diuji:



Gambar 4.17 *Puzzle* pada Level Sulit



Gambar 4.18 Solusinya *Puzzle* pada Level Sulit

Setelah melakukan pengujian 10 kali pada level sulit dengan parameter *Linear Congruential Generator* yaitu $a=7$, $c=6$ dan $m=11$, maka didapatkan hasil sebagai berikut:

Tabel 4.3 Hasil Uji Coba Puzzle Level Sulit

No	Percobaan	Posisi Kotak Soal	Waktu (detik)
1	Pertama	2,3,4,5,6,7,8,9,11,13,15,17,19,20,21,24,26,27,29,30,31,32,33,34,35,36,38,41,44,46,47,48,50,51,53,54,56,57,59,60,61,62,63,65,68,70,71,73,74,75,76,77,78,81.	813,111
2	Kedua	1,2,3,5,6,8,9,10,12,14,17,21,22,23,24,25,26,27,28,29,30,32,33,35,36,37,39,41,44,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,67,69,71,75,78,80,81.	67,113
3	Ketiga	1,2,3,5,6,7,8,9,10,11,14,16,17,21,22,23,24,27,28,29,30,31,32,33,34,35,36,38,40,42,44,47,48,51,53,54,55,56,57,58,59,60,61,62,63,64,65,67,69,71,75,78,81.	59,191
4	Keempat	2,3,4,5,6,8,9,11,13,15,17,19,20,21,24,25,26,27,28,29,30,31,32,33,34,35,36,37,39,41,43,45,48,50,51,54,55,56,57,58,59,60,61,62,63,64,66,68,70,72,75,77,78,81.	54,341
5	Kelima	1,2,3,4,5,6,8,9,10,12,14,17,21,23,24,25,26,27,28,29,30,32,33,34,35,36,38,41,43,44,47,48,49,50,51,54,55,56,57,58,59,60,61,62,63,65,67,68,70,71,74,75,78,81.	365,269
6	Keenam	1,2,3,4,5,6,7,8,9,11,13,14,16,18,20,21,24,27,28,29,30,31,32,33,34,35,36,37,38,40,41,43,44,48,51,54,56,57,58,59,60,61,62,63,65,67,69,70,72,73,74,75,78,81.	59,191
7	Ketujuh	1,2,3,4,5,6,7,8,9,10,11,14,16,18,21,23,24,27,28,29,30,31,32,33,34,35,36,37,38,41,43,45,48,50,51,54,55,56,57,58,59,60,61,62,63,65,68,70,72,74,75,77,78,81.	54,341
8	Kedelapan	1,2,3,4,5,6,7,8,9,10,12,13,14,16,18,21,24,27,28,29,30,31,32,33,34,35,36,37,38,40,41,44,48,51,53,54,55,56,57,58,59,60,61,62,63,65,68,70,72,74,75,77,78,81.	58,741
9	Kesembilan	2,3,4,5,6,7,8,9,11,13,14,16,17,19,20,21,24,27,28,29,30,31,32,33,34,35,36,38,41,43,45,47,48,50,51,54,55,56,57,58,59,60,61,62,63,64,66,67,69,70,72,75,78,81.	62,183
10	Kesepuluh	1,2,3,4,5,6,7,8,9,11,13,14,17,20,21,24,26,27,29,30,32,33,35,36,38,41,44,46,47,48,49,50,5	99,938

No	Percobaan	Posisi Kotak Soal	Waktu (detik)
		1,52,53,54,55,56,57,58,59,60,61,62,63,64,65, 67,68,71,75,78,80,81.	

Berdasarkan data dari tabel 4.3, maka dapat diketahui bahwa waktu untuk mendapatkan solusi dengan jumlah kotak kosong sebanyak 54 adalah antara 54,341-813,111 detik.

Berdasarkan hasil pengujian yang telah dilakukan, terdapat perbedaan waktu pencarian solusi karena adanya perbedaan jumlah kotak kosong. Semakin banyak kotak kosong yang harus dicari solusinya, maka semakin lama waktu yang dibutuhkan. Terutama pada level sulit yang pada waktu maksimal pencarian solusi dari pengujiannya ada yang mencapai 813 detik. Ini dimungkinkan karena banyaknya kemungkinan angka. Tetapi lamanya waktu pencarian tidak menghalangi pembuktian bahwa algoritma *Branch And Bound* mampu sebagai *solver* permainan *Sudoku Sharaf*.

4.3.2 Uji Coba Pengguna

Pengujian aplikasi *Sudoku Sharaf* dilakukan pada responden dari MTs Mubtadul ‘Abidin Karangploso. Pengujian dilakukan pada tanggal 28 Maret 2013 pukul 10.30 WIB. Terdapat 30 responden dari kelas VIII C yang dijadikan responden untuk mengetahui penilaian pengguna terhadap kelayakan dan manfaat dari aplikasi *Sudoku Sharaf*.

Berikut ini adalah rekapitulasi hasil uji coba aplikasi *Sudoku Sharaf* yang berdasarkan penilaian responden:

Tabel 4.4 Rekapitulasi Hasil Uji Coba

NO	PERTANYAAN	HASIL		PROSENTASE	
		YA	TIDAK	YA	TIDAK
1	Apakah pemain mengetahui kegunaan setiap menu pada halaman utama?	22	8	73 %	27%
2	Apakah pemain mengerti penjelasan dari menu Bantuan ?	28	2	93%	7%
3	Apakah pemain mengerti penjelasan dari menu Sharaf ?	26	4	87%	13%
4	Apakah pemain mengetahui cara memilih <i>wazn</i> ?	24	6	80%	20%
5	Apakah pemain mengetahui cara memilih <i>fi'il</i> ?	26	4	87%	13%
6	Apakah pemain mengetahui cara memilih <i>shighah</i> ?	22	8	73%	27%
7	Apakah pemain mengetahui cara mengisi kotak kosng dalam <i>puzzle</i> ?	29	1	96%	4%
8	Apakah pemain mengetahui cara penggunaan menu Selesaikan jika tidak mampu mengisi setiap kotak kosong dalam <i>puzzle</i> ?	29	1	96%	4%
9	Apakah pemain merasa bosan memainkan <i>game Sudoku Sharaf</i> ?	8	22	27%	73%
10	Apakah <i>game Sudoku Sharaf</i> menyenangkan dan menarik?	25	5	83%	17%
11	Apakah pemain merasa lebih mudah menghafal <i>wazn</i> melalui <i>game Sudoku Sharaf</i> ?	25	5	83%	17%
12	Apakah pemain merasa lebih mudah menghafal <i>shighah</i> melalui <i>game Sudoku Sharaf</i> ?	24	6	80%	20%
13	Apakah pemain mengetahui pelafalan <i>shighah</i> yang benar dari <i>game Sudoku Sharaf</i> ?	24	6	80%	20%
14	Apakah pemain ingin memainkan <i>game Sudoku Sharaf</i> lagi?	29	1	96%	4%

Berdasarkan tabel diatas, dari jumlah keseluruhan 30 responden, 73% responden mengetahui fungsi setiap menu pada halaman utama dan 27% menyatakan tidak mengetahui. Untuk penjelasan pada halaman Bantuan, 93% mengerti dan 7% tidak mengerti. 87% responden juga mengerti penjelasan pada halaman *Sharaf* dan 13% yang tidak mengerti.

Responden yang mengetahui cara memilih *wazn* sebanyak 80% dan 20% tidak mengetahui. Untuk memilih fi'il pada halaman pilih fi'il, sebanyak 13% tidak mengetahui dan 87% mengetahui cara memilihnya. Responden yang mengetahui cara memilih *shighah* sebanyak 73% dan 27% tidak mengetahui cara memilih *shighah*. Sebanyak 96% responden mengerti cara mengisi kotak kosong pada halaman puzzle dan hanya 4% yang tidak mengetahui.

Untuk penggunaan menu selesaikan, 96% responden mengerti cara penggunaannya dan 4% yang tidak mengetahuinya. Responden yang merasa bosan memainkan *game Sudoku Sharaf* sebanyak 27% dan yang menyatakan tidak bosan sebanyak 73%. *Game Sudoku Sharaf* juga dianggap menarik oleh 83% responden dan 17% merasa tidak menarik.

Game Sudoku Sharaf juga mampu membuat responden lebih mudah untuk menghafal *wazn*. Sebanyak 83% responden merasa lebih mudah menghafal *wazn* melalui *game* ini dan 17% merasa tidak lebih mudah. Untuk menghafal *shighah*, sebanyak 80% merasa lebih mudah dan 20% tidak merasa lebih mudah. 80% responden mengetahui pelafalan *shighah* yang benar dari *game* ini dan 20% tidak mengetahui. Responden yang ingin memainkan *game Sudoku Sharaf* lagi sebanyak 96% dan hanya 4% yang tidak ingin memainkan lagi.

Tabel 4.5 Prosentase Berdasarkan Parameter Penilaian

NO	Parameter Penilaian	Parameter	Prosentase
1	Pemahaman <i>Game</i>	Halaman Utama	73%
		Halaman Bantuan	93%
		Halaman <i>Sharaf</i>	87%
		Pilih <i>Wazn</i>	80%
		Pilih Fi'il	87%
		Pilih <i>Shighah</i>	73%
		Isi Puzzle	96%
		Menu Selesaikan	96%
		Rata-rata	86%

NO	Parameter Penilaian	Parameter	Prosentase
2	Minat <i>Game</i>	Tidak Bosan	73%
		Menarik	83%
		Main Lagi	96%
		Rata-rata	84%
3	Minat Belajar	Hafal Wazn	83%
		Hafal Shighah	80%
		Pelafalan Shighah	80%
		Rata-rata	81%
Rata-rata			84%

Dari tabel diatas, penilaian baik responden terhadap ujicoba *game Sudoku Sharaf* yang dibagi berdasarkan pemahaman *game*, minat *game* dan minat belajar. Untuk fitur yang merupakan bagian dari pemahaman *game* adalah pertanyaan tentang halaman utama, halaman bantuan, halaman *Sharaf*, pilih wazn, pilih fi'il, pilih shighah, isi puzzle dan menu selesaikan yang merupakan menu solver permainan. Pada parameter penilaian pemahaman *game*, rata-rata prosentase baik dari responden sebanyak 86%.

Pada parameter penilaian minat *game* yang berisi pertanyaan kebosanan, ketertarikan dan keinginan memainkan *game* lagi, didapat prosentase penilaian baik dari responden sebanyak 84%. Untuk parameter penilaian minat belajar didapat prosentase sebanyak 81% penilaian baik dari responden.

Dari data hasil pengujian *game Sudoku Sharaf* diatas, dapat disimpulkan bahwa *game* ini dapat diterima oleh siswa Madrasah Tsanawiyah. Prosentase penilaian baik oleh responden menjadi bukti bahwa *game* ini dapat menarik para siswa untuk memainkannya.

Para siswa mampu mengerti dalam menerapkan navigasi *game* mulai dari halaman utama sampai pada penyelesaian permainan ini. Dari sisi minat untuk

memainkan *game* ini juga cukup besar, karena hanya 27% dari para siswa yang merasakan kebosanan untuk memainkan *game* ini.

Game Sudoku Sharaf juga mampu menjadi media pembelajaran ilmu *Sharaf* para siswa. Mulai dari pengenalan definisi ilmu *Sharaf*, pembagian *wazn*, *shighah* dan pelafalan *shighah* yang benar. Para siswa yang merasa lebih mudah menghafal *wazn* melalui permainan *Sudoku Sharaf* mencapai 83%.

Para siswa juga merasa lebih mudah menghafal *shighah* dari *fi'il* yang dipilih melalui *game Sudoku Sharaf* ini, hal tersebut ditunjukkan dengan prosentase responden yang merasa lebih mudah mencapai 80%. Sedangkan untuk pelafalan *shighah*, para siswa juga mendapatkan pembelajaran tentang pelafalan *shighah* yang benar. Responden yang merasa terbantu dengan pelafalan *shighah* dari *game Sudoku Sharaf* mencapai 80%.

Jadi, *game Sudoku Sharaf* dapat menjadi salah satu media pembelajaran Ilmu *Sharaf* dengan total rata-rata penilaian baik responden yang mencapai 84% terhadap *game Sudoku Sharaf*.

4.4 Integrasi *Game Sudoku Sharaf* dengan Islam

Al-Qur'an sebagai kitab suci umat islam memegang peran penting sebagai wahyu Ilahi yang harus menjadi pedoman hidup umat islam. Al-Qur'an diturunkan Allah SWT dengan menggunakan bahasa arab agar manusia memahaminya. Seperti firman Allah SWT sebagai berikut:

إِنَّا جَعَلْنَاهُ قُرْءَانًا عَرَبِيًّا لَّعَلَّكُمْ تَعْقِلُونَ ﴿٢٦﴾

“*Sesungguhnya Kami menjadikan Al Quran dalam bahasa Arab supaya kamu memahaminya*” (QS. Az-Zukhruf: 3).

Berdasarkan arti ayat tersebut, sudah jelas bahwa apabila umat islam ingin memahami kandungan dalam AL-Quran, maka mempelajari bahasa arab adalah konsekuensinya.

Dalam salah satu hadits yang diriwayatkan At-Thabrani pada kitab *Majma' Al-Bayan Fi Tafsir Al-Qur'an* halaman 206 , Rasulullah SAW berkata:

أَحِبُّوا الْعَرَبَ لِثَلَاثٍ: لِأَنِّي عَرَبِيٌّ وَالْقُرْآنَ عَرَبِيٌّ وَكَلَامَ أَهْلِ الْجَنَّةِ عَرَبِيٌّ

"*Cintailah bahasa Arab karena tiga hal, yaitu bahwa aku adalah orang Arab, bahwa al-Quran adalah bahasa Arab, dan bahasa penghuni syurga di dalam syurga adalah bahasa Arab.*" (HR. at-Thabrani).

Dari arti hadits diatas, jelas sekali bahwa mempelajari bahasa arab sangat penting karena 3 hal yang Rasulullah sebutkan:

1. Nabi Muhammad SAW adalah keturunan arab
2. Al-Qur'an diurunkan dalam bahasa arab
3. Bahasa Ahli Surga adalah bahasa arab.

Mempelajari bahasa arab tentunya dimulai secara bertahap, karena ilmu yang harus dikuasai untuk memahami bahasa arab tidak sedikit. Salah satu ilmu yang menjadi dasar untuk mempelajari bahasa arab adalah ilmu *Sharaf*. Dengan mempelajari ilmu *Sharaf*, maka akan membantu dalam pemahaman akan perubahan kata dalam bahasa arab.

Berbagai macam cara dapat dilakukan untuk mempelajari ilmu *Sharaf*, salah satunya melalui permainan *Sudoku Sharaf*. Permainan yang berbasis

Android tersebut merupakan permainan *Sudoku* yang menjadikan *shighah* sebagai pengganti angka untuk diisi dalam kotak kosong. Jadi, sebagaimana penjelasan hadits Rasulullah tentang pentingnya belajar bahasa arab, maka permainan ini dapat memberikan pengenalan sekaligus pembelajaran ilmu *sharaf*.



BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil implementasi dan uji coba yang telah peneliti lakukan dapat disimpulkan bahwa Algoritma *Linear Congruential Generator* dan *Branch And Bound* merupakan algoritma yang mampu dijadikan sebagai pembangkit dan penyelesai *puzzle Sudoku Sharaf*. Hal ini terbukti dengan pembangkitan *puzzle* yang dinamis baik pada level mudah, sedang maupun sulit. Begitu juga untuk *solver* permainan yang mampu memberikan solusi untuk menyelesaikan permainan dengan waktu pencarian solusi yang berbeda-beda sesuai dengan tingkatan level permainan.

Pembelajaran *tashrif* pada *Ilmu Sharaf* dapat dilakukan dengan menggunakan aplikasi ini karena hasil kuesioner uji coba di Madrasah Tsanawiyah Hidayatul Mubtadi'in Malang menyimpulkan bahwa, hasil kuesioner pada 30 siswa menunjukkan 84% responden menyatakan penilaian baik pada aplikasi ini. Oleh karena itu, aplikasi *Sudoku Sharaf* dapat dijadikan media pembelajaran ilmu *sharaf*.

5.2 Saran

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

1. Mengembangkan aplikasi *Sudoku Sharaf* dengan lebih menarik, baik dari segi tampilan maupun alur permainan, khususnya sebaga media pembelajaran.
2. Mengembangkan aplikasi *Sudoku Sharaf* 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)
- Al-Mahalli, Jalaluddin Muhammad dan As-Suyuthi, Jalaluddin Abdurrahman. 2010. *Tafsir Jalalain*. Surabaya: Pustaka elBa.
- Aprilia, Shieny. 2007. *Aplikasi Algoritma Branch And Bound Untuk Menyelesaikan Integer Programming*. Program Studi Teknik Informatika Institut Teknologi Bandung. Bandung.
- Farida, Luluk. 2008. *Mengenal Kata Dalam Bahasa Arab*. Yogyakarta: Aziziyah.
- Grace, Lindsay. 2005. *Game Type and Game Genre*. diakses 30 April 2013.
- Hasan, Ahmad Bin. 2010. *Kitab Tasrif Juz Pertama*. Bangil: Raihan.
- H S, Simon. 2007. *Pencarian Solusi Sudoku dengan Algoritma Berbasis Branch and Bound*. Departemen Teknik Informatika Institut Teknologi Bandung. Bandung.
- Idris, Muhammad Jauhari. 2012. *Al-Qawaa'id Ash-Sharfiyyah*. Sumenep: Al-Amien Press
- Ismail, Andang. 2006. *Education Games*. Yogyakarta: Pilar Media.
- L, Riani. 2010. *Pembangkit Bilangan Acak*. Mata Kuliah Pemodelan & Simulasi. Jurusan Teknik Informatika Universitas Komputer Indonesia. Bandung.
- Mahtarami, Affan dan Ifansyah, Noor , M. 2010. *Pengembangan Game Pembelajaran Otomata Finit*. Seminar Nasional Informatika. Jurusan Teknik Informatika. Yogyakarta: Universitas Islam Indonesia.
- Ma'shum, Muhammad. 1965. *Al-Amtsilatut Tashrifiyah*. Semarang: Maktabah Alawiyah.
- Mufarakhah, Eva Rizqiyatul. 2011. *Rancang Bangun Game Dasar-Dasar Kitab Al-Amtsilatut Tashrifiyah*. Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim. Malang.
- Omimura, Satsuko. 2009. *222 Puzzle Sudoku Untuk Melejitkan Potensi Kreatif Otak Anda*. Jakarta: Prestasi Pustakaraya.

- Ramadhan, Andresta. Tanpa Tahun. *Perbandingan Algoritma Linear Congruential Generators, BlumBlumShub, dan MersenneTwister Untuk Membangkitkan Bilangan Acak Semu*. Program Studi Teknik Informatika Institut Teknologi Bandung. Bandung.
- Safaat, Nazruddin. 2011. *Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika.
- 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.
- Shihab, Quraish. 2009. *Tafsir Al-Mishbah*. Jakarta: Lentera Hati.
- Sidabutar, Juandi. 2008. *Pengembangan Algoritma Pencarian Untuk Menyelesaikan Problema Program Taklinier Integer Campuran*. Sekolah Pasca Sarjana Universitas Sumatera Utara. Medan.
- Sumitro, Yoel Krisnanda. 2007. *Penyusunan Jarkom HMIF ITB dengan Menggunakan Algoritma Branch and Bound*. Departemen Teknik Informatika Institut Teknologi Bandung. Bandung.
- Syam, Umi Fadilah Wardati. 2007. *Perumusan Sudoku Dengan Solusi Unik Menggunakan Algoritma Percabangan dan Pembatasan (Branch and Bound)*. Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung. Bandung.
- Thabrasi, Aminuddin Abu Ali Al-Fadhl ibn Hasan ibn Al-Fadhl. 1997. *Majma' Al-Bayan Fi Tafsir Al-Qur'an Jilid-2*. Beirut: Dar Al Kutub Al-Ilmiyah.
- Tim EMS. 2012. *Panduan Cepat Pemrograman Android*. Jakarta: Elex Media Komputindo.
- Zaman, Saiful. Dyah R Helmi. 2009. *12 Permainan Untuk Meningkatkan Intelligensi Anak*. Jakarta: Visimedia.

LAMPIRAN

Data Responden Uji Coba

No	Nama	Kelas	Sekolah
1	M. Puji Haryono	VIII C	MTs. Hidayatul Mubtadi'in
2	Agus Susilo	VIII C	MTs. Hidayatul Mubtadi'in
3	Anjas Kurniawan	VIII C	MTs. Hidayatul Mubtadi'in
4	Singgih Arif Setyanto	VIII C	MTs. Hidayatul Mubtadi'in
5	Dimas Andre Pratama	VIII C	MTs. Hidayatul Mubtadi'in
6	M. Fafan w	VIII C	MTs. Hidayatul Mubtadi'in
7	Ahmad Ghufron	VIII C	MTs. Hidayatul Mubtadi'in
8	M. Alfian Aziz	VIII C	MTs. Hidayatul Mubtadi'in
9	Ahmad Zainuri	VIII C	MTs. Hidayatul Mubtadi'in
10	Maulid	VIII C	MTs. Hidayatul Mubtadi'in
11	Angga Prasetyo	VIII C	MTs. Hidayatul Mubtadi'in
12	Dwi Megawati	VIII C	MTs. Hidayatul Mubtadi'in
13	Alif Lathifah	VIII C	MTs. Hidayatul Mubtadi'in
14	Khoirunnisa'	VIII C	MTs. Hidayatul Mubtadi'in
15	Dewi nur fita	VIII C	MTs. Hidayatul Mubtadi'in
16	Miftahul Jannah	VIII C	MTs. Hidayatul Mubtadi'in
17	M. Saifuddin	VIII C	MTs. Hidayatul Mubtadi'in
18	Robiansyah	VIII C	MTs. Hidayatul Mubtadi'in
19	Ahmad Febri	VIII C	MTs. Hidayatul Mubtadi'in
20	Siti Maria Ulfah	VIII C	MTs. Hidayatul Mubtadi'in
21	Ambar	VIII C	MTs. Hidayatul Mubtadi'in
22	Eny	VIII C	MTs. Hidayatul Mubtadi'in
23	Rosalinda Putri W	VIII C	MTs. Hidayatul Mubtadi'in
24	Indri	VIII C	MTs. Hidayatul Mubtadi'in
25	Sefti Aeni R.A	VIII C	MTs. Hidayatul Mubtadi'in
26	Ina Sulfiah	VIII C	MTs. Hidayatul Mubtadi'in
27	Isnaini Sevia Putri	VIII C	MTs. Hidayatul Mubtadi'in
28	Izzatul Lutfiah	VIII C	MTs. Hidayatul Mubtadi'in
29	Fitri Aisyah	VIII C	MTs. Hidayatul Mubtadi'in
30	Sri Wahyuni	VIII C	MTs. Hidayatul Mubtadi'in