APLIKASI GAME PUZZLE SUDOKU DENGAN ALGORITMA BACKTRACKING

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika



Oleh :

Nama	:	Kurnia Wahyuni S Putri
No. Mahasiswa	:	06 523 236

TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2011

LEMBAR PENGESAHAN PEMBIMBING

APLIKASI GAME PUZZLE SUDOKU DENGAN ALGORITMA BACKTRACKING

TUGAS AKHIR

Oleh :

Nama : **Kurnia Wahyuni S Putri**

No. Mahasiswa : **06 523 236**

Yogyakarta, 27 April 2011

Pembimbing

Ami Fauzijah, ST., MT.

LEMBAR PERNYATAAN KEASLIAN

HASIL TUGAS AKHIR

Saya yang bertanda tangan dibawah ini,

Nama : Kurnia Wahyuni S Putri

No. Mahasiswa : 06 523 236

Menyatakan bahwa semua komponen dan isi dalam laporan Tugas Akhir ini adalah hasil karya saya sendiri. Apabila kemudian hari terbukti bahwa ada beberapa bagian dari karya ini bukan hasil karya saya sendiri, maka saya siap menanggung resiko dan konsekuensi apapun.

Demikianlah pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 27 April 2011

Kurnia Wahyuni S Putri

LEMBAR PENGESAHAN PENGUJI

APLIKASI GAME PUZZLE SUDOKU DENGAN ALGORITMA BACKTRACKING

TUGAS AKHIR

Oleh :

Nama : Kurnia Wahyuni S Putri

No. Mahasiswa : 06 523 236

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta,

Tim Penguji

Ami Fauzijah, ST., MT. .....
Ketua

Beni Suranto, ST. .....
Anggota 1

Affan Mahtarami, S.Kom., MT. .....
Anggota II

Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia

(Yudi Prayudi, S.Si, M.Kom)

PERSEMBAHAN

Rasa puji syukur sedalam dalamnya saya panjatkan kepada **ALLAH SWT**… yang telah memberikan nikmat dan hidayah_Nya sehingga sehingga skripsi ini terselesaikan dengan baik…

Sholawat dan Salam tak lupa saya haturkan kepada **NABI MUHAMMAD SAW**…karena beliau semoga saya dapat menjadi orang yang selalu benar dalam melangkah dan di ridhoi oleh Allah….

Papa dan **Mama**…terima kasih atas kasih sayang yang selama ini diberikan kepadaku, terima kasih atas lautan maaf yang telah diberikan kepadaku,
terima kasih atas doa, nasehat serta dukungan yang telah kalian berikan baik materiil, moral maupun spiritual sehingga menjadikanku sebagai anak yang lebih baik…

Abang dan adek serta **semua keluarga besarku**…terima kasih atas doa, bantuan dan support yang telah kalian berikan kepadaku…

Teman, **Kawan**, **Sahabat**…terima kasih atas persahabatan yang terjalin selama ini dan terima kasih atas support dan motivasi yang telah kalian berikan…

MOTTO

“ Sesungguhnya Allah SWT akan membantu orang-orang yang berusaha, sekali pun ia tidak memiliki kekuatan dan kemampuan, melainkan kemauan yang kuat serta niat yang tulus dan ikhlas”

“Bekerjalah untuk duniamu seakan akan kamu akan hidup selamanya, dan beribadah lah untuk akhiratmu seakan akan kamu akan mati besok“
(Al Hadist)

“Di dalam masalah, terdapat emas dan salah.
Ambillah emasnya, engkau akan berhasil”
(Anonim)

“Bahwa tiada yang orang dapatkan, kecuali yang ia usahakan”
(Q.S. 53, Surat An Najm : 39)

“Sesungguhnya sesudah kesulitan itu ada kemudahan, maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain”
(Q.S. Asy Syarh ayat 6 dan 7)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamualaikum Wr.Wb.

Alhamdulillaahirabbil'aalamin, segala puji bagi Allah SWT atas segala rahmat dan hidayah yang diberikanNya, sehingga penulisan laporan tugas akhir yang berjudul "Aplikasi Game Puzzle Sudoku dengan Algoritma Backtracking" dapat penulis selesaikan dengan baik. Shalawat serta salam juga dipanjatkan kepada Rasulullah SAW, keluarganya, sahabatnya dan orang-orang yang memberikan loyalitas kepadanya.

Laporan tugas akhir ini disusun sebagai salah satu syarat guna memperoleh gelar Sarjana Teknik Informatika pada Universitas Islam Indonesia. Dan juga sebagai sarana untuk mempraktekkan secara langsung ilmu dan teori yang telah diperoleh selama menjalani masa studi di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Sehubungan dengan terselesaikanya penyusunan tugas akhir ini, penyusun mengucapkan banyak terima kasih kepada pihak-pihak yang telah memberikan dukungannya baik secara langsung maupun tidak. Dengan penuh rasa syukur penulis ucapkan terima kasih kepada :

1 Bapak Ir. Gumbolo Hadi Susanto, M.Sc., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.

2 Bapak Yudi Prayudi, S.Si., M.Kom., selaku Ketua Jurusan Teknik Informatika.

3 Ibu Ami Fauzijah, ST., MT. selaku Dosen Pembimbing, yang telah banyak memberikan masukkan, bimbingan, bantuan serta waktunya dalam penyelesaian tugas akhir dan laporan tugas akhir ini.

4 Keluargaku tercinta, Papa, Mama, Kakak serta adikku yang telah memberikan doa, kasih sayang dan dukungan sehingga tugas akhir ini terselesaikan dengan baik.

5 Untuk Mas Arif sebagai tempat bertanya, terima kasih banyak atas bantuan dan masukkanya.

6 Untuk para sahabatku Kurnia Arivanty, Riska Amalia, Andi Fadlun Mulyani terima kasih atas bantuan kalian selama pelaksanaan tugas akhir ini.

7 Teman-teman Teknik Informatika UII seluruhnya dan semua pihak yang tidak dapat saya sebutkan satu per satu.

Semoga Allah SWT melimpahkan rahmat dan hidayahnya kepada semua pihak yang telah membantu terselesaikannya penulisan laporan tugas akhir ini. Penulis menyadari bahwa dalam penyusunan laporan tugas akhir ini masih banyak terdapat kekeliruan dan kekurangan. Untuk itu penulis menyampaikan permohonan maaf sebelumnya serta sangat diharapkan kritik dan saran yang sifatnya membangun untuk penyempurnaan laporan tugas akhir ini dimasa mendatang. Harapan saya semoga karya ini dapat menjadi sumbangan yang

berarti bagi kampus tercinta Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta.

Akhir kata semoga laporan ini dapat memberikan manfaat bagi penulis dan semua pembaca.

Wassalamu'alaikum Wr.Wb.

Yogyakarta, 27 April 2011

Penyusun

ABSTRAKSI

Seiring dengan perkembangan teknologi, fungsi komputer juga mengalami perkembangan untuk dapat menjalankan aplikasi *game*. *Game* menjadi sangat beragam jenisnya dan populer karena minat dari *user* sendiri. Saat ini sudah banyak sekali jenis *game puzzle* yang ada, salah satunya yaitu *game* Sudoku yang pada bidang informatika termasuk kedalam permasalahan yang sulit untuk dipecahkan. Dan salah satu cara untuk menyelesaikan permasalahan *game puzzle* Sudoku ini yaitu dengan menggunakan algoritma runut-balik (*backtracking*).

"Aplikasi Game Puzzle Sudoku dengan Algoritma Backtracking" ini dibangun dengan menggunakan bahasa pemrograman Visual Basic 6.0 dan microsoft Access 2003 sebagai database. Pada aplikasi ini juga terdapat penerapan algoritma runut-balik (*backtracking*) dalam penyelesaian permainan sudoku. Dengan menggunakan algoritma runut-balik, solusi dapat ditemukan lebih cepat tanpa harus mencoba semua kemungkinan solusi. Input pada aplikasi ini berupa angka yang dipilih oleh pemain untuk dimasukkan kedalam sel papan permainan. Sedangkan outputnya adalah skor yang diperoleh pemain saat menyelesaikan permainan.

Hasil penelitian dan pengujian menunjukkan bahwa aplikasi ini sudah memiliki kriteria game yang baik. Aplikasi game puzzle sudoku ini sudah mampu menerapkan algoritma backtracking pada proses penyelesaiannya dan memberikan hasil akhir berupa skor yang diperoleh pemain saat menyelesaikan permainan.

Kata kunci : Sudoku, Backtracking, *Game puzzle*

TAKARIR

Sudoku	Jenis permainan logika angka ringan yang dimainkan dengan cara mengisikan angka 1-9 pada kotak kosong sehingga semua kotak kosong tersebut terisi semua dengan ketentuan pada masing-masing kolom, baris dan subgrid tidak ada angka yang sama.
Grid	Keseluruhan kotak pada papan permainan yang terdiri dari 9 x 9 kotak.
Subgrid	Bagian dari *grid* yang terdiri dari 3 x 3 kotak, dimana 3 x 3 *subgrid* merupakan satu *grid*.
Gameboard	Papan permainan dimana permainan akan berlangsung.
Beginner	Tingkat kesulitan yang diperuntukkan kepada pemain pemula.
Intermediate	Tingkat kesulitan yang diperuntukkan kepada pemain yang sudah cukup mahir.
Advance	Tingkat kesulitan yang diperuntukkan kepada pemain yang sudah mahir.
Algoritma Backtracking	Algoritma runut-balik. Dimana pencarian solusi suatu permasalahan hanya dengan mempertimbangkan pencarian yang mengarah ke solusi saja.

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING i

LEMBAR PERNYATAAN KEASLIAN ii

LEMBAR PENGESAHAN PENGUJI iii

HALAMAN PERSEMBAHAN iv

HALAMAN MOTTO v

KATA PENGANTAR vi

SARI..... ix

TAKARIR..... x

DAFTAR ISI... xi

DAFTAR TABEL xiv

DAFTAR GAMBAR xv

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 2

1.3 Batasan Masalah 3

1.4 Tujuan Penelitian 3

1.5 Manfaat Penelitian 4

1.6 Metodologi Penelitian 4

1.7 Sistematika Penulisan 4

BAB II LANDASAN TEORI 6

2.1 Sejarah Game 6

2.1.1 Jenis-jenis Game 8

2.1.2 Sudoku 10

2.2 Visual Basic 6.0 17

2.3 Microsoft Access 18

2.4 Teori dalam Pembuatan Game 19

BAB III METODOLOGI 22

3.1 Analisis Sistem 22

3.2 Hasil Analisis 24

3.3 Perancangan Perangkat Lunak 27

3.3.1 Perancangan Diagram Alir (*Flowchart*) 27

3.3.2 Perancangan Antarmuka 36

3.4 Implementasi Perangkat Lunak 47

3.4.1 Implementasi 47

3.4.2 Aturan Permainan 57

BAB IV HASIL DAN PEMBAHASAN 59

4.1 Hasil Program 59

4.2 Analisis Kinerja Sistem 59

4.2.1 Proses pada Halaman Papan Permainan 59

4.2.2 Proses pada Halaman Regoster 66

4.2.3 Proses pada Halaman Menu 68

4.3 Pengujian Data Analisis 70

4.4 Kelebihan dan Kekurangan Sistem 72

4.4.1 Kelebihan Sistem 72

4.4.2 Kekurangan Sistem 73

4.5 Analisis Perbandingan 73

BAB V PENUTUP 76

5.1 Kesimpulan 76

5.2 Saran 77

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 4.1 Hasil Analisis Responden yang Belum Pernah Memainkan *Game* Sudoku 71

Tabel 4.2 Hasil Analisis Responden yang Sudah Pernah Memainkan *Game* Sudoku 71

Tabel 4.3 Hasil Perbandingan 74

DAFTAR GAMBAR

Gambar 3.1 Flowchart Sistem 29

Gambar 3.2 Flowchart Pemilihan Papan Permainan 30

Gambar 3.3 Flowchart Proses *Undo* 30

Gambar 3.4 Flowchart Proses *Check* 32

Gambar 3.5 Flowchart Proses *Hint* 33

Gambar 3.6 Flowchart Proses *Solve* 35

Gambar 3.7 Flowchart Proses Registrasi 36

Gambar 3.8 Rancangan Antarmuka Main Menu 37

Gambar 3.9 Rancangan Antarmuka Pemilihan Type Grid 38

Gambar 3.10 Rancangan Antarmuka Papan Permainan 4 x 4 39

Gambar 3.11 Rancangan Antarmuka Papan Permainan 6 x 6 40

Gambar 3.12 Rancangan Antarmuka Papan Permainan 9 x 9 42

Gambar 3.13 Rancangan Antarmuka Menu 43

Gambar 3.14 Rancangan Antarmuka Top Score 44

Gambar 3.15 Rancangan Antarmuka Register 45

Gambar 3.16 Rancangan Antarmuka How To Play 46

Gambar 3.17 Rancangan Antarmuka Credit 46

Gambar 3.18 Halaman Main Menu 49

Gambar 3.19 Halaman Pilih Grid 50

Gambar 3.20 Halaman papan permainan 4 x 4 51

Gambar 3.21 Halaman papan permainan 6 x 6 51

Gambar 3.22 Halaman papan permainan 9 x 9 52

Gambar 3.23 Halaman Menu 54

Gambar 3.24 Halaman Top Score 55

Gambar 3.25 Halaman Register..... 56

Gambar 3.26 Halaman How to Play 56

Gambar 3.27 Halaman Credit..... 57

Gambar 4.1 Pilihan angka pada papan permainan 4x4..... 60

Gambar 4.2 Pilihan angka pada papan permainan 6x6..... 60

Gambar 4.3 Pilihan angka pada papan permainan 9x9..... 60

Gambar 4.4 Angka jawaban 61

Gambar 4.5 Angka penanda sementara..... 61

Gambar 4.6 Penghapusan angka jawaban..... 61

Gambar 4.7 Penghapusan angka penanda sementara 62

Gambar 4.8 Pesan pemberitahuan permainan telah diselesaikan 62

Gambar 4.9 Pesan konfirmasi *solve*..... 63

Gambar 4.10 Pesan pemberitahuan *solve* selesai 63

Gambar 4.11 Pesan konfirmasi *solve game* dari awal 63

Gambar 4.12 Pesan pernyataan game tidak dapat diselesaikan 64

Gambar 4.13 Proses *hint* 65

Gambar 4.14 Pesan pemberitahuan ada angka yang sama..... 65

Gambar 4.15 Pesan pemberitahuan pengisian sudah benar 65

Gambar 4.16 Proses *Undo*..... 66

Gambar 4.17 Pengisian nama..... 67

Gambar 4.18 Penghapusan nama..... 67

Gambar 4.19 Pesan konfirmasi tidak menyimpan skor 68

Gambar 4.20 Pesan konfirmasi keluar dari permainan..... 69

Gambar 4.21 Pesan konfirmasi mengulang *game* 70

Gambar 4.22 Tampilan antarmuka *game* Hodoku..... 73

Gambar 4.23 Tampilan antarmuka *main* menu *game* Sudoku Pagoda..... 73

Gambar 4.24 Tampilan antarmuka pilih *grid game* Sudoku Pagoda..... 74

Gambar 4.25 Tampilan antarmuka permainan *game* Sudoku Pagoda..... 75

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sudoku adalah permainan (teka-teki) logika angka yang ringan yang dapat dimainkan semua orang yang sudah bisa berhitung dari 1 sampai 9. Dimana ada 9 x 9 *grid* yang dibagi lagi menjadi 9 *subgrid* yang berisi 3 x 3 kotak. Pada 9 x 9 kotak tersebut diisikan beberapa angka yang acak, yang nantinya pemain diminta mengisi semua kotak yang belum terisi angka dengan ketentuan:

1. Tiap *subgrid* harus diisi angka 1-9 dengan tidak ada angka yang sama
2. Tiap baris harus diisi angka 1-9 dengan tidak ada angka yang sama
3. Tiap kolom harus diisi angka 1-9 dengan tidak ada angka yang sama

Saat ini permainan Sudoku adalah salah satu *puzzle* angka yang paling banyak digemari. Selain menjadi hiburan yang menantang, *game* ini ternyata juga memberi manfaat yang baik bagi yang memainkan, diantaranya yaitu merangsang sel otak agar selalu aktif, meningkatkan ketelitian dan logika berpikir, menambah kemampuan memecahkan masalah dengan cepat, dan mempertajam imajinasi, kreativitas, serta inovasi. Dengan demikian selain dapat menghibur, bermain *game puzzle* Sudoku ini dapat membantu kerja otak agar lebih baik.

Selain digemari, *game* Sudoku juga merupakan salah satu permasalahan yang cukup sulit di bidang informatika. Permasalahan *puzzle* Sudoku ini sulit

untuk dipecahkan karena termasuk dalam permasalahan *NP-complete*, sehingga tidak bisa diselesaikan dalam waktu yang sama. Hingga saat ini banyak programmer yang mencari algoritma yang tepat untuk menyelesaikan *puzzle* ini. Salah satu cara untuk menyelesaikan *game* ini yaitu dengan menggunakan algoritma *Backtracking* (runut-balik). Algoritma ini merupakan perbaikan dari algoritma *Brute Force*, dimana solusi dapat ditemukan dengan penelusuran yang lebih sedikit dan dapat mencari solusi permasalahan secara lebih mangkus karena tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang perlu dipertimbangkan.

Karena alasan inilah penulis terdorong untuk mencoba membuat suatu aplikasi *game puzzle* Sudoku dengan menerapkan algoritma *Backtracking*. Dan mengacu pada judul tugas akhir yang pernah diterima sebelumnya, yaitu Aplikasi Game Sudoku dengan Menggunakan Pemprograman J2ME pada Perangkat Mobile, aplikasi *game* sudoku yang penulis buat akan lebih mengutamakan kepada interface dan fitur-fiturnya, seperti skoring dan batas waktu, dimana tidak terdapat pada aplikasi yang pernah dibuat sebelumnya.

1.2 Rumusan Masalah

Bagaimana membuat *Game Puzzle* Sudoku yang menarik dan memiliki fitur skoring serta batas waktu yang dapat menambah kesulitan bagi *user* dalam menyelesaikan *puzzle* sudoku tersebut.

1.3 Batasan Masalah

Adapun batasan masalah yang ditetapkan adalah :

1. Menggunakan sistem skoring
2. Terdapat batas waktu dalam menyelesaikan permainan
3. Hanya dapat dimainkan oleh satu orang saja dalam satu waktunya (single player)
4. Dapat menentukan jumlah grid puzzle yang akan dimainkan, yaitu 4x4, 6x6 , atau 9x9
5. Terdapat tiga tingkat kesulitan, yaitu *beginner*, *intermediate*, dan *advance*, dimana dibedakan dari jumlah angka yang diberikan pada awal permainan. Semakin sulit tingkat kesulitannya, semakin sedikit jumlah angka yang diberikan pada awal permainan.

1.4 Tujuan Penelitian

Adapun tujuan akhir dari penelitian tugas akhir ini adalah merealisasikan sebuah program aplikasi berupa *game puzzle* sudoku yang memiliki *interface* yang menarik serta memiliki fitur-fitur yang dapat membantu dan menambah ketegangan dalam bermain. Aplikasi *game puzzle* Sudoku ini akan dibuat dengan menggunakan bahasa pemrograman Visual Basic 6.0.

1.5 Manfaat Penelitian

Manfaat penelitian tugas akhir ini adalah memberikan alternatif hiburan sekaligus mampu melatih mengembangkan kreatifitas, logika, dan nalar dalam bentuk *game* sudoku kepada penggunanya.

1.6 Metodologi Penelitian

Untuk melancarkan penelitian tersebut, maka perlu dilakukan hal-hal sebagai berikut:

1. Analisis kebutuhan, meliputi penenetuan data dan *tools* yang dibutuhkan dalam pembuatan aplikasi ini, yang meliputi kebutuhan *input,* proses, dan *output*, serta antarmuka yang diinginkan.
2. Perancangan, meliputi pembuatan desain yang akan diimplementasikan pada program ini berdasarkan data dan *tools* yang telah dianalisis dalam bahasa yang dimengerti *user*
3. Pemrograman, meliputi penerjemahan masalah yang telah dirancang ke dalam bahasa pemrograman yang telah ditentukan
4. Pengujian program yang telah jadi sebagai bentuk uji coba terhadap program yang telah selesai dibuat, untuk menemukan kesalahan-kesalahan yang masih terdapat pada program tersebut.

1.7 Sistematika Penulisan

Sistem penulisan laporan berguna untuk memberikan gambaran umum dari keseluruhan isi laporan. Penulisan laporan ini dibagi dalam beberapa bab, yaitu :

Bab I berisi pendahuluan yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab II berisi landasan teori yang meliputi gambaran umum tentang teori yang diterapkan dalam pembuatan *software* ini. Dalam bab ini dijelasakan tentang teori-teori Visual Basic 6.0.

Bab III berisi metodologi yang menjelaskan tentang berbagai langkah dan metode yang digunakan dalam pengaplikasian *game* sudoku yang dibangun

Bab IV berisi hasil dan pembahasan yang menjelaskan tentang kerja sistem dan beberapa *capture* dari sistem yang dibangun

Bab V berisi penutup yang meliputi kesimpulan dan saran.

BAB II

LANDASAN TEORI

2.1 Sejarah Game

Sejarah teknologi *game* komputer secara langsung berhubungan dengan perkembangan komputer itu sendiri [NUR08]. Komputer dengan kecepatan *processor* tinggi, grafis yang lebih mendekati realita, dan media penyimpanan yang lebih besar sebenarnya dimaksudkan untuk memenuhi kebutuhan dalam bermain *games.*

Arti dari *game* (*game* komputer) sendiri adalah sebuah permainan interaktif yang membutuhkan komputer untuk bermain. Program komputer menerima input dari si pemain melalui pengendali dan menampilkan lingkungan buatan melalui TV atau layar monitor.

Game generasi pertama

Tahun 1952, di Universitas Cambridge, A.S Douglas menulis sebuah tesis untuk gelar sebuah PhD-nya mengenai interaksi antara komputer dan manusia. Dalam tesisnya itu ia menciptakan *game* komputer dari sebuah permainan tradisional bernama Tic-Tac-Toe. *Game* ini diprogram dengan memakai komputer *EDVAC vacuum tube* yang memiliki *layer* berupa *cathode ray tube* (*CRT*).

Kemudian di tahun 1958. William Haginbotham menciptakan video *game* pertamanya. Berbeda dengan Douglas, video *game* pertamanya yang berjudul Tennis for Two diciptakan dan dimainkan di *osciloscope.*

Tapi *game* komputer pertama yang benar-benar diciptakan menggunakan komputer sungguhan adalah *Spacewar*. Saat itu di tahun 60-an, komputer tergolong barang sangat mewah dan biasanya komputer dipakai untuk kepentingan riset dalam dunia militer. Tapi seseorang bernama Steve Russel memiliki ketertarikan akan hal lain. Dia dan teman-temannya sangat menyukaikisah fiksi ilmiah berjudul *Skylark* karangan Edward E Smith. Dari situ mereka membuat sebuah *game* bernama *Spacewar*. Kebetulan juga, Steve Russel bekerja menggunakan sebuah komputer *mainframe* bernama *MIT PDP-1* yang biasa dipakai untuk perhitungan statistik. Dengan komputer itulah dia membuat *Spacewar* di tahun 1961.

Game generasi kedua

Di tahun 1971, Nolan Bushnell bersama dengan Ted Dabney menciptakan *game* ber-*genre* arcade yang pertama. Dinamai Komputer *Space*, *game* itu didasari oleh *Spacewar*. Tahun 1972, Nolan dan Ted memulai Atari komputer. Kemudian dia mengembangkan *game* berjudul *Pong* yaitu *game* pertama yang tersedia untuk publik. Karena *game-game* sebelumnya hanya ada di dalam komputer *mainframe* untuk kesenangan sendiri saja. Asal usul Pong dimulai saat Nolan ingin membuat *game* sederhana dan mudah dimengerti. Dengan memori dan *micro processor* kelas rendah, kemampuan proses yang terbatas dan grafis

yang sederhana, akhirnya dia membuat versi elektronik dari permainan ping pong yang kemudian menjadi Pong.

Pong kemudian berevolusi menjadi sebuah game bernama *Breakout. Game* itu diciptakan oleh Steve Jobs untuk Atari. Dari situ, Steve Jobs dan temannya Steve Wozniak mulai berpikir untuk menciptakan sebuah *Portable Computer* (PC). Kemudian mereka meminjam semua peralatan yang dipakai dalam proyek *Breakout* dan membuat sebuah *prototype* bernama Apple I yang merupakan cikal bakal dari komputer Apple Macintosh yang ada sekarang.

Pada tahun 1980, Atari mengeluarkan *game* berjudul *Asteroid* dan *Lunar Lander*. Kedua *game* tersebut adalah *game* pertama yang didaftarkan pada kantor hak cipta untuk mendapatkan paten. *Asteroid* merupakan *game* yang penuh inovasi baru dalam grafisnya. Daripada menggunakan metode *raster*, *game* ini merupakan grafis *vector line* seperti yang ada pada *osciloscope.*

Pengenalan Atari Video *computer System* (Atari 2600) dengan CPU biasa dan slot untuk kasetnya, menjadi suatu era baru dalam dunia game. Di tahun 1980 itu juga menunjukkan penjualan yang meningkat dari PC yang biasa dipakai untuk *game*.

Dan kini komputer bersaing dengan mesin konsol seperti PS atau Xbox. Dan komputer tidak hanya sebagai mesin untuk kepentingan bisnis semata, tapi juga untuk hiburan seperti *game*.

2.1.1 Jenis-jenis *Game* [PUT09]

a) Berdasarkan Jenis *platform* atau alat yang digunakan :

1. Arcade *games*, atau yang sering disebut ding-dong di Indonesia, biasanya berada di daerah / tempat khusus dan memiliki *box* atau mesin yang memang khusus di design untuk jenis video *games* tertentu dan tidak jarang bahkan memiliki fitur yang dapat membantu pemainnya dalam permainan tertentu, seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil.
2. PC *games*, yaitu video *game* yang dimainkan menggunakan *Personal Computers.*
3. *Console games*, yaitu video *games* yang dimainkan menggunakan *console* tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.
4. *Handheld game*s yaitu yang dimainkan di *console* khusus video *game* yang dapat dibawa kemana-mana, contoh Nintendo DS dan Sony PSP.
5. *Mobile games*, yaitu yang dapat dimainkan atau khusus untuk *mobile phone* atau PDA.

b) Berdasarkan *genre* permainannya :

1. Aksi-*Shooting* (tembak-tembakan, atau bisa juga perkelahian, tergantung cerita dalam *game* tersebut).
2. Fighting (pertarungan)
3. Aksi-Petualangan
4. Petualangan
5. Simulasi, Konstruksi dan Manajemen

6. Role Playing
7. Strategi
8. Puzzle

   Video *game* jenis ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, sampai mendorong-dorong kotak masuk ke tempat yang seharusnya, itu semua termasuk dalam jenis ini. Sering pula permainan jenis ini adalah juga unsur permainan dalam video *game* petualangan maupun game edukasi. Contohnya Sudoku, Tetris, Minesweeper, Bejeweled, Sokoban dan Bomberman.

9. Simulasi kendaraan
10. Olahraga

c) Berdasarkan kategori lainnya :

1. *Multiplayer Online*.
2. *Casual games*.
3. Edugames.
4. Advergames.

2.1.2 Sudoku

a) Sejarah Sudoku

Sudoku juga dikenal sebagai *Number Place* atau *Nanpure*, adalah sejenis teka-teki logika [MUS10]. Tujuannya adalah untuk mengisikan

angka-angka dari 1 sampai 9 ke dalam jaring-jaring 9×9 yang terdiri dari 9 kotak 3×3 tanpa ada angka yang berulang di satu baris, kolom atau kotak. *Sudoku* adalah sebuah *puzzle* yang didasarkan pada konsep *Latin Square*. *Latin Square* sendiri diperkenalkan pada tahun 1783 oleh Leonhard Euler, seorang matematikawan asal Swiss.

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 1986, ketika penerbit Nikoli menemukan teka-teki ini yang diciptakan Howard Garns seorang mantan arsitek yang meninggal 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.

Sudoku menjadi benar-benar mewabah di Inggris ketika *The Daily Telegraph* mengenalkannya pada pembacanya pada bulan Februari 2005. Media lain pun kemudian mengikuti dengan menyediakan permainan ini di edisinya masing-masing. Pertengahan Mei 2005 adalah awal demam *Sudoku*. Segala hal tentang *Sudoku* menjadi ladang uang, pemuatan di

koran harian, penerbitan buku, penerbitan majalah, pertunjukkan televisi, siaran radio, pelayanan langganan lewat *email*, dan penyediaan layanan di telepon genggam. Khusus untuk buku, *News* & *Star* mencatat bahwa 6 dari 10 buku nonfiksi yang paling laris saat ini adalah buku tentang *Sudoku*. Dari Inggris, *Sudoku* kemudian menyebar di daratan Eropa, dari Prancis sampai Slowakia, lalu menular pula ke Australia dan Amerika.

b) Aturan Permainan Sudoku

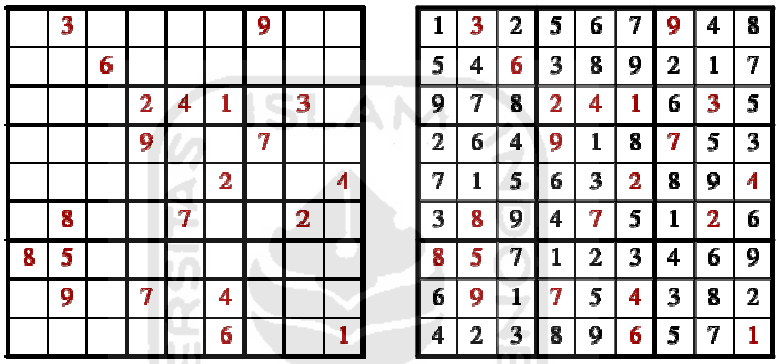
Aturan permainan untuk *puzzle* ini sangat sederhana, untuk menyelesaikan permainan ini tidak diperlukan pengetahuan umum, kepandaian atas bahasa tertentu, juga kemampuan matematika. Tetapi hanya memerlukan kecermatan, kesabaran, dan logika.

Papan *Sudoku* terbuat dari sembilan buah kotak berukuran 3×3 (disebut blok/ *subgrid*) yang disusun sedemikian rupa sehingga menghasilkan kotak besar berukuran 9×9. Beberapa kotak sudah diisi sebagai petunjuk awal dan tugas pemain adalah melengkapi angka-angka pada kotak yang lain sehingga keseluruhan papan permainan terisi angka secara lengkap. Aturan permainannya sangatlah sederhana:

1. Kotak-kotak pada setiap baris, kolom, dan blok/ *subgrid* harus berisi sebuah angka.
2. Angka-angka yang diisikan harus unik dari 1 hingga 9 sehingga dalam 1 blok/ *subgrid* hanya terdiri atas angka 1-9 yang tidak

berulang dan tidak ada angka yang berulang dalam 1 baris maupun kolom.

Angka-angka ini sebenarnya tidak memiliki hubungan aritmetis satu sama lain. Karena itu selain dapat menggunakan angka, bisa juga diganti dengan 9 huruf, lambang, atau warna yang berbeda. Contoh bentuk papan permainan sudoku dapat dilihat pada Gambar 2.1.



Gambar 2.1 Bentuk papan Sudoku

c) Strategi Umum Penyelesaian Teka-teki Sudoku

1. Pemindahan (*scanning*)

   Berupa proses memindahkan baris atau kolom untuk mengindentifikasi baris mana dalam suatu blok yang terdapat angka-angka tertentu. Proses ini kemudian diulang pada setiap kolom (atau baris) secara sistematis. Kemudian menentukan nilai dari suatu sel dengan membuang nilai-nilai yang tidak mungkin.

2. Penandaan (*marking*)

Berupa analisa logika, dengan menandai kandidat angka yang dapat dimasukkan dalam sebuah sel.

3. Analisa (*Analysing*)

   Berupa eliminasi kandidat, dimana kemajuan dicapai dengan mengeliminasi kandidat angka secara berturut-turut hingga sebuah sel hanya punya 1 kandidat.

d) Algoritma *Backtracking* (runut-balik)

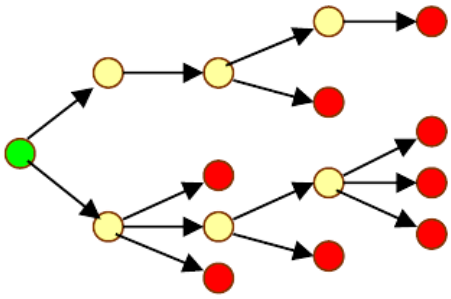
Algoritma *backtracking* pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950 menyajikan uraian umum tentang *backtracking* dan penerapannya dalam berbagai persoalan dan aplikasi. *Backtracking* adalah algoritma yang berbasis pada algoritma DFS (*Depth-First Search*) yang dapat mencari solusi sebuah persoalan dengan lebih mangkus [DIA09]. Algoritma ini dapat menemukan solusi sebuah persoalan tanpa perlu memeriksa semua kemungkinan solusi dan hanya mempertimbangkan pencarian yang mengarah kesolusi. Algoritma Backtracking merupakan algoritma yang berbasiskan DFS (Depth First Search). Yang dilakukan oleh algoritma ini adalah mencari kemungkinan solusi dengan menelusuri hingga node terdalam. Kemudian dilakukan perjalanan kembali dengan melalui node-node calon solusi yang telah dikunjungi untuk menemukan jalur solusi lain yang lebih sesuai. Dengan kata lain, algoritma ini akan melakukan pencarian solusi secara berurutan

dari jalur solusi satu ke jalur solusi lain. Namun akan berhenti bila solusi yang sesuai telah ditemukan.

Algoritma *backtracking* adalah suatu algoritma yang merupakan perbaikan dari algoritma *brute force*, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. *Backtracking* merupakan bentuk tipikal dari algoritma rekursif dan berbasis pada DFS dalam mencari solusi yang tepat. Selain itu, algoritma ini juga merupakan metode yang mencoba-coba beberapa keputusan sampai kita menemukan salah satu yang ”berjalan”. Kita tidak perlu memeriksa semua kemungkinan solusi yang ada, tetapi cukup yang mengarah kepada solusi saja. Dengan memangkas (*pruning*) simpul-simpul yang tidak mengarah ke solusi, sehingga waktu pencarian dapat dihemat. Algoritma ini banyak diterapkan untuk program games dan permasalahan pada bidang kecerdasan buatan.

e) Prinsip Pencarian Solusi dengan Metode *Backtracking*

Seperti yang telah dijelaskan bahwa pencarian solusi dengan menggunakan algoritma backtracking digunakan pohon ruang status. Cara kerjanya adalah dengan membentuk lintasan dari akar ke daun, seperti terlihat pada Gambar 2.2.



Gambar 2.2 Pohon ruang status Algoritma *Backtracking*

Langkah-langkah pencarian solusi pada pohon ruang status:

1. 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.
2. 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.
3. 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 *backtracking* ke simpul hidup terdekat (simpul orang tua). Selanjutnya simpul ini menjadi simpul-E yang baru. Lintasan baru dibangun kembali sampai lintasan tersebut membentuk solusi.

4. Pencarian dihentikan bila solusi telah ditemukan atau tidak ada simpul hidup untuk *backtracking* atau simpul yang dapat diperluas.

2.2 Visual Basic 6.0

Visual Basic 6.0 merupakan bahasa pemrograman yang cukup populer dan mudah untuk dipelajari [MAD01]. Bahasa pemrograman Visual Basic, yang dikembangkan oleh Microsoft sejak tahun 1991, merupakan pengembangan dari pendahulunya yaitu bahasa pemrograman BASIC (*Beginner's All-purpose Symbolic Instruction Code*) yang dikembangkan pada era 1950-an. Visual Basic merupakan salah satu *Development Tool* yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi Windows. Visual Basic merupakan salah satu bahasa pemrograman komputer yang mendukung object (*Object Oriented Programming* = OOP).

Visual Basic merupakan bahasa yang mendukung Pemrograman berorientasi objek, namun tidak sepenuhnya, Beberapa karakteristik obyek tidak dapat dilakukan pada Visual Basic, seperti *Inheritance* tidak dapat dilakukan pada

class module, *Polymorphism* secara terbatas bisa dilakukan dengan mendeklarasikan *class module* yang memiliki *Interface* tertentu. Visual Basic (VB) tidak bersifat case sensitif.

Visual Basic menjadi populer karena kemudahan desain form secara visual dan adanya kemampuan untuk menggunakan komponen-komponen *ActiveX* yang dibuat oleh pihak lain. Namun komponen *ActiveX* memiliki masalahnya tersendiri yang dikenal sebagai *DLL hell*, Pada Visual Basic .NET, Microsoft mencoba mengatasi masalah *DLL hell* dengan mengubah cara penggunaan komponen (menjadi independen terhadap registry).

2.3 Microsoft Access

Microsoft Access (atau Microsoft Office Access) merupakan program database yang populer dan banyak digunakan saat ini. Karena kemudahannya dalam pengolahan berbagai jenis database serta hasil akhir berupa laporan dengan tampilah desain yang menarik [MAD07]. Program aplikasi ini merupakan anggota dari beberapa aplikasi Microsoft Office, selain tentunya Microsoft Word, Microsoft Excel, dan Microsoft PowerPoint.

Microsoft Access dapat menggunakan data yang disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Para pengguna/*programmer* yang mahir dapat menggunakannya untuk mengembangkan perangkat lunak aplikasi yang kompleks, sementara para programmer yang kurang mahir dapat menggunakannya untuk mengembangkan

perangkat lunak aplikasi yang sederhana. Access juga mendukung teknik-teknik pemprograman berorientasi objek, tetapi tidak dapat digolongkan ke dalam perangkat bantu pemrograman berorientasi objek.

Salah satu keunggulan Microsoft Access dilihat dari perpektif *programmer* adalah kompatibilitasnya dengan bahasa pemrograman Structured Query Language (SQL); query dapat dilihat dan disunting sebagai statement-statement SQL, dan statemen SQL dapat digunakan secara langsung di dalam Macro dan VBA Module untuk secara langsung memanipulasi tabel data dalam Access. Para pengguna dapat mencampurkan dan menggunakan kedua jenis bahasa tersebut (VBA dan Macro) untuk memprogram *form* dan logika dan juga untuk mengaplikasikan konsep berorientasi objek.

2.4 Teori dalam Pembuatan Game

Urutan pembuatan game dan pengembangan program game adalah sebagai berikut :

1. Tentukan tipe game yang ingin dibuat

   Penentuan ini sebagai dasar mulai bekerja sampai mendapatkan ide yang bagus untuk dibuat game.

2. Definisikan model game dan tujuannya

   Pada tahap ini sebaiknya model game ditulis secara jelas sehingga bila terjadi perubahan maka gamenya tetap konsisten dan tidak membingungkan.

3. Definisikan secara jelas *Game Worlds*-nya

Game Worlds adalah elemen-elemen utama yang terdapat dalam suatu program game yang terdiri dari :

a) Game Board

Game Board merupakan bentuk tampilan, latar belakang, dan lainnya.

b) Instruksi untuk game

Instruksi untuk game harus jelas supaya tidak membingungkan dan pemain dapat menentukan langkah dalam menyelesaikan game tersebut.

c) Informasi untuk pemain

Informasi ini penting ditampilkan dalam program game ketika sedang berjalan.

d) Penghargaan

Penghargaan digunakan sebagai rangsangan untuk pemain ketika menyelesaikan level tertentu dan mendorong untuk melanjutkan ke level berikutnya.

e) Variasi

Variasi digunakan agar pemain tidak cepat merasa bosan, tetapi variasi tidak boleh berlebihan karena dapat membuat alur cerita menjadi tidak konsisten dan membingungkan.

f) Tingkat kesulitan

Tingkat kesulitan dari program game akan membuat pemain menjadi bergairah jika melewati tingkat kesulitan yang diberikan.

4. Pastikan game bisa dimainkan

   Game yang tidak mungkin diselesaikan akan membuat program game tidak dapat dimainkan.

5. Rancang program sebaik mungkin

   Gunakan teknik pemprograman yang sesuai dalam pembuatan dan buat program yang mudah dimodifikasi dan dikembangkan.

6. Pengujian program

   Program yang telah selesai harus diuji. Pertama, program diuji oleh perancangnya sendiri (*alpha test*) untuk menetukan kesalahan logika yang mungin terjadi. Kedua, program diuji oleh orang lain *(betaa test)* untuk mengetahui kesalahan logika yang tidak terlihat oleh perancangnya, sehingga dapat member masukan, saran, atau ide dalam pengembangan program game tersebut.

BAB III

METODOLOGI

3.1 Analisis Sistem

Tujuan dibuatnya sistem aplikasi ini adalah sebagai salah satu hiburan yang dapat dimainkan oleh semua orang melalui komputer. Dimana selain dapat menghibur aplikasi ini juga dapat membantu melatih kerja otak agar lebih baik, yaitu dengan merangsang sel otak agar selalu aktif, meningkatkan ketelitian dan logika berpikir, menambah kemampuan memecahkan masalah dengan cepat, dan mempertajam imajinasi, kreativitas, serta inovasi.

Sebelum dibuatnya sistem ini, penulis terlebih dahulu mencari tahu beberapa macam *game* sudoku yang pernah dibuat sebagai bahan perbandingan dalam pembuatan sistem aplikasi ini. *Game-game* sudoku tersebut yaitu Hodoku yang merupakan *game* sudoku berbasis *open source* dengan bahasa pemrograman java, Sudoku Pagoda yang merupakan *game* komersil buatan Revlexive.com, dan *game* sudoku buatan Rahman Adiarto [Adiarto, 2007] yang merupakan tugas akhir yang dibuatnya yang berjudul ‘Aplikasi Game Sudoku dengan menggunakan pemrograman J2ME pada Perangkat Mobile’.

Pada Hodoku hanya terdapat papan permainan 9x9, namun memiliki 5 tingkat kesulitan. Hodoku memiliki banyak proses yang dapat dilakukan, diantaranya *random* soal sudoku, *save* data permainan dan *open* data yang pernah disimpan, *restart* dan *reset game*, *hint*, dan langkah-langkah penyelesaian untuk soal sudoku yang dimainkan. Selain itu juga terdapat 3 mode permainan, yaitu *playing*, *learning*, dan *training*. Juga terdapat pilihan untuk menampilkan opsi jawaban pada setiap sel sudoku yang ada.

Pada Sudoku Pagoda, tampilan *game* sudah sangat bagus dimana terdapat *background* dan *backsound*, juga animasi tampilan. Pada tampilan awal terdapat beberapa menu pilihan, yaitu pembuatan profil nama pemain, 3 tipe permainan, yaitu *Puzzle*, *Puzzle Creator*, dan *Arcade*, juga terdapat menu untuk melanjutkan permainan terakhir yang dilakukan, *Option*, *How to Play*, *Top Score*, dan *Quit*. Selain itu terdapat 3 pilihan bentuk angka isian, angka dan huruf, kanji, dan bentuk geometri, dan juga ada 3 pilihan bentuk pemasukan angka isian, pop-up, scroll-over, dan panel. Pada Sudoku Pagoda ini terdapat 6 tipe papan permainan, 3 tingkat kesulitan dan ribuan bentuk soal sudoku yang dapat dipilih. Dan pada proses permainan terdapat menu bantuan yaitu waktu permainan, *undo*, *hint*, pengecekan, dan *solve*.

Sedangkan g*ame* sudoku buatan Rahman Adiarto merupakan game untuk perangkat mobile dimana tampilannya masih sangat sederhana dengan tidak memiliki *background* dan juga *backsound*. Sedangkan untuk proses yang dapat dilakukan oleh game ini yaitu menampilkan papan permainan 9x9, menampilkan angka pada papan permainan 9x9, menampilkan jawaban pemecahan, dan proses

pengecekan. Selain ini terdapat proses *hint* dan buka soal yang disediakan. Pada *game* ini pemain dapat memilih soal permainan yang telah disediakan atau dapat memainkan sudoku tanpa menggunakan soal.

Setelah membandingkan *game-game* sudoku tersebut, maka untuk tampilan sistem yang akan dibuat lebih banyak meniru dari Sudoku Pagoda, dimana pada tampilan awal aplikasi akan terdapat beberapa menu, diantaranya *Play, Top Score, How to play, Credit,* dan *Exit*. Pada aplikasi yang dibuat ini hanya akan memiliki 3 jenis pilihan papan permainan yaitu 4x4, 6x6, dan 9x9, dan juga 3 tingkat kesulitan, yaitu *Begginer*, *Intermediate*, dan *Advance*. Dan pada proses permainan pada sistem ini akan memiliki beberapa menu bantuan, yaitu *Menu, Solve, Hint, Check, Undo,* dan pemberitahuan sisa waktu permainan serta jumlah sel yang telah terisi. Pada menu *Menu* sendiri akan terdapat beberapa pilihan yaitu *Resume, Restart, New Game,* dan kembali ke halaman awal aplikasi. Pada aplikasi ini terdapat sistem skoring, sehingga pada halaman Top Score nantinya yang ditampilkan adalah nama serta skor yang didapat.

3.2 Hasil Analisis

Hasil analisis dari kebutuhan-kebutuhan di dalam pembuatan perangkat lunak meliputi input, proses, output, serta fungsi-fungsi yang dibutuhkan dan antarmuka yang diinginkan.

a. Analisis kebutuhan *Input*/masukan

Input dari sistem ini adalah pengaturan atau *setting* yang merupakan pengaturan terhadap *game* yang akan atau sedang dimainkan. Pengaturan tersebut antara lain :

1. Menentukan jumlah *grid* yang akan dimainkan serta tingkat kesulitannya
2. Buka soal lainnya / *New Game*
3. Mengulang persoalan yang sama / *Restart Game*
4. Melanjutkan permainan yang sedang berlangsung / *Resume*
5. Bantuan jawaban / *Hint*

   Penggunaan fungsi penampilan bantuan jawaban pemecahan untuk membantu pemain dalam pemecahan puzzle sudoku yang sedang dimainkan. Dengan menggunakan fungsi ini, sistem akan menampilkan satu bantuan jawaban dari soal sudoku yang sedang dimainkan.

b. Analisis kebutuhan Proses

Beberapa proses yang dapat dilakukan oleh sistem adalah sebagai berikut :

1. Proses penampilan *grid* papan permainan *(gameboard).* Pada proses ini dilakukan pengambilan gambar papan permainan sesuai dengan pilihan *user* sebelum memasuki permainan, yaitu 4x4, 6x6, atau 9x9.
2. Proses penampilan angka pada *grid* papan permainan *(gameboard).* Pada proses ini game akan menampilkan angka yang menjadi soal dari game sudoku ini.

3. Proses penampilan jawaban pemecahan. Proses ini dilakukan untuk menampilkan jawaban dari soal sudoku yang sedang dimainkan dengan metode *backtracking*.
4. Proses pengecekan apakah angka yang dimasukkan benar atau salah. Pada proses ini jika ternyata angka yang dimasukkan salah dan tidak sesuai dengan ketentuan maka angka tersebut akan berwarna merah, dimana menandakan bahwa terjadi kesalahan dalam peletakan angka tersebut pada kolom tersebut.
5. Proses penghapusan angka. Proses ini dilakukan untuk menghapus angka yang paling terakhir dimasukkan ke dalam papan permainan.
6. Proses game berjalan. Proses ini dilakukan untuk menampilkan angka yang dimasukkan.
7. Proses perhitungan *score*. Proses ini dilakukan untuk menghitung jumlah *score* yang dicapai oleh *user*.

c) Analisis kebutuhan *Output*/keluaran

Output pada sistem ini adalah status game pada suatu kondisi tertentu ketika game tersebut sedang berlangsung.

1. Angka yang sedang dimainkan baik pada proses bermain maupun pada penampilan jawaban soal sudoku
2. Huruf yang ditampilkan pada mode 'Tentang Sudoku' dan 'Cara Bermain'
3. Angka dan Huruf yang ditampilkan pada mode ‘Top Score’.

d) Analisis kebutuhan antarmuka yang diinginkan

Suatu sistem yang memiliki antarmuka yang menarik, informatif dan praktis dengan penampilan yang inovatif dan komunikatif.

Kebutuhan antarmuka yang akan dibuat adalah sebagai berikut:

1. Main Menu
2. Pemilihan *Type Grid*
3. Papan permainan
4. Menu
5. *Top Score*
6. *Register*
7. *How to Play*
8. *Credit*

3.3 Perancangan Perangkat Lunak

3.3.1 Diagram Alir (*Flowchart*)

1. *Flowchart* Sistem

   Alur proses sistem dari memulai permainan sampai dengan proses inti dan hasilnya digambarkan sebagai sebuah diagram alir secara keseluruhan. Diagram alir sistem ini dapat dilihat pada Gambar 3.1.

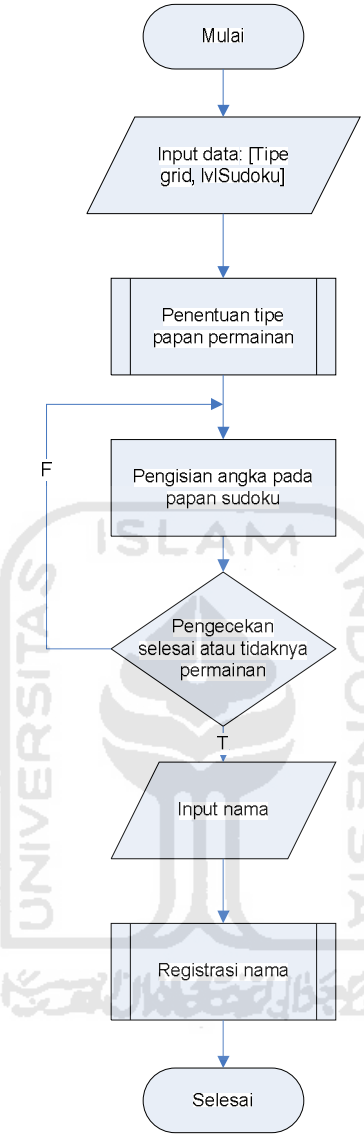
2. *Flowchart* pemilihan papan permainan

Flowchart untuk pemilihan tipe papan permainan dapat dilihat pada Gambar 3.2. Inisialisasi dari *flowchart* ini adalah tipe *grid* dan tingkat kesulitan. Sistem akan melihat apakah tipe *grid* yang dipilih adalah 4x4, jika benar maka akan memeriksa tingkat kesulitan yang dipilih, jika salah maka sistem akan memeriksa apakah tipe *grid* yang dipilih 6x6, jika benar maka selanjutnya sistem akan memeriksa tingkat kesulitan yang dipilih, jika salah maka sistem akan memeriksa apakah tipe *grid* yang dipilih 9x9, jika benar maka selanjutnya sistem akan memeriksa tingkat kesulitan yang dipilih. Selanjutnya jika tingkat kesulitan adalah *begginer*, maka soal yang dipakai adalah tipe soal *begginer*, jika salah sistem akan memeriksa apakah tingkat kesulitan yang dipilih adalah *intermediate*, jika benar maka soal yang dipakai adalah tipe soal *intermediate*, jika salah sistem akan memeriksa apakah tingkat kesulitan yang dipilih adalah *advance*, jika benar maka soal yang dipakai adalah tipe soal *advance*. Setelah itu sistem akan membuka papan permainan sesuai dengan tipe *grid* dan tingkat kesulitan yang telah diinisialisasi.

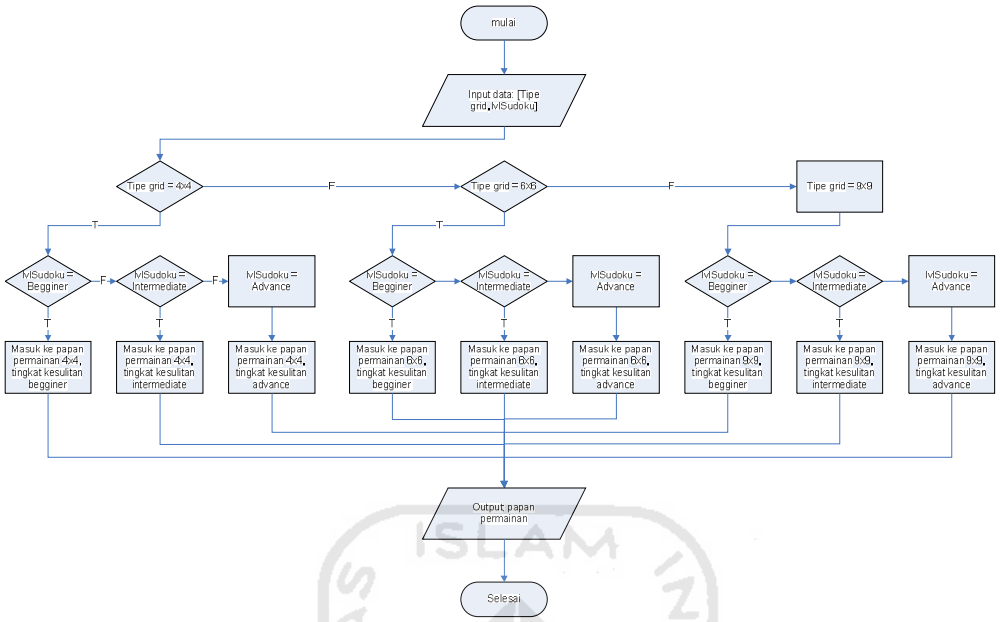
3. *Flowchart* proses *undo*

Proses ini dimulai dengan pengecekan apakah terdapat sejarah pengisian atau penghapusan pada papan permainan yang terdapat pada UndoStack. Jika UndoStack = 0 berarti tidak terdapat sejarah pengisian maupun penghapusan pada papan permainan sehingga tidak akan ada proses yang terjadi. Namun bila UndoStack > 0 yang berarti terdapat sejarah

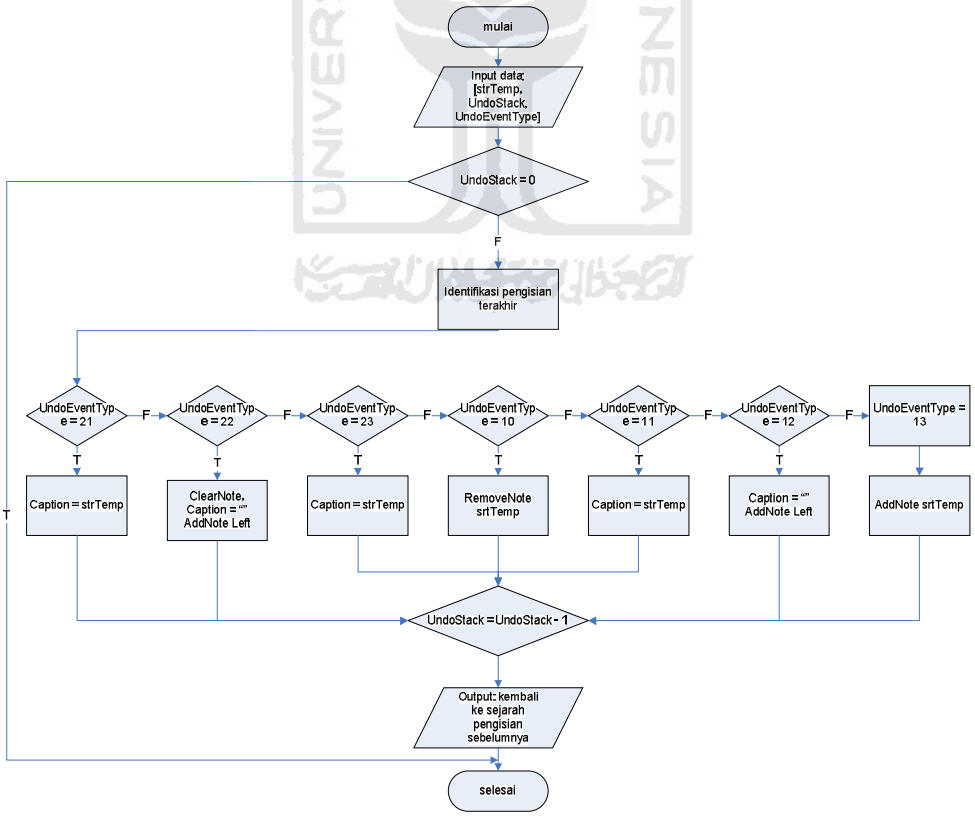
pengisian maupun penghapusan maka proses akan dilanjutkan dengan pengidentifikasian sejarah aktifitas yang terakhir dilakukan. Selanjutnya akan dilakukan pengecekan aktifitas yang terakhir dilakukan tersebut apa dan kemudian dilakukan proses *undo* sesuai dengan aktifitas terakhir tersebut, seperti yang terlihat pada Gambar 3.3. Setelah itu *UndoStack* yang ada akan dikurangi dengan 1.



Gambar 3.1 *Flowchart* Sistem



Gambar 3.2 *Flowchart* Pemilihan Papan Permainan



Gambar 3.3 *Flowchart* Proses *Undo*

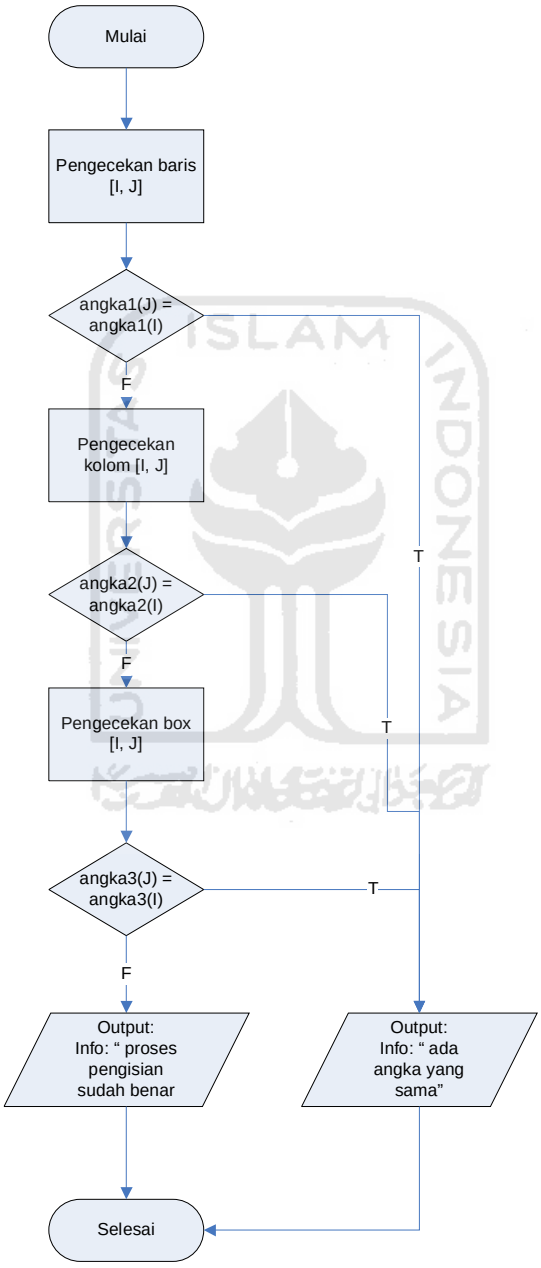
4. *Flowchart* proses *check*

Proses ini dilakukan untuk pengecekan apakah terdapat angka yang sama pada baris, kolom, ataupun kotak sudoku. Dapat dilihat pada Gambar 3.4. Dimulai dengan pengecekan pada tiap baris papan sudoku, apabila Angka1(J) = Angka1(I) maka akan keluar output info yang menyatakan bahwa terdapat angka yang sama pada papan sudoku. Jika salah maka kemudian dilakukan pengecekan untuk setiap kolom sudoku, yang apabila Angka2(J) = Angka2(I) maka akan keluar output info yang menyatakan bahwa terdapat angka yang sama pada papan sudoku. Jika salah maka kemudian dilakukan pengecekan untuk setiap kotak sudoku yang apabila Angka3(J) = Angka3(I) maka akan keluar output info yang menyatakan bahwa terdapat angka yang sama pada papan sudoku. Jika salah maka akan keluar output yang menyatakan bahwa pengisian angka sudah benar.

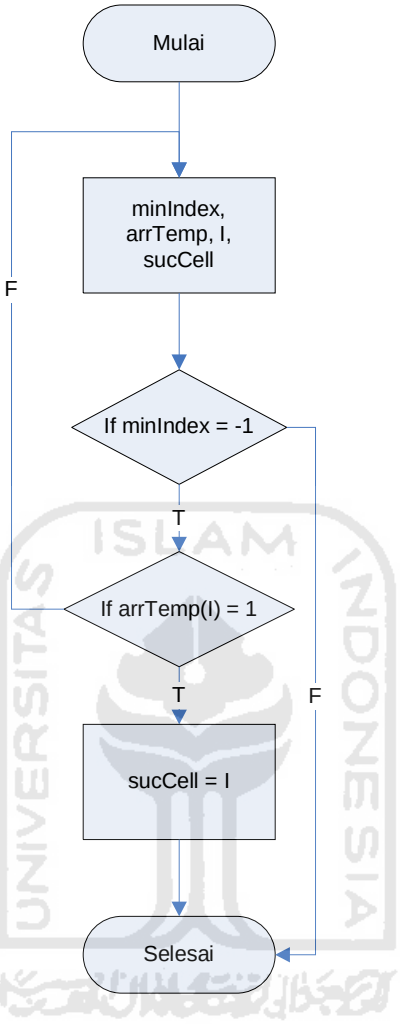
5. *Flowchart* proses *hint*

Proses ini dilakukan untuk mencari angka isian yang dapat diisi selanjutnya. Dapat dilihat pada Gambar 3.5. Dimulai dengan pendeklarasian fungsi minIndex, arrTemp, I, dan sucCell. Kemudian akan dilakukan pengecekan pada sel sudoku apakah minIndex = -1 yang berarti bahwa ada sel kosong, jika salah maka semua sel telah terisi. Setelah didapat sel yang kosong maka akan dicek berapakah opsi yang tersisa untuk sel tersebut, dan

jika arrTemp(I) = 1 maka sel tersebut akan diisi dengan nilai I (sucCell = I), jika arrTemp(I) masih lebih dari 1 maka akan dicari sel yang lain.



Gambar 3.4 *Flowchart* Proses *Check*



Gambar 3.5 *Flowchart* Proses *Hint*

6. *Flowchart* proses *solve*

Proses ini dilakukan untuk menyelesaikan permainan puzzle sudoku yang berlangsung dengan mengunakan algoritma *backtracking*. Dapat dilihat pada Gambar 3.6. Dimulai dengan pendeklarasian fungsi minIndex, arrTemp, I, dan sucCell. Kemudian akan dilakukan pengecekan pada sel sudoku apakah minIndex = -1 yang berarti bahwa ada sel kosong, jika salah

yang berarti minIndex = -2 maka semua sel telah terisi dan puzzle sudoku selesai. Setelah didapat sel yang kosong maka akan dicek berapakah opsi yang tersisa untuk sel tersebut, dan jika arrTemp(I) = 0 maka sel tersebut tidak memiliki opsi pilihan yang berarti minIndex tetap -1 sehingga sudoku tidak dapat diselesaikan, jika arrTemp(I) = 1 sel tersebut akan diisi dengan nilai I (sucCell = I) dan proses akan diulang dari awal kembali, jika arrTemp(I) masih lebih dari 1 maka akan dicari sel yang lain.

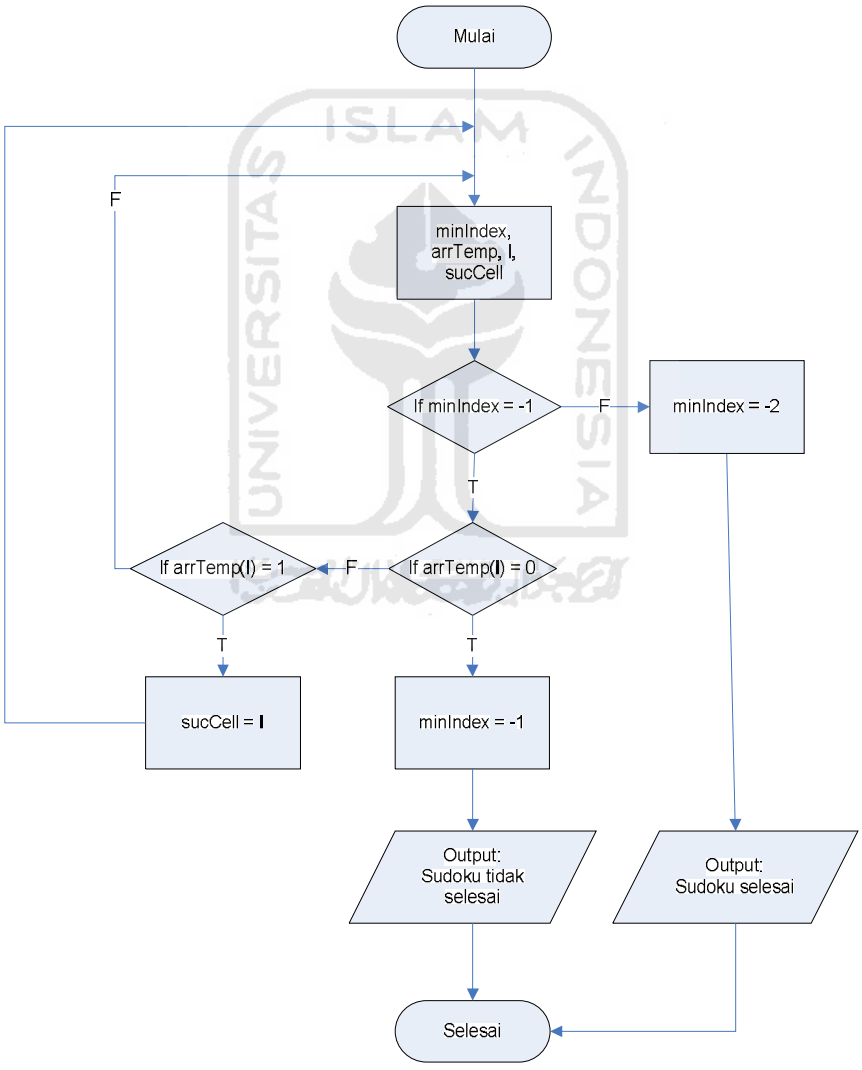
7. *Flowchart* proses register

Proses ini dilakukan untuk meyimpan data untuk skor sudoku. Dapat dilihat pada Gambar 3.7. Dimulai dengan pengambilan id nama yang telah dimasukkan *user* pada akhir permainan, kemudian melakukan koneksi ke database. Setelah itu akan dicari table database untuk data tersebut, jika tipe = 1 maka selanjutnya akan di cek apakah lvlSudoku = B, jika benar maka tabel databasenya adalah TopScore 9x9B, jika lvlSudoku = I maka tabel databasenya adalah TopScore 9x9I, jika lvlSudoku = A maka tabel databasenya adalah TopScore 9x9A. Namun jika tipe tidak sama dengan 1 maka akan dicek apakah tipe = 2, jika benar maka selanjutnya akan di cek apakah lvlSudoku = B, jika benar maka tabel databasenya adalah TopScore 4x4B, jika lvlSudoku = I maka tabel databasenya adalah TopScore 4x4I, jika lvlSudoku = A maka tabel databasenya adalah TopScore 4x4A. Namun jika tipe tidak sama dengan 2 maka akan akan dicek apakah tipe = 3 jika benar maka selanjutnya akan di cek apakah lvlSudoku = B, jika benar maka

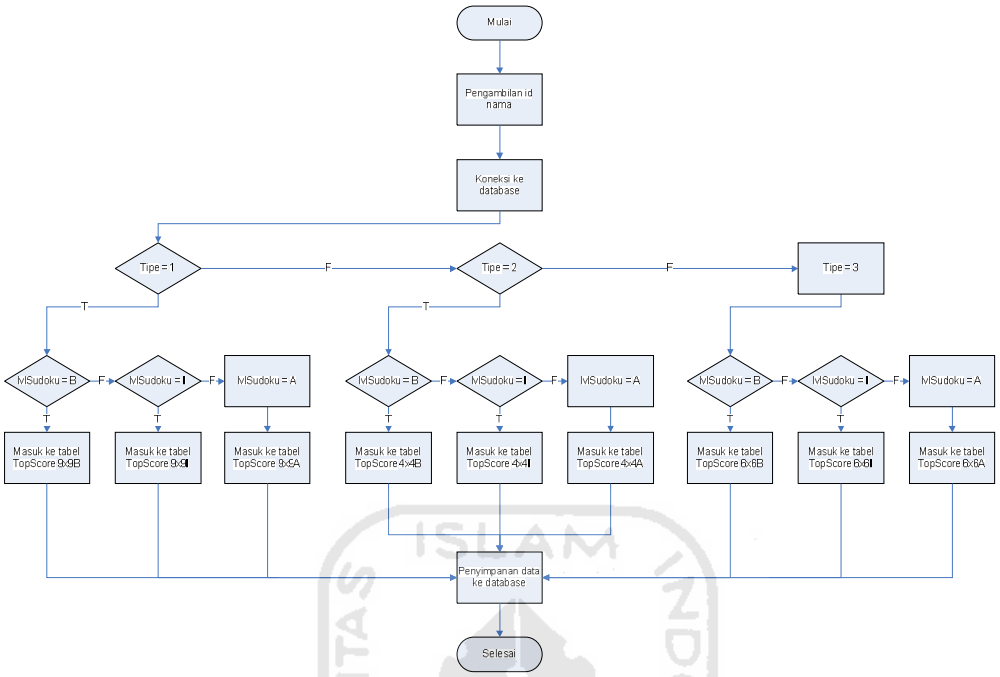
tabel databasenya adalah TopScore 6x6B, jika lvlSudoku = I maka tabel databasenya adalah TopScore 6x6I, jika lvlSudoku = A maka tabel databasenya adalah TopScore 6x6A.

Rumus perhitungan skor sudoku pada aplikasi ini yaitu :

((waktu * 10) / 4 – (hintcounter * 10))



Gambar 3.6 *Flowchart* Proses *Solve*



Gambar 3.7 *Flowchart* Proses Registrasi

3.3.2 Perancangan Antarmuka

a. Rancangan Antarmuka Main Menu

Rancangan antarmuka *main menu* merupakan halaman yang pertama kali muncul ketika *user* membuka sistem ini. Perancangan antarmuka main menu dapat dilihat pada Gambar 3.8.

Beberapa menu yang terdapat pada halaman main menu yaitu :

1. *Play*, merupakan menu dimana *user* dapat memulai permainan dengan memilih tipe grid dan tingkat kesulitannya terlebih dahulu.
2. *Top Score*, merupakan menu dimana *user* dapat melihat data *score* 3 tertinggi setiap tipe grid dan tingkat kesulitan yang ada.

3. *How To Play*, merupakan menu yang di dalamnya berisi penjelasan mengenai cara penggunaan sistem.
4. *Credit*, merupakan menu yang di dalamnya berisi penjelasan mengenai game ini dibuat.
5. *Exit*, digunakan untuk keluar dari sistem.

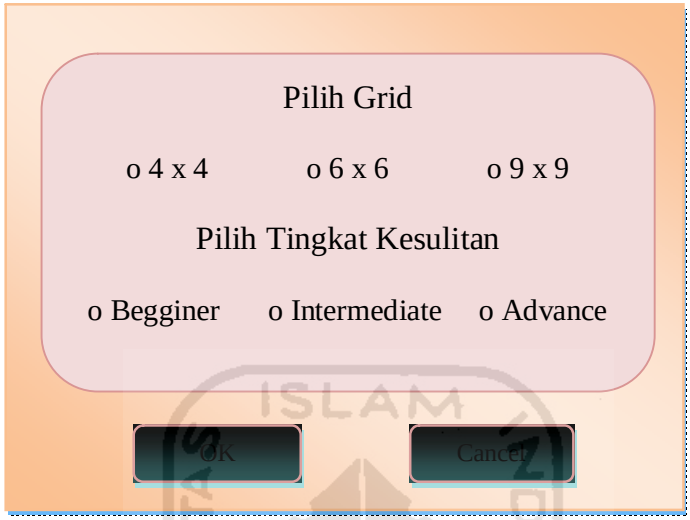


Gambar 3.8 Rancangan Antarmuka Main Menu

b. Rancangan Antarmuka Pemilihan Type Grid

Rancangan antarmuka pemilihan tipe grid merupakan halaman yang digunakan untuk menentukan tipe grid dari papan permaian sudoku dan juga tingkat kesulitan permainan yang akan dimainkan. Perancangan antarmuka pemilihan *type grid* dapat dilihat pada Gambar 3.9. Pada halaman ini, *user* dapat

menentukan jumlah grid papan permainan yang tersedia, yaitu 4x4, 6x6, dan 9x9, dan juga tingkat kesulitannya *begginer, intermediate,* atau *advance*.



Gambar 3.9 Rancangan Antarmuka Pemilihan Type Grid

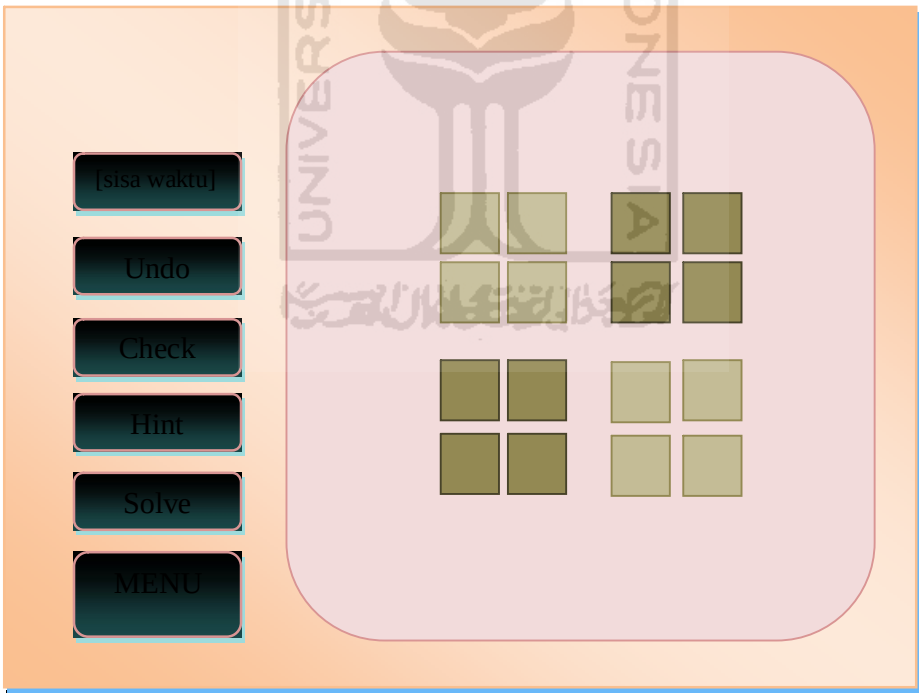
c. Rancangan Antarmuka Papan Permainan 4 x 4

Rancangan antarmuka papan permainan 4 x 4 merupakan halaman yang dimana *user* akan memainkan permainan sudoku ini, dimana pada halaman ini papan permainannya terdiri dari kotak 4x4. Perancangan antarmuka papan permainan 4 x 4 dapat dilihat pada Gambar 3.10.

Pada halaman ini terdapat beberapa menu yang dapat membantu jalannya permainan, yaitu:

1. *Menu*, merupakan menu dimana akan ditampilkan menu-menu lain yang berhubungan dengan permainan yang ada.

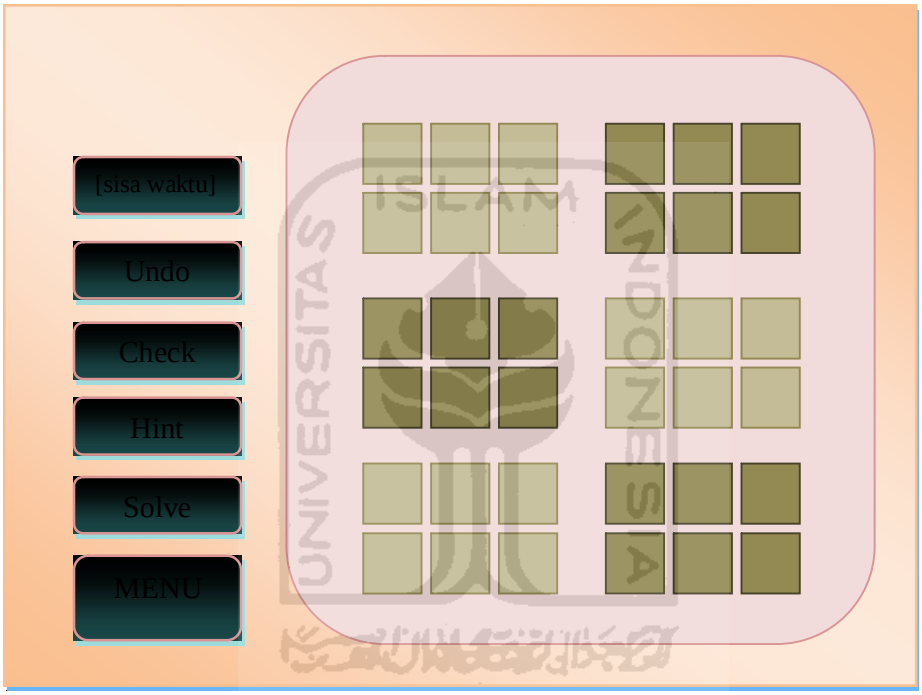
2. *Solve*, merupakan menu dimana sistem akan menyelesaikan permainan secara otomatis dengan menampilkan langkah-langkah penyelesaiannya sesuai dengan metode *backtracking*.
3. *Hint*, merupakan menu dimana sistem akan memberikan bantuan jawaban berupa dimana angka selanjutnya harus diisikan.
4. *Check*, merupakan menu dimana sistem akan mengecek apakah terdapat angka yang sama dalam satu baris, kolom, atau kotaknya.
5. *Undo*, merupakan menu dimana sistem akan menghapus angka paling terakhir yang diisikan oleh *user*.



Gambar 3.10 Rancangan Antarmuka Papan Permainan 4 x 4

d. Rancangan Antarmuka Papan Permainan 6 x 6

Rancangan antarmuka papan permainan 6 x 6 merupakan halaman yang dimana *user* akan memainkan permainan sudoku ini, dimana pada halaman ini papan permainannya terdiri dari kotak 6x6. Perancangan antarmuka papan permainan 6 x 6 dapat dilihat pada Gambar 3.11.



Gambar 3.11 Rancangan Antarmuka Papan Permainan 6 x 6

Pada halaman ini terdapat beberapa menu yang dapat membantu jalannya permainan, yaitu:

1. *Menu*, merupakan menu dimana akan ditampilkan menu-menu lain yang berhubungan dengan permainan yang ada.

2. *Solve*, merupakan menu dimana sistem akan menyelesaikan permainan secara otomatis dengan menampilkan langkah-langkah penyelesaiannya sesuai dengan metode *backtracking*.
3. *Hint*, merupakan menu dimana sistem akan memberikan bantuan jawaban berupa dimana angka selanjutnya harus diisikan.
4. *Check*, merupakan menu dimana sistem akan mengecek apakah terdapat angka yang sama dalam satu baris, kolom, atau kotaknya.
5. *Undo*, merupakan menu dimana sistem akan menghapus angka paling terakhir yang diisikan oleh *user*.

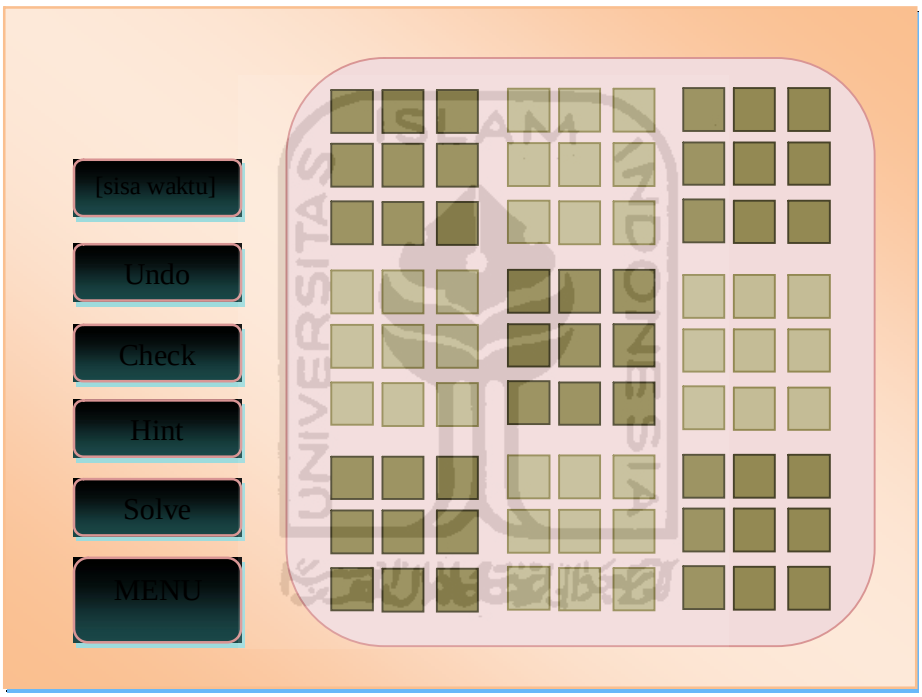
e. Rancangan Antarmuka Papan Permainan 9 x 9

Rancangan antarmuka papan permainan 9 x 9 merupakan halaman yang dimana *user* akan memainkan permainan sudoku ini, dimana pada halaman ini papan permainannya terdiri dari kotak 9x9. Perancangan antarmuka papan permainan 9 x 9 dapat dilihat pada Gambar 3.12.

Pada halaman ini terdapat beberapa menu yang dapat membantu jalannya permainan, yaitu:

1. *Menu*, merupakan menu dimana akan ditampilkan menu-menu lain yang berhubungan dengan permainan yang ada.
2. *Solve*, merupakan menu dimana sistem akan menyelesaikan permainan secara otomatis dengan menampilkan langkah-langkah penyelesaiannya sesuai dengan metode *backtracking*.

3. *Hint*, merupakan menu dimana sistem akan memberikan bantuan jawaban berupa dimana angka selanjutnya harus diisikan.
4. *Check*, merupakan menu dimana sistem akan mengecek apakah terdapat angka yang sama dalam satu baris, kolom, atau kotaknya.
5. *Undo*, merupakan menu dimana sistem akan menghapus angka paling terakhir yang diisikan oleh *user*.



Gambar 3.12 Rancangan Antarmuka Papan Permainan 9 x 9

f. Rancangan Antarmuka Menu

Rancangan antarmuka menu merupakan halaman yang berisi menu-menu yang dapat dipilih setelah pemain memulai permainan dengan cara menklik tombol '*Menu*' yang berada di sebelah kiri bawah papan permainan. Perancangan antarmuka menu dapat dilihat pada Gambar 3.13.



Gambar 3.13 Rancangan Antarmuka Menu

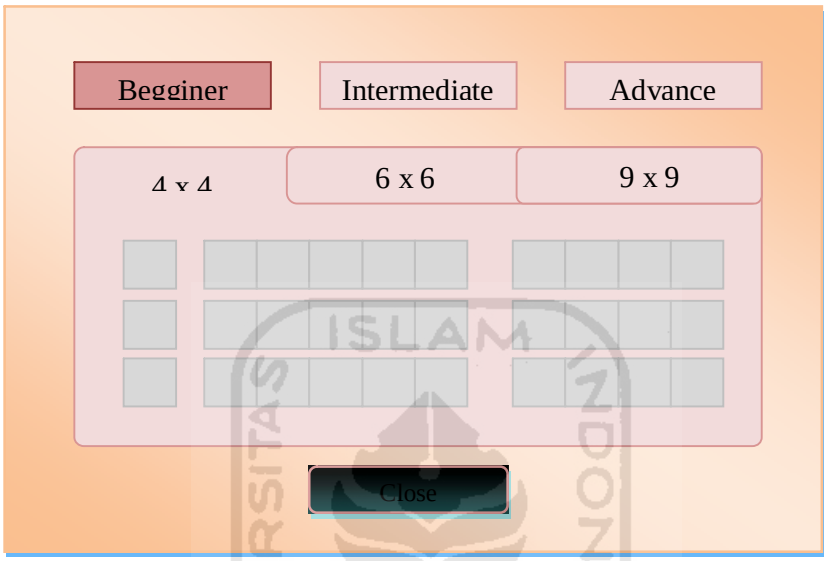
Menu-menu yang terdapat pada halaman ini yaitu:

1. *Resume*, merupakan menu dimana pemain akan kembali ke halaman permainan tanpa ada perubahan dari sebelum masuk ke halaman menu.
2. *Restart*, merupakan menu dimana sistem akan mengembalikan papan permainan seperti awal permainan dan batas waktu akan di*reset* kembali ke semula.
3. *New Game*, merupakan menu dimana pemain akan memainkan soal permainan yang berbeda dengan sebelumnya.
4. *Main Menu*, merupakan menu untuk kembali ke halaman '*Main Menu*'.

g. Rancangan Antarmuka Top Score

Rancangan antarmuka *top score* merupakan halaman yang berisi *record* skor yang diperoleh pemain setiap menyelesaikan permainan. *Record* skor yang tercatat merupakan 3 skor tertinggi untuk setiap tingkat kesulitan dan tipe grid

permainan, jadi skor yang lebih kecil atau rendah dari pada 3 skor yang telah tercatat tidak akan masuk dalam *record* skor yang ada. Perancangan antarmuka *top score* dapat dilihat pada Gambar 3.14.



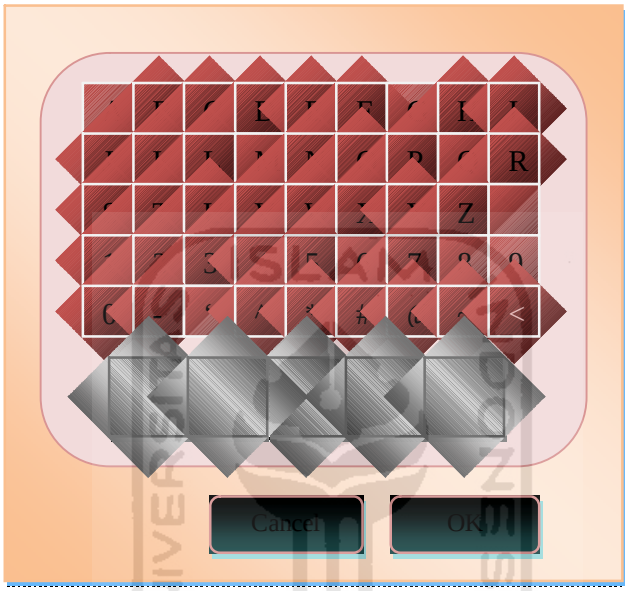
Gambar 3.14 Rancangan Antarmuka Top Score

Saat pertama membuka halaman antarmuka *top score* ini, pemain akan langsung dihadapkan pada 3 skor tertinggi pada tingkat kesulitan *begginer* grid 4x4. Untuk melihat skor pada grid 6x6 dan 9x9 pemain dapat langsung memilih pada *tab* selanjutnya. Dan untuk melihat skor pada tingkat kesulitan *intermediate* dan *advance*, pemain dapat mengklik tombol yang ada untuk berpindah kehalaman *top score* yang dimaksud.

h. Rancangan Antarmuka Register

Rancangan antarmuka *register* merupakan halaman dimana muncul ketika pemain telah berhasil menyelesaikan permainan. Di halaman ini pemain diminta

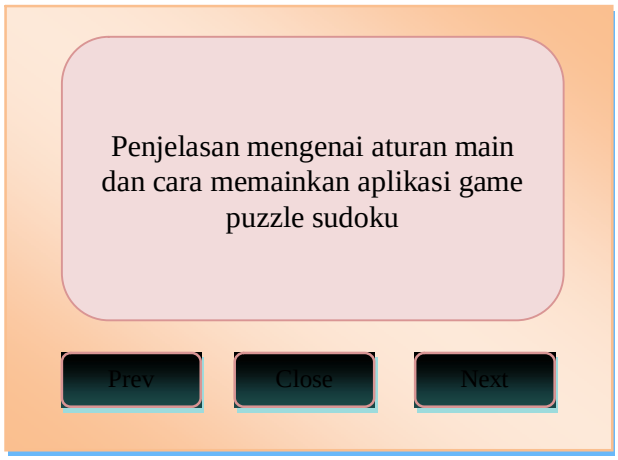
untuk mengisikan namanya untuk dimasukkan kedalam *record* skor yang jika termasuk ke dalam 3 skor tertinggi maka nama pemain tersebut akan tampil di halaman antarmuka *top score*. Perancangan antarmuka *register* dapat dilihat pada Gambar 3.15.



Gambar 3.15 Rancangan Antarmuka Register

i. Rancangan Antarmuka How To Play

Rancangan antarmuka *How To Play* merupakan halaman yang berisi tentang penjelasan singkat bagi *user* mengenai cara pengggunaan sistem aplikasi *game puzzle* sudoku ini. Perancangan antarmuka *How To Play* dapat dilihat pada Gambar 3.16.



Gambar 3.16 Rancangan Antarmuka How To Play

j. Rancangan Antarmuka Credit

Perancangan antarmuka *credit* digunakan untuk melihat alasan dibuatnya aplikasi ini dan juga nama pembuat aplikasi ini. Perancangan antarmuka *credit* dapat dilihat pada Gambar 3.17.



Gambar 3.17 Rancangan Antarmuka Credit

3.4 Implementasi Perangkat Lunak

3.4.1 Implementasi

Implementasi aplikasi *game* Sudoku ini dibuat dengan menggunakan bahasa pemprograman Visual Basic 6.0 dan database yang digunakan adalah Microsoft Access 2003. Implementasi aplikasi *game* ini dibagi menjadi tiga bagian, yaitu batasan implementasi, implementasi sistem dan implementasi antarmuka.

a. Batasan Implementasi

Dalam proses pembuatan aplikasi ini terdapat beberapa keterbatasan yaitu:

1. Pada tampilan antarmuka hanya menggunakan sedikit animasi saja dan hanya memiliki satu tingkat resolusi saja yaitu 1024 x 768.
2. Belum terdapat proses *save* permainan yang dapat membantu *user* menyelesaikan permainannya di lain waktu.

b. Implementasi Sistem

Implementasi merupakan tahap dimana sistem siap dioperasikan pada tahap yang sebenarnya, sehingga akan diketahui apakah sistem yang telah dibuat benar-benar sesuai dengan yang direncanakan. Pada implementasi perangkat lunak ini akan dijelaskan bagaimana sistem ini bekerja dengan tampilan yang telah dibuat.

c. Implementasi Antarmuka

Pada bagian ini memuat gambaran penjelasan antarmuka yang terdapat pada sistem perangkat lunak *game* Sudoku.

a) Antarmuka Main Menu

Halaman main menu akan muncul pertama kali ketika *user* membuka sistem. Dimana pada halaman ini terdapat 5 pilihan menu yang dapat di pilih oleh *user*. Pilihan-pilihan yang ada yaitu *Play* untuk memulai permainan yang dimulai dengan pemilihan tipe *grid* dan tingkat kesulitan, *Top Score* untuk melihat 3 skor tertinggi untuk setiap tingkat kesulitan dan tipe grid permainan, *How to Play* untuk melihat aturan dasar dari permainan sudoku dan bagaimana cara memainkan permainan sudoku pada sistem yang dibuat, *Credit* untuk melihat pembuat serta alasan dibuatnya sistem ini, dan *Exit* untuk keluar dari permainan. Implemantasi halaman antarmuka main menu dapat dilihat pada Gambar 3.18.

b) Antarmuka Pilih Grid

Halaman antarmuka pilih grid akan muncul saat *user* memilih menu 'Play' pada halaman antarmuka 'Main Menu'. Pada *form* antarmuka pilih grid ini terdapat 2 buah parameter yang berguna sebagai penentu tipe papan permainan yang akan dimainkan. Parameter yang pertama adalah pilih *grid* yang berfungsi sebagai penentu tipe grid papan permainan, diantaranya *grid* 4x4, 6x6, dan 9x9. Parameter yang kedua yaitu pilih tingkat kesulitan, dimana semakin sulit tingkat kesulitan yang dipilih maka semakin sedikit angka bantuan yg muncul pada awal permainan. Implemantasi halaman antarmuka pilih grid dapat dilihat pada Gambar 3.19.



Gambar 3.18 Halaman Main Menu



Gambar 3.19 Halaman Pilih Grid

c) Antarmuka Papan Pemainan

Halaman antarmuka papan permainan terdiri dari papan permainan dan menu bantuan yang berada di sebelah kiri halaman halaman. Untuk papan permainannya sendiri akan mengikuti dari pilihan yang diminta *user* pada halaman antarmuka pilih grid, yaitu 4x4, 6x6, atau 9x9. Sedangkan untuk menu bantuan yang berada di kiri halaman tetap sama untuk setiap pilihan *grid*nya. Di halaman ini juga terdapat petunjuk batas waktu yang diberikan dan juga petunjuk jumlah *cell* yang telah terisi. Implementasi antarmuka papan permainan 4 x 4 dapat dilihat pada Gambar 3.20. Untuk implementasi antarmuka papan permainan 6 x 6 dapat dilihat pada Gambar 3.21. Sedangkan implementasi antarmuka papan permainan 9 x 9 dapat dilihat pada Gambar 3.22.



Gambar 3.20 Halaman papan permainan 4 x 4



Gambar 3.21 Halaman papan permainan 6 x 6



Gambar 3.22 Halaman papan permainan 9 x 9

1. *Pseudocode* proses *Solve* dengan algoritma *backtracking*

```
Input :  MaxSudokuOption <- jumlah angka jawaban;
         arrTemp
         I <- opsi jawaban yang tersedia
         sucCell <- sel sudoku
Inisialisasi : MinIndex = -1

Cek sucCell
   Jika minIndex = -1 maka
      For I = 1 to MaxSudokuOption
      a. Input ArrTemp(I) <- jumlah opsi jawaban
      b. Case ArrTemp(I) :
          • ArrTemp(I) = 0 maka minIndex = -1
            Output : "Sudoku tidak selesai";
          • ArrTemp(I) = 1 maka sucCell = I
            Cek sucCell berikutnya;
      else
         Cek sucCell berikutnya
   else
   Jika MinIndex = -2 maka
      Output : "Sudoku selesai"
```

2. *Pseudocode* proses *Check*

```
Input: I;
       J;
       MaxSudokuOption;
       Angka1 <- angka jawaban pada tiap baris;
       Angka2 <- angka jawaban pada tiap kolom;
       Angka3 <- angka jawaban pada tiap box

For I = 1 to MaxSudokuOption
For J = 1 to MaxSudokuOption
a. Input : Angka1(I); Angka1(J)
b. Jika Angka1(I) = Angka1(J) maka
      Output : "Ada angka yang sama"
      else
      For I = 1 to MaxSudokuOption
      For J = 1 to MaxSudokuOption
      a. Input : Angka2(I); Angka2(J)
      b. Jika Angka2(I) = Angka2(J) maka
            Output : "Ada angka yang sama"
            else
            For I = 1 to MaxSudokuOption
            For J = 1 to MaxSudokuOption
            a. Input : Angka3(I); Angka3(J)
            b. Jika Angka3(I) = Angka3(J) maka
                  Output : "Ada angka yang sama"
                  else
            Output "Proses pengisian sudah benar"
```

d) Antarmuka Menu

Pada tampilan halaman antarmuka menu terdapar 4 menu yang dapat dipilih. Menu-menu tersebut yaitu *Resume, Restart, New Game,* dan *Main Menu.* Implemantasi halaman antarmuka menu dapat dilihat pada Gambar 3.23.



Gambar 3.23 Halaman Menu

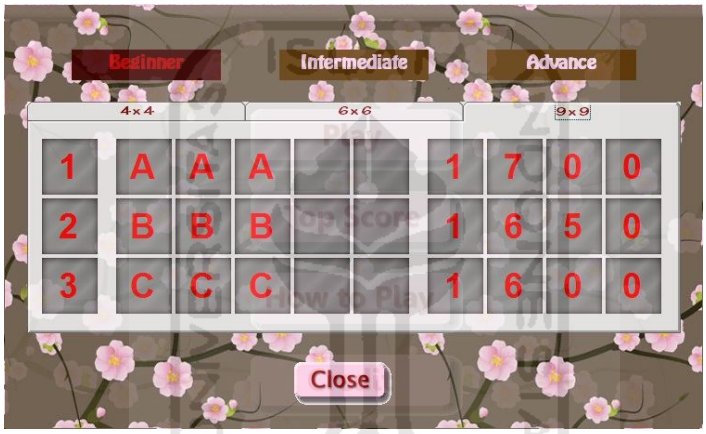
e) Antarmuka Top Score

Pada halaman antarmuka top score ini berisikan 3 skor tertinggi dari tiap tipe grid dan tingkat kesulitan permainan yang ada. Pada halaman ini pemain hanya dapat melihat 3 nama serta skor tertinggi saja. Implementasi halaman antarmuka top score dapat dilihat pada Gambar 3.24.

f) Antarmuka Register

Pada halaman antarmuka register ini pemain diminta untuk mengisikan namanya untuk dimasukkan kedalam *record* skor, yang jika termasuk

kedalam 3 skor tertinggi maka nama pemain tersebut akan tampil di halaman halaman top score. Setelah mengisikan namanya, *user* harus mengklik tombol OK untuk mengeksekusi perintah. Sedangkan tombol CANCEL digunakan bila user tidak ingin memasukkan namanya ke dalam *record* skor. Implementasi halaman antarmuka register dapat dilihat pada Gambar 3.25.



Gambar 3.24 Halaman Top Score



Gambar 3.25 Halaman Register

g) Antarmuka How to Play

Pada halaman antarmuka *how to play* ini berisi penjelasan singkat mengenai sudoku dan aturan permainannya serta bagaimana cara memainkan game sudoku di sistem ini. Implementasi antarmuka *how to play* dapat dilihat pada Gambar 3.26.



Gambar 3.26 Halaman How to Play

h) Antarmuka Credit

Halaman antarmuka credit digunakan untuk melihat alasan dibuatnya aplikasi ini dan juga nama pembuat aplikasi ini. Implemantasi halaman antarmuka credit dapat dilihat pada Gambar 3.27.



Gambar 3.27 Halaman Credit

3.4.2 Aturan Permainan

Saat aplikasi ini pertama dibuka maka akan muncul beberapa pilihan menu yang salah satunya yaitu *play* yang merupakan menu untuk memulai permainan. Namun sebelum masuk kedalam permainan, *user* terlebih dahulu diminta untuk menentukan tipe grid papan permainan dan tingkat kesulitan permainan tersebut. Kemudian papan permainan akan muncul sesuai dengan kriteria yang dipilih oleh *user* tadi.

Setelah itu *user* dapat memulai permainan dengan mulai mengisikan angka dengan aturan sudoku yang berlaku, yaitu tidak boleh ada angka yang sama dalam satu baris, kolom, dan *subgrid* (kotak kecil yang masing-masing berisi 9 sel angka). Ketika bermain *user* dapat menggunakan menu yang ada di bagian kiri papan permainan maupun di dalam tampilan halaman menu. Dalam tampilan menu, *user* dapat memilih menu "Resume" untuk kembali ke papan permainan, "Restart Game" untuk mengulang game yang sama mulai dari awal, "New Game" untuk memainkan game dengan pilihan grid dan level yang sama namun angka yang berbeda, maupun "Main Menu" yang merupakan menu untuk kembali ke halaman awal game pertama kali dibuka.

Selain menu yang terdapat pada tampilan halaman menu, *user* dapat menggunakan menu yang berada di sisi kiri papan permainan seperti "Hint". Dalam hal ini ketika *user* memilih menu *hint* maka sistem akan menunjukkan kotak selanjutnya yang bisa diisi berserta angka yang harus diisikan. Selain *hint* terdapat juga menu "Solve" yang berfungsi untuk menyelesaikan permainan yang sedang *user* mainkan secara otomatis. Ada juga menu "Check" yang ketika *user*

memilih menu tersebut, sistem akan mengecek apakah di papan permainan yang sedang dimainkan oleh *user* terdapat angka yang sama baik di satu baris, kolom, maupun *subgrid*nya. Selain itu terdapat menu "Undo" yang berfungsi untuk menghapus angka yang terakhir yang dimasukkan oleh *user* kedalam papan permainan.

Lalu saat semua sel kosong sudah terisi semua dengan benar yaitu tidak ada angka yang sama di baris, kolom, dan *subgrid* yang ada maka akan muncul pernyataan bahwa *game* telah terselesaikan dan kemudian *user* diminta untuk memasukkan namanya untuk ditampilkan di halaman Top Score. Namun bila pernyataan bahwa *game* telah terselesaikan tersebut tidak muncul maka itu merupakan pertanda bahwa masih terdapat angka yang sama pada satu baris, kolom, atau pada sebuah *subgrid*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Program

Pada tahap ini menjelaskan mengenai pengujian sistem ”Aplikasi Game Puzzle Sudoku dengan Algoritma *Backtracking*”. Pengujian dilakukan secara menyeluruh untuk mengetahui kinerja sistem, serta kelemahan ataupun kesalahan yang mungkin terjadi pada saat sistem ini dijalankan.

4.2 Analisis Kinerja Sistem

4.2.1 Proses pada Halaman Papan Permainan

1. Proses Pengisian dan Penghapusan Angka

   Pada pengujian normal, *user* akan memasukkan angka pada kotak-kotak papan permainan yang kosong. Saat *user* mengklik kiri pada kotak tersebut maka akan muncul pilihan angka yang dapat diisikan. Untuk pilihan angka pada papan permainan 4x4 dapat dilihal pada Gambar 4.1, untuk papan permainan 6x6 dapat dilihat pada Gambar 4.2, dan untuk pilihan angka papan permainan 9x9 dapat dilihat pada Gambar 4.3.

   Dari pilihan angka yang muncul tersebut, *user* dapat memilih jawaban angka untuk kotak itu atau memilih angka penanda sementara (*penciling*) sebagai salah satu kemungkinan angka jawaban kotak tersebut. Untuk memilih jawaban angka dari pilihan angka yang ada *user* dapat mengklik

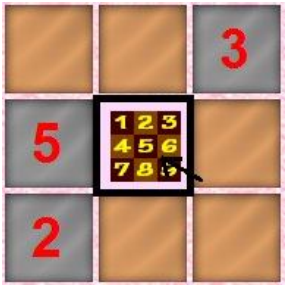
kiri pada pilihan angka, dapat dilihat pada Gambar 4.4. Angka jawaban ini ditandai dengan warna kuning. Sedangkan untuk memilih angka penanda sementara, *user* dapat mengklik kanan pada pilihan angka yang ada, dapat dilihat pada Gambar 4.5. Angka penanda sementara ini ditandai dengan angka kecil di pojok kiri atas kotak berwarna abu-abu. Untuk angka penanda sementara ini *user* dapat memilih hingga maksimal 5 angka penanda sementara.



Gambar 4.1 Pilihan angka pada papan permainan 4x4



Gambar 4.2 Pilihan angka pada papan permainan 6x6



Gambar 4.3 Pilihan angka pada papan permainan 9x9

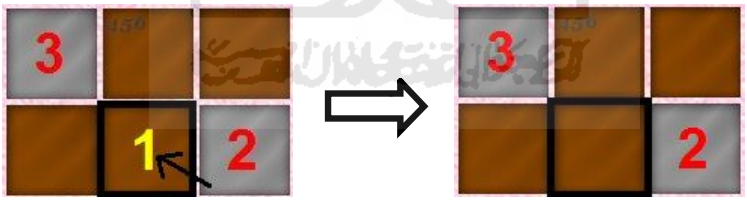


Gambar 4.4 Angka jawaban



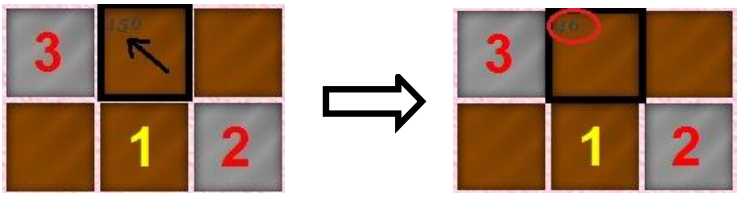
Gambar 4.5 Angka penanda sementara

Untuk menghapus angka yang telah diisikan oleh *user* baik berupa angka jawaban, maupun angka penanda sementara dapat dilakukan dengan mengklik kanan pada angka yang ingin dihapuskan. Penghapusan angka jawaban dapat dilihat pada Gambar 4.6. sedangkan penghapusan angka penanda sementara dapat dilihat pada Gambar 4.7.



Gambar 4.6 Penghapusan angka jawaban

Setelah melakukan proses pengisian angka hingga selesai dan sudah salam kondisi yang benar, yaitu tidak ada angka yang sama pada baris, kolom, atau kotak sudoku, maka akan muncul pesan konfirmasi yang menandakan bahwa permainan telah berhasil di selesaikan. Seperti yang terlihat pada Gambar 4.8.



Gambar 4.7 Penghapusan angka penanda sementara



Gambar 4.8 Pesan pemberitahuan permainan telah diselesaikan

2. Proses *Solve*

Jika tombol *Solve* ditekan, maka akan muncul sebuah pesan konfirmasi yang menanyakan apakah *user* ingin melakukan perintah *solve*, dapat dilihat pada Gambar 4.9. Jika *user* memilih tombol *No* maka pesan tersebut akan hilang dan *user* dapat melanjutkan permainannya. Sedangkan tombol *Yes* ditekan apabila *user* ingin menjalankan perintah *solve* tersebut, dimana setelah tombol *Yes* ditekan maka angka jawaban akan otomatis muncul satu persatu pada papan permainan.



Gambar 4.9 Pesan konfirmasi *solve*

Setelah sistem melakukan eksekusi perintah *solve* dan permainan selesai maka akan muncul pesan pemberitahuan seperti yang terlihat pada Gambar 4.10, namun jika permainan tidak dapat diselesaikan maka akan muncul pesan konfirmasi baru seperti yang terlihat pada Gambar 4.11.



Gambar 4.10 Pesan pemberitahuan *solve* selesai



Gambar 4.11 Pesan konfirmasi *solve game* dari awal

Apabila setelah dilakukan proses *solve game* dari awal permainan dan *game* tetap tidak dapat diselesaikan maka akan muncul pesan pernyataan bahwa *game* tersebut tidak dapat diselesaikan, seperti yang terlihat pada Gambar 4.12.



Gambar 4.12 Pesan pernyataan game tidak dapat diselesaikan

3. Proses *Hint*

Jika tombol *hint* ditekan maka akan muncul angka jawaban pada kotak yang sebelumnya kosong selama beberapa detik, dan kemudian akan hilang kembali. Proses tersebut dapat dilihat pada Gambar 4.13.

4. Proses *Check*

Jika terdapat angka yang sama pada baris, kolom, atau kotak pada papan permainan dan kemudian *user* menekan tombol *check* maka akan muncul pesan pemberitahuan bahwa ada angka yang sama. Seperti yang terlihat pada Gambar 4.14. Namun jika tidak ada angka yang sama, maka pesan pemberitahuan yang muncul akan menyatakan bahwa pengisian yang dilakukan *user* sudah tepat. Seperti yang terlihat pada Gambar 4.15.



Gambar 4.13 Proses *hint*



Gambar 4.14 Pesan pemberitahuan ada angka yang sama



Gambar 4.15 Pesan pemberitahuan pengisian sudah benar

5. Proses *Undo*

Jika sebelumnya *user* telah melakukan proses pengisian atau penghapusan angka pada papan permainan dan kemudian menekan tombol *Undo*, maka kegiatan terakhir yang dilakukan akan dimundurkan ke keadaan

sebelumnya. Misalnya, kegiatan terakhir yang dilakukan *user* adalah menghapus isian angka jawaban pada sebuah sel, maka angka yang dihapus tersebut akan dimunculkan kembali pada sel tersebut. Dapat dilihat pada Gambar 4.16. Namun bila sebelumnya *user* belum melakukan aktifitas apapun, maka tombol *undo* tidak aktif dan tidak dapat ditekan.



Gambar 4.16 Proses *Undo*

4.2.2 Proses pada Halaman Register

1. Proses pengisian dan penghapusan nama

Proses pengisian nama pada *form* register dapat dilakukan dengan mengklik pada tombol huruf, angka dan simbol yang tersedia maksimal sebanyak 5 karakter. Karakter pada tombol yang dipilih tersebut akan berubah warna dari merah menjadi putih. Karakter yang telah dipilih tersebut akan muncul pada 5 kotak kosong yang berada dibawah, dapat dilihat pada Gambar 4.17. Untuk menghapus karakter yang telah dipilih dapat digunakan tombol yang memiliki karakter “<”. Seperti yang terlihat pada Gambar 4.18.

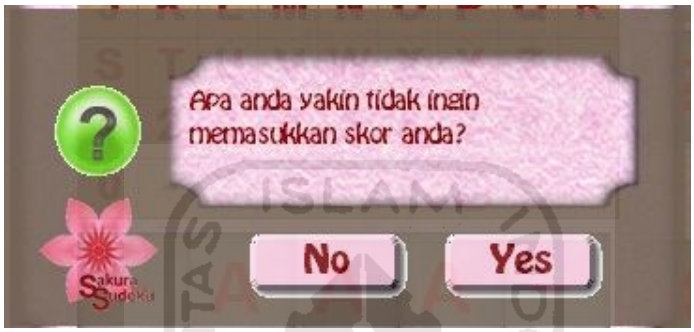


Gambar 4.17 Pengisian nama



Gambar 4.18 Penghapusan nama

Setelah memasukkan nama yang dikehendaki *user* harus mengklik tombol *OK* sebagai perintah untuk memasukkan nama yang telah dibuat terebut kedalam table database. Namun jika *user* menekan tombol *Cancel* maka akan muncul pesan konfirmasi apakah *user* tidak akan memasukkan namanya. Seperti yang terlihat pada Gambar 4.19.



Gambar 4.19 Pesan konfirmasi tidak menyimpan skor

4.2.3 Proses pada Halaman Menu

1. Proses *New Game*

Jika menekan tombol *New Game* maka akan muncul pesan konfirmasi apakah *user* ingin mengakhiri permainan yang sedang berlangsung, seperti yang terlihat pada Gambar 4.20. Apabila *user* memilih *Yes* maka sistem akan menutup permainan yang sedang berlangsung dan membuka papan permainan yang baru. Sedangkan tombol *No* untuk membatalkan perintah dan kembali ke papan permainan sebelumnya.



Gambar 4.20 Pesan konfirmasi keluar dari permainan

2. Proses *MainMenu*

Jika menekan tombol *MainMenu* pada *form* menu maka akan muncul pesan konfirmasi apakah *user* ingin mengakhiri permainan yang sedang berlangsung, seperti yang terlihat pada Gambar 4.20. Apabila *user* mengklik *Yes* maka papan permainan akan ditutup dan *user* akan kembali ke halaman awal. Sedangkan apabila *user* mengklik *No* maka *user* akan kembali ke halaman papan permainan.

3. Proses *Restart*

Jika menekan tombol *restart* maka akan muncul pesan konfirmasi apakah *user* ingin mengulang game yang sedang berlangsung, seperti yang terlihat pada gambar 4.21. Jika *user* menekan tombol *Yes* maka semua angka isian yang sudah dimasukkan *user* pada papan permainan akan dihapus dan waktu akan di*reset* ke keadaan semula. Namun jika *user* menekan tombol *No* maka pesan akan hilang dan *user* dapat melanjutkan permainannya kembali.



Gambar 4.21 Pesan konfirmasi mengulang *game*

4.3 Pengujian Data Analisis

Sebuah sistem belum bisa dikatakan bagus apabila belum dilakukan pengujian. Pengujian ini dilakukan untuk mengetahui kekurangan pada sistem yang telah dibuat. Maka untuk mengetahui kekurangan pada *game* yang telah dibuat ini, sebanyak 20 orang dengan usia 15-25 tahun dimana diminta untuk memainkan *game* yang telah dibuat maupun *game* perbandingan. Kemudian setelah memainkan *game-game* tersebut mereka diminta untuk mengisi kuisioner yang telah disediakan. Kuisioner ini digunkan untuk mengetahui kekurangan dari *game* yang telah dibuat. Bentuk kuisioner dapat dilihat pada halaman lampiran.

Dari kuisioner tersebut diberikan beberapa nilai untuk memudahkan perhitungan hasil analisisnya, yaitu:

Nilai 1 untuk jawaban tidak baik.

Nilai 2 untuk jawaban kurang.

Nilai 3 untuk jawaban cukup.

Nilai 4 untuk jawaban baik.

Nilai 5 untuk jawaban sangat baik.

Dari nilai tersebut dapat digunakan untuk menghitung nilai rata-rata dari jawaban responden. Pada pengujian ini respondennya akan dipisah antara responden yang pernah memainkan *game* sudoku sebelumnya dan responden yang belum pernah memainkan *game* sudoku sebelumnya. Rumus untuk menghitung nilai rata-ratanya adalah:

$$\text{Rata-rata} = \frac{\sum_{i=1}^{N} a_i}{N}$$

Keterangan: a = nilai jawaban responden

N = jumlah responden

Tabel 4.1 Hasil analisis responden yang belum pernah memainkan *game* sudoku

Pertanyaan	Jawaban responden										
	A	B	C	D	E	F	G	H	I	J	Rata-rata
Tampilan antarmuka?	4	3	5	3	5	3	4	4	3	4	3.8
Tingkat kesulitan?	3	3	3	4	3	3	3	2	4	4	3.2
Variasi papan permainan?	4	4	4	3	4	3	3	4	4	4	3.7
Apakah menghibur?	3	4	4	4	4	3	4	4	3	3	3.6
Apakah akan memainkannya lagi?	2	4	4	3	5	2	5	4	3	3	3.5

Tabel 4.2 Hasil analisis responden yang sudah pernah memainkan *game* sudoku

Pertanyaan	Game	Jawaban responden										
		A	B	C	D	E	F	G	H	I	J	Rata-rata
Tampilan antarmuka?	Sakura Sudoku	4	4	4	3	3	4	4	4	3	4	3.7
	Hodoku	3	2	2	2	2	2	2	2	3	2	2.2
	Sudoku pagoda	5	5	5	4	5	4	5	4	4	5	4.6

Tingkat kesulitan?	Sakura Sudoku	3	3	3	3	3	3	3	3	3	5	3.2
	Hodoku	3	3	3	2	3	3	3	3	4	3	3.0
	Sudoku pagoda	3	5	3	4	4	4	5	4	4	2	3.8
Variasi tipe papan permainan?	Sakura Sudoku	4	4	3	3	3	4	4	4	4	5	3.8
	Hodoku	3	2	2	2	2	2	2	3	1	4	2.3
	Sudoku pagoda	5	5	4	4	5	4	4	5	5	2	4.3
Menu-menu yang disediakan	Sakura Sudoku	3	3	4	3	4	3	3	2	3	5	3.3
	Hodoku	3	3	3	3	3	3	3	2	4	3	3.0
	Sudoku pagoda	4	5	4	4	4	4	4	4	4	3	4.0
Apakah menghibur?	Sakura Sudoku	4	2	4	3	4	3	3	4	3	4	3.4
	Hodoku	3	2	3	3	3	3	3	3	2	3	2.8
	Sudoku pagoda	5	2	5	3	4	3	3	4	4	3	3.6
Apakah akan memainkannya lagi?	Sakura Sudoku	5	2	3	2	3	2	3	5	3	3	3.1
	Hodoku	2	2	2	2	2	2	3	3	3	3	2.4
	Sudoku pagoda	5	2	5	2	5	2	3	5	3	3	3.5

4.4 Kelebihan dan Kekurangan Sistem

Setelah menyelesaikan tahap analisis, maka dapat diketahui kinerja dari sistem yang dibuat secara keseluruhan sehingga dapat diketahui kelebihan dan kekurangan sistem ”Aplikasi Game Puzzle Sudoku dengan Algoritma *Backtracking*” yang dibangun. Kelebihan dan kekurangan sistem ini adalah sebagai berikut.

a. Kelebihan Sistem

1. Terdapat 3 pilihan papan permainan dan 3 tingkat kesulitan yang dapat dimainkan.
2. Pada sistem ini soal-soal yang ditampilkan pada setiap tipe papan permainan merupakan hasil dari rotasi soal yang sama, sehingga pemain dapat mengulang-ulang permainan pada tipe papan permainan yang sama dengan bentuk soal yang berbeda.

3. Adanya skoring yang dapat membuat *user* semakin tertantang untuk mempercepat penyelesaian permainan agar skor yang dihasilkan lebih tinggi.
4. Saat permainan sedang berlangsung, *user* dapat mengisi kotak papan permainan dengan angka penanda sementara (*penciling*) sebagai penanda angka-angka yang mungkin untuk diisikan di kotak tersebut.

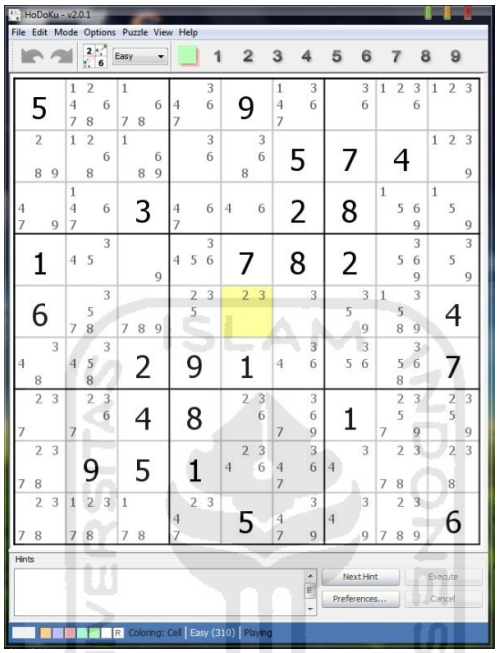
b. Kekurangan Sistem

1. Soal yang terdapat pada tiap *grid* yaitu 4x4, 6x6, dan 9x9 belum pas dengan rotasi soal yang dibuat, hal ini mengakibatkan sering terjadinya soal yang tidak dapat diselesaikan.
2. Tidak terdapat efek suara (*backsound*) pada sistem yang dibuat.

4.5 Analisis Perbandingan

Untuk mengetahui apakah *game* yang dibuat sudah memiliki standar *game* yang baik, maka dilakukan analisis perbandingan terhadap beberapa *game* sudoku lainnya, yaitu Hodoku, Sudoku Pagoda, dan Sudoku Mobile. Hodoku yang merupakan *game* sudoku berbasis *open source* dengan bahasa pemrograman java yang tampilah antarmukanya dapat dilihat pada Gambar 4.22, sedangkan Sudoku Pagoda merupakan *game* komersil buatan Revlexive.com yang bentuk tampilah antarmukanya dapat dilihat pada Gambar 4.23 sampai Gambar 4.25, dan Sudoku Mobile merupakan penamaan sementara untuk *game* sudoku buatan Rahman Adiarto yang merupakan tugas akhir yang dibuatnya yang berjudul ‘Aplikasi

Game Sudoku dengan menggunakan pemrograman J2ME pada Perangkat Mobile'. Perbandingan *game* yang dibuat, Sakura Sudoku, terhadap Hodoku, Sudoku Pagoda, dan Sudoku Mobile dapat dilihat pada Tabel 4.3.



Gambar 4.22 Tampilan antarmuka *game* Hodoku



Gambar 4.23 Tampilan antarmuka *main* menu *game* Sudoku Pagoda



Gambar 4.24 Tampilan antarmuka pilih *grid game* Sudoku Pagoda



Gambar 4.25 Tampilan antarmuka permainan *game* Sudoku Pagoda

Tabel 4.3 Hasil Perbandingan

No.	Jenis Perbandingan	Sakura Sudoku	Hodoku	Sudoku Pagoda	Sudoku Mobile
1	Variasi Grid	Ada	Tidak ada	Ada	Tidak ada
2	Variasi Level	Ada	Ada	Ada	Ada
3	Background	Ada	Tidak ada	Ada	Tidak ada

4	Backsound	Tidak ada	Tidak ada	Ada	Tidak ada
5	Pengurangan soal (*Hint*)	Ada	Ada	Ada	Tidak ada
6	Pemberian solusi (*Solve*)	Ada	Ada	Ada	Ada
7	Proses pengecekan	Ada	Tidak ada	Ada	Tidak ada
8	Pengisian soal sendiri oleh pemain	Tidak ada	Ada	Ada	Ada
9	Perintah simpan game	Tidak ada	Ada	Ada	Tidak ada
10	Skoring	Ada	Tidak ada	Ada	Tidak ada

Dari hasil perbandingan diatas, dapat disimpulkan bahwa aplikasi *game* Sudoku yang dibuat yaitu Sakura Sudoku sudah memiliki fitur-fitur yang lebih banyak dari *game* Hodoku dan Sudoku Mobile, namun masih kalah banyak dibandingkan dengan *game* Sudoku Pagoda. Pada *game* Sakura Sudoku sudah memiliki fitur variasi *grid*, variasi level, *background*, *hint, solve,* proses pengecekan dan scoring, namun tidak memiliki fitur *backsound*, pengisian soal sendiri dan *save game.* Sedangkan pada *game* Hodoku memiliki fitur variasi level, *hint, solve* pengisian soal sendiri dan *save game*, namun tidak memiliki fitur variasi *grid*, *background*, *backsound*, proses pengecekan , dan scoring. Untuk *game* Sudoku Mobile hanya memiliki fitur variasi level, *solve,* dan pengisian soal sendiri. Sedangkan pada *game* Sudoku Pagoda sudah memiliki semua fitur yang dibandingkan.

BAB V

PENUTUP

5.1 Kesimpulan

Setelah dilakukan pengujian terhadap *game* yang telah dibuat tersebut, maka dapat disimpulkan hal-hal sebagai berikut:

1. Dengan aplikasi ini dapat dimainkan *game puzzle* sudoku yang memiliki fitur skoring dan batas.
2. Aplikasi *game* Sudoku ini memiliki interfaceyang menarik serta memiliki fitur-fitur yang dapat membantu dan menambah kesulitan dalam bermain.
3. Kelebihan dari sistem yang dibuat ini yaitu memiliki tiga tipe papan permainan dan 3 tingkat kesulitan yang dapat dipilih sesuai keinginan *user*. Selain itu pemain dapat mengulang-ulang permainan yang sama dengan tipe soal yang berbeda dan sistem ini memiliki fitur skoring dan penanda sementara (*pencilling*) yang dapat membantu permainan.

5.2 Saran

Setelah melihat hasil yang dicapai dalam aplikasi *game* sudoku ini, maka ada beberapa saran yang perlu diperhatikan, yaitu:

1. Mengembangkan perancangan *game* sudoku pada bagian rotasi soal dimana agar tidak terjadi soal yang tidak dapat diselesaikan.
2. Mengembangkan perancangan *game* sudoku dengan menambahkan efek suara (*backsound*) dan *setting* suara.

DAFTAR PUSTAKA

[MAN03] Mangkulo, Hengky Alexander. *Membangun sistem Database dengan Visual Basic 6.0 dan Access 2000*, Jakarta: PT Elex Media Komputindo, 2003.

[MAD01] Madcoms. *Seri Panduan Pemrograman Microsoft Visual Basic 6.0*, Madiun: Penerbit Andi, 2001.

[MAD07] Madcoms. *Panduan Lengkap Microsoft Access 2007*, Madiun: Penerbit Andi, 2007.

[NUR05] Nugroho, Bunafit. *Membuat Animasi dan Tampilan Cantik pada Interface Form*, Yogyakarta: Penerbit Gava Media, 2005.

[DIA09] Dianto, Biraman. *Jaringan Komputer Algoritma Routing*, http://biraman-dianto.blogspot.com/2009/01/jaringan-komputer-algoritma-routing.html terakhir diakses 29 Desember 2010

[MUS10] Musadik, Rihan. *Pengantar dan Sejarah Sudoku*, http://universologi.blogspot.com/2010/03/permainan-seperti-sudoku-sudah-dikenal.html, terakhir diakses 29 Desember 2010

[NUR08] Nur Rochmaddi, Ichsan. *Sejarah Game*, http://inron01.blogspot.com/2008/11/sejarah-game.html, terakhir diakses 14 Desember 2010

[PUT09] Putrafunky. *Jenis-jenis Game di sekitar kita*, http://www.gamexeon.com/forum/console-gaming/57020-jenis-jenis-game-sekitar-kita.html, teakhir diakses 14 Desember 2010

LAMPIRAN