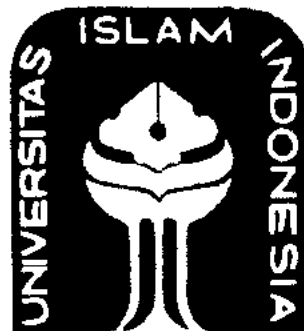


# **APLIKASI GAME PUZZLE SUDOKU DENGAN ALGORITMA BACKTRACKING**

**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**



**Nama : Kurnia Wahyuni S Putri**

**No. Mahasiswa : 06 523 236**

الجامعة الإسلامية  
الابدية لا تتبدل  
Yogyakarta, 27 April 2011

**Pembimbing**

**Ami Fauziah, ST., MT. xy/99**

**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 Fauziah, ST., MT.****Ketua****Beni Suranto, ST.****Anggota I****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...

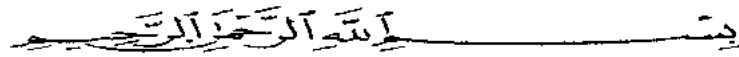
Sholawat dan Salam tak lupa saya haturkan kepada NABI MUHAMMAD SAW... karena beliau semoga saya dapat menjadi orang yang selalu benar dalam melangkah dan di ridhoi oleh Allah....

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 materil, 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...

## KATA PENGANTAR



*Assalamualaikum Wr.Wb.*

Alhamdulillah rabbil`aalam, 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 terselesaikannya 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 Fauziah, ST., MT. selaku Dosen Pembimbing, yang telah banyak memberikan masukan, 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

## 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*

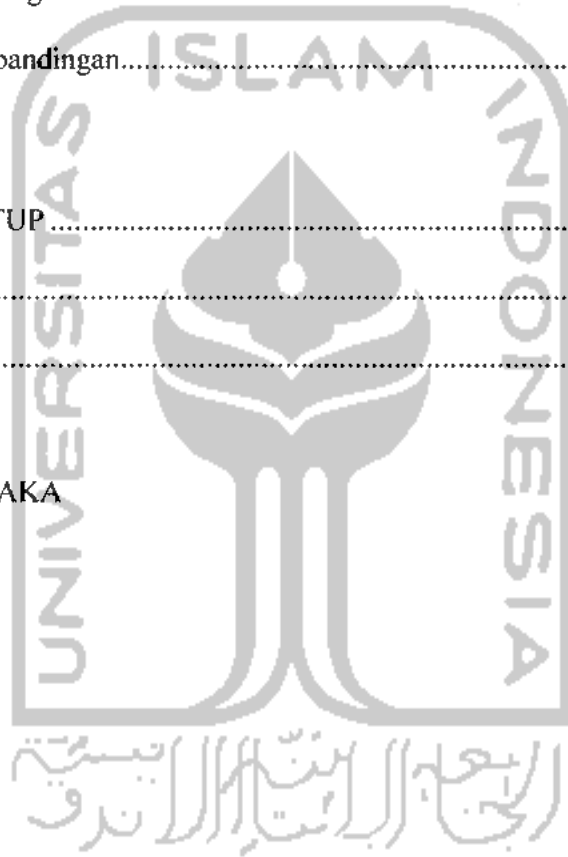
## TAKARIR

<b>Sudoku</b>	Jenis permainan logika angka ringan yang dimainkan dengan cara mengisi 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.
<b>Grid</b>	Keseluruhan kotak pada papan permainan yang terdiri dari 9 x 9 kotak.
<b>Subgrid</b>	Bagian dari <i>grid</i> yang terdiri dari 3 x 3 kotak, dimana 3 x 3 <i>subgrid</i> merupakan satu <i>grid</i> .
<b>Gameboard</b>	Papan permainan dimana permainan akan berlangsung.
<b>Beginner</b>	Tingkat kesulitan yang diperuntukkan kepada pemain pemula.
<b>Intermediate</b>	Tingkat kesulitan yang diperuntukkan kepada pemain yang sudah cukup mahir.
<b>Advance</b>	Tingkat kesulitan yang diperuntukkan kepada pemain yang sudah mahir.
<b>Algoritma Backtracking</b>	Algoritma runut-balik. Dimana pencarian solusi suatu permasalahan hanya dengan mempertimbangkan pencarian yang mengarah ke solusi saja.



<b>BAB II LANDASAN TEORI .....</b>	<b>6</b>
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
 <b>BAB III METODOLOGI.....</b>	 <b>22</b>
3.1 Analisis Sistem.....	22
3.2 Hasil Analisis.....	24
3.3 Perancangan Perangkat Lunak .....	27
3.3.1 Perancangan Diagram Alir ( <i>Flowchart</i> ).....	27
3.3.2 Perancangan Antarmuka.....	36
3.4 Implementasi Perangkat Lunak.....	47
3.4.1 Implementasi.....	47
3.4.2 Aturan Permainan .....	57
 <b>BAB IV HASIL DAN PEMBAHASAN.....</b>	 <b>59</b>
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
<b>BAB V PENUTUP</b> .....	<b>76</b>
5.1 Kesimpulan .....	76
5.2 Saran .....	77
<b>DAFTAR PUSTAKA</b>	
<b>LAMPIRAN</b>	



**DAFTAR TABEL**

Tabel 4.1	Hasil Analisis Responden yang Belum Pernah Memainkan <i>Game</i>	
	Sudoku .....	71
Tabel 4.2	Hasil Analisis Responden yang Sudah Pernah Memainkan <i>Game</i>	
	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 <i>Undo</i> .....	30
Gambar 3.4	Flowchart Proses <i>Check</i> .....	32
Gambar 3.5	Flowchart Proses <i>Hint</i> .....	33
Gambar 3.6	Flowchart Proses <i>Solve</i> .....	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

# 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

### 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.

## 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 *oscilloscope*.

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 menyukai kisah 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



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. *Advergimes*.

### 2.1.2 Sudoku

#### a) Sejarah Sudoku

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



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  $3 \times 3$  (disebut blok/ *subgrid*) yang disusun sedemikian rupa sehingga menghasilkan kotak besar berukuran  $9 \times 9$ . 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

### 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.

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.

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 :

## 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 *game* sudoku buatan Rahman Adiartha 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

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

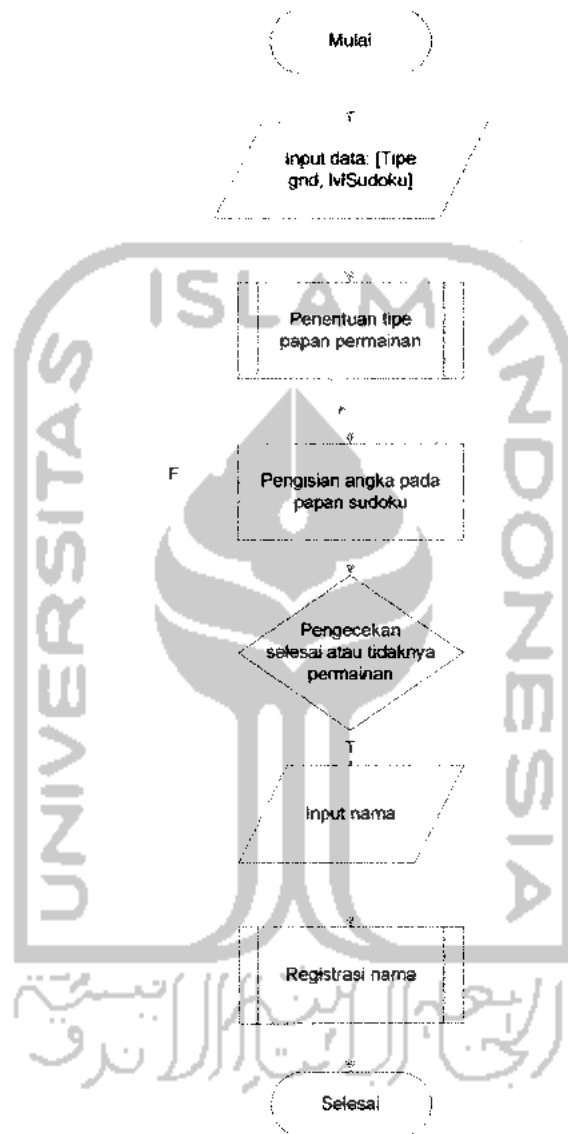
#### 3.3.1 Diagram Alir (*Flowchart*)

1. *Flowchart* Sistem

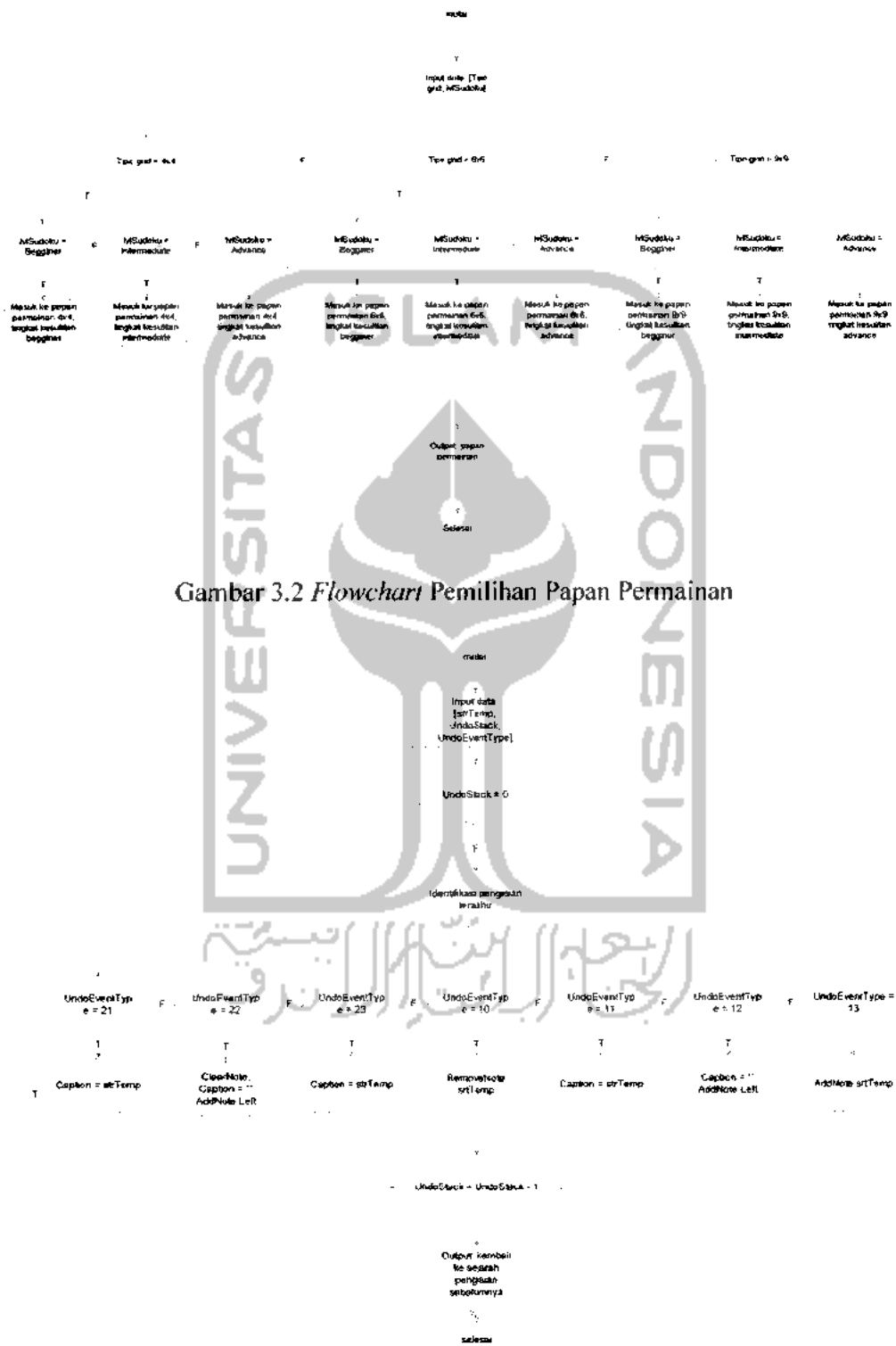
Alur proses sistem dari memulai permainan sampai dengan proses inti dan hasilnya digambarkan sebagai sebuah diagram alir secara keseluruhan. Diagram alir sistem ini dapat dilihat pada Gambar 3.1.

2. *Flowchart* pemilihan papan permainan

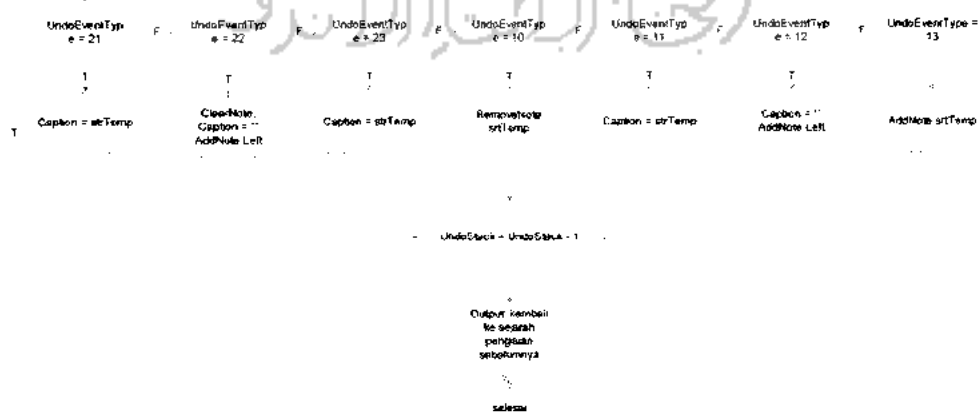
*Flowchart* untuk pemilihan tipe papan permainan dapat dilihat pada Gambar 3.2. Inisialisasi dari *flowchart* ini adalah tipe *grid* dan tingkat kesulitan. Sistem akan melihat apakah tipe *grid* yang dipilih adalah 4x4, jika benar maka akan memeriksa tingkat kesulitan yang dipilih, jika salah maka sistem akan memeriksa apakah tipe *grid* yang dipilih 6x6, jika benar maka selanjutnya sistem akan memeriksa tingkat kesulitan yang dipilih, jika salah



Gambar 3.1 *Flowchart* Sistem



Gambar 3.2 Flowchart Pemilihan Papan Permainan



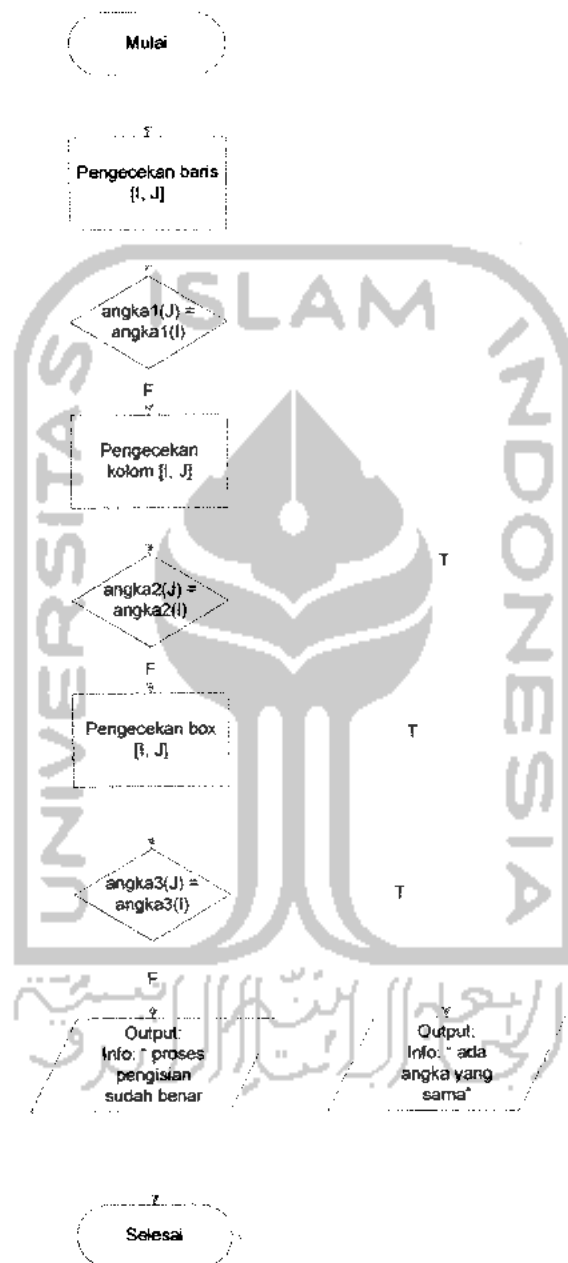
Gambar 3.3 Flowchart Proses Undo

#### 4. *Flowchart proses check*

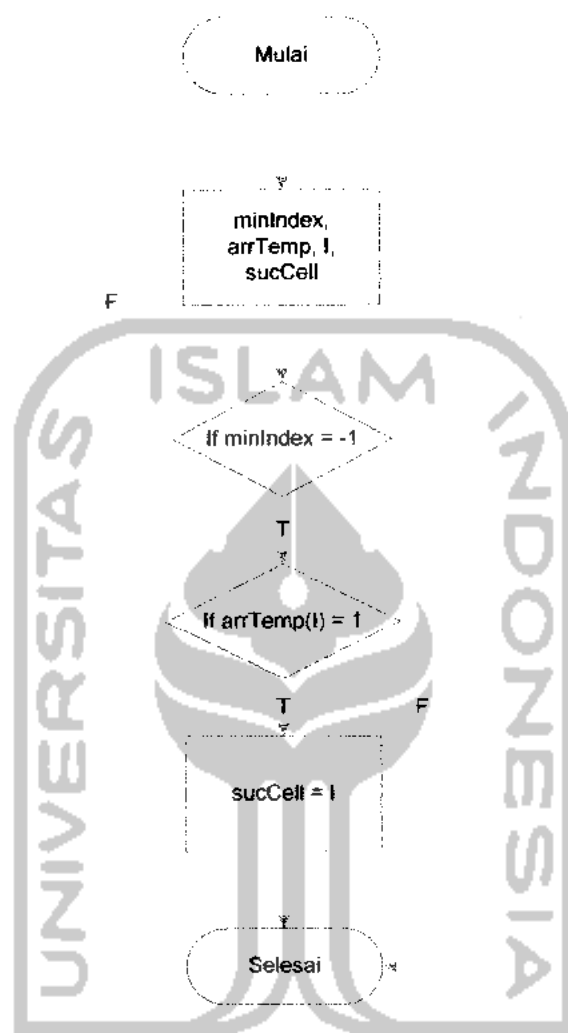
Proses ini dilakukan untuk pengecekan apakah terdapat angka yang sama pada baris, kolom, ataupun kotak sudoku. Dapat dilihat pada Gambar 3.4. Dimulai dengan pengecekan pada tiap baris papan sudoku, apabila  $\text{Angka1}(J) = \text{Angka1}(I)$  maka akan keluar output info yang menyatakan bahwa terdapat angka yang sama pada papan sudoku. Jika salah maka kemudian dilakukan pengecekan untuk setiap kolom sudoku, yang apabila  $\text{Angka2}(J) = \text{Angka2}(I)$  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  $\text{Angka3}(J) = \text{Angka3}(I)$  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  $\text{minIndex}$ ,  $\text{arrTemp}$ ,  $I$ , dan  $\text{sucCell}$ . Kemudian akan dilakukan pengecekan pada sel sudoku apakah  $\text{minIndex} = -1$  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  $\text{arrTemp}(I) = 1$  maka sel tersebut akan diisi dengan nilai 1 ( $\text{sucCell} = 1$ ), jika  $\text{arrTemp}(I)$  masih lebih dari 1 maka akan dicari sel yang lain.



Gambar 3.4 *Flowchart* Proses *Check*

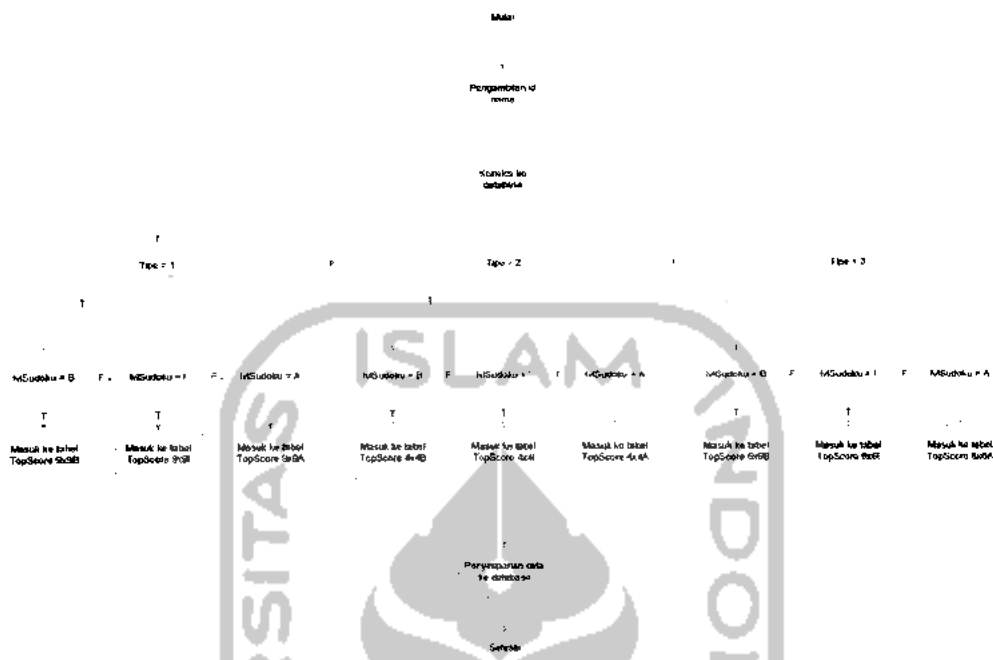


Gambar 3.5 Flowchart Proses Hlint

#### 6. Flowchart proses solve

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





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.

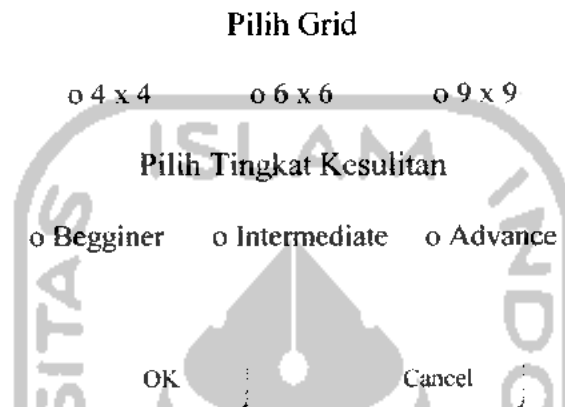


Gambar 3.8 Rancangan Antarmuka Main Menu

b. Rancangan Antarmuka Pemilihan Tipe Grid

Rancangan antarmuka pemilihan tipe grid merupakan halaman yang digunakan untuk menentukan tipe grid dari papan permainan 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 *beginner*, *intermediate*, atau *advance*.



Gambar 3.9 Rancangan Antarmuka Pemilihan Type Grid

c. Rancangan Antarmuka Papan Permainan 4 x 4

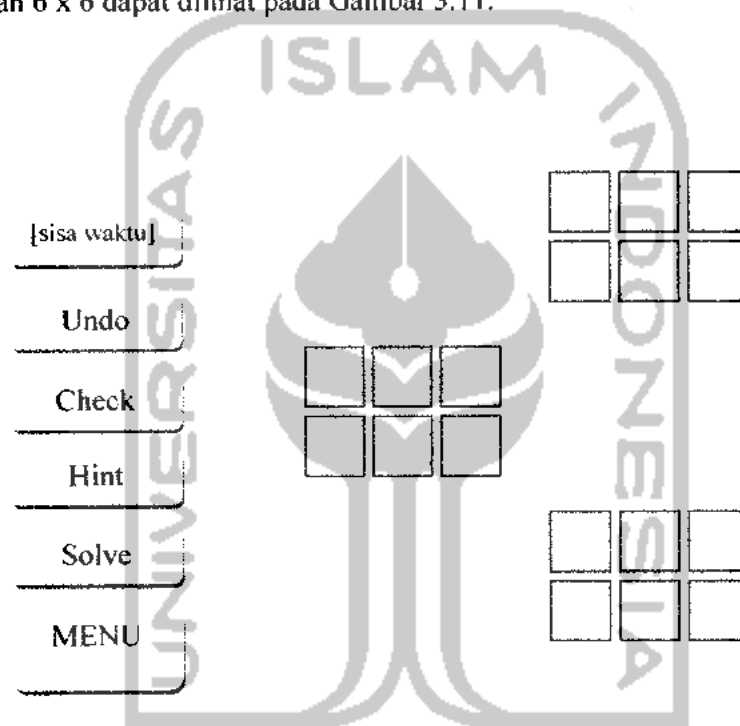
Rancangan antarmuka papan permainan 4 x 4 merupakan halaman yang dimana *user* akan memainkan permainan sudoku ini, dimana pada halaman ini papan permainannya terdiri dari kotak 4x4. Perancangan antarmuka papan permainan 4 x 4 dapat dilihat pada Gambar 3.10.

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.

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.

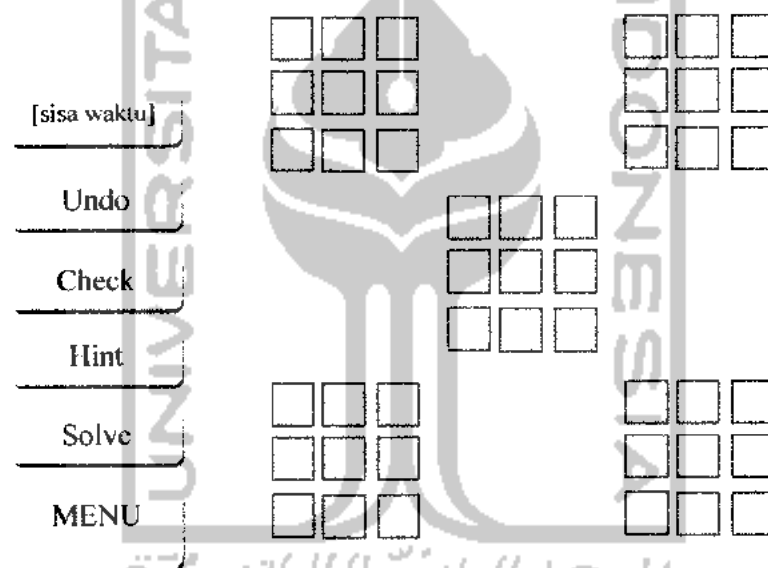


Gambar 3.11 Rancangan Antarmuka Papan Permainan 6 x 6

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.

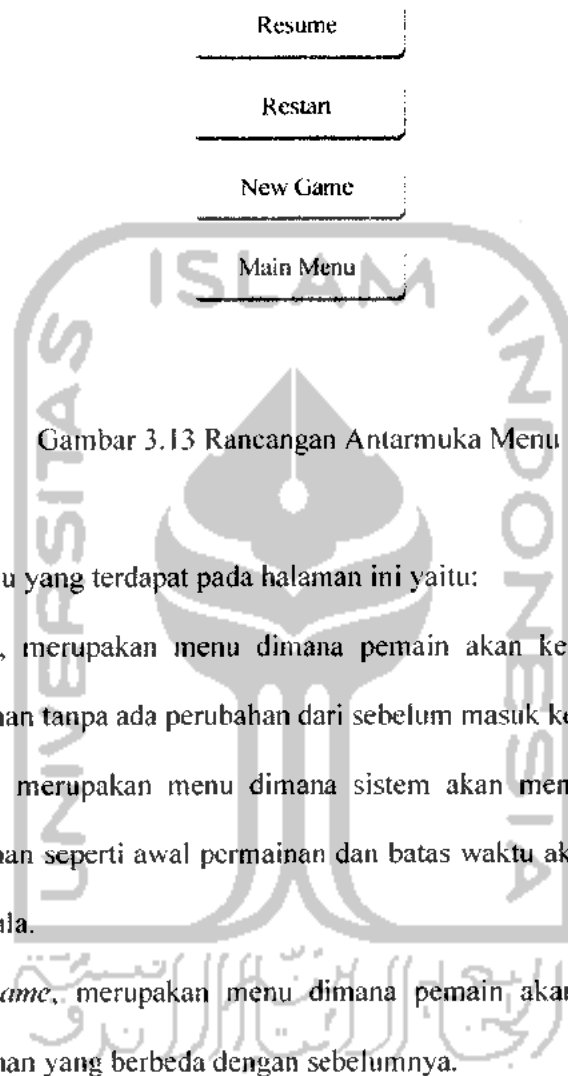
3. *Hint*, merupakan menu dimana sistem akan memberikan bantuan jawaban berupa dimana angka selanjutnya harus diisikan.
4. *Check*, merupakan menu dimana sistem akan mengecek apakah terdapat angka yang sama dalam satu baris, kolom, atau kotaknya.
5. *Undo*, merupakan menu dimana sistem akan menghapus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

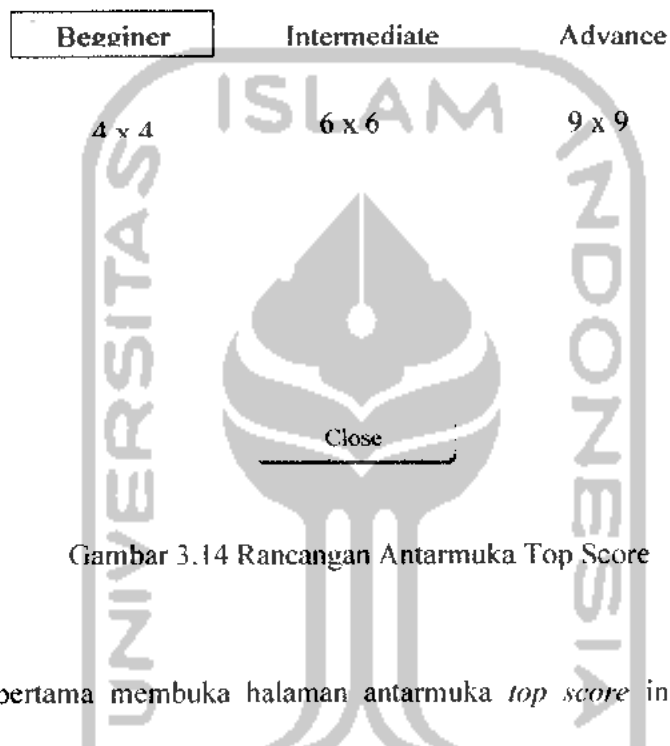
Menu-menu yang terdapat pada halaman ini yaitu:

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 *direset* 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.



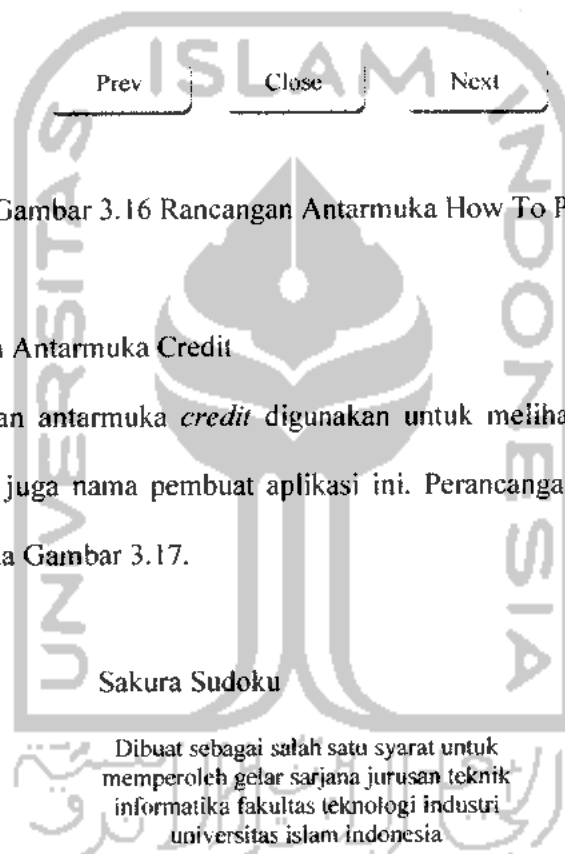
Gambar 3.14 Rancangan Antarmuka Top Score

Saat pertama membuka halaman antarmuka *top score* ini, pemain akan langsung dihadapkan pada 3 skor tertinggi pada tingkat kesulitan *beginner* 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

Penjelasan mengenai aturan main  
dan cara memainkan aplikasi game  
puzzle sudoku



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 pemrograman 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:

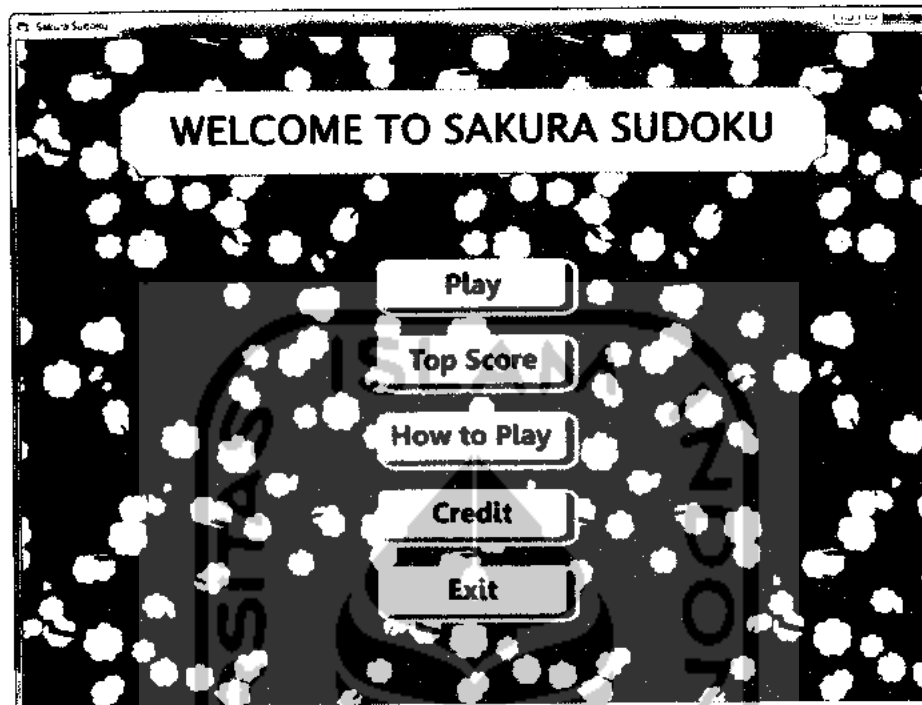
1. Pada tampilan antarmuka hanya menggunakan sedikit animasi saja.
2. Belum terdapat proses *save* permainan yang dapat membantu *user* menyelesaikan permainannya di lain waktu.

##### b. Implementasi Sistem

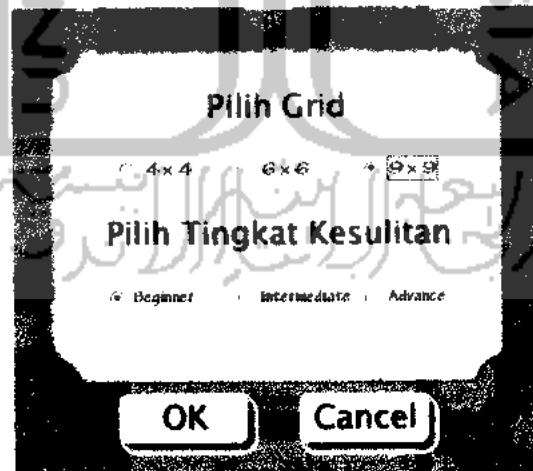
Implementasi merupakan tahap dimana sistem siap dioperasikan pada tahap yang sebenarnya, sehingga akan diketahui apakah sistem yang telah dibuat benar-benar sesuai dengan yang direncanakan. Pada implementasi perangkat lunak ini akan dijelaskan bagaimana sistem ini bekerja dengan tampilan yang telah dibuat.

##### c. Implementasi Antarmuka

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



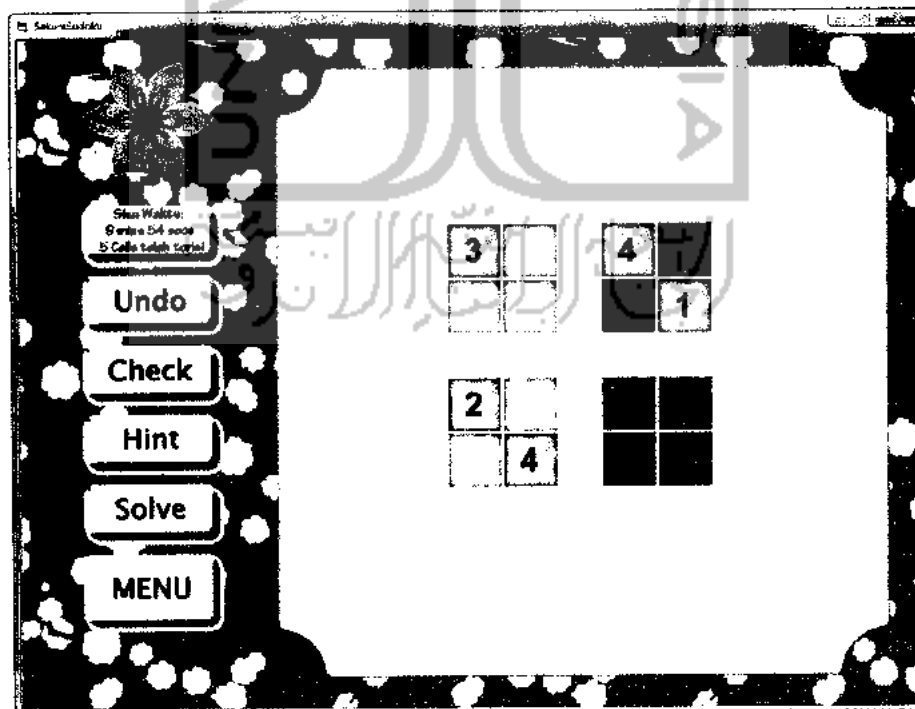
Gambar 3.18 Halaman Main Menu



Gambar 3.19 Halaman Pilih Grid

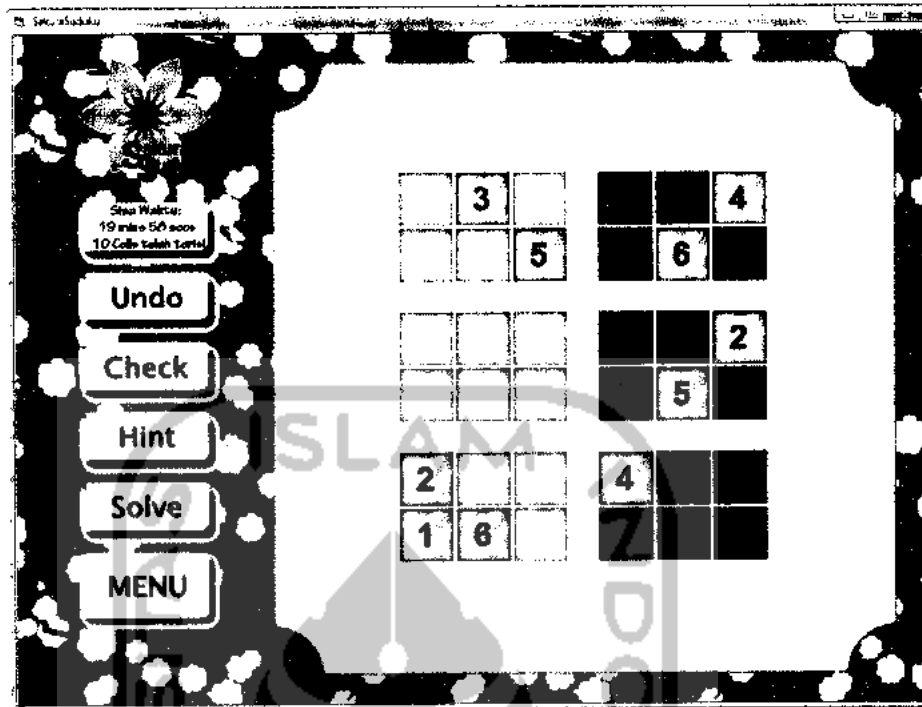
c) Antarmuka Papan Permainan

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.



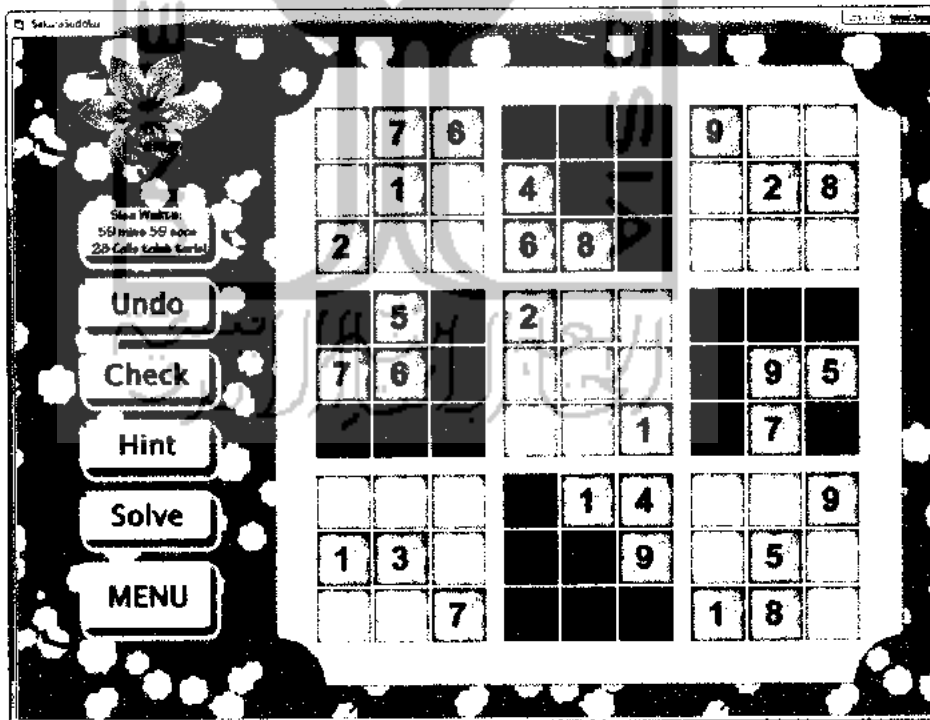
Gambar 3.20 Halaman papan permainan 4 x 4

dengan alga  
 tion <- j  
 awaban yar  
 sel sudoku  
 dex - -1  
 l maka  
 MaxSudokuOp  
 mp(I) <- j  
 mp(I) :  
 I) = 0 mak  
 : "Sudoku  
 I) = 1 maka  
 Cell berik  
 berikutnya.  
 2 maka  
 ku selesai"



Gambar 3.21 Halaman papan permainan 6 x 6

on;  
 ka jawaban  
 ka jawaban  
 ka jawaban  
 kuOption  
 kuOption  
 ; Angka(J)  
 Angka(J) me  
 angka yang :  
 MaxSudokuOpti  
 MaxSudokuOpti  
 ka2(I); Angk  
 (I) = Angka2  
 "Ada angka  
 I to MaxSudo  
 1 to MaxSudo  
 : Angka3(I)  
 Angka3(I) =  
 tput : "Ada  
 se  
 Proses pengi



Gambar 3.22 Halaman papan permainan 9 x 9

d) Antarmuka Menu

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



Gambar 3.23 Halaman Menu

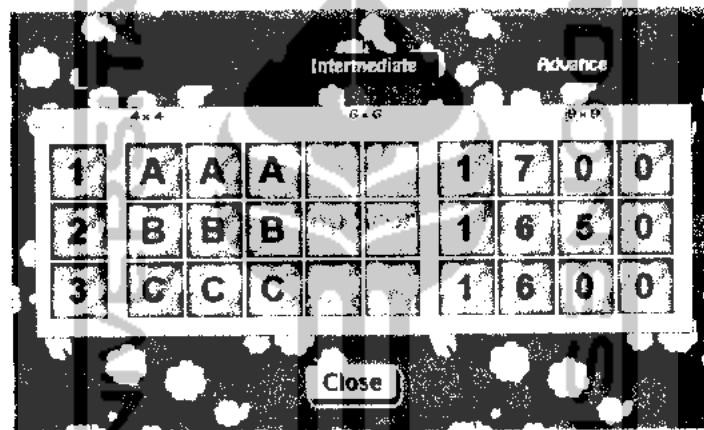
e) Antarmuka Top Score

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.

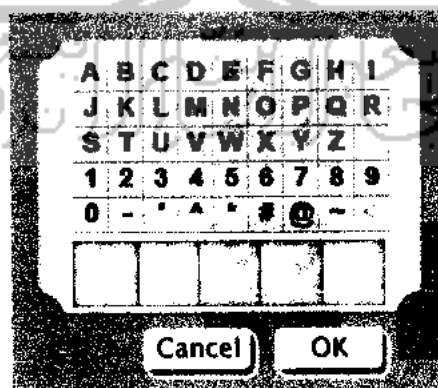
f) Antarmuka Register

Pada halaman antarmuka register ini pemain diminta untuk mengisi 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.



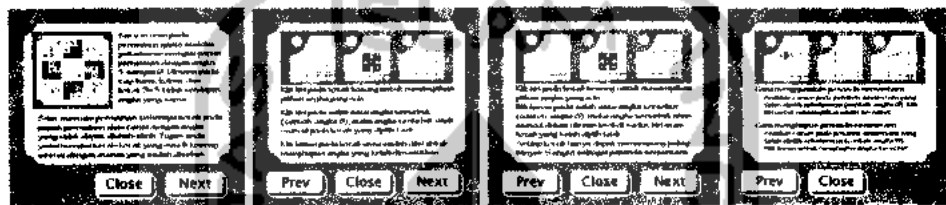
Gambar 3.24 Halaman Top Score



Gambar 3.25 Halaman Register

g) Antarmuka How to Play

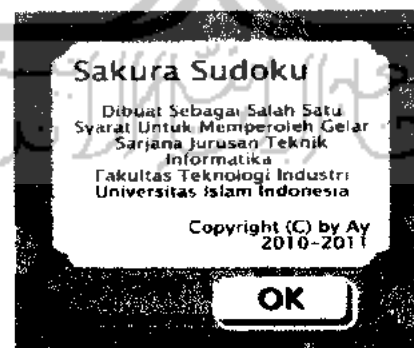
Pada halaman antarmuka *how to play* ini berisi penjelasan singkat mengenai sudoku dan aturan permainannya serta bagaimana cara memainkan game sudoku di sistem ini. Implementasi antarmuka *how to play* dapat dilihat pada Gambar 3.26.



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. Implementasi 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 mengisi 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 beserta angka yang harus diisi. Selain *hint* terdapat juga menu “Solve” yang berfungsi untuk menyelesaikan permainan yang sedang *user* mainkan secara otomatis. Ada juga menu “Check” yang ketika *user*



## 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 dilihat 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



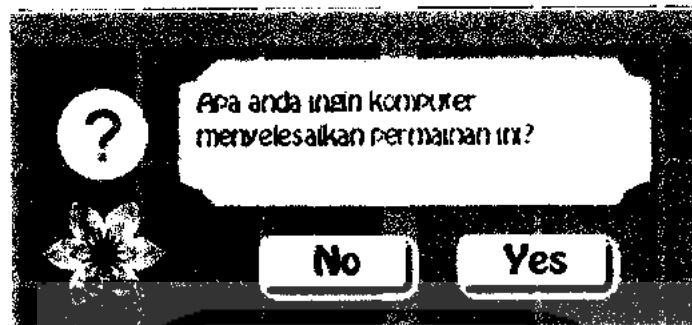
Gambar 4.5 Angka penanda sementara

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 dalam 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.

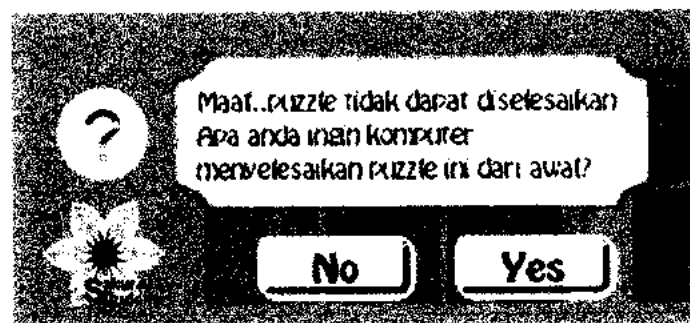


Gambar 4.9 Pesan konfirmasi *solve*

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

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.



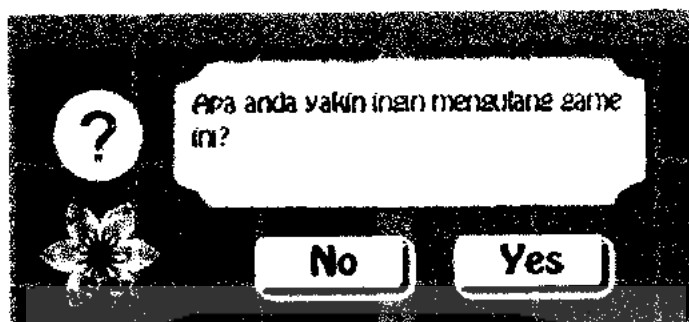
Gambar 4.16 Proses *Undo*

#### 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.

ut dapat di  
 Pada pengi  
 memainkan  
 kan game s  
 h:



Gambar 4.21 Pesan konfirmasi mengulang game

i jawaban res  
 lah responde

sis responden

	B	C
	3	5
	3	3
	4	4
	4	4
	4	4

sis responden y

Game	f
ura Sudoku	4
doku	3
oku pagoda	5

### 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 digunakan 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.

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

background  
pengurangan  
(hint)  
pemberian so  
(Solve)  
proses pengec  
pengisian soa  
sendiri oleh pe  
perintah simp  
ame  
koring

ri hasil perb

yang dibuat

dari game F

ngkan dengan

i fitur varia

kan dan scori

an save game.

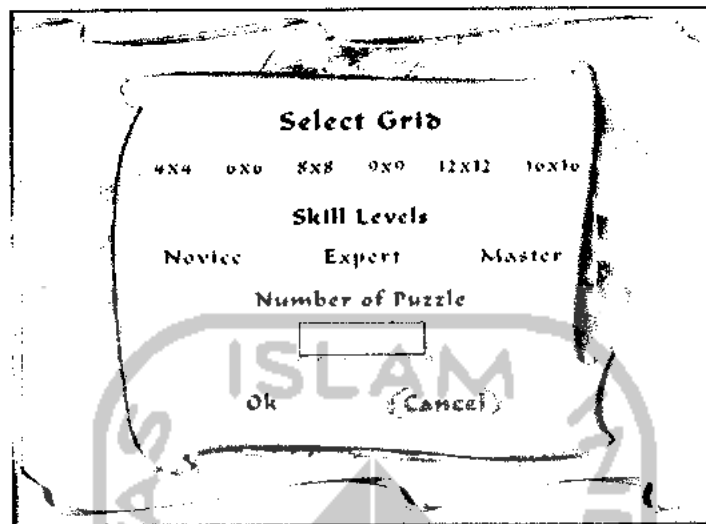
ve pengisian

grid, backgrou

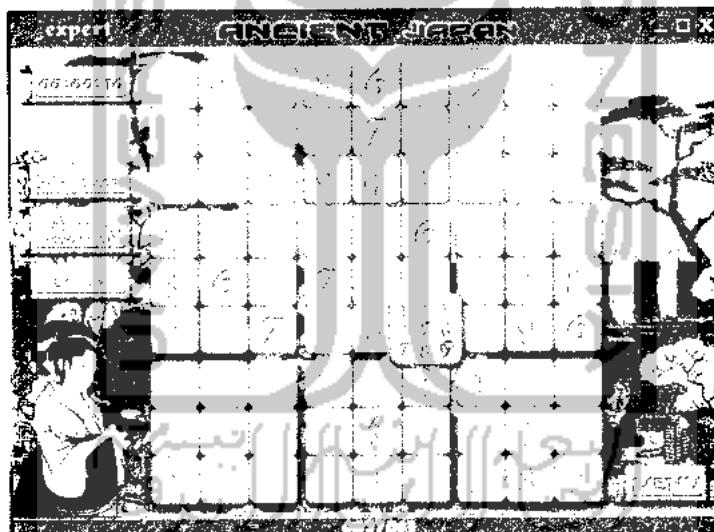
udoku Mobile t

Sedangkan pac

ngkan.



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



[NUR01

## DAFTAR PUSTAKA

- [MAN03] Mangkulo, Hengky Alexander. *Membangun sistem Database dengan Visual Basic 6.0 dan Access 2000*, Jakarta: PT Elex Media Komputindo, 2003.
- [PUT09]
- [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-scperti-sudoku-sudah-dikenal.html>, terakhir diakses 29 Desember 2010



SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : KURNIA WAHYU  
No. Mhs. : 06323236  
Judul TA : \_\_\_\_\_

- o Sering konsultasi
- o Buat timeline pengerjaan, biar g molor.
- o Perjelas <sup>siapa</sup> pengguna. also. backtup key-nya.
- o Lakukan stud. pustaka, ttg masalah & metode ts. and krayikan!

Nilai kemajuan Tugas Akhir: \_\_\_\_\_ (0 - 100)  
(studi pustaka, perancangan, penguasaan materi, keepatan)

Yogyakarta, 11/11

Dosen,

W. Desh

IRWANI PRAMUSINTO  
(nama terang)

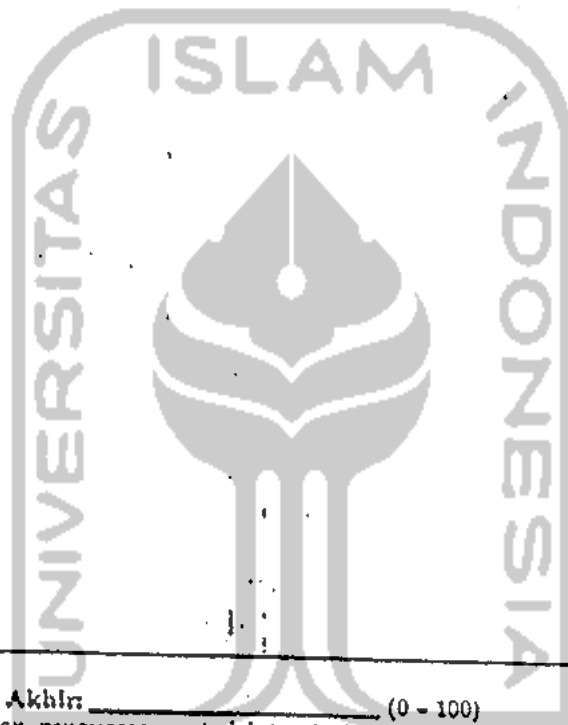
Dilampirkan pada Laporan TA yang diajukan untuk pendadaran



SARAN/USULAN PRESENTASI KEMAJUAN TUGAS AKHIR

Nama Mhs. : Kurnia Wahyuni S P  
No. Mhs. : 06523236  
Judul TA : \_\_\_\_\_

- Lanjutkan x Selesaikan . secepatnya .
- Konsultasikan kemajuan pada pembimbing secepatnya mungkin



Nilai kemajuan Tugas Akhir: \_\_\_\_\_ (0 - 100)  
(studi pustaka, perancangan, penguasaan materi, ketepatan)

Yogyakarta, 11/1

Dosen,

(nama terang)

Dilampirkan pada Laporan TA yang diajukan untuk pendadaran