

**RANCANG BANGUN GAME CATUR UNTUK SMARTPHONE  
DENGAN METODE FORWARD CHAINING PRUNING**

**SKRIPSI**

Oleh:  
**TEGUH SANTOSO**  
NIM. 08650051



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

**2013**

**RANCANG BANGUN GAME CATUR UNTUK SMARTPHONE  
DENGAN METODE FORWARD CHAINING PRUNING**

**SKRIPSI**

Oleh:  
**TEGUH SANTOSO**  
NIM. 08650051



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

**2013**

**RANCANG BANGUN GAME CATUR UNTUK SMARTPHONE  
DENGAN METODE FORWARD CHAINING PRUNING**

**SKRIPSI**

Oleh:  
**TEGUH SANTOSO**  
NIM. 08650051

Telah Disetujui untuk Diuji  
Malang, 18 Januari 2013

Dosen Pembimbing I,

Muhammad Faisal, M.T  
NIP. 197405102005011007

Dosen Pembimbing II,

A'la Syaqui, M.Kom  
NIP. 197712012008011007

Mengetahui,  
Ketua Jurusan Teknik Informatika

Ririen Kusumawati, M.Kom  
NIP. 197203092005012002

**RANCANG BANGUN GAME CATUR UNTUK SMARTPHONE  
DENGAN METODE FORWARD CHAINING PRUNING**

**SKRIPSI**

Oleh:  
**TEGUH SANTOSO**  
**NIM. 08650051**

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

Tanggal, 26 Januari 2013

<b>Susunan Dewan Penguji:</b>		<b>Tanda Tangan</b>
<b>1. Penguji Utama</b>	: <b><u>HANI NURHAYATI, M.T</u></b> <b>NIP. 197806252008012006</b>	( )
<b>2. Ketua Penguji</b>	: <b><u>SUHARTONO, M.Kom</u></b> <b>NIP. 196805192003121001</b>	( )
<b>3. Sekretaris Penguji</b>	: <b><u>MUHAMMAD FAISAL, M.T</u></b> <b>NIP. 197405102005011007</b>	( )
<b>4. Anggota Penguji</b>	: <b><u>A'LA SYAUQL, M.Kom</u></b> <b>NIP. 197712012008011007</b>	( )

**Mengetahui dan Mengesahkan,  
Ketua Jurusan Teknik Informatika**

**Ririen Kusumawati, M.Kom**  
**NIP. 197203092005012002**

## SURAT PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : Teguh Santoso  
NIM : 08650051  
Fakultas / Jurusan : Sains Dan Teknologi / Teknik Informatika  
Judul Penelitian : Rancang Bangun *Game* Catur Untuk *Smartphone* Dengan Metode *Forward Chaining Pruning*

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

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

Malang, 30 Januari 2013  
Yang Menyatakan,

Teguh Santoso  
NIM. 08650051

## Motto

*Nothing is not no, and I believe a miracle will come to the people wanted*



## Lembar Persembahan

Kupersembahkan karyaku yang sederhana ini  
Kepada ayah dan ibu tercinta

Susah payah engkau membiayaiku kuliah selama ini  
Tapi aku tidak bisa memberikan apa-apa  
hanya sebuah buku dan lembaran kertas ini  
yang bisa kupersembahkan padamu  
Yang dalam buku itu terdapat hasil karyaku  
Dan pada lembaran kertas itu tertulis nama anakmu  
Inilah permintaanmu yang baru bisa aku capai selama ini

Wahai ayah ibu  
Terima kasih telah mendoakanku disetiap sholatmu  
Disetiap tetes keringatmu terdapat perjuangan yang sangat besar  
Untuk mendidikku dari seorang anak yang tak tahu apa-apa menjadi seorang sarjana  
Terima kasih sebesar-besarnya kuucapkan untuk ayah dan ibu.....

Dan juga tak lupa untuk kakak perempuanku tercinta  
Yang selalu mendukungku selama ini  
Terima kasih kakak,,,,,

Dan tak lupa untuk teman-temanku yang selalu memberikanku semangat  
I love u all.....

## KATA PENGANTAR

*Alhamdulillah rabbil 'alamin.* Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas segala rahmat dan limpahan hidayahnya Skripsi yang berjudul “Rancang Bangun Game Catur Untuk Smartphone Dengan Metode *Forward Chaining Pruning*” ini dapat diselesaikan. Dan semoga Allah melimpahkan rahmat atas Nabi Muhammad SAW yang senantiasa memberikan cahaya petunjuk kepada kita.

Selanjutnya penulis haturkan ucapan terima kasih seiring do'a dan harapan *jazakumullah ahsanal jaza'* kepada semua pihak yang telah membantu terselesaikannya skripsi ini. Dalam kesempatan ini penulis ingin mengucapkan terima kasih yang tak terhingga kepada:

1. Bapak Muhammad Faisal, M.T dan Bapak A'la Syauqi, M.Kom selaku dosen pembimbing skripsi, yang telah banyak memberikan bimbingan serta memberikan masukan positif kepada penulis dalam menyelesaikan skripsi ini.
2. Ibu Ririen Kusumawati, M.Kom selaku ketua jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Prof. Dr. H. Mudjia Rahardjo, M.Si selaku Rektor UIN Maulana Malik Ibrahim Malang yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
4. Prof. Drs. Sutiman Bambang Sumitro, SU., DSc. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
5. Seluruh Dosen Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan ilmu yang tak terbatas sebagai bekal hidup.
6. Seluruh keluarga yang selalu mendoakan, memberikan motivasi dan dorongan dalam penyelesaian skripsi ini.
7. Teman-teman Teknik Informatika angkatan 2008 khususnya kelas B. Terima kasih atas segala bantuan, dukungan, motivasi, dan kebersamaannya selama ini. Semoga Allah SWT memberikan balasan yang setimpal atas jasa dan bantuan yang telah diberikan.
8. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah banyak membantu dalam penyelesaian skripsi ini.

Semoga penulisan laporan tugas akhir ini bermanfaat bagi pembaca sekalian. Penulis menyadari bahwa tugas akhir ini masih jauh dari kesempurnaan, sehingga dengan segala kerendahan hati penulis mengharapkan kritik dan saran yang membangun.

Malang, 28 Januari 2013

Penulis



## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PERSETUJUAN I.....	ii
LEMBAR PERSETUJUAN II.....	iii
SURAT PERNYATAAN .....	iv
MOTTO .....	v
LEMBAR PERSEMBAHAN .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR .....	xi
ABSTRAK.....	xiii
<b>BAB I: PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan Skripsi .....	5
<b>BAB II: TINJAUAN PUSTAKA .....</b>	<b>7</b>
2.1 Game .....	7
2.1.1 Pengertian Game .....	8
2.1.2 Sejarah Perkembangan Game.....	9
2.1.3 Klasifikasi Game .....	11
2.2 Android.....	20
2.2.1 Kerjasama Dengan Androdi Inc.....	21
2.2.2 Produk awal.....	22
2.2.3 Fitur .....	26
2.2.4 Android bagi Komunitas Open Source .....	26
2.3 <i>Forward Chaining</i> .....	27
2.3.1 Verifikasi Sistem Pakar .....	30
2.3.2 Block Diagram .....	31
2.3.3 Dependency Diagram.....	31
2.3.4 Decision Table.....	31
2.2.5 Rule Base.....	32
2.4 Catur .....	32
2.5 <i>Eclipse, ADT Plugins, Dan SDK</i> .....	46
2.6 <i>Pruning</i> .....	48
2.6.1 Alpha Beta Pruning .....	48
<b>BAB III: DESAIN DAN PERANCANGAN SISTEM .....</b>	<b>51</b>
3.1 Metode Penelitian.....	51
3.2 Analisis Kebutuhan .....	54
3.3 Pengumpulan Data Dan Analisa Masalah.....	56
3.4 Strategi Pemecahan Masalah.....	57
3.5 Analisis Dan Perancangan Desain Sistem.....	58
3.6 Analisis Dan Perancangan Sistem .....	61

3.6.1 Flowchart Permainan.....	62
3.6.2 Flowchart Akses Menu.....	64
3.6.3 Flowchart Permainan Baru.....	64
3.6.4 Flowchart Option.....	66
3.6.5 Flowchart Strategi Permainan Komputer.....	67
3.6.6 Flowchart Strategi Permainan Manusia.....	73
3.6.7 Flowchart Check Status Permainan.....	73
3.6.8 Algoritma Permainan.....	75
3.6.9 Algoritma Permainan Baru.....	75
3.6.10 Algoritma Konfigurasi.....	76
3.6.11 Algoritma Strategi Permainan Komputer.....	76
3.6.12 Algoritma Strategi Manusia Memindahkan Bidak.....	77
3.6.13 Algoritma Check Status Permainan.....	77
<b>BAB IV: HASIL DAN PEMBAHASAN.....</b>	<b>78</b>
4.1 Implementasi Sistem.....	78
4.1.1 Kebutuhan <i>Hardware</i> .....	78
4.1.2 Kebutuhan <i>Software</i> .....	78
4.1.3 Implementasi Desain <i>Interface</i> .....	79
4.1.4 Tampilan Layar Form Awal Aplikasi.....	79
4.1.5 Tampilan Layar Form Menu Permainan.....	80
4.1.6 Tampilan Layar Form Cara Bermain.....	82
4.1.7 Tampilan Layar Form Tentang.....	84
4.1.8 Tampilan Layar Form Mulai Permainan.....	85
4.1.9 Tampilan Layar Pemain Memindahkan Bidak.....	87
4.1.10 Tampilan Akhir Permainan.....	90
4.2 Pengujian Program.....	91
4.2.1 Penginstalan Software Yang Dibutuhkan.....	92
4.2.2 Cara Menjalankan Program Catur.....	92
4.2.3 Uji Coba Desain Interface, Kemudahan penggunaan, Hiburan.....	93
4.2.4 Uji Coba Sistem.....	95
4.2.5 Implementasi Metode <i>Forward Chaining Dan Pruning</i> .....	96
4.3 Pembahasan.....	98
<b>BAB V: PENUTUP.....</b>	<b>100</b>
5.1 Kesimpulan.....	100
5.2 Saran.....	100

DAFTAR PUSTAKA  
LAMPIRAN

## DAFTAR TABEL

Tabel 2.1 Decision Tabel .....	31
Tabel 4.1 Hasil Uji Coba Desain Interface, Kemudahan Penggunaan, Hiburan .....	94
Tabel 4.2 Hasil Uji Coba Sistem.....	95



## DAFTAR GAMBAR

Gambar 2.1 Notasi Papan Catur.....	29
Gambar 2.2 Posisi Awal Permainan .....	33
Gambar 2.3 Posisi Awal Benteng .....	34
Gambar 2.4 Langkah Benteng .....	34
Gambar 2.5 Posisi Awal Menteri.....	35
Gambar 2.6 Langkah Menteri .....	35
Gambar 2.7 Posisi Awal Ratu.....	36
Gambar 2.8 Langkah Ratu .....	36
Gambar 2.9 Posisi Awal Kuda.....	37
Gambar 2.10 Langkah Kuda.....	37
Gambar 2.11 Posisi Awal Pion .....	38
Gambar 2.12 Langkah Pion .....	38
Gambar 2.13 Posisi Awal Raja .....	39
Gambar 2.14 Langkah Raja .....	39
Gambar 2.15 Langkah Yang Mungkin Bagi Raja Ketika Dipojokan Oleh Bidak Lawan .....	40
Gambar 2.16 Posisi Sebelum Dan Setelah Melakukan Rokade .....	41
Gambar 2.17 Putih Tidak Dapat Melakukan Rokade .....	42
Gambar 2.18 Langkah Dua Kotak Diikuti Langkah En-Passant .....	43
Gambar 2.19 Pion Sebelum Dan Setelah Promosi.....	43
Gambar 2.20 Gambar Skak.....	44
Gambar 2.21 Skakmate .....	45
Gambar 2.22 Saat Hitam Melangkah, Permainan Jadi Buntu .....	46
Gambar 2.23 Eclipse.....	47
Gambar 2.24 Prinsip Algoritma Alpha Beta Pruning .....	49
Gambar 3. Metodologi Penelitian .....	51
Gambar 3.1 Rancangan Layar Form Menu.....	59
Gambar 3.2 Rancangan Layar Form Mulai Permainan .....	59
Gambar 3.3 Rancangan Layar Form Cara Bermain.....	60
Gambar 3.4 Rancangan Layar Form About.....	60
Gambar 3.5 Block Diagram Permainan Catur .....	61
Gambar 3.6 Flowchart Permainan .....	63
Gambar 3.7 Flowchart Akses Menu .....	64
Gambar 3.8 Flowchart Mulai Permainan Baru .....	65
Gambar 3.9 Flowchart Option .....	66
Gambar 3.10 Flowchart Computer Melangkah .....	68
Gambar 3.11 Flowchart Manusia Melangkah.....	73
Gambar 3.12 Flowchart Status Permainan.....	74
Gambar 4.1 Tampilan Layar Form Awal Aplikasi .....	79
Gambar 4.2 Tampilan Layar Form Menu .....	81
Gambar 4.3 Tampilan Layar Form Cara Bermain .....	83
Gambar 4.4 Tampilan Layar Form Tentang .....	84
Gambar 4.5 Tampilan Layar Form Mulai Permainan.....	86
Gambar 4.6 Tampilan Layar Pemain Putih Memindahkan Bidak .....	88
Gambar 4.7 Tampilan Layar Pemain Hitam Memindahkan Bidak .....	88
Gambar 4.8 Tampilan Layar Akhir Permainan.....	89
Gambar 4.9 Grafik Dari Pertanyaan A.1 .....	94
Gambar 4.10 Grafik Dari Pertanyaan A.2 .....	94

Gambar 4.11 Grafik Dari Pertanyaan A.3 ..... 94



## ABSTRAK

Teguh Santoso. 2013. 08650051. **Rancang Bangun Game Catur Untuk Smartphone Dengan Metode *Forward Chaining Pruning***. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Pembimbing : (I) Muhammad Faisal, M.T (II) A'la Syauqi, M.Kom

---

Kata Kunci : Catur, *Forward Chaining*, *Pruning*

Permainan catur adalah permainan strategi yang melibatkan dua pemain, sebidang papan catur dan beberapa bidak. Dalam permainan catur, kedua pemain secara bergantian melakukan langkah dengan memindahkan bidak dari satu kotak ke kotak lainnya. Sebelum melangkah, pemain akan menentukan langkah terbaik yang bisa diambil. Dengan berkembangnya teknologi komputer, maka komputer dapat bermain catur sebagaimana manusia. Komputer seolah-olah dapat berpikir sebagaimana manusia dalam menentukan langkah yang akan diambil.

Dalam penentuan langkah yang akan diambil, komputer menggunakan perhitungan dengan formulasi-formulasi (algoritma) yang sudah dikenal secara umum. Algoritma seperti *forward chaining* dan *pruning* merupakan cabang dari kecerdasan buatan. *Forward Chaining* (runut maju) menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan dengan menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil. Sedangkan *Pruning* (pemangkasan *tree*) merupakan bagian dari proses pembentukan *decision tree*. Saat pembentukan *decision tree*, beberapa node merupakan *outlier* maupun hasil dari *noise* data. Penerapan *pruning* pada *decision tree* dapat mengurangi *outlier* maupun *noise* data pada *decision tree* awal sehingga dapat meningkatkan akurasi pada klasifikasi data. Dalam hal ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Dengan menggunakan algoritma *forward chaining* dan *pruning* diperoleh hasil pencarian langkah terbaik dengan waktu berpikir pemain komputer pada level pemula antara 4 – 5 detik, level menengah antara 8 – 10 detik, lanjut antara 29 – 30 detik, dan master antara 59 – 60 detik. Sehingga semakin tinggi level yang digunakan dalam permainan, maka semakin banyak pula waktu permainan komputer yang dibutuhkan untuk mencari langkah terbaik.

## ABSTRACT

Teguh Santoso. 2013. 08650051. **Build Designing Chess Game Application For Smartphones with Forward Chaining Pruning Method.** Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

Pembimbing : (I) Muhammad Faisal, M.T (II) A'la Syauqi, M.Kom

---

Keyword: Chess, *Forward Chaining*, *Pruning*

Chess is a strategy game that involves two players, chess piece and a few pawns. In the game of chess, both players move alternately perform by moving the pawn from one box to another. Before moving, the player will determine the best steps that can be taken. With the development of computer technology, the computer can play chess as humans. Computer as though you may think as humans in determining the steps to be taken.

In determining the measures to be taken, using computer calculations formulations (algorithm) that is known in general. Algorithms such as forward chaining and pruning a branch of artificial intelligence. Forward Chaining (trace forward) using a set of condition-action rules. In this method, the data is used to determine which rules are to be executed, then the rule is implemented by adding data to the working memory. The process is repeated until you find a result. While Pruning (tree trimming) is part of the process of formation of decision tree. When the formation of decision tree, some nodes are outliers and the results of the noise data. Application of the decision tree pruning can reduce outliers and noise data on the decision tree early so as to improve the accuracy of the data classification. In this case begins tracking the input information, and then try to describe conclusions. By using the forward chaining and pruning algorithm obtained the best results with a time step thinking computer players at beginner level between 4-5 seconds, intermediate level between 8-10 seconds, advanced level between 29-30 seconds, and the master level between 59-60 seconds. So the higher level used in the game, more time game to computer is needed to find the best move.

# BAB I

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Perkembangan game di dunia semakin pesat, tidak terkecuali di Indonesia. Game saat ini sudah menjadi alternatif hiburan bagi tua, muda, pria maupun wanita. Industri dan bisnis pengembangan game juga sudah menjadi suatu hal yang menjanjikan, terbukti dengan banyaknya perusahaan pengembang game di Amerika, Eropa dan Asia. Meski industri *game* berkembang sangat pesat saat ini (penjualannya mencapai USD 9,3 Milyar di Amerika pada tahun 2011, berdasarkan data dari salah satu badan *retail tracking* terbesar di Amerika Serikat, NPD), tetapi sampai sekarang masih saja banyak menimbulkan pro dan kontra soal baik – buruknya *game digital* bagi *user* nya. Mulai dari di kalangan politikus, orang tua, sampai *gamer* itu sendiri tidak berhenti memperlmasalahkan dampak dari permainan *game* seperti yang tertulis dalam Surat Al Qhososh ayat 77.

وَأَبْتَعْ فِي مَاءِ آتَنَكَ اللَّهُ الْدَّارَ الْآخِرَةَ ۖ وَلَا تَنْسَ نَصِيبَكَ مِنَ الدُّنْيَا ۗ  
وَأَحْسِنَ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ ۖ وَلَا تَبْغِ الْفَسَادَ فِي الْأَرْضِ ۗ إِنَّ اللَّهَ لَا يُحِبُّ  
الْمُفْسِدِينَ

Artinya : “Dan carilah pada apa yang telah dianugerahkan Allah kepadamu (kebahagiaan) negeri akhirat, dan janganlah kamu melupakan bahagianmu dari



(kenikmatan) duniawi dan berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik, kepadamu, dan janganlah kamu berbuat kerusakan di (muka) bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan. (Q.S. Al Qoshosh : 77).

Dalam Al Qur'an Allah SWT juga berfirman:

يُؤْتِي الْحِكْمَةَ مَنْ يَشَاءُ ۚ وَمَنْ يُؤْتَ الْحِكْمَةَ فَقَدْ أُوتِيَ خَيْرًا كَثِيرًا ۗ وَمَا

يَذَّكَّرُ إِلَّا أُولُو الْأَلْبَابِ ﴿١٦٦﴾

Artinya : “Allah menganugerahkan Al-hikmah (Kefahaman yang dalam tentang Al Qur'an dan As Sunnah kepada siapa yang dikehendakinya, dan barang siapa yang dianugrahi hikmah, ia benar – benar telah dianugrahi karunia yang banyak, dan hanya orang – orang yang berakal lah yang dapat mengambil pelajaran (dari firman Allah)” (Qs. Al Baqarah, 2:269).

Catur adalah permainan mental yang dimainkan oleh dua orang. Pecatur adalah orang yang memainkan catur, baik dalam pertandingan satu lawan satu maupun satu melawan banyak orang (dalam keadaan informal). Sebelum bertanding, pecatur memilih biji catur yang akan ia mainkan. Terdapat dua warna yang membedakan bidak atau biji catur, yaitu hitam dan putih. Pemegang buah putih memulai langkah pertama, yang selanjutnya diikuti oleh pemegang buah hitam secara bergantian sampai permainan selesai (Harroffie, 2011). Permainan ini dikenal di seluruh dunia. Sebagai permainan dunia, catur diminati oleh banyak

orang, bahkan permainan ini dipertandingkan pada turnamen-turnamen tingkat dunia.

Seiring perkembangan perangkat lunak dan kemajuan dalam bidang kecerdasan buatan, maka dikembangkanlah aplikasi catur dengan kemampuan komputer sebagai pemain. Dalam permainan melawan komputer, komputer diharapkan dapat berpikir dalam menentukan langkah yang akan diambil. Catur dengan nyata dapat merangsang perhatian dan konsentrasi, penilaian dan pengendalian, imajinasi dan antisipasi memori, keinginan untuk menang, kesabaran dan pengendalian diri, semangat dalam membuat keputusan dan keberanian, logika matematika, berpikir analitik dan sintetik, kreativitas, kecerdasan, studi metode organisasi, dan bahasa asing. Catur juga dapat meningkatkan kecerdasan karena permainan ini melatih otak para pemainnya.

Tujuan dari penelitian ini adalah untuk merancang, membuat, mengaplikasikan serta mengembangkan aplikasi game berupa *board game* berjudul *Catur* dibuat ke dalam *game computer* agar permainan ini diminati oleh anak – anak dan dimainkan dalam bentuk *game computer*.

## 1.2 RUMUSAN MASALAH

Perumusan masalah yang ada dalam penelitian ini yaitu Bagaimana merancang dan membuat aplikasi *board game Catur* dengan mengimplementasikan metode *forward chaining dan pruning*.

## 1.3 BATASAN MASALAH

Batasan masalah pada penelitian ini adalah:

1. *Game* Catur ini merupakan permainan yang dapat merangsang perhatian dan konsentrasi, penilaian dan pengendalian, imajinasi dan antisipasi memori, keinginan untuk menang, kesabaran dan pengendalian diri, semangat dalam membuat keputusan dan keberanian, logika matematika, berpikir analitik dan sintetik, kreativitas, kecerdasan, studi metode organisasi, dan bahasa asing.
2. *Game* ditujukan untuk dimainkan oleh semua orang, dan diklasifikasikan berdasarkan level permainan (pemula, menengah, lanjut, master).
3. *Game* dirancang dan dibuat untuk *smartphone* berbasis sistem operasi *android gingerbread*.

#### 1.4 TUJUAN

Tujuan dari penelitian ini adalah merancang dan membuat aplikasi *board game Catur* agar bisa digunakan dan diaplikasikan dengan baik.

#### 1.5 MANFAAT

Manfaat yang dapat diambil dari penelitian ini adalah membuat game untuk permainan *Catur*. Agar bisa dimainkan dalam *smartphone* yang berbasis sistem operasi *Android Gingerbread*.

#### 1.6 SISTEMATIKA PENULISAN

Dalam penulisan skripsi ini, secara keseluruhan terdiri dari lima bab yang masing – masing bab disusun dalam sistematika sebagai berikut:

BAB I : PENDAHULUAN

Bab ini memuat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

## BAB II : TINJAUAN PUSTAKA

Pada bab ini menjelaskan tentang teori – teori yang berhubungan dengan pengerjaan skripsi ini, di antaranya: (1) Pengertian Game, (2) Sistem Operasi Android, (3) Metode Forward Chaining, (4) Game Catur, (5) Eclipse, ADT Plugin, Dan SDK (6) Pruning

## BAB III : DESAIN DAN PERANCANGAN SISTEM

Bab ini menjelaskan tentang metode penelitian yang dilakukan dalam pembuatan Rancang Bangun Aplikasi *Board Game Catur* Dengan Menggunakan *Metode Forward Chaining dan Pruning*. Yang membahas tentang pembuatan desain dan perancangan sistem aplikasi.

## BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan hasil yang dicapai dari perancangan sistem dan implementasi program, yaitu dengan melakukan pengujian sehingga dapat ditarik kesimpulan.

## BAB V : PENUTUP

Pada bab terakhir berisi kesimpulan berdasarkan hasil yang telah dicapai. Serta berisi saran sebagai bahan pertimbangan bagi pihak – pihak yang akan melakukan pengembangan terhadap program aplikasi ini kedepannya.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Game

Game berasal dari kata bahasa Inggris yang memiliki arti dasar Permainan. Permainan dalam hal ini merujuk pada pengertian “kelincahan intelektual” (intellectual playability). Game juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target yang ingin dicapai pemainnya. Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran sejauh mana game itu menarik untuk dimainkan secara maksimal. Pada awalnya, game identik dengan permainan anak-anak. Kita selalu berpikir game merupakan suatu kegiatan yang dilakukan oleh anak-anak yang dapat menyenangkan hati mereka.

Dengan kata lain, segala bentuk kegiatan yang memerlukan pemikiran, kelincahan intelektual dan pencapaian terhadap target tertentu dapat dikatakan sebagai game. Tetapi yang akan dibahas pada kesempatan ini adalah game yang terdapat di komputer, baik off line maupun online. Saat ini perkembangan games di komputer sangat cepat. Para pengelola industri game berlomba-lomba untuk menciptakan game yang lebih nyata dan menarik untuk para pemainnya. Hal inilah yang membuat perkembangan games di komputer sangat cepat. Sehingga games bukan hanya sekedar permainan untuk mengisi waktu luang atau sekedar hobi. Melainkan sebuah cara untuk meningkatkan kreatifitas dan tingkat intelektual para penggunanya.

Jadi, bermain game adalah suatu proses “fine tuning” (atau penyamaan frekuensi) dari logika berpikir anak-anak kita dengan logika berpikir aplikasi

komputer yang canggih tadi. Pada saat bersamaan, game juga secara nyata mempertajam daya analisis para penggunanya untuk mengolah informasi dan mengambil keputusan cepat yang jitu (Retno, 2011). Namun, tentu saja kenyataan juga harus kita masukkan kedalam perhitungan. Kenyataan itu diantaranya adalah kecanduan para pemain / penggunanya yang akut terhadap permainan komputer semacam ini. Mereka bisa lupa segala-galanya akan tugas mereka yang lain termasuk tugas menuntut ilmu.

### 2.1.1 Pengertian Game

Game adalah kegiatan antara dua lebih independen (pemain) membuat keputusan untuk mencapai tujuan mereka dalam beberapa konteks yang membatasi (*rules*). Lebih definisi konvensional menjelaskan bahwa *game* merupakan suatu konteks dengan aturan antar lawan mencoba untuk memenangkan tujuan (Abt, 1970).

Definisi lain menjelaskan, *game* adalah program perangkat lunak dimana satu atau lebih pemain berusaha untuk membuat keputusan melalui kontrol terhadap obyek dan sumber daya guna memenuhi satu tujuan tertentu (Prayudi, 2008).

Menurut Agustinus Nilwan dalam bukunya “Pemrograman Animasi dan Game Profesional” terbitan Elex Media Komputindo, game di artikan sebagai suatu aktivitas terstruktur atau juga digunakan sebagai alat pembelajaran. Sebuah game bisa dikarakteristikan dari apa pemain lakukan misalnya :

#### A. Peralatan

Misal : bola, kartu, papan, atau sebuah Komputer.

## **B. Peraturan**

Peraturan digunakan untuk menentukan giliran pemain, hak dan keharusan masing-masing pemain, dan tujuan permainan.

## **C. Skill, Strategi dan Keberuntungan**

Game dengan dengan skill, contohnya dengan kekuatan fisik, misal gulat, menembak dan kekuatan mental seperti catur.

## **D. Single Player Game (pemain satu orang) dan Double Player (lebih dari satu pemain)**

Jika pemain tunggal, pemain harus bermain dengan keahlian, berpacu dengan waktu dan keberuntungan sedangkan pemain double, pemain diharuskan untuk menggunakan suatu strategi dan kekompakan sesama pemain, untuk mencapai tujuan tertentu atau sebaliknya pemain harus berlomba dengan pemain lainnya untuk mencapai sesuatu tujuan.

### **2.1.2 Sejarah Perkembangan Game**

Pada tahun 1947 adalah tahun pertama di mana game didesain untuk dimainkan dengan layar CRT (cathode ray tube). Game sederhana ini dirancang oleh Thomas T. Goldsmith Jr. dan Estle Ray Mann. Aplikasi ini dipatenkan pada tanggal 14 Desember 1948. Sistem yang dibuatnya terdiri dari 8 vacum tubes dan mensimulasikan peluru ditembakkan pada target, ide ini berasal dari display radar pada Perang Dunia II. Beberapa knop disediakan untuk mengatur kurva dan kecepatan titik yang mewakili peluru. Karena pada waktu itu grafik belum bisa dibuat, target penembakan digambarkan pada sebuah lapisan yang kemudian

ditempelkan pada CRT. Hal ini adalah sistem pertama yang secara spesifik didesain untuk game pada layar CRT.

Banyak yang menyebutkan bahwa penemu video game adalah William Higinbotham. Pada tahun 1952, A.S. Douglas membuat OXO, game grafis noughts and crosses atau nol dan silang, di University of Cambridge untuk mendemonstrasikan tesisnya tentang interaksi komputer dan manusia. Permainan ini bekerja pada komputer besar yang menggunakan CRT display. Perangkat game portable genggam yang pertama dibuat adalah Tic Tac Toe di tahun 1972 oleh Waco Company. Tahun 1958 menciptakan game Tennis for Two pada osiloskop. Game ini menampilkan lapangan tenis sederhana dipandang dari samping. Bola seakan dipengaruhi oleh gravitasi dan harus melewati net/jaring.

Dengan dua kontrol yang masing-masing dilengkapi knop untuk mengarahkan bola dan sebuah tombol untuk memukul bola sampai melewati net. Tahun 1972 dirilis perangkat video game pertama untuk pasar rumahan, Magnavox Odyssey, dihubungkan dengan televisi. Meski tidak sukses besar, perusahaan lain dengan produk yang sama harus membayar lisensi. tetapi, kesuksesan menjemput sejak Atari meluncurkan Pong sebuah video game ping-pong pada 29 November 1972. Berangkat dari sini, video dan komputer game menjadi populer dan hobi baru di saat PC baru saja mulai dikenal dan dipakai secara luas. Mistery House, rancangan ibu rumah tangga, Roberta Williams dipercaya sebagai game petualangan pertama dengan grafis pada Apple II. Meski interface untuk input perintah masih berupa teks, ilustrasi grafik hitam putih sebuah rumah bergaya victoria merupakan gebrakan baru di masa itu. Game ini



begitu populer dan mendorong Roberta Williams mendirikan Sierra On-Line bersama suaminya dan terus memproduksi game khususnya petualangan.

Pada awal tahun 1980-an ditandai oleh Nintendo, Kesuksesan LCD genggam ini menciptakan banyak pengikut untuk membuat yang sama dengan mengadopsi game-game populer, awal tahun 1980-an juga ditandai dengan hadirnya media penyimpanan CD-ROM yang dalam waktu singkat menjadi populer. Era game 3 dimensi (3D) dengan perspektif orang pertama dan multiplayer game mulai muncul di era ini. Suara dan musik semakin berkembang di pertengahan 1980-an seiring dengan hadirnya produk sound card. Jadi, memang terasa bahwa pasar game komputer semakin berkembang sejalan dengan perkembangan teknologi pendukungnya.

### 2.1.3 Klasifikasi Game

Board Games (Permainan Papan) Game pada katagori ini membutuhkan suatu papan yang terbagi dalam sektor-sektor tertentu (dengan garis-garis) dan didalamnya terdapat sejumlah alat main yang dapat digerakkan, yang termasuk game dalam katagori ini adalah catur. Dalam permainan ini dua pemain akan saling berhadapan dan saling mengadu strategi untuk mempertahankan daerah sendiri dan mengalahkan lawan.

Card Games (Permainan Kartu) Game ini akan memanfaatkan simbol dari 52 kartu yang terbagi dalam dua faktor : suit (4 nilai) dan rank (13 nilai). Sejumlah ketentuan dibuat untuk mengatur bagaimana cara-cara untuk membuat kombinasi tersebut, yang termasuk game dalam katagori ini adalah permainan kartu bridge/truf.

Athletic Games (Permainan Atletik) Permainan games jenis ini lebih cenderung pada penggunaan fisik. Peraturan game mewajibkan pemain untuk melakukan aksi tertentu. Kekuatan badan, kecepatan, ketepatan dan kerjasama menjadi bagian utama dari game atletik.

Children Games (Permainan Anak) Contoh dari game anak – anak seperti berlari, sembunyi, melempar dan menangkap. Pada umumnya game ini menekankan pada aktifitas kelompok sebagai latihan untuk berkehidupan sosial.

Computer Games (Permainan Komputer) Game ini dimainkan lewat bantuan alat komputer. Terdapat 5 alat yang dapat dikategorikan sebagai komputer, yaitu :

1. Expensive dedicated machine, mesin dengan cara memasukkan koin untuk memainkannya.
2. Inexpensive dedicated machine, disebut juga dengan hand held machine. Alat game watch termasuk dalam katagori ini.
3. Multiprogram home, mesin seperti Atari, Nintendo termasuk dalam kelompok komputer ini.
4. Personal computer
5. Mainframe computer

Computer game berbeda dengan jenis game yang lain karena tidak ada pergerakan secara fisik atau interaksi langsung dengan object kecuali lewat perantaraan komputer. Software yang dibuat harus dapat menangkap reaksi yang cepat dari interaksi yang dihasilkan dengan pemain dan juga harus bersifat real time. Kompleksitas game bergantung dari kemampuan menjelaskan aturan atau cara kerja game dan lingkungan game dalam program yang dibuat.

Berdasarkan Jenis “*Platform*” Atau Alat Yang Di Gunakan :

1. ***Arcade games***, yaitu permainan 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 membuat pemainnya lebih merasa “masuk” dan “menikmati”, seperti pistol, kursi khusus, sensor gerakan, sensor injakkan dan stir mobil (beserta transmisinya tentunya).
2. ***PC Games*** , yaitu video game yang dimainkan menggunakan Personal Computer.
3. ***Console games***, yaitu video games yang dimainkan menggunakan console tertentu, seperti Playstation 2, Playstation 3, XBOX 360, dan Nintendo Wii.
4. ***Handheld games***, 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.

Berdasarkan “Genre” Permainannya :

1. **Aksi – Shooting**, (tembak-tembakkan, atau hajar-hajaran bisa juga tusuk – tusukan, tergantung cerita dan tokoh di dalamnya), video game jenis ini sangat memerlukan kecepatan refleks, koordinasi mata-tangan, juga timing, inti dari game jenis ini adalah tembak menembak. Termasuk didalam-nya :
  - a. *First person shooting (FPS)* seperti Counter Strike dan Call of Duty
  - b. *Drive n’ shoot*, menggunakan unsur simulasi kendaraan tetapi tetap dengan tujuan utama menembak dan menghancurkan lawan, contoh : Spy Hunter, Rock and Roll Racing, Road Rash.

- c. *Shoot em' up*, seperti Raiden, 1942, dan gradius.
- d. *Beat 'em up* (tonjok hajar) seperti Double Dragon dan Final Fight, lalu ***hack and slash*** (tusuk tebas) seperti Shinobi dan Legend of Kage.
- e. *Light gun shooting*, yang menggunakan alat yang umumnya berbentuk seperti senjata, seperti Virtua Cop dan Time Crisis.

**2. Fighting ( pertarungan )** Ada yang mengelompokan video game fighting di bagian Aksi, jenis ini memang memerlukan kecepatan refleks dan koordinasi mata-tangan, tetapi inti dari game ini adalah penguasaan jurus (hafal caranya dan lancar mengeksekusinya), pengenalan karakter dan timing sangatlah penting, *combo*-pun menjadi esensial untuk mengalahkan lawan secepat mungkin. Dan berbeda seperti game aksi pada umumnya yang hanya melawan Artificial Intellegence atau istilah umumnya *melawan komputer* saja, pemain jenis *fighting game* ini baru teruji kemampuan sesungguhnya dengan melawan pemain lainnya. Seri Street Fighter, Tekken, Mortal Kombat, Soul Calibur dan King of Fighter adalah contohnya.

**3. Aksi – Petualangan.** Memasuki gua bawah tanah, melompati bebatuan di antara lahar, bergelayutan dari pohon satu ke pohon lain, bergulat dengan ular sambil mencari kunci untuk membuka pintu kuil legendaris, atau sekedar mencari telepon umum untuk mendapatkan misi berikutnya, itulah beberapa dari banyak hal yang karakter pemain harus lakukan dan lalui dalam video game jenis ini. Game jenis ini sudah berkembang jauh hingga menjadi genre campuran *action beat-em up* juga, dan sekarang, di tahun 2000 an, jenis ini cenderung untuk memiliki visual 3D dan sudut pandang orang ke-tiga. Tomb Rider, Grand Theft Auto dan Prince of Persia termasuk didalamnya.

**4. Petualangan.** Bedanya dengan jenis video game aksi-petualangan, refleks dan kelihaiian pemain dalam bergerak, berlari, melompat hingga memecut atau menembak tidak diperlukan di sini. Video Game murni petualangan lebih menekankan pada jalan cerita dan kemampuan berpikir pemain dalam menganalisa tempat secara visual, memecahkan teka-teki maupun menyimpulkan rangkaian peristiwa dan percakapan karakter hingga penggunaan benda-benda tepat pada tempat yang tepat. Termasuk didalamnya:

- a. *Petualangan dengan teks* atau sistem tunjuk dan klik, contoh: Kings Quest, Space Quest, Heroes Quest, Monkey Island, Sam and Max,
- b. *Novel atau film interaktif*, seperti game “*dating*” yang banyak beredar di Jepang, Dragons Lair dan Night Trap.

**5. Simulasi, Konstruksi dan manajemen.** Video Game jenis ini seringkali menggambarkan dunia di dalamnya sedekat mungkin dengan dunia nyata dan memperhatikan dengan detil berbagai faktor. Dari mencari jodoh dan pekerjaan, membangun rumah, gedung hingga kota, mengatur pajak dan dana kota hingga keputusan memecat atau menambah karyawan. Dunia kehidupan rumah tangga sampai bisnis membangun konglomerasi, dari jualan limun pinggir jalan hingga membangun laboratorium cloning. Video Game jenis ini membuat pemain harus berpikir untuk mendirikan, membangun dan mengatasi masalah dengan menggunakan dana yang terbatas. Contoh: Sim City, The Sims, Tamagotchi.

**6. Role Playing.** Video game jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain ( biasanya menjadi semakin

hebat, semakin kuat, semakin berpengaruh, dll) dalam berbagai parameter yang biasanya ditentukan dengan naiknya **level**, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun mahluk peliharaan. Secara kebudayaan, pengembang game Jepang biasanya membuat Role Playing Game (**RPG**) ke arah cerita linear yang diarahkan seolah karakter kita adalah tokoh dalam cerita itu, seperti Final Fantasy, Dragon Quest dan Xenogears. Sedangkan pengembang game **RPG** Eropa, cenderung membuat karakter kita bebas memilih jalan cerita sendiri secara non-linear, seperti Ultima, Never Winter Nights, Baldur's Gate, Elder Scroll, dan Fallout.

**7. Strategi.** Kebalikan dari video game jenis action yang berjalan cepat dan perlu refleksi secepat kilat, video game jenis strategi, layaknya bermain catur, justru lebih memerlukan keahlian berpikir dan memutuskan setiap gerakan secara hati-hati dan terencana. Video game strategi biasanya memberikan pemain atas kendali tidak hanya satu orang tapi minimal sekelompok orang dengan berbagai jenis tipe kemampuan, sampai kendaraan, bahkan hingga pembangunan berbagai bangunan, pabrik dan pusat pelatihan tempur, tergantung dari tema ceritanya. Pemain game strategi melihat dari sudut pandang lebih luas dan lebih kedepan dengan waktu permainan yang biasanya lebih lama dan santai dibandingkan game action. Unsur-unsur permainannya biasanya berkisar sekitar, prioritas pembangunan, peletakan pasukan, mencari dan memanfaatkan sumberdaya (uang, besi, kayu, minyak, dll), hingga ke pembelian dan peng-upgrade-an pasukan atau teknologi. Game jenis ini terbagi atas:

a. *Real time Strategy*, game berjalan dalam waktu sebenarnya dan serentak antara semua pihak dan pemain harus memutuskan setiap langkah yang diambil saat itu

juga berbarengan mungkin saat itu pihak lawan juga sedang mengeksekusi strateginya. Contoh: Starcraft, Warcraft , dan Command and Conquer.

b. *Turn based Strategy* , game yang berjalan secara bergiliran, saat kita mengambil keputusan dan menggerakkan pasukan, saat itu pihak lawan menunggu, begitu pula sebaliknya, layaknya catur. contoh: Front Mission, Super robot wars, Final Fantasy tactics, Heroes of might and magic, Master of Orion. Sebenarnya ada yang memilah lagi menjadi jenis tactical dan strategi, namun penulis cenderung untuk menggabungkannya karena perbedaannya hanya ada di masalah skala dan ke-kompleks-an dalam manajemen sumber dayanya saja.

**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 kota 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. Tetris, Minesweeper, Bejeweled, Sokoban dan Bomberman.

**9. Simulasi kendaraan.** Video Game jenis ini memberikan pengalaman atau interaktifitas sedekat mungkin dengan kendaraan yang aslinya, meskipun terkadang kendaraan tersebut masih eksperimen atau bahkan fiktif, tapi ada penekanan khusus pada detil dan pengalaman realistik menggunakan kendaraan tersebut. Terbagi atas beberapa jenis:

a. **Perang.** Video game simulasi kendaraan yang sempat tenar di tahun 90-an ini mengajak pemain untuk menaiki kendaraan dan berperang melawan kendaraan lainnya. Dan kebanyakan diantaranya memiliki judul sama dengan nama

kendaraannya. Contoh : Apache 64, Comanche, Abrams, YF-23, F-16 fighting eagle. Tetapi game kehidupan bajak laut seperti 'Pirates!' pun dapat dikategorikan disini.

b. **Balapan.** Dari namanya sudah jelas, siapa sampai duluan di garis finish dialah pemenangnya! Terkadang malah pemain dapat memilih kendaraan, mendandani, upgrade mesin bahkan mengecatnya. Contoh: Top Gear, Test Drive, Sega Rally Championship, Daytona, Grand Turismo, Need For Speed, Mario Kart, ManXTT.

c. **Luar Angkasa.** Walau masih dapat dikategorikan simulasi kendaraan perang, tetapi segala unsur fiksi ilmiah dan banyaknya judul yang beredar membuat subgenre ini pantas dikategorikan diluar simulasi kendaraan perang. Jenis ini memungkinkan pemain untuk menjelajah luar angkasa, berperang dengan makhluk alien, mendarat di planet antah berantah atau sekedar ingin merasakan bagaimana menjadi kapten di film fiksi ilmiah kesayangan kamu. Contoh: Wing Commander, Freelancer , Star Wars X-Wing, Star Wars Tie Fighter, dll.

d. **Mecha.** Pendapat bahwa hampir tidak ada orang yang terekspos oleh film robot jepang saat kecilnya tidak memimpikan ingin mengendalikan robot, memang sulit dibantah. Dipopulerkan oleh serial Mechwarrior oleh Activision, subgenre Simulasi Mecha ini memungkinkan pemainnya untuk mengendalikan robot dan menggunakannya untuk menghancurkan gedung, helikopter dan tentu saja robot lainnya. Contoh: Mechwarrior, Gundam Last war Chronicles, dan Armored Core.

**10. Olahraga.** Singkat padat jelas, bermain sport di PC atau konsol anda. Biasanya permainannya diusahakan serealistik mungkin walau kadang ada yang menambah unsur fiksi seperti NBA JAM. Contohnya pun jelas, Seri Winning



Eleven, seri NBA, seri FIFA, John Madden NFL, Lakers vs Celtics, Tony hawk pro skater, dll.

Kategori – Kategori Lainnya :

**1. Multiplayer Online.** Game yang lagi trend di Indonesia bahkan dunia, menjadi salah satu titik balik mengapa dunia game dan internet di Indonesia dapat berkembang. Dan karena dimainkan online dan dengan sistem pembayaran menggunakan voucher, pembajakan sudah tidak menjadi masalah lagi. Game yang dapat dimainkan secara bersamaan oleh lebih dari 2 orang (bahkan dapat mencapai puluhan ribu orang dalam satu waktu) membuat pemain dapat bermain bersama dalam satu dunia virtual dari sekedar chatting hingga membunuh naga bersama teman yang entah bermain di mana. Umumnya permainan tipe ini dimainkan di PC dan bertema RPG, walau ada juga yang bertema music atau action. Contoh: Ragnarok online, O2jam, World of Warcraft, Ayo Dance, Lineage, Rose online

**2. Casual games.** Sesuai namanya, game yang casual itu tidak kompleks, mainnya rileks dan sangat mudah untuk dipelajari ( bahkan cenderung langsung bisa dimainkan ). Jenis ini biasanya memerlukan spesifikasi komputer yang standar pada jamannya dan ukurannya tidak lebih dari 100 MB karena biasanya dapat di download versi demo-nya di website resminya. Genre permainannya biasanya puzzle atau action sederhana dan umumnya dapat dimainkan hanya menggunakan mouse ( biasanya game lain menggunakan banyak tombol tergantung game-nya ). Contoh: Diner Dash, Sally Salon, Bejeweled, Zuma, Feeding Frenzy, Insaniquarium.

**3. Edugames.** Video Game jenis ini dibuat dengan tujuan spesifik sebagai alat pendidikan, entah untuk belajar mengenal warna untuk balita, mengenal huruf dan angka, matematika, sampai belajar bahasa asing. Developer yang membuatnya, harus memperhitungkan berbagai hal agar game ini benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan ketrampilan yang memainkannya. Target segmentasi pemain harus pula disesuaikan dengan tingkat kesulitan dan design visual ataupun animasinya. Contoh edugames : Bobi Bola, Dora the explorer, Petualangan Billy dan Tracy.

**4. Advergames.** Anda pasti menemukan game-game yang dapat dimainkan lalu dapat anda beritahukan/mengundang langsung ke teman-teman anda. jenis game yang biasanya mudah dimainkan ini mengusung dan menampilkan produk atau brand mereka baik secara gamblang maupun tersembunyi. Di era tumbuhnya media-media baru berteknologi tinggi sekarang ini, dunia periklanan memang sudah tidak lagi terbatas pada TV, koran, majalah, billboard dan radio, video *game* sekarang telah menjadi sarana beriklan atau membangun *brand-awareness* yang efektif. Baik melalui internet maupun di mainkan di event-event mereka, edugames terasa semakin dibutuhkan untuk menjaring calon konsumen bagi produk yang menggunakan advergames ini. Contoh produk di indonesia yang membuat advergames: A-Mild, Rexona teens, Axe.

## 2.2 Android

**Android** adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (Wikipedia, 2012).

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

### **2.2.1 Kerjasama dengan Android Inc.**

Pada Juli 2000, Google bekerjasama dengan Android Inc., perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri Android Inc. bekerja pada Google, di antaranya Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi Android Inc. hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa Google hendak memasuki pasar telepon seluler. Di perusahaan Google, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh

kernel Linux. Hal ini menunjukkan indikasi bahwa Google sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

Versi Android terbaru yaitu versi 4.2 Android juga sudah bergabung dengan beberapa smart mobile seperti Samsung, Sony Ericsson dan lainnya.

### 2.2.2 Produk awal

Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikasi telepon seluler (akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (*mobile*) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

### **Android versi 1.1**

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

### **Android versi 1.5 (Cupcake)**

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

### **Android versi 1.6 (Donut)**

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

### **Android versi 2.0/2.1 (Eclair)**

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

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

### **Android versi 2.2 (Froyo: Frozen Yoghurt)**

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

kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

### **Android versi 2.3 (Gingerbread)**

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

### **Android versi 3.0/3.1 (Honeycomb)**

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

### **Android versi 4.0 (ICS :Ice Cream Sandwich)**

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu

kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

### 2.2.3 Fitur

Fitur yang tersedia di Android adalah:

- Kerangka aplikasi: itu memungkinkan penggunaan dan penghapusan komponen yang tersedia.
- Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat mobile.
- Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.
- SQLite: untuk penyimpanan data.
- Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM, Bluetooth, EDGE, 3G, dan WiFi (hardware dependent)
- Kamera, Global Positioning System (GPS), kompas, dan *accelerometer* (tergantung hardware)

### 2.2.4 Android bagi komunitas sumber terbuka (*open source*)

Android memiliki berbagai keunggulan sebagai software yang memakai basis kode komputer yang bisa didistribusikan secara terbuka (*open source*) sehingga pengguna bisa membuat aplikasi baru di dalamnya. Android memiliki aplikasi *native* Google yang terintegrasi seperti *pushmail* Gmail, Google Maps, dan Google Calendar.

Para penggemar open source kemudian membangun komunitas yang membangun dan berbagi Android berbasis firmware dengan sejumlah



penyesuaian dan fitur-fitur tambahan, seperti FLAC lossless audio dan kemampuan untuk menyimpan download aplikasi pada microSD card. Mereka sering memperbaharui paket-paket firmware dan menggabungkan elemen-elemen fungsi Android yang belum resmi diluncurkan dalam suatu carrier-sanction firmware.

### 2.3 Forward Chaining

*Forward Chaining* (runut maju) berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil (Wilson, 1998).

Metode *Forward Chaining* disebut juga penalaran dari bawah ke atas karena penalaran dari fakta pada *level* bawah menuju konklusi pada *level* atas yang didasarkan pada fakta. Dalam hal ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan.

Perbedaan antara sistem pakar yang menggunakan metode *forward chaining* dengan sistem pakar yang sudah ada yaitu sistem pakar dengan menggunakan metode *forward chaining* pelacakan dimulai dari informasi masukan dan selanjutnya memperoleh sebuah keputusan dan proses *inference engine*. Akan tetapi pada proses sistem pakar ini dimulai dari pelacakan informasi masukan *user* dan mencoba mencari sebuah rule yang didasarkan pada informasi masukan user yang selanjutnya mendapatkan suatu konklusi sementara. Setelah memperoleh konklusi sementara proses *inference engine* masih berjalan yaitu dengan mencari

konklusi sementara tersebut yang bertindak sebagai premis di rule lainnya. Maka program akan melanjutkan pertanyaan selanjutnya sesuai dengan rule-rule yang memenuhi konklusi sementara sampai ditemukan sebuah konklusi akhir (Wilson, 1998).

Contoh permasalahan menggunakan *Forward Chaining*, diketahui kaidah tipe *IF..THEN..* berikut:

*IF A THEN B*

*IF B THEN C*

*IF C THEN D*

Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Untuk langkah-langkahnya sebagai berikut :

Catat fakta pertama yang telah diinputkan oleh *user*.

1. Cari *rule* yang bagian premisnya sesuai dengan fakta yang diinputkan.
2. Jika *rule* tidak ada maka cari fakta selanjutnya, kemudian kembali ke langkah no 2.
3. Jika *rule* ada maka akan ketemu konklusi sementara
4. Cari *rule* yang menggunakan konklusi sementara sebagai premis
5. Jika *rule* tidak ada maka konklusi tidak ditemukan.
6. Namun jika *rule* terpenuhi cari fakta selanjutnya yang sesuai dengan konklusi sementara.
7. Cari *rule* yang menggunakan konklusi sementara dan fakta selanjutnya sebagai premis.

8. Jika rule tidak terpenuhi maka konklusi tidak ditemukan, namun jika rule terpenuhi maka konklusi ditemukan.

Contoh implementasi *Forward Chaining* pada permainan Catur.

State awal dalam permainan Catur. Permainan ini dimainkan pada sebuah papan catur yang terdiri dari 64 kotak: delapan baris dan delapan kolom dengan warna terang dan gelap secara bergantian. Papan harus datar dengan kotak berwarna gelap pada sudut kiri bawah. Untuk memudahkan notasi setiap langkah, maka setiap kotak diberi sebuah notasi. Dari sisi pemain putih, baris diberi angka 1, 2, 3, 4, 5, 6, 7, 8. Baris terbawah diberi angka 1 dan kolom paling atas diberi angka 8. Setiap kolom diberi notasi, dari kiri ke kanan a, b, c, d, e, f, g, h. Setiap kotak mempunyai notasi yang merupakan kombinasi huruf kolom dan angka baris yang dimilikinya. Contoh : notasi kotak kiri bawah (dari sisi putih) adalah a1.



Gambar 2.1 : Notasi Papan Catur

Para pemain secara bergantian melangkah dimulai dengan langkah pemain putih (pemain yang bermain dengan bidak putih). Sebuah langkah

merupakan perpindahan sebuah bidak pemain ke kotak berbeda sesuai aturan langkah dari bidak tersebut. Ada satu pengecualian, dinamakan rokade (*castling*), dimana pemain memindahkan dua bidak secara bersamaan.

Pemain dapat mengambil sebuah bidak lawan dengan memindahkan sebuah bidaknya ke kotak yang berisi sebuah bidak lawan. Bidak lawan tersebut dikeluarkan dari papan, dan tidak dapat digunakan sampai akhir permainan (gerakan ini bersifat opsional).

### 2.3.1 Verifikasi Sistem Pakar

*Verifikasi* merupakan sekumpulan aktivitas yang memastikan suatu sistem apakah telah berlaku dalam kondisi yang diterapkan, verifikasi itu sendiri terdiri dari dua proses. Pertama, memeriksa pelaksanaan sistem. Kedua, memeriksa konsistensi dan kelengkapan basis pengetahuan. *Verifikasi* dijalankan ketika ada penambahan atau perubahan pada *rule*, karena *rule* tersebut sudah ada pada sistem. Sedangkan tujuan *verifikasi* adalah untuk memeriksa adanya kecocokan antara sistem dengan sistem yang terbebas dari error.

Berikut ini adalah beberapa metode pemeriksaan *rule* dalam suatu basis pengetahuan :

1. *Redundant Rules*, yaitu jika dua buah *rule* atau lebih mempunyai premis dan konklusi yang sama.
2. *Conflicting Rules*, yaitu terjadi jika dua buah *rule* atau lebih mempunyai premis yang sama, tetapi memiliki konklusi yang berlawanan.
3. *Circular Rules*, yaitu suatu keadaan dimana terjadinya proses perulangan dari suatu *rule*. Ini dikarenakan suatu premis dari salah satu *rule* merupakan konklusi dari *rule* yang lain atau kebalikannya.

### 2.3.2 Block Diagram

*Blok Diagram* merupakan susunan *rule-rule* yang terdapat didalam sebuah bidang ilmu. Dengan Membuat *blok diagram* didalam sistem pakar maka dapat diketahui urutan kerja sistem dalam mencari keputusan.

### 2.3.3 Dependency Diagram

*Dependency diagram* didalam sistem pakar bersifat untuk menunjukkan hubungan atau ketergantungan antara inputan pertanyaan, *rule-rule* dan rekomendasi yang dibuat oleh prototype sistem berbasis pengetahuan. Dari *blok Diagram* diatas apabila diteruskan menjadi *dependency diagram*.

### 2.3.4 Decision Table

*Decision table* dibuat untuk menunjukkan hubungan antar nilai-nilai pada hasil fase atau rekomendasi akhir knowledge Base System (KBS). Sebagai contoh pembuatan *decision table* dapat dilihat pada tabel dibawah ini.

*Step 1 : Plan*

*Condition : Member\_status (ok, not\_ok) = 2*

*Reason (new\_case, follow\_up\_case, information\_other) = 3*

*Problem (serious, non\_serious) = 2*

*Rows : 2 x 3 x 2*

*Step 2 : Completed Decision Table*

Tabel 2.1 *Decision Tabel*

Rule	Member_Status	Reason	Problem	Recommended Support
A1	Ok	New_case	Serious	Level_1
A2	Ok	New_case	Non_serious	Level_2
A3	Ok	Follow_up_case	Serious	Level_1

A4	Ok	Follow_up_case	Non_serious	Level_3
A5	Ok	Information_other	Serious	Information_other
A6	Ok	Information_other	Non_serious	Information_other
A7	Not_Ok	New_case	Serious	Non_member
A8	Not_Ok	New_case	Non_serious	Non_member
A9	Not_Ok	Follow_up_case	Serious	Non_member
A10	Not_Ok	Follow_up_case	Non_serious	Non_member
A11	Not_Ok	Information_other	Serious	Non_member
A12	Not_Ok	Information_other	Non_serious	Non_member

### 2.3.5 Rule Base

Pada dasarnya basis aturan (*rule base*) terdiri dari 2 pokok, yaitu bagian premis atau kondisi bagian conclusion atau kesimpulan. Struktur *rule* secara logika menghubungkan satu atau lebih kondisi (premis) pada bagian *if* (yang akan menguji kebenaran dari serangkaian data) dengan satu atau lebih kesimpulan (*conclusion*) yang terdapat pada bagian *then*.

## 2.4 Catur

Catur adalah permainan yang dimainkan oleh dua pemain. Salah satu pemain memegang bidak putih, sedangkan pemain yang lain memegang bidak hitam. Pada awal permainan, setiap pemain memiliki enam belas bidak : satu raja, satu ratu, dua benteng, dua menteri, dua kuda dan delapan pion.

### a. Aturan Main (*Rules*)

Para pemain secara bergantian melangkah dimulai dengan langkah pemain putih (pemain yang bermain dengan bidak putih). Sebuah langkah merupakan perpindahan sebuah bidak pemain ke kotak berbeda sesuai aturan langkah dari bidak tersebut. Ada satu pengecualian, dinamakan rokade (*castling*), dimana pemain memindahkan dua bidak secara bersamaan.

Pemain dapat mengambil sebuah bidak lawan dengan memindahkan sebuah bidaknya ke kotak yang berisi sebuah bidak lawan. Bidak lawan tersebut dikeluarkan dari papan, dan tidak dapat digunakan sampai akhir permainan (gerakan ini bersifat opsional).

Pada awal permainan, posisi tiap bidak terlihat pada gambar berikut :



Gambar 2.2 : Posisi Awal Permainan

Pada baris kedua, terdapat delapan pion putih, pada baris ketujuh, terdapat delapan pion hitam. Pada baris pertama, dari kiri ke kanan, terdapat : benteng, kuda, menteri, ratu, raja, menteri, kuda dan benteng.

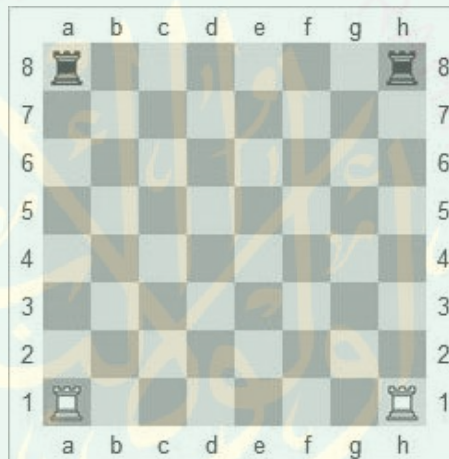
Posisi awal Ratu berada pada kotak sewarna dengan bidak yang dimainkan, dengan kotak gelap pada sudut kanan tiap pemain.

## 1) Langkah Bidak

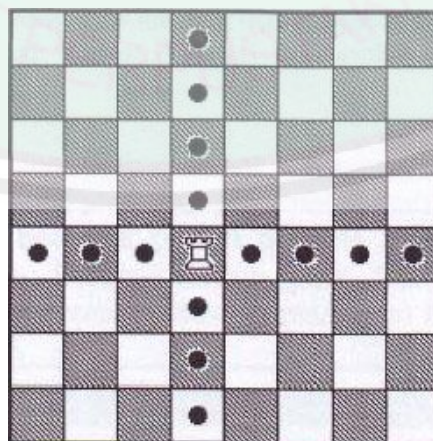
### a) Benteng (*Rook*)

Benteng bergerak lurus secara horizontal atau vertikal.

Benteng tidak dapat bergerak melewati bidak lain (kawan atau lawan), artinya: semua kotak antara kotak awal dan kotak tujuan harus kosong dan kotak tujuan tidak terisi bidak sewarna.



Gambar 2.3 : Posisi Awal Benteng

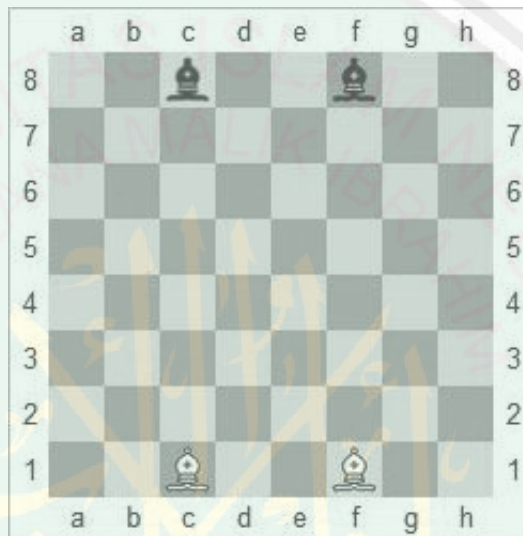


Gambar 2.4 : Langkah Benteng

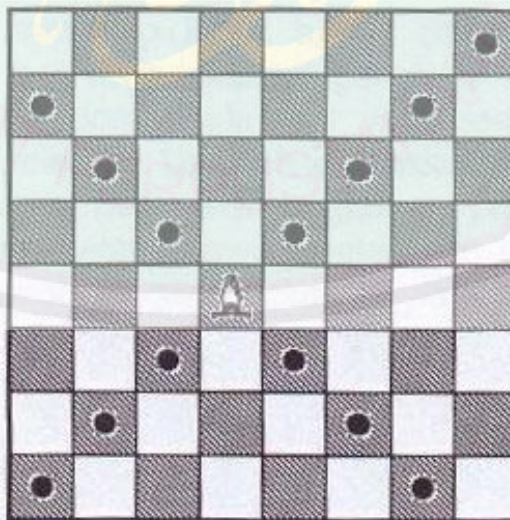


**b) Menteri (Bishop)**

Menteri bergerak lurus secara diagonal. Menteri tidak dapat bergerak melewati bidak lain (kawan atau lawan), artinya: semua kotak antara kotak awal dan kotak tujuan harus kosong dan kotak tujuan tidak terisi bidak sewarna.



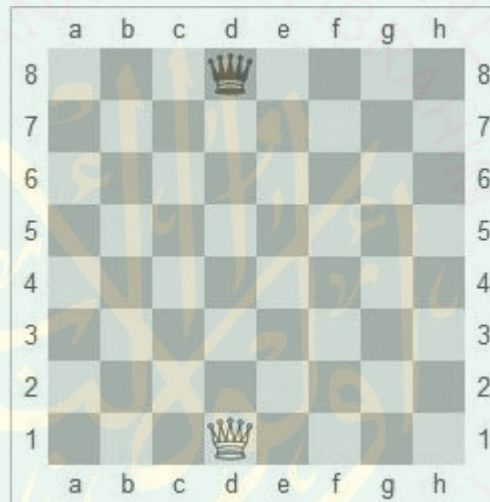
Gambar 2.5 : Posisi Awal Menteri



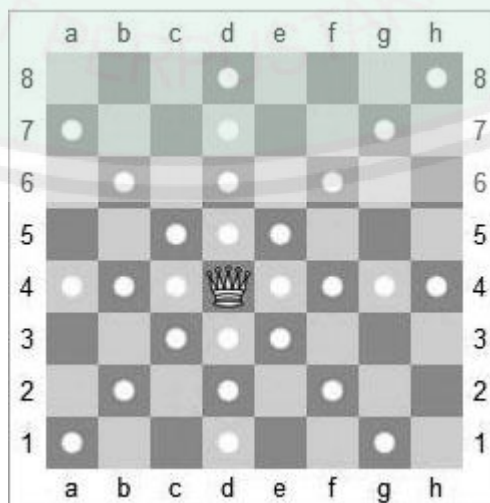
Gambar 2.6 : Langkah Menteri

**c) Ratu (Queen)**

Ratu mempunyai kemampuan bergerak seperti Benteng dan Menteri, sehingga Ratu dapat melangkah garis lurus secara horizontal, vertikal atau diagonal. Ratu tidak dapat bergerak melewati bidak lain (kawan atau lawan), artinya: semua kotak antara kotak awal dan kotak tujuan harus kosong dan kotak tujuan tidak terisi bidak sewarna.



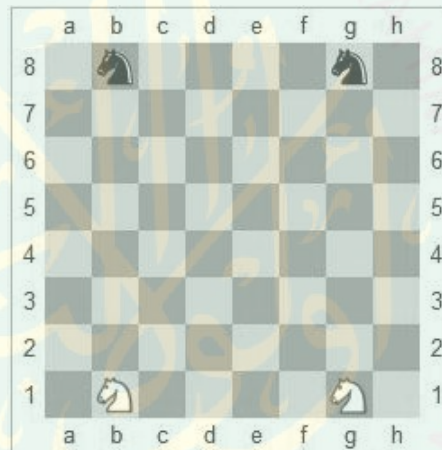
Gambar 2.7 : Posisi Awal Ratu



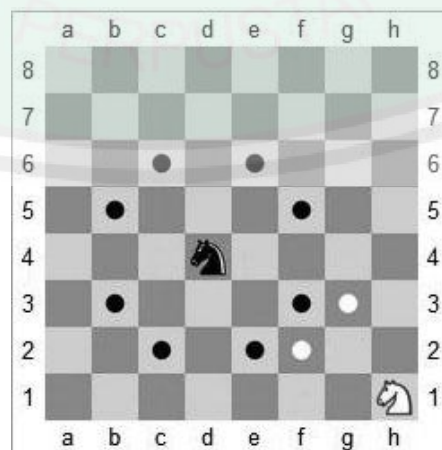
Gambar 2.8 : Langkah Ratu

#### d) Kuda (*Knight*)

Gerakan kuda terdiri dari satu langkah horizontal atau vertikal, dan satu langkah diagonal ke arah luar. Kuda bergerak melompat. Kuda dapat melewati bidak (kawan atau lawan). Sebagai contoh, pemain putih dapat memulai permainan dengan menggerakkan kuda dari kotak b1 ke kotak c3. Bidak yang dilewati tidak berubah statusnya. Seperti biasa, kuda dapat mengambil bidak lawan dengan melangkah ke kotak tempat bidak lawan tersebut.



Gambar 2.9 : Posisi Awal Kuda



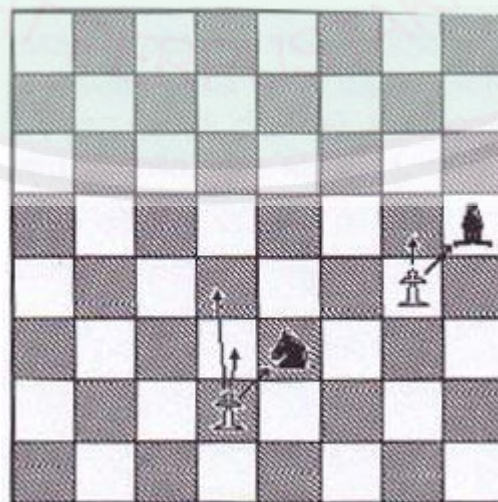
Gambar 2.10 : Langkah Kuda

### e) Pion (*Pawn*)

Gerakan pion berbeda saat mengambil bidak lawan atau melangkah maju. Saat melangkah maju, pion melangkah satu kotak ke depan, jika pion belum pernah bergerak (masih di baris kedua), pion dapat melangkah 2 kotak ke depan. Sebagai contoh, pion dapat melangkah dari kotak d2 ke kotak d4. Saat mengambil bidak lawan, pion melangkah satu kotak ke depan secara diagonal.



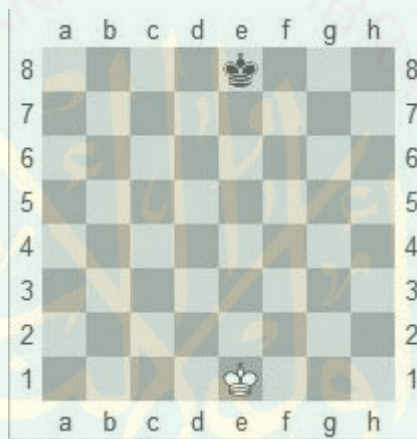
Gambar 2.11 : Posisi Awal Pion



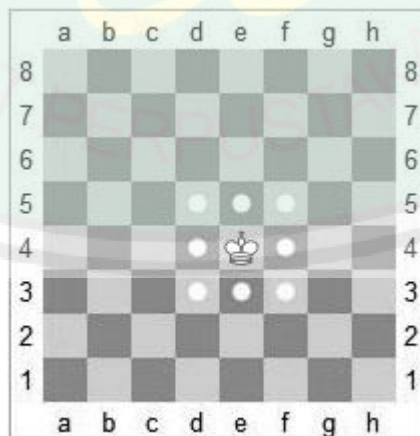
Gambar 2.12 : Langkah Pion

### f) Raja (*King*)

Raja dapat melangkah satu kotak ke segala arah, horizontal, vertikal atau diagonal. Raja memiliki satu jenis langkah yang khusus yang dilakukan oleh raja dan benteng secara bersamaan. Langkah ini disebut rokade (*castling*). Raja merupakan bidak yang paling penting dalam permainan, dan langkah yang dibuat seorang pemain tidak boleh membuat posisi rajanya dalam keadaan skak (*check*).



Gambar 2.13 : Posisi Awal Raja



Gambar 2.14 : Langkah Raja



Gambar 2.15: Langkah Yang Mungkin Bagi Raja Ketika Dipojokkan Oleh Bidak

Lawan

## 2) Rokade (*Castling*)

Terdapat aturan khusus dimana raja dan benteng dapat bergerak bersamaan (satu langkah) dengan gerakan (*castling*).

Syarat-syarat yang harus dipenuhi dalam melakukan gerakan rokade adalah sebagai berikut :

- (1) Raja belum pernah melakukan gerakan.
- (2) Benteng yang akan melakukan rokade belum pernah melakukan gerakan.
- (3) Raja tidak sedang dalam kondisi skak.
- (4) Raja tidak melalui kotak yang dapat diserang oleh bidak lawan pada saat melakukan langkah rokade, contoh pada saat melakukan langkah rokade, tidak terdapat bidak lawan yang dapat melangkah ke kotak yang dilalui oleh raja.
- (5) Raja tidak dapat melangkah ke kotak yang dapat diserang oleh bidak lawan pada saat melakukan langkah rokade, contoh pemain

tidak dapat melakukan rokade sehingga posisi raja dalam keadaan skak.

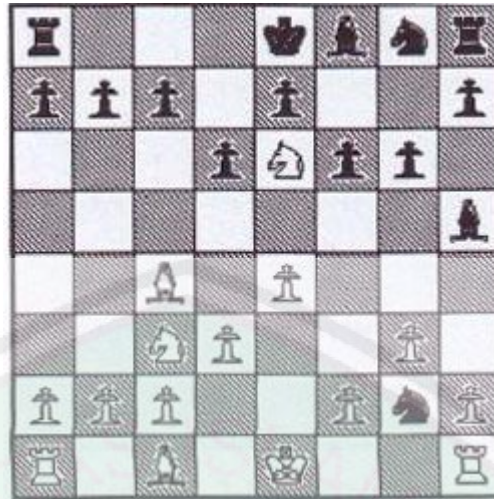
- (6) Semua kotak antara benteng dan raja sebelum langkah rokade dilakukan dalam keadaan kosong.
- (7) Raja dan benteng harus berada pada baris yang sama.

Saat rokade, raja melangkah dua kotak ke arah benteng, dan benteng melangkah melewati raja ke kotak di samping raja. Contoh : raja hitam pada kotak e8 dan benteng hitam pada kotak a8, raja hitam melangkah ke kotak c8 dan benteng hitam melangkah ke kotak d8 (rokade panjang), raja putih pada kotak e1 dan benteng putih pada kotak h1, raja putih melangkah ke kotak g1 dan benteng putih melangkah ke kotak f1 (rokade pendek).



Gambar 2.16 : Posisi Sebelum dan Setelah Melakukan Rokade

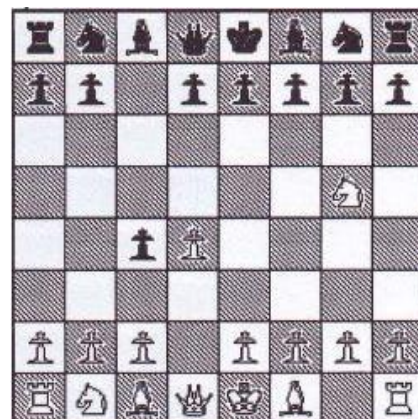
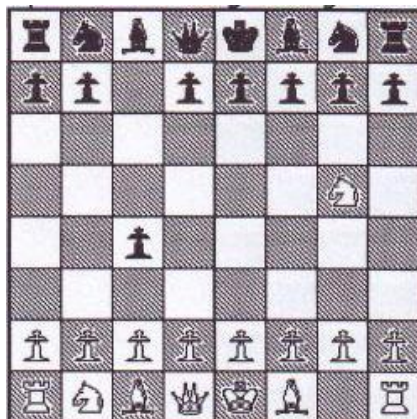
Putih Rokade Pendek, Hitam Rokade Panjang Raja Putih Dalam Keadaan Skak dan Raja Hitam Tidak Dapat Melakukan Rokade Pendek.



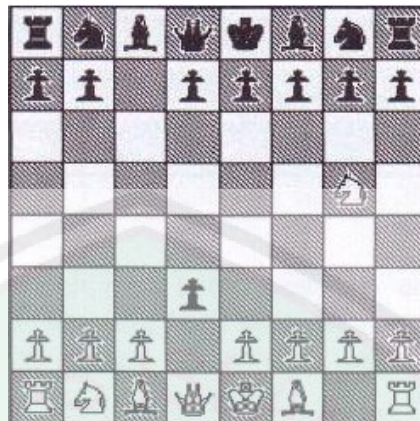
Gambar 2.17 : Putih Tidak Dapat Melakukan Rokade :

3) *En-Passant*

Pion memiliki satu aturan khusus yang disebut langkah *en-passant*. Saat pion melangkah dua kotak dari baris kedua ke baris keempat, dan terdapat pion lawan di samping kotak tujuan (baris keempat), maka pion lawan selanjutnya dapat melangkah diagonal ke kotak yang dilewati oleh pion tersebut (baris ketiga). Pada saat yang sama, pion yang melangkah dua kotak tersebut diambil. Langkah *en-passant* ini harus dilakukan segera, jika pemain yang dapat melakukan langkah *en-passant* tidak melakukannya pada langkah pertama setelah lawan melangkah dua kotak, maka pion tersebut tidak dapat diambil dengan langkah *en-passant*.



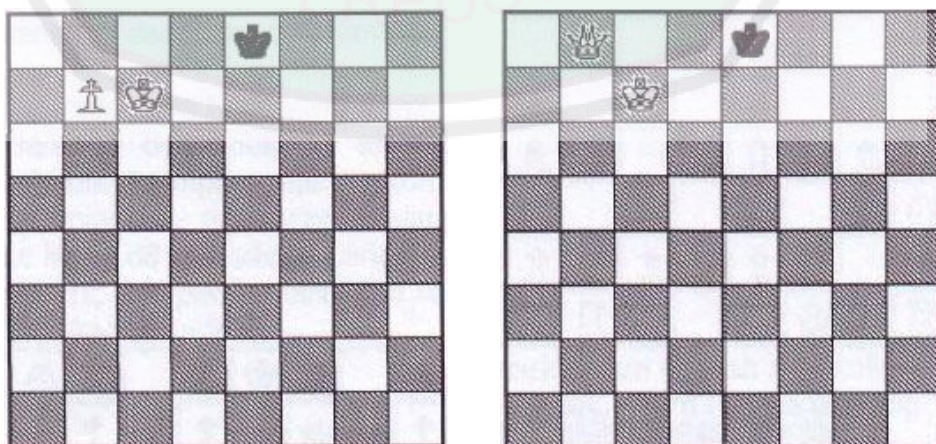




Gambar 2.18 : Langkah Dua Kotak Diikuti Langkah *En-Passant*

#### 4) Promosi (*Promotion*)

Pion yang mencapai baris terakhir mendapatkan *promosi*. Saat pemain malangkahkan pion ke baris terakhir (baris pertama lawan), maka pion tersebut berubah menjadi menteri, benteng, kuda, atau ratu (dengan warna yang sama). Biasanya pemain mempromosikan pion menjadi ratu, tetapi bisa juga dengan bidak jenis lain (selain pion dan raja). Promosi pion tidak harus dari bidak yang sudah diambil, sehingga mungkin saja seorang pemain pada saat tertentu memiliki dua ratu.



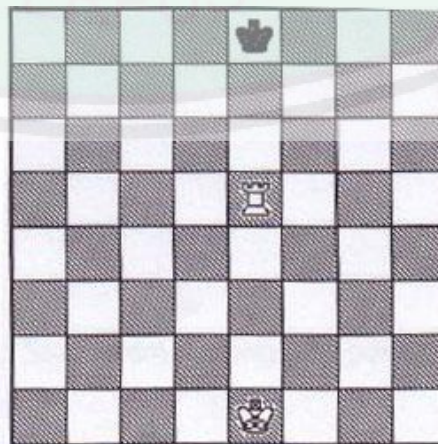
Gambar 2.19 : Gambar Pion Sebelum dan Setelah Promosi

## 5) Skak, Skakmat dan Buntu

### a) Skak (*Check*)

Posisi dimana raja dari seorang pemain dapat diambil oleh bidak pemain lawan disebut posisi skak. Sebagai contoh, pemain putih menggerakkan bentengnya pada posisi dimana benteng tersebut menyerang raja, jika pemain hitam tidak melakukan usaha untuk mengamankan raja, maka benteng tersebut dapat mengambil raja hitam pada langkah berikutnya. Dikatakan bahwa benteng putih melakukan skak. Dianjurkan untuk mengucapkan Skak saat seorang pemain melakukan skak ke lawannya.

Pemain tidak boleh melakukan langkah yang menyebabkan rajanya dalam kondisi skak. Jika pemain secara tidak sengaja melakukan hal ini, pemain tersebut harus membatalkan langkahnya dan melakukan langkah lain (aturan ini mengikuti aturan pemain harus melangkah dengan bidak yang sudah dipegang).



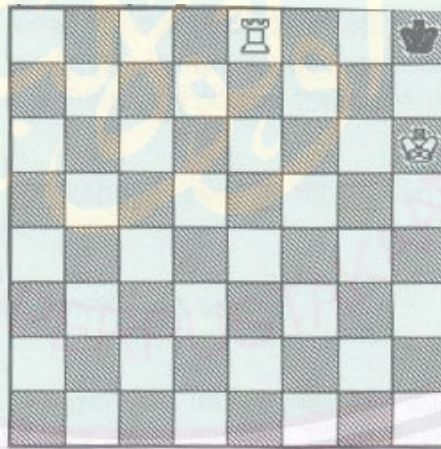
Gambar 2.20 : Gambar Skak

### b) Skakmat (*Checkmate*)

Saat pemain dalam kondisi skak, dan pemain tersebut tidak dapat melakukan langkah sehingga raja tidak dalam kondisi skak, maka pemain tersebut dalam kondisi skakmat. Pemain dalam kondisi ini mengalami kekalahan dan pemain yang melakukan skakmat menjadi pemenang.

Ada tiga cara berbeda untuk menghilangkan kondisi skak :

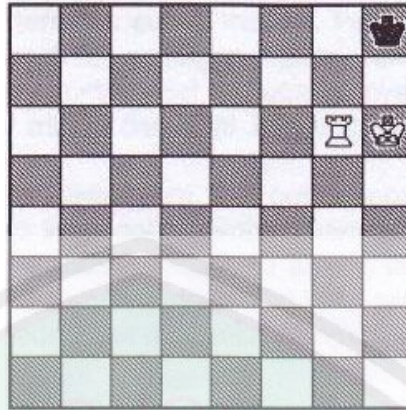
1. Pindahkan raja ke kotak yang tidak diserang lawan.
2. Ambil bidak yang menyebabkan skak.
3. Untuk kasus skak oleh benteng, ratu atau menteri :  
gerakkan sebuah bidak ke kotak antara bidak yang melakukan skak dan raja.



Gambar 2.21 : Skakmat

### c) Buntu (*Stalemate*)

Saat pemain tidak melangkah dan tidak dalam keadaan skak, maka pemain tersebut dalam kondisi buntu. Dalam kasus buntu, permainan menjadi seri.



Gambar 2.22: Saat Hitam Melangkah, Permainan Jadibuntu

## 2.5 Eclipse, ADT Plugin, Dan SDK

Sebelumnya, terlebih dahulu Anda harus memiliki 3 buah komponen dibawah ini untuk menjalankan emulator android.

1. Eclipse
2. ADT Plugin
3. SDK

**Eclipse** adalah sebuah IDE (integrated Development Environment) untuk pengembangan perangkat lunak dan dapat dijalankan di semua platform. Pada awalnya eclipse dikembangkan oleh IBM sendiri untuk menggantikan perangkat lunak IBM visual age for java yang telah menghabiskan dana pengembangan sebanyak US \$40 juta. Semenjak itu konsorsium eclipse foundation mengambil alih untuk pengembangan eclipse lebih lanjut dan pengaturan organisasinya.

Eclipse merupakan komunitas open source yang bertujuan menghasilkan platform pemrograman terbuka. Eclipse terdiri dari framework yang dapat dikembangkan lebih lanjut, peralatan bantu untuk membuat dan manage software sejak awal hingga diluncurkan. Platform Eclipse didukung oleh

ekosistem besar yang terdiri dari vendor teknologi, start-up inovatif, universitas, riset institusi serta individu.

Kelebihan eclipse sendiri antara lain adalah :

- Multi-role

Selain sebagai IDE untuk pengembangan aplikasi, eclipse juga digunakan untuk aktifitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, UML dan lain-lain.

- Multi-language

Eclipse dikembangkan dengan bahasa pemrograman Java, mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya seperti C, C++, PHP, Perl, Python dan lain-lain.

- Multi-platform

Target sistem operasi Eclipse adalah Microsoft Windows, GNU/Linux, Solaris, HP-UX, AIX dan Mac OS X



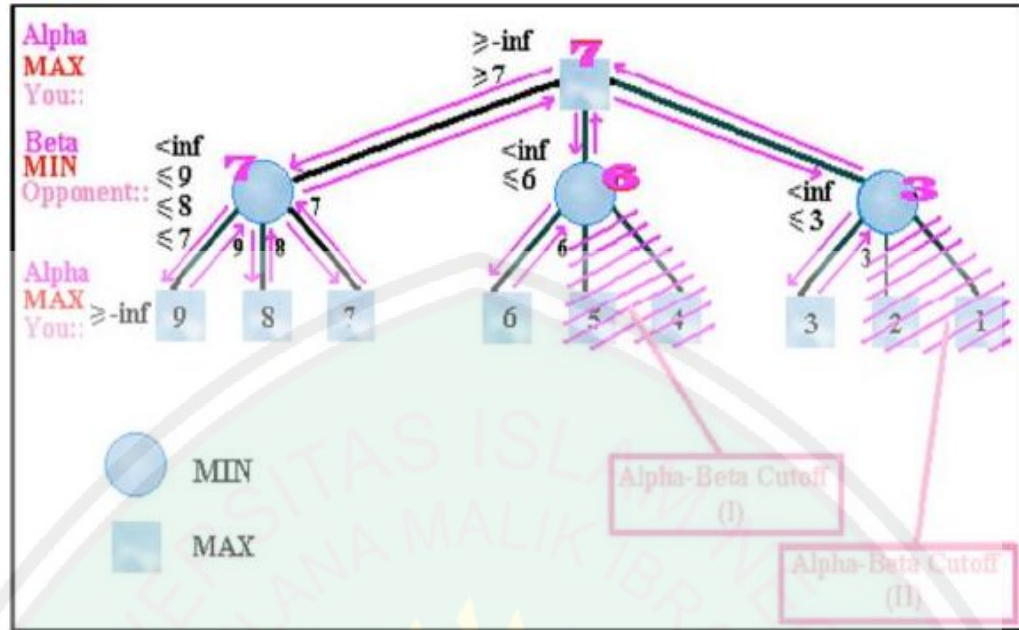
Gambar 2.23 Eclipse

## 2.6 Pruning

*Pruning* (pemangkasan *tree*) merupakan bagian dari proses pembentukan *decision tree*. Saat pembentukan *decision tree*, beberapa node merupakan *outlier* maupun hasil dari *noise data*. Penerapan *pruning* pada *decision tree* dapat mengurangi *outlier* maupun *noise data* pada *decision tree* awal sehingga dapat meningkatkan akurasi pada klasifikasi data (Han & Kamber 2001 dalam Budi 2010). Prinsip *pruning* terbagi menjadi dua: *Pre-pruning* dan *post pruning*. *Pre-pruning* merupakan proses pemangkasan saat *tree* belum terbentuk secara sempurna. Berbeda prinsip dengan *pre-pruning*, *post pruning* bekerja setelah *tree* terbentuk dengan sempurna.

### 2.6.1 Alpha-Beta Pruning

Alpha-Beta Pruning merupakan modifikasi dari algoritma Minimax, yang akan mengurangi jumlah node yang dievaluasi oleh pohon pencarian. Pencarian untuk node berikutnya akan dipikirkan terlebih dahulu. Algoritma ini akan berhenti mengevaluasi langkah ketika terdapat paling tidak satu kemungkinan yang ditemukan dan membuktikan bahwa langkah tersebut lebih buruk jika dibandingkan dengan langkah yang diperiksa sebelumnya. Sehingga, langkah berikutnya tidak perlu dievaluasi lebih jauh. Dengan algoritma ini hasil optimasi dari suatu algoritma tidak akan berubah. Gambar 2-8 merupakan pohon dengan algoritma Alpha-Beta Pruning.



Gambar 2.24 Prinsip Algoritma Alpha Beta Pruning

Diperlihatkan, pada pohon tersebut, terdapat pemotongan pencarian dengan menggunakan algoritma Alpha-Beta Pruning. Pada algoritma ini, terdapat dua nilai yang diatur, yaitu Alpha dan Beta, yang merepresentasikan nilai minimum dari max yang diyakini dan nilai maksimum dari min yang diyakini. Nilai awal alpha adalah tak hingga negatif dan nilai awal beta adalah tak hingga positif. Sebagai hasil dari proses rekursif, area pencarian akan semakin kecil. Ketika beta menjadi lebih kecil dari alpha, akan berarti posisi saat itu tidak dapat menjadi hasil terbaik permainan untuk kedua pemain dan pencarian tidak perlu dilakukan lebih jauh.

Pada Game Catur, metode *Alpha-beta Pruning* akan diterapkan pada pohon n-ary yang telah terbentuk. Pada setiap cabang akan memiliki nilai yang bertujuan memaksimalkan pencarian dan meminimalkan waktu pencarian pada pohon n-ary. Sedangkan fungsi evaluasi sendiri adalah inialisasi pertama dalam metode ini

sebelum melakukan permainan. Berikut nilai fungsi evaluasi dari metode *Alpha-beta Pruning*.

- 0 → MIN (player) menang

- 1 → MAX (komputer) menang

Berikut merupakan aturan untuk *Alpha-beta Pruning*:

- Pemangkasan *Alpha* : Pencarian dapat dihentikan untuk simpul turunan selanjutnya jika setiap simpul MIN memiliki nilai *beta* kurang dari atau sama dengan nilai *alpha* apapun dari simpul MAX sebelumnya pada induk yang sama.

- Pemangkasan *Beta* : Pencarian dapat dihentikan untuk simpul turunan selanjutnya jika setiap simpul MAX memiliki nilai *alpha* lebih dari atau sama dengan nilai *beta* apapun dari simpul MIN sebelumnya pada induk yang sama.



## BAB III

### DESAIN DAN PERANCANGAN SISTEM

#### 3.1 Metode Penelitian

Dalam melakukan penelitian ini peneliti membagi menjadi beberapa tahap pengerjaan yang digunakan sebagai acuan desain dan perancangan sistem *board game Catur*. Berikut adalah tahap-tahap dalam melakukan penelitian:

1. Pengumpulan Data

Studi literatur, dengan membaca dan mencari sumber bacaan, berupa buku, paper, *e-book*, jurnal, tutorial, media internet dan referensi lain yang mendukung penelitian dengan tujuan memperoleh teori dan gambaran dari permasalahan yang akan dibahas.

2. Desain dan Perancangan Game

Tahapan ini merupakan tahapan desain dan perancangan game dengan mengacu pada data yang telah diperoleh dan dianalisa pada tahapan sebelumnya, meliputi skenario, *gameplay*, *rule play* dan lain-lain.

3. Implementasi

Dalam proses ini dilakukan pembuatan *game* berdasarkan desain dan perancangan, meliputi implementasi metode penelitian yaitu berbasis sistem operasi *android* dengan menggunakan metode *forward chaining*. Pada pembuatan game ini difokuskan pada desain *game* dan *rule play* yang ada dalam permainan *Catur*.

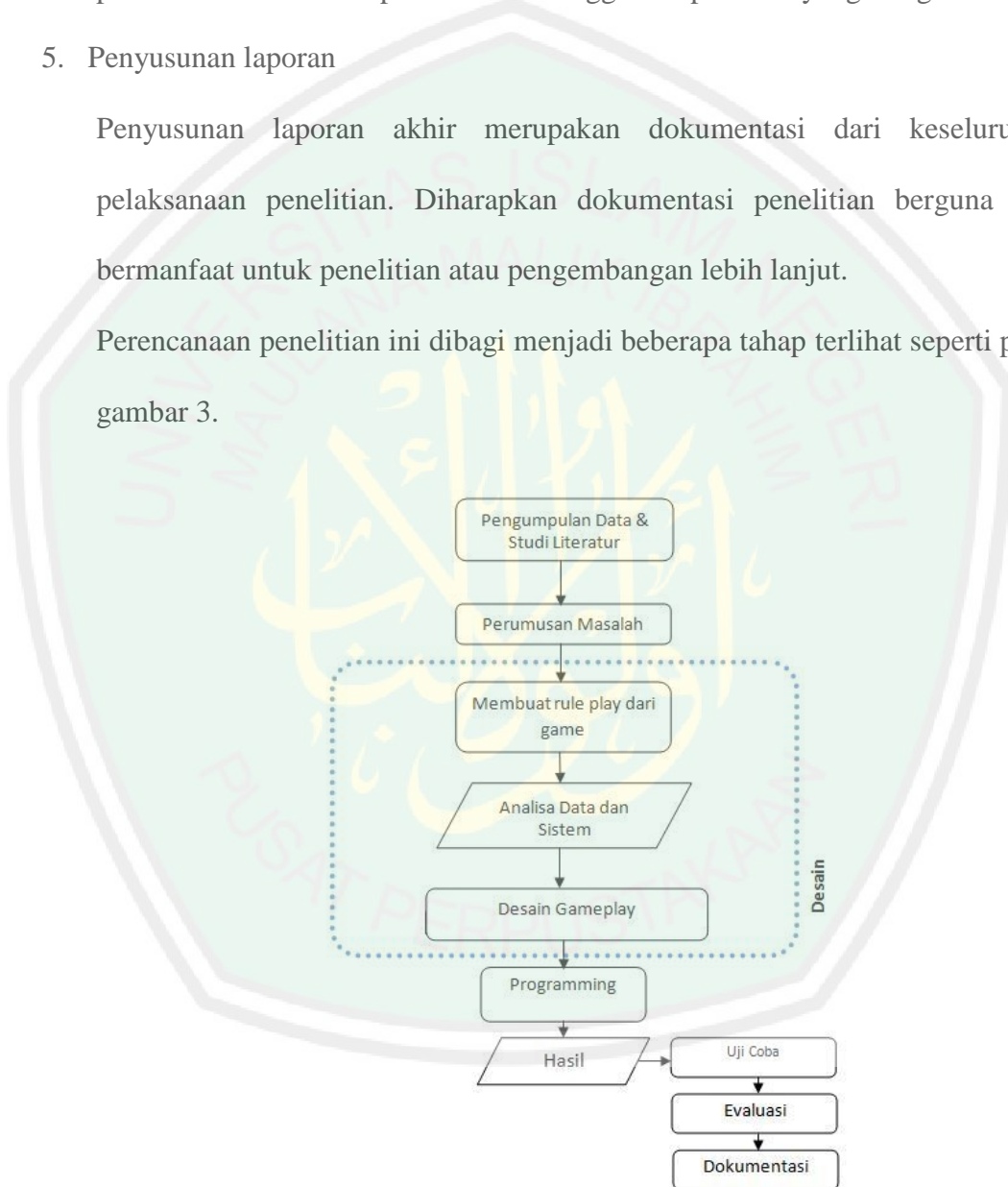
#### 4. Uji Coba dan Evaluasi

Uji coba dan evaluasi dilakukan terhadap tahapan aplikasi terhadap game yang sudah dibuat. Langkah ini dilakukan untuk perbaikan jika ada permasalahan dalam implementasi sehingga didapat hasil yang diinginkan.

#### 5. Penyusunan laporan

Penyusunan laporan akhir merupakan dokumentasi dari keseluruhan pelaksanaan penelitian. Diharapkan dokumentasi penelitian berguna dan bermanfaat untuk penelitian atau pengembangan lebih lanjut.

Perencanaan penelitian ini dibagi menjadi beberapa tahap terlihat seperti pada gambar 3.



Gambar 3. Metodologi Penelitian

### Tahap Studi Literatur

- Pengumpulan data

- Pengumpulan software yang diperlukan, diantaranya: Android SDK, pastikan di komputer anda sudah terinstall JDK versi 5 atau 6 dan Eclipse versi 3.4 atau 3.5
- Menginstall ADT untuk Eclipse
- Menentukan spesifikasi program yang akan dibuat

#### **Tahap Analisis Kebutuhan**

- Menganalisa hardware dan software yang digunakan
- Menganalisa kebutuhan user
- Menganalisa waktu perencanaan

#### **Tahap Pengumpulan Data**

- Mengumpulkan data-data terkait
- Menggunakan metode observasi, wawancara dan studi pustaka untuk mendapatkan data-data terkait.

#### **Tahap Analisa Data dan Sistem**

- Mengidentifikasi fungsi sistem
- Identifikasi dan Desain Sistem
- Menganalisa *Data Flow Diagram* (DFD)

#### **Tahap Desain**

- Pembuatan desain sistem
- Penyempurnaan rancangan dan desain

#### **Tahap Membangun Aplikasi**

- Pembuatan aplikasi *Catur* dengan menggunakan Eclipse.

#### **Tahap Programming**

- Memulai pembuatan aplikasi dengan *Eclipse*

- Menganalisa kesalahan dari program
- Menyempurnakan program
- Mencatat semua yang berkaitan dengan program

#### **Tahap Ujicoba**

- Menguji cobakan kepada pengguna
- Melakukan wawancara pada pengguna
- Mencatat semua yang berkaitan dengan program

### **3.2 Analisis Kebutuhan**

Dalam melakukan perancangan dan pembuatan aplikasi ini maka perlu melakukan pengumpulan data-data awal yang perlu di tampilkan dalam aplikasi yang akan dibuat, dan dari data-data itu pun kemudian diolah menjadi informasi. Disamping itu perlu menyiapkan perangkat lain yang diperlukan, diantaranya: hardware meliputi PC/laptop dan melakukan instalasi software meliputi Android SDK, pastikan di komputer sudah terinstall JDK versi 5 atau 6 keatas, dan Eclipse versi 3.4 atau 3.5 keatas dan menginstall ADT untuk Eclipse.

Kebutuhan-kebutuhan tersebut dapat dijabarkan menjadi dua, yaitu:

#### **a. Identifikasi Kebutuhan Fungsional**

Mengidentifikasi kebutuhan fungsional bagi aplikasi dengan cara mengidentifikasi data dan informasi. Diantara kebutuhan fungsional tersebut adalah:

- Fasilitas memulai permainan baru.
- Fasilitas konfigurasi permainan.
- Fasilitas menampilkan histori langkah.

- Fasilitas menyerah dalam permainan.

b. Identifikasi Kebutuhan Non Fungsional

Dalam hal ini juga memerlukan kebutuhan yang tak terduga misalnya, kebutuhan non fungsional.

1. Aplikasi yang didesain lebih menarik hati user
2. Aplikasi yang mudah digunakan dan dipergunakan user maupun oleh administrator.

Dan ada juga kebutuhan non fungsional lainnya misalnya, hardware, software, dll yang akan dijelaskan lebih rinci dibawah ini:

a. Perangkat Keras

Berikut ini adalah perangkat keras yang digunakan selama tahap pengimplementasian sistem keaman ini, diantaranya:

- 1) Desktop PC.
- 2) Processor Intel Dual Core T2080 @ 2.4 GHz.
- 3) DDRII 2048 MB, Hitachi 2.5" *harddisk drive* 160 GB.
- 4) Layar LCD : 1280x800.

b. Perangkat Lunak

Perangkat Lunak yang digunakan selama tahap implementasi sistem ini, diantaranya:

- 1) Microsoft Windows Vista sebagai sistem operasi.
- 2) Eclipse Indigo Version: 3.7.1 sebagai editor yang digunakan untuk menulis bahasa program JAVA.
- 3) JDK 1.7.0\_02, JRE 7

### 3.3 Pengumpulan Data Dan Analisa Masalah

Jika seseorang pertama kali bermain catur, maka dapat dipastikan bahwa orang tersebut bermain dengan kemampuan yang sangat terbatas. Saat seorang pemain mulai mengerti aturan dasar untuk tiap bidak, maka dia dapat bermain catur. Untuk meningkatkan kemampuan, pemain biasanya membaca pola-pola permainan. Strategi dan taktik dikembangkan bertujuan membantu pemain dalam setiap permainan.

Bagi manusia, permainan catur mengandung konsep pemikiran abstrak tingkat tinggi. Tidak demikian dengan komputer. Komputer bermain catur tidak berpikir, tetapi melakukan perhitungan dengan beberapa formulasi yang menghasilkan langkah terbaik yang dapat diambil. Dalam bermain, komputer harus memiliki kemampuan dalam melakukan langkah dari tiap jenis bidak, menentukan langkah terbaik, dan menentukan hasil akhir permainan. Saat ini terdapat beberapa algoritma pencarian yang umum digunakan dalam menentukan langkah terbaik. Algoritma-algoritma menggunakan metode pencarian *depth-first* yang merupakan sub bagian dari kecerdasan buatan.

Dalam skripsi ini, metode yang digunakan adalah metode *forward chaining* dan *pruning* untuk menentukan langkah terbaik berdasarkan bidak lawan. Alasan penggunaan algoritma ini karena metode *forward chaining* dan *pruning* mampu menganalisis segala kemungkinan posisi permainan untuk menghasilkan keputusan yang terbaik dan karena metode *forward chaining* dan *pruning* ini bekerja secara rekursif dengan mencari langkah yang akan membuat lawan mengalami kerugian minimum. Semua strategi lawan akan dihitung dengan algoritma yang sama dan seterusnya. Ini berarti, pada langkah pertama komputer

akan menganalisis seluruh langkah permainan. Dan untuk setiap langkahnya, komputer akan memilih langkah yang paling membuat lawan mendapatkan keuntungan minimum, dan yang paling membuat komputer itu sendiri mendapatkan keuntungan maksimum.

Dengan algoritma ini tingkat kedalaman (*depth*) pencarian dapat ditentukan. Kemampuan komputer dalam berpikir untuk beberapa langkah ke depan sangat menentukan tingkat kepandaian komputer dalam menentukan langkah terbaik yang akan diambil. Semakin dalam tingkat pencarian maka semakin pandai komputer dalam menentukan langkah terbaik.

Dalam bermain, komputer harus memiliki kemampuan :

- a. Dalam membuat langkah dan berpikir (menentukan jalan terbaik).
- b. Komputer harus memiliki kemampuan dalam membuat langkah berdasarkan permintaan pemain (manusia).
- c. Komputer harus memiliki kemampuan dalam menentukan akhir permainan dan hasilnya.

### 3.4 Strategi Pemecahan Masalah

Solusi permasalahan hasil analisa adalah dengan membuat aplikasi permainan yang memenuhi kriteria-kriteria yang disebutkan pada hasil analisa. Dalam pembuatan aplikasi, dibutuhkan perancangan yang bertujuan untuk memberikan gambaran awal tentang aplikasi yang akan dibuat. Perancangan ini juga memberikan kemudahan dalam tahap implementasi.

Rancangan aplikasi ini dibagi menjadi dua bagian. Rancangan yang pertama adalah *user interface* dan rancangan yang kedua adalah rancangan alur proses dalam bentuk *flowchart* dan algoritma.

### 3.5 Analisis dan Perancangan Desain Sistem

#### a. User Interface

Agar suatu aplikasi mudah digunakan, maka diperlukan *user interface* yang dapat dengan mudah dimengerti oleh *user*. Untuk menghasilkan *user interface* yang mudah dimengerti oleh *user* maka diperlukan rancangan layar sebelum diimplementasikan dalam bentuk program. Berikut ini adalah rancangan layar untuk program game catur.

##### 1.) Rancangan Layar *Form* Awal Permainan

Rancangan layar *form* awal permainan merupakan *form* yang akan ditampilkan pada saat pertama kali aplikasi dijalankan. Pada *form* awal ini akan muncul splashscreen sebagai pembuka untuk memulai aplikasi.

##### 2.) Rancangan Layar *Form* Menu

Rancangan layar *form* menu merupakan *form* yang akan ditampilkan pada saat *user* akan memulai permainan baru. Pada *form* ini terdapat menu-menu yang akan memudahkan *user* untuk memulai permainan atau mengatur permainan baru.

Rancangan layar *form* mulai permainan baru dapat dilihat pada gambar 3.1.

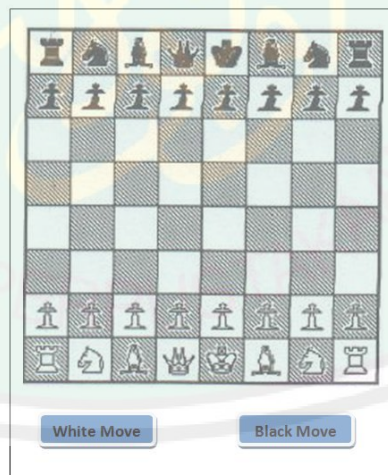


Gambar 3.1 Rancangan Layar *Form* Menu.

### 3.) Rancangan Layar *Form* Mulai Permainan/Play

Rancangan layar *form* mulai permainan merupakan *form* yang akan ditampilkan pada saat *user* mengklik menu mulai permainan. Pada *form* ini bidak-bidak catur ditempatkan pada posisi awalnya masing-masing.

Rancangan layar *form* mulai permainan dapat dilihat pada gambar 3.2

Gambar 3.2 Rancangan Layar *Form* Mulai Permainan.

### 4.) Rancangan Layar *Form* Cara Bermain/How To Play

Rancangan layar *form* cara bermain merupakan *form* yang akan ditampilkan pada saat *user* mengklik menu cara bermain. Dalam *form* ini *user* bisa mengetahui aturan atau tata cara dalam permainan catur.

Rancangan layar *form* cara bermain dapat dilihat pada gambar 3.3

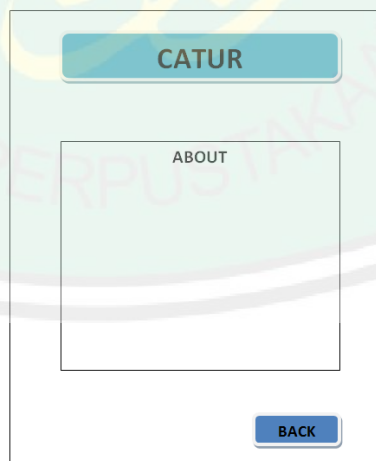


Gambar 3.3 Rancangan Layar *Form* Cara Bermain.

#### 5.) Rancangan Layar *Form* About

Rancangan layar *form* about merupakan *form* yang akan ditampilkan pada saat *user* mengklik menu about. Dalam *form* ini *user* bisa melihat pembuat atau author game ini.

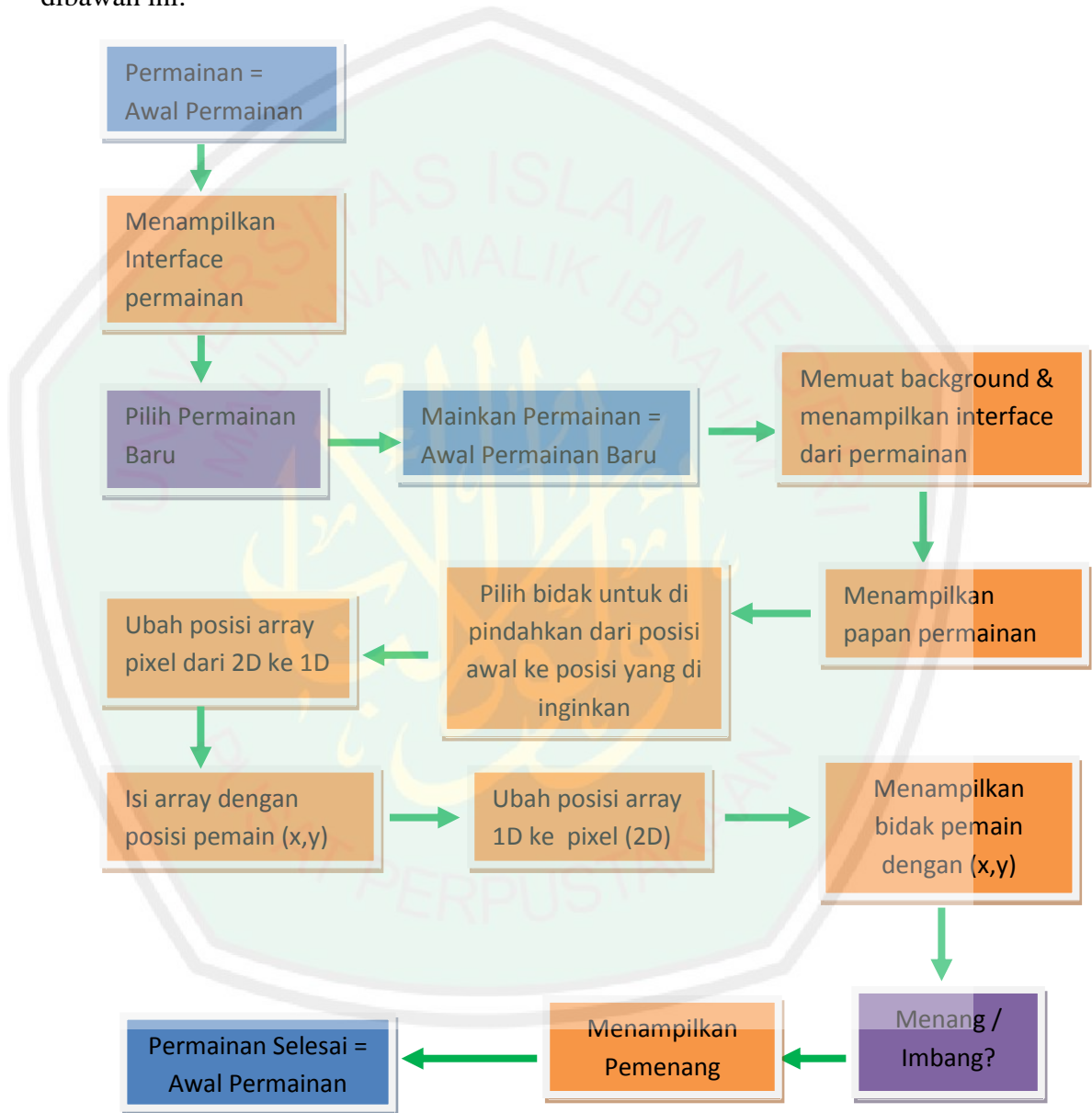
Rancangan layar *form* about dapat dilihat pada gambar 3.4



Gambar 3.4 Rancangan Layar *Form* About.

### 3.6 Analisis dan Perancangan Sistem

Dalam sistem ini terdapat satu aktor, yaitu *player*. Saat *player* akan memulai permainan maka alur atau rancangan sistem akan dijelaskan seperti gambar dibawah ini.



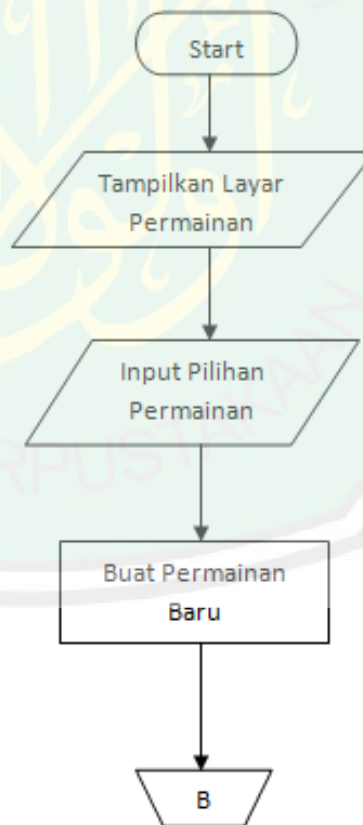
Gambar 3.5 Block Diagram dari Permainan Catur

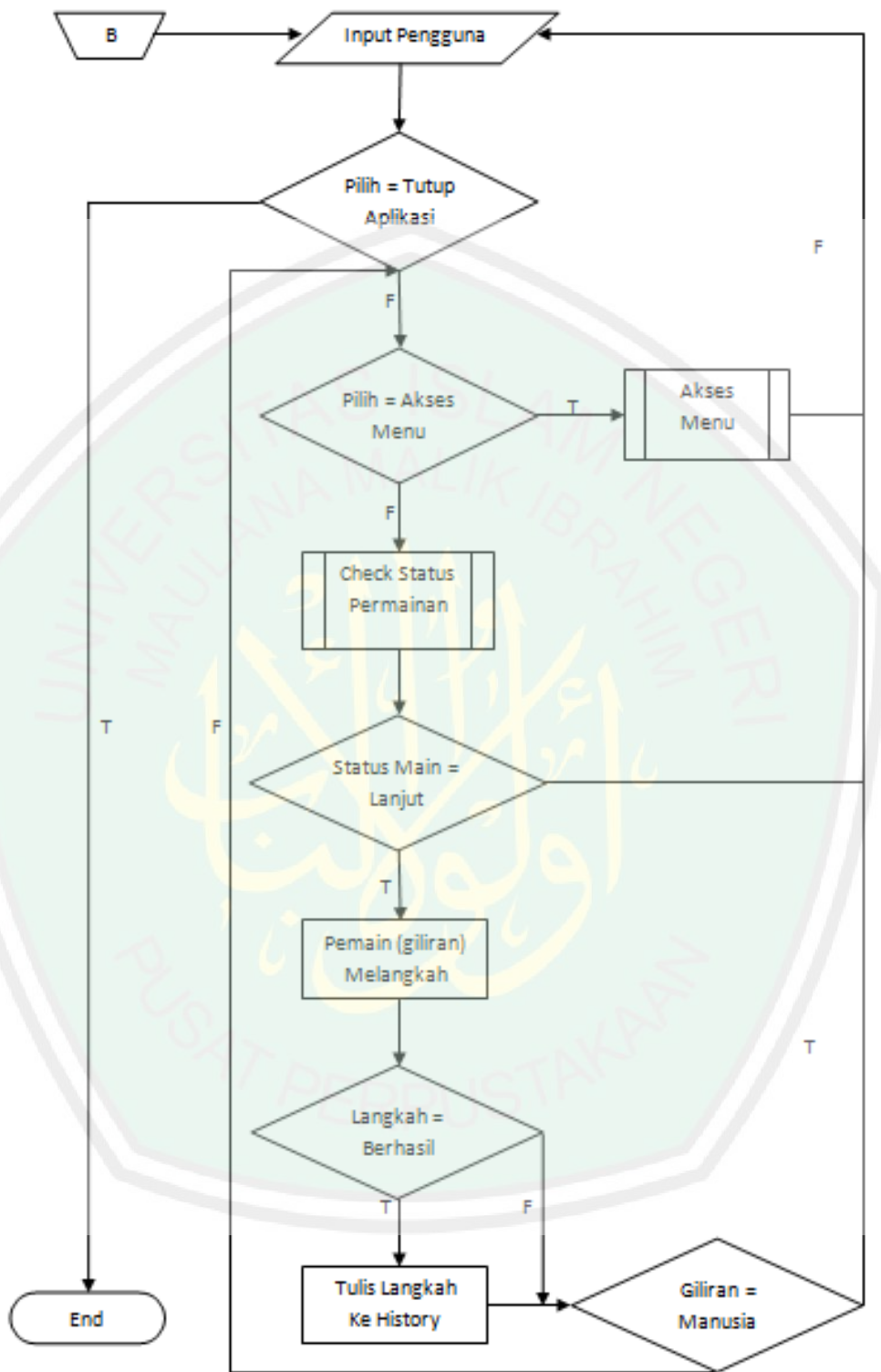
### a.) *Flowchart* (Diagram Alur)

Untuk menggambarkan urutan-urutan proses pada sistem ini, maka digunakan sebuah *flowchart* sebagai penjelas. Berikut ini akan diberikan beberapa *flowchart* untuk masing-masing proses.

#### 1.) *Flowchart* Permainan

*Flowchart* ini menggambarkan proses yang dijalankan pada saat awal aplikasi berjalan dan sepanjang aplikasi dijalankan. Pada *flowchart* permainan, sistem akan mengecek semua inputan pengguna. Inputan ini dapat berupa permintaan keluar, akses menu atau membuat langkah. Berikut diagram alur permainan.

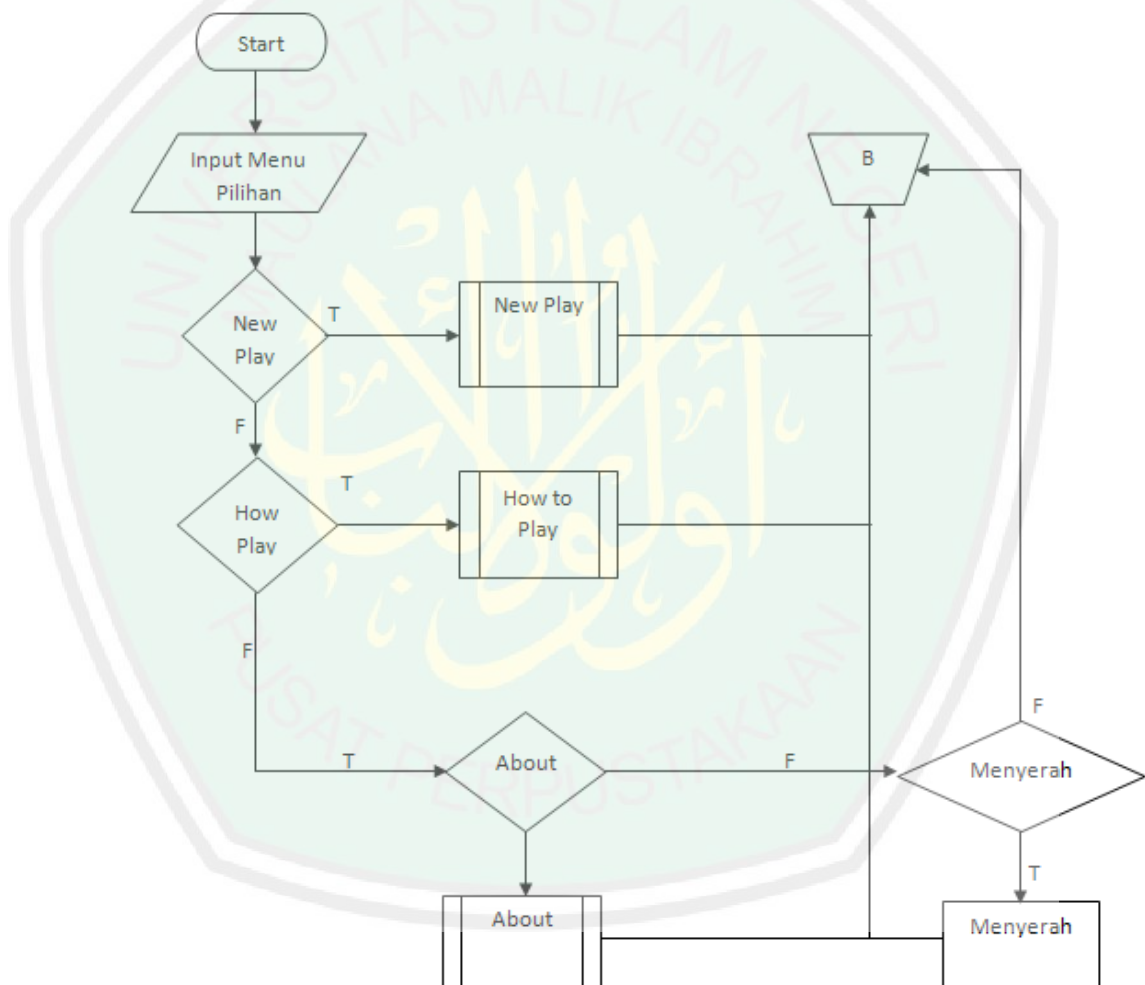




Gambar 3.6 Flowchart Permainan.

## 2.) Flowchart Akses Menu

Pada *flowchart* Permainan, terdapat *predefine* proses yang disebut Akses Menu dan *Check Status Permainan*. *flowchart Check Status Permainan* akan dibahas kemudian. *Flowchart Akses Menu* merupakan rancangan dalam mengendalikan akses pengguna terhadap menu aplikasi. Berikut *flowchart* Akses Menu:

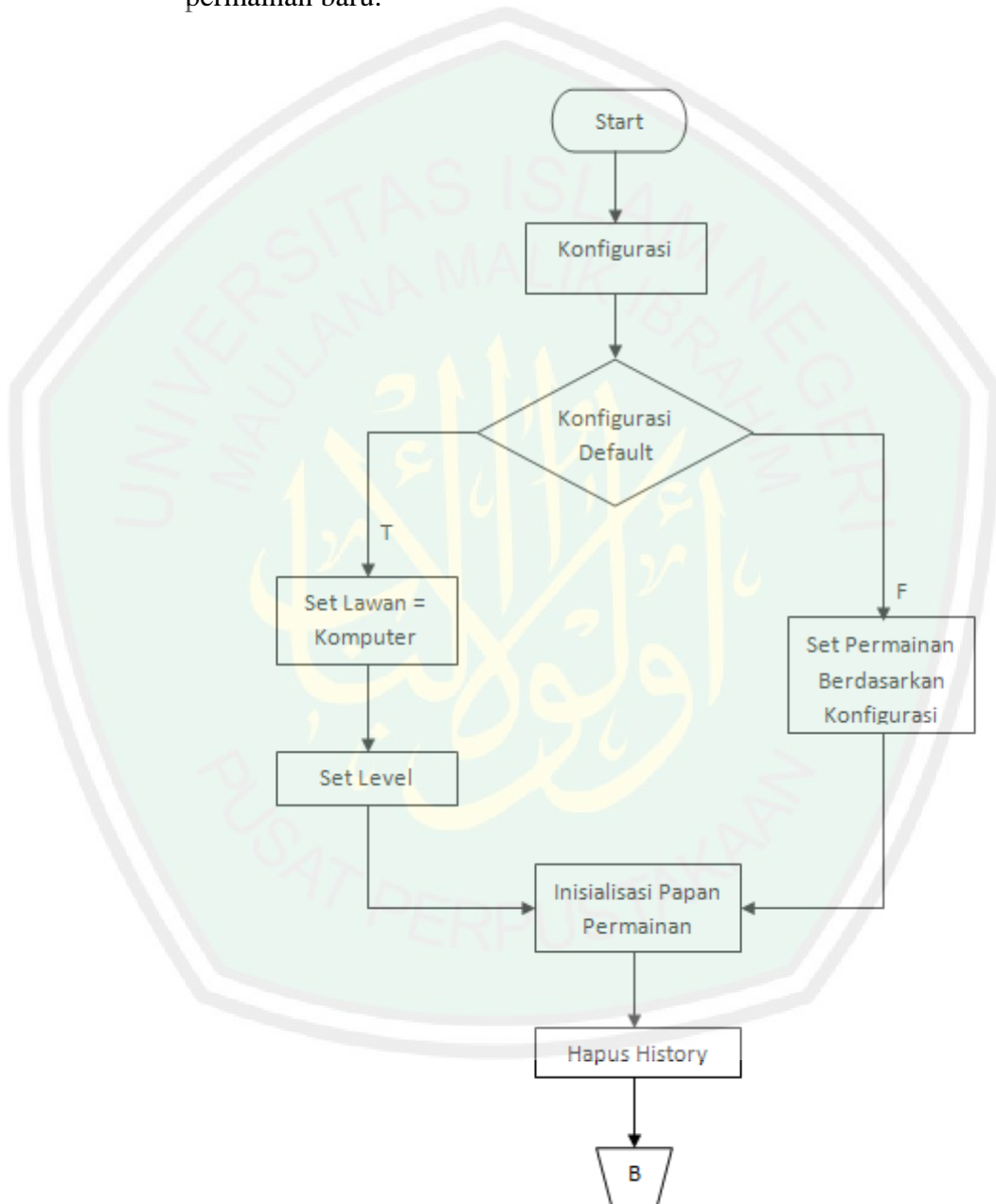


Gambar 3.7 Flowchart Akses Menu.

## 3.) Flowchart Mulai Permainan Baru

Pada *flowchart* Akses Menu, terdapat beberapa *predefine* proses yaitu : Mulai Permainan Baru, Simpan Permainan, Muat Permainan

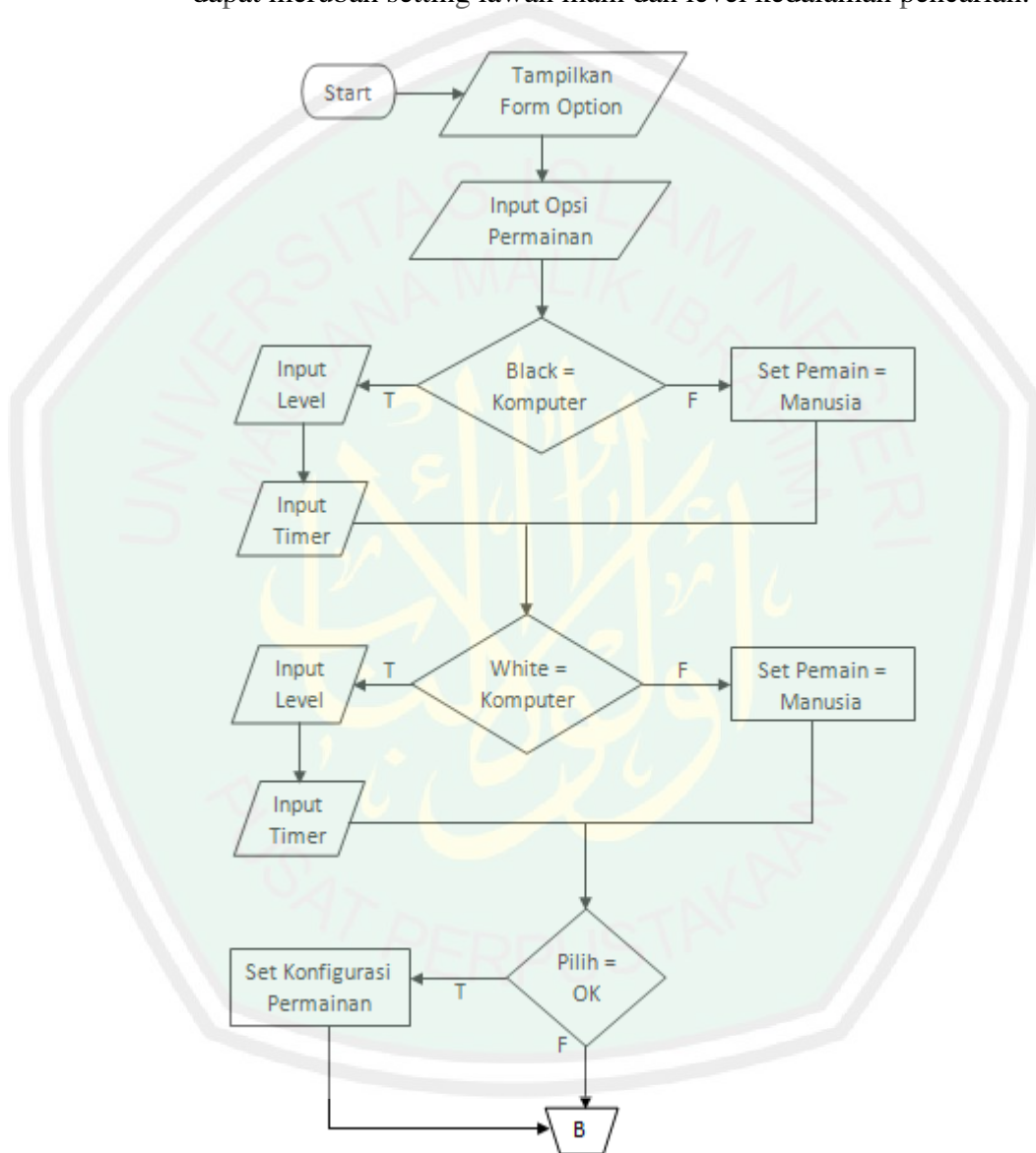
dan Konfigurasi. *Flowchart* Simpan Permainan, Muat Permainan dan Konfigurasi akan dibahas kemudian. *Flowchart* Mulai Permainan Baru menggambarkan proses yang dijalankan ketika *user* memulai permainan baru.



Gambar 3.8 *Flowchart* Mulai Permainan Baru.

#### 4.) Flowchart Option

Flowchart ini menjelaskan proses yang dijalankan ketika pengguna menggunakan fasilitas menu Option, dimana pengguna dapat merubah setting lawan main dan level kedalaman pencarian.



Gambar 3.9 Flowchart Option.



## 5.) *Flowchart* Strategi Permainan Komputer Menggunakan Metode Forward Chaining Pruning

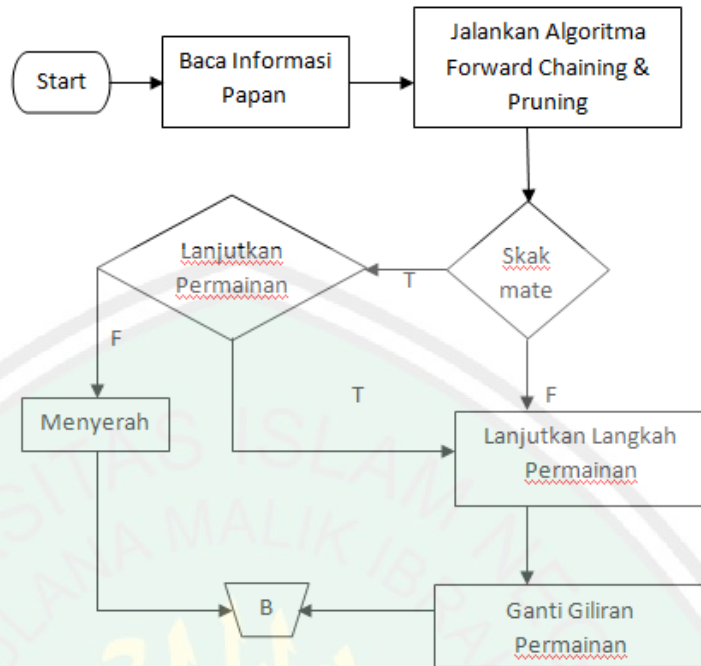
Dalam permainan catur, pemain melangkah dengan memindahkan bidak dari kotak asal ke kotak tujuan pada papan catur yang digunakan.

Perbedaan mendasar antara langkah manusia dan langkah komputer adalah pada proses penentuan langkah terbaik. Pada penentuan langkah terbaik, manusia menggunakan pikirannya sedang komputer menggunakan *formula-formula* dengan menghitung nilai dari tiap langkah yang diambil.

Dengan demikian, alur proses dari komputer melangkah adalah sebagai berikut:

- a.) Menentukan langkah terbaik.
- b.) Periksa validitas langkah.
- c.) Lakukan pemindahan bidak dari kotak asal ke kotak tujuan.
- d.) Periksa kondisi bidak yang lain setelah pemindahan bidak. Jika ada bidak dalam keadaan das, batalkan langkah.

Berdasarkan alur proses di atas, maka dapat digambarkan *flowchart* seperti pada gambar berikut ini :



Gambar 3.10 *Flowchart* Komputer Melangkah.

Dalam *flowchart* komputer melangkah, terdapat proses yaitu Jalankan Algoritma *Forward Chaining*.

Dalam pencarian langkah terbaik, komputer menggunakan beberapa algoritma umum. Algoritma yang dimaksud adalah sebagai berikut :

a.) ***Forward Chaining dan Pruning***

*Forward Chaining* (runut maju) berarti menggunakan himpunan aturan kondisi-aksi. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil (Wilson,1998). Dalam hal ini pelacakan dimulai dari informasi masukan, dan selanjutnya mencoba menggambarkan kesimpulan. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut

dijalankan. Contoh permasalahan menggunakan *Forward Chaining*, diketahui kaidah tipe *IF..THEN..* berikut:

*IF A THEN B*

*IF B THEN C*

*IF C THEN D*

*Pruning* (pemangkasan *tree*) merupakan bagian dari proses pembentukan *decision tree*. Saat pembentukan *decision tree*, beberapa node merupakan *outlier* maupun hasil dari *noise* data. Penerapan *pruning* pada *decision tree* dapat mengurangi *outlier* maupun *noise* data pada *decision tree* awal sehingga dapat meningkatkan akurasi pada klasifikasi data.

*Function set* pada program aplikasi ini merupakan kumpulan fungsi yang melakukan analisa pemeriksaan kondisi pada papan catur sewaktu permainan berlangsung sebelum bidak putih memutuskan akan menggerakkan bidak raja putih atau bidak benteng putih. Terdapat 13 buah fungsi yang akan digunakan oleh algoritma *forward chaining* dan *pruning* untuk program aplikasi ini. Masing – masing fungsi diberi nama sesuai dengan kondisi apa yang dicek oleh fungsi tersebut. 13 fungsi tersebut antara lain:

*F1-KingsInOpposition*, memeriksa apakah posisi bidak raja putih dan raja hitam berada pada posisi oposisi yaitu posisi saling berhadapan antara keduanya dan dipisahkan hanya oleh satu petak pada papan catur.

*F2-KingsCloseToOpposition*, memeriksa apakah posisi bidak raja putih dan hitam hamper mendekati posisi oposisi.

*F3-BlackKingOnEdge*, memeriksa apakah posisi bidak raja hitam berada di sudut-sudut pojok dari papan catur.

*F4-OneEmptyFieldToBlackKing*, memeriksa apakah bidak raja hitam hanya bisa melakukan satu langkah saja.

*F5-NoEmptyFieldToBlackKing*, memeriksa apakah bidak raja hitam tidak bisa melangkah kemana-mana lagi.

*F6-DistanceBKWK3*, memeriksa apakah posisi bidak raja putih dan hitam terpisah sebanyak tiga petak pada papan catur.

*F7-PossibleControlOfEscape-Line*, memeriksa apakah bidak benteng putih dapat membatasi gerak bidak raja hitam.

*F8-PossibleLateralCheck*, memeriksa apakah bidak benteng putih dapat mengunci bidak raja hitam.

*F9-ControlOfLineBetweenKings*, memeriksa apakah bidak benteng putih menguasai atau mengontrol satu jalur (baris atau kolom) di antara bidak raja putih dan hitam.

*F10-ThreatenedRook*, memeriksa apakah bidak benteng putih terancam untuk dimakan oleh bidak raja hitam.

*F11-ProtectedRook*, memeriksa apakah bidak benteng putih terlindungi oleh bidak raja putih atau dengan kata lain posisi bidak raja putih berada di dekat bidak benteng putih.

*F12-PossibleProtectionOfRook*, memeriksa apakah dengan posisinya sekarang, bidak raja putih dapat melindungi bidak benteng putih.

*F13-PossibleProtectionOfRook-WithOpposition*, memeriksa apakah dengan posisinya sekarang, bidak raja putih dapat melindungi

bidak benteng putih dengan cara bergerak ke posisi oposisi dengan bidak raja hitam.

*Terminal set* pada program aplikasi ini merupakan kumpulan langkah – langkah khusus yang dapat dilakukan oleh bidak raja putih atau bidak benteng putih. Terdapat delapan buah *terminal* untuk program aplikasi catur ini dan masing – masing *terminal* ini diberi nama sesuai dengan langkah apa yang dilakukan dan jenis bidak putih apa yang akan digerakkan. Delapan buah *terminal* ini antara lain:

1. *T1-ProtectRook*, menggerakkan bidak raja putih untuk melindungi bidak benteng putih.
2. *T2-NearlyOpposition*, menggerakkan bidak raja putih untuk mendekati posisi oposisi dengan bidak raja hitam.
3. *T3-DistantOpposition*, menggerakkan bidak raja putih ke posisi oposisi dengan bidak raja hitam.
4. *T4-ControlEscapeLine*, menggerakkan bidak benteng putih untuk membatasi ruang gerak bidak raja hitam.
5. *T5-LateralMove*, menggerakkan bidak benteng putih untuk bergerak sepanjang jalur baris atau kolom yang dapat dilalui oleh bidak raja hitam.
6. *T6-ControlLineBetweenKings*, menggerakkan bidak benteng putih untuk menguasai satu jalur di antara bidak raja putih dan hitam.
7. *T7-LateralCheck*, menggerakkan bidak benteng putih untuk mengunci bidak raja hitam.
8. *T8-MoveAwayToFreeBK*, menggerakkan bidak benteng putih untuk menjauhi bidak raja hitam.

Kemudian catat fakta pertama yang telah diinputkan oleh *user*.

1. Cari *rule* yang bagian premisnya sesuai dengan fakta yang diinputkan.
2. Jika *rule* tidak ada maka cari fakta selanjutnya, kemudian kembali ke langkah no 2.
3. Jika *rule* ada maka akan ketemu konklusi sementara
4. Cari *rule* yang menggunakan konklusi sementara sebagai premis
5. Jika *rule* tidak ada maka konklusi tidak ditemukan.
6. Namun jika *rule* terpenuhi cari fakta selanjutnya yang sesuai dengan konklusi sementara.
7. Cari *rule* yang menggunakan konklusi sementara dan fakta selanjutnya sebagai premis.

Jika *rule* tidak terpenuhi maka konklusi tidak ditemukan, namun jika *rule* terpenuhi maka konklusi ditemukan. Berikut merupakan *pseudo code* untuk algoritma *Forward Chaining dan Pruning*.

```

int langkahTerbaik(Papan papan,int kedalaman,Pemain pemain,Pemain lawan)
    pemainAwal=pemain
    langkah= forward(board,kedalaman,pemain,lawan)
    return langkah

int forward(Papan papan,int kedalaman,Pemain pemain,Pemain lawan)
    langkahTerbaik=0
    if(kedalaman=0 atau tidak bisa melangkah)
        langkahTerbaik = evaluasi papan untuk pemain awal
        return langkahTerbaik
    end if

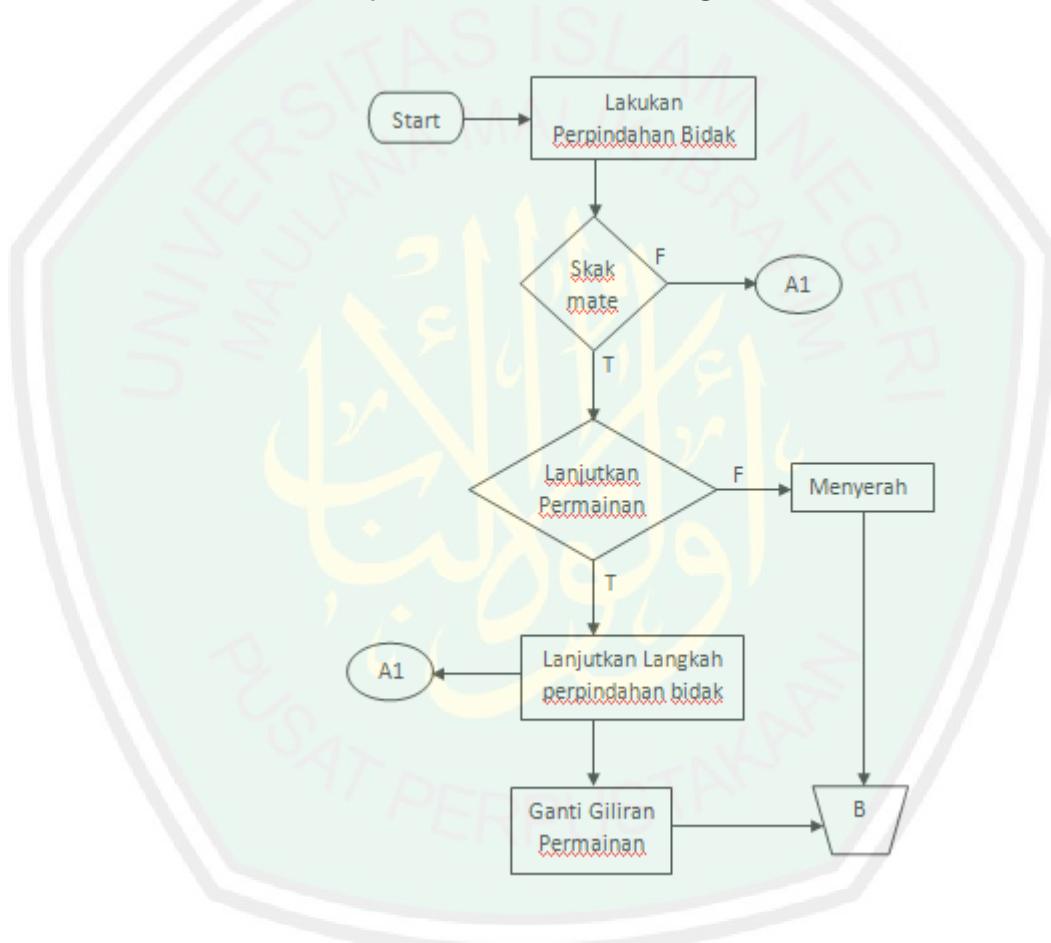
    foreach langkah in listLangkah loop
        jalankan langkah
        langkah = forward(p,kedalaman-1,lawan,pemain)
        mundur langkah
        if(pemain is pemainAwal) then
            if(langkah>langkahTerbaik) then
                langkahTerbaik=langkah
            end if
        else
            if(langkah<langkahTerbaik) then
                langkahTerbaik=langkah
            end if
        end for

    return langkahTerbaik
  
```

### 6.) Flowchart Strategi Permainan Manusia

Pada proses strategi manusia melangkah(*move*), komputer hanya mengecek validitas langkah sebelum membuat langkah. Pengecekan validitas langkah terdiri dari pengecekan apakah langkah dari bidak yang dipilih sesuai aturan.

Berikut *flowchart* manusia melangkah :



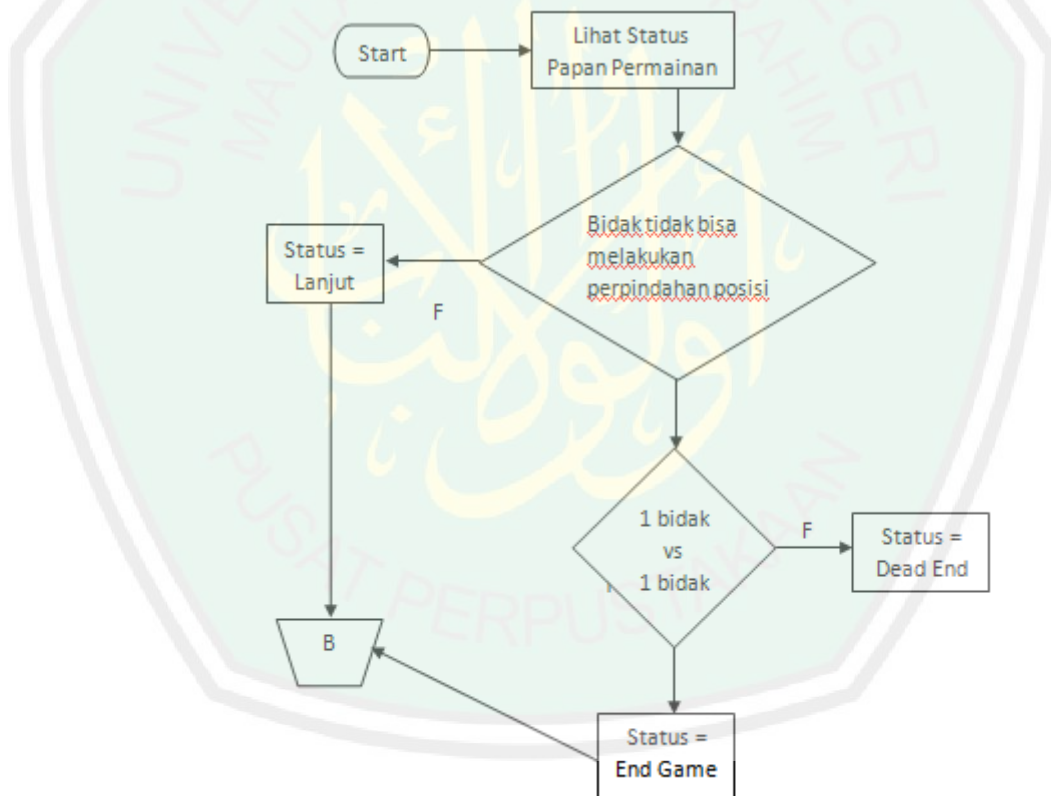
Gambar 3.11 *Flowchart* Manusia Melangkah

### 7.) Flowchart Check Status Permainan

Akhir permainan catur ada dua, salah satu pemain menang atau kedua pemain seri. Seorang pemain dinyatakan menang apabila pemain tersebut sedang menyerang raja lawan dan lawan tidak menemukan langkah yang dapat menyelamatkan raja dari serangan

(Skakmat). Permainan dinyatakan seri apabila raja tidak diserang dan pemain yang mendapat giliran tidak dapat menjalankan satu bidak pun (Buntu). Pada *flowchart* Permainan (Gambar 3.12), terdapat *predefine* proses yang disebut *Check Status Permainan*. Proses ini digunakan untuk memeriksa apakah permainan masih bisa dilanjutkan atau sudah berakhir. Bila sudah berakhir, maka aplikasi akan menampilkan status akhir permainan.

Berikut ini adalah *flowchart* *Check Status Permainan*:



Gambar 3.12 *Flowchart* *Check Status Permainan*

### b. Algoritma

Algoritma digunakan untuk mempermudah dalam pembuatan dan perancangan suatu program. Dalam sistem ini terdiri dari beberapa



algoritma yang digunakan untuk menjalankan proses pada program. Berikut adalah algoritma dari setiap proses yang ada pada aplikasi.

### 1.) Algoritma Permainan

Algoritma ini menjelaskan proses yang dilakukan disaat awal dan sepanjang permainan.

```

1.      Tampilkan Layar Utama
2.      Periksa Permainan Sebelumnya
3.      Buat Permainan Baru
4.      End if
5.      Baca Input Pengguna
6.      While Input <> Keluar
7.          If Input = Akses Menu then
8.              Pilih Akses Menu
9.          End if
10.         Check Status Permainan
11.         While Status Main = Lanjut
12.             Pemain Melangkah
13.             If Langkah Berhasil then
14.                 Tulis History Langkah
15.             End if
16.             If Giliran Manusia then
17.                 Lakukan Langkah
18.                 Tulis History Langkah
19.             Else
20.                 Check Status Permainan
21.             End if
22.         End while
23.     BacaInputPengguna
24.     End while
  
```

### 2.) Algoritma Mulai Permainan Baru

Algoritma dari proses Mulai Permainan Baru adalah sebagai berikut:

```

1.  Baca Konfigurasi
2.  If Konfigurasi = Default then
3.      Set Lawan = Komputer
4.      Set Level = Level
5.      Set Timer = Timer
6.  Else
7.      Set Pemain Berdasarkan Konfigurasi
8.  End if
9.  Inisialisasi Papan
10. Hapus History
  
```

### 3.) Algoritma Konfigurasi

Algoritma dari proses Konfigurasi adalah sebagai berikut :

```

1. Tampilkan Form Konfigurasi
2. Input Pemain
3. If Pemain Black = Komputer
4.     Input Level
5.     Input Timer
6. Else
7.     Set Pemain = Manusia
8.     Else If Pemain White = Komputer
9.         Input Level
10.        Input Timer
11.    Else
12.        Set Pemain = Manusia
13.        Else If Pilih = Ok
14.            Set Konfigurasi Permainan
15.        Else
16.            Tutup Form konfigurasi
17.        End If
18.    End If
19. End If

```

### 4.) Algoritma Strategi Permainan Komputer Menggunakan Metode Forward Chaining Pruning

Algoritma dari proses Strategi Komputer Memindahkan bidak adalah sebagai berikut :

```

1. Baca Informasi Papan
2. Jalankan Algoritma Forward Chaining & Pruning
3. If Skakmate
4.     If Bisa Melangkah (melakukan perpindahan posisi bidak) =
        Lanjut Permainan
5.         Lakukan Perpindahan posisi bidak
6.     Else
7.         Menyerah
8.     End If
9. Else
10.    Lakukan Perpindahan posisi bidak
11. End If
12. Ganti Giliran Perpindahan posisi bidak

```

### 5.) Algoritma Strategi Manusia Memindahkan Bidak

Algoritma dari proses Strategi Permainan Manusia memindahkan bidak adalah sebagai berikut :

1. Lakukan Perpindahan posisi bidak
2. If Skakmate
3.     If Bisa Memindahkan bidak
4.         Lakukan Perpindahan posisi bidak
5.     Else
6.         Menyerah
7.     End If
8. Else
9.     Lakukan Perpindahan posisi bidak
10. End If
11.     Ganti Giliran Melangkah untuk memindahkan posisi bidak

### 6.) Algoritma Check Status Permainan

Algoritma dari proses Komputer Melangkah adalah sebagai berikut :

1. Baca Informasi Papan
2. If Tidak Ada Kemungkinan Perpindahan posisi bidak
3.     If 1Bidak vs 1bidak lokasi berjauhan
4.         Status = End Game
5.     Else
6.         Status = Dead End
7.     End If
8. Else
9.     Status = Lanjut
10. End If

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Implementasi

Pada bab ini akan dijelaskan bagaimana pengimplementasian *game* catur ini terhadap rancangan program yang telah dibuat, termasuk penjelasan mengenai perangkat lunak dan perangkat keras yang digunakan. Serta pengujian program yang berfungsi untuk mengevaluasi sistem agar mengetahui kelebihan dan kekurangan sistem ini.

##### 4.1.1 Spesifikasi Perangkat Keras dan Lunak

###### a. Perangkat Keras

Berikut ini adalah perangkat keras yang digunakan selama tahap pengimplementasian sistem, diantaranya:

1. Desktop PC.

Processor Intel Dual Core T2080 @ 2.4 GHz.

DDRII 2048 MB, Hitachi 2.5" *harddisk drive* 160 GB.

Layar LCD : 1280x800.

###### b. Perangkat Lunak

Perangkat Lunak yang digunakan selama tahap implementasi sistem, diantaranya:

1. Microsoft Windows Vista sebagai sistem operasi.

2. Eclipse Indigo Version: 3.7.1 sebagai editor yang digunakan untuk menulis bahasa program JAVA.
3. JDK 1.7.0\_02, JRE 7

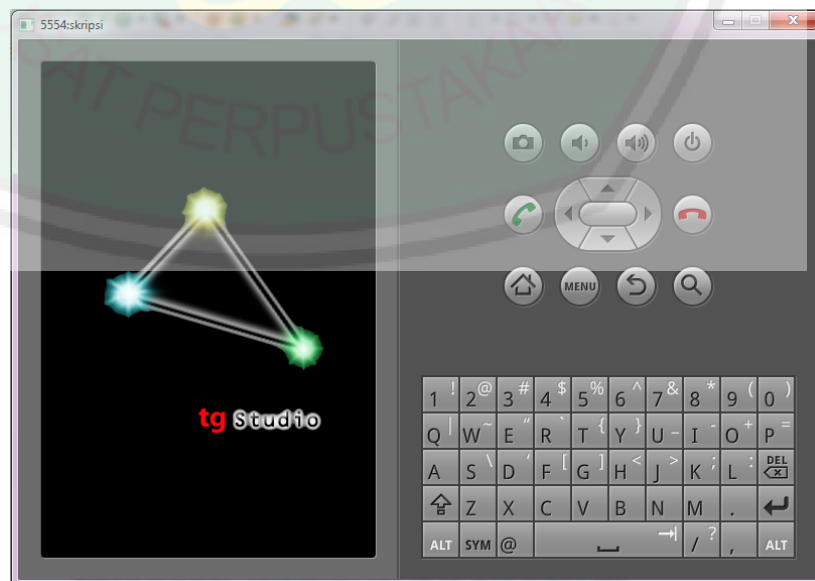
#### 4.1.2 Implementasi Desain Interface

Agar sistem dapat digunakan dengan mudah, maka *interface* atau tampilan layar yang dibuat haruslah mudah dimengerti. Hal itu bertujuan agar pemakai merasa nyaman dan tidak jenuh dalam menjalankan sistem ini. Berikut ini adalah tampilan layar setiap menu yang ada pada sistem ini.

##### a. Tampilan Layar *Form* Awal Aplikasi

*Form* Awal Aplikasi adalah *form* yang pertama kali muncul saat aplikasi dijalankan. Pada *form* ini menampilkan *splashscreen* dari aplikasi permainan ini kemudian akan muncul *form menu* dari tampilan aplikasi permainan Catur.

Berikut adalah tampilan layar *form* awal aplikasi :



Gambar 4.1 : Tampilan Layar *Form* Awal Aplikasi

Berikut ini potongan source code xml awal aplikasi (splashscreen)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10dip"
    android:gravity="center"
    android:id="@+id/coba"
    android:background="@drawable/splash">
</RelativeLayout>
```

Berikut ini potongan source code java awal aplikasi (splashscreen)

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.animasi);
    Thread timer = new Thread() {
        @Override
        public void run() {
            long startTime = System.currentTimeMillis();
            // lakukan sesuatu pada saat memulai game

            long now = System.currentTimeMillis();
            if (now - startTime < 5000){
                try {
                    sleep(5000 - (now - startTime));
                } catch (InterruptedException iEx){
                }
            }
            SplashScreen.this.finish();
            startActivity(new
            Intent(SplashScreen.this.getApplicationContext(),
            Catur_JawaActivity.class));
        }
    };
    timer.start();
}
```

## b. Tampilan Layar *Form* Menu Permainan

*Form* menu permainan adalah *form* yang menampilkan beberapa menu yang ada pada aplikasi permainan, pada menu permainan terdapat menu : permainan baru, cara bermain, tentang, dan keluar.

Berikut adalah tampilan layar *form* menu permainan :

Gambar 4.2 : Tampilan Layar *Form Menu*

Berikut ini potongan source code xml menu permainan

```

<AbsoluteLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/main_layout"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:background="@drawable/main_menu2"
android:gravity="center"
android:padding="10dp" >

  <LinearLayout
    android:id="@+id/main_linear"
    android:layout_width="237dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="48dp"
    android:layout_x="29dp"
    android:layout_y="120dp"
    android:gravity="center_horizontal"
    android:orientation="vertical" >

    <Button
      android:id="@+id/read_button"
      android:layout_width="wrap_content"
      android:layout_height="match_parent"
      android:layout_margin="10dp"
      android:layout_marginTop="19dp"
      android:background="@drawable/bg_button"
      android:text="Permainan Baru"
      android:textColor="#ffffff" />

```

Berikut ini potongan source code java menu permainan

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.mainmenu);
    Button mRead = (Button)findViewById(R.id.read_button);
    mRead.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            startActivity(new
Intent(Catur_JawaActivity.this, ChessIt.class));}
    });
    Button m4Read = (Button)findViewById(R.id.Button01);
    m4Read.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            startActivity(new Intent(Catur_Activity.this,
HowPlay.class));}
    });
    Button m5Read =
(Button)findViewById(R.id.favorite_button);
    m5Read.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            startActivity(new Intent(Catur_Activity.this,
About.class)); }
    });
    Button0 = (Button) findViewById(R.id.keluar);
    Button0.setOnClickListener(this); }
    public void onClick(View view) {
        if (view == Button0) {
            AlertDialog.Builder builder = new
AlertDialog.Builder(this);
            builder.setMessage("Apakah Anda Benar-Benar ingin
keluar?")
                .setCancelable(false)
                .setPositiveButton("Ya",
new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog,
int id) {
                        Intent exit = new Intent(Intent.ACTION_MAIN);
                        exit.addCategory(Intent.CATEGORY_HOME);
                        exit.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                        startActivity(exit);
                        Catur_Activity.this.finish();}})
                .setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog,
int id) {dialog.cancel();}
                }).show();
        } else {

```

### c. Tampilan Layar Form Cara Bermain

*Form* Cara bermain adalah *form* yang menunjukkan atau menjelaskan aturan atau tata cara dalam permainan catur.



Berikut adalah tampilan layar *form* Cara bermain :



Gambar 4.3 : Tampilan Layar *Form* Cara Bermain

Berikut ini potongan source code xml cara bermain

```
<AbsoluteLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main_layout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/cara_bermain"
    android:gravity="center"
    android:padding="10dp" >
<ScrollView
    android:layout_width="wrap_content"
    android:layout_height="274dp"
    android:layout_x="20dp"
    android:layout_y="90dp"
    >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="262dp"
        android:layout_height="684dp"
        android:text="Catur adalah permainan yang dimainkan
antara dua pemain. Permainan ini dimainkan pada papan catur ,
yang merupakan papan kotak-kotak persegi dengan 64 kuadrat
diatur dalam sebuah-oleh-delapan grid delapan. Pada awalnya,
```

Berikut ini potongan source code java menu permainan

```
protected void onResume() {
    super.onResume();
    Music.play(this, R.raw.main); }
protected void onPause() {
    super.onPause();
    Music.stop(this);}
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.howplay);
    Button mAwal = (Button)findViewById(R.id.howplay);
    mAwal.setOnClickListener(new View.OnClickListener() {
        public void onClick(View arg0) {
            startActivity(new Intent(HowPlay.this,
                Catur_Activity.class));
            finish();
        }
    });
}
```

#### d. Tampilan Layar *Form* Tentang

*Form* Awal Aplikasi adalah *form* merupakan *form* yang akan ditampilkan pada saat *user* mengklik menu about. Dalam *form* ini *user* bisa melihat pembuat atau author game ini.

Berikut adalah tampilan layar *form* tentang :



Gambar 4.4 : Tampilan Layar *Form* Tentang

Berikut ini potongan source code xml tentang

```
<AbsoluteLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main_layout"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/tentang"
    android:gravity="center"
    android:padding="10dip" >

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="64dp"
    android:layout_y="181dp"
    android:text="TEGUH SANTOSO"
    android:textAppearance="?android:attr/textAppearanceLarge"
/>

</AbsoluteLayout>
```

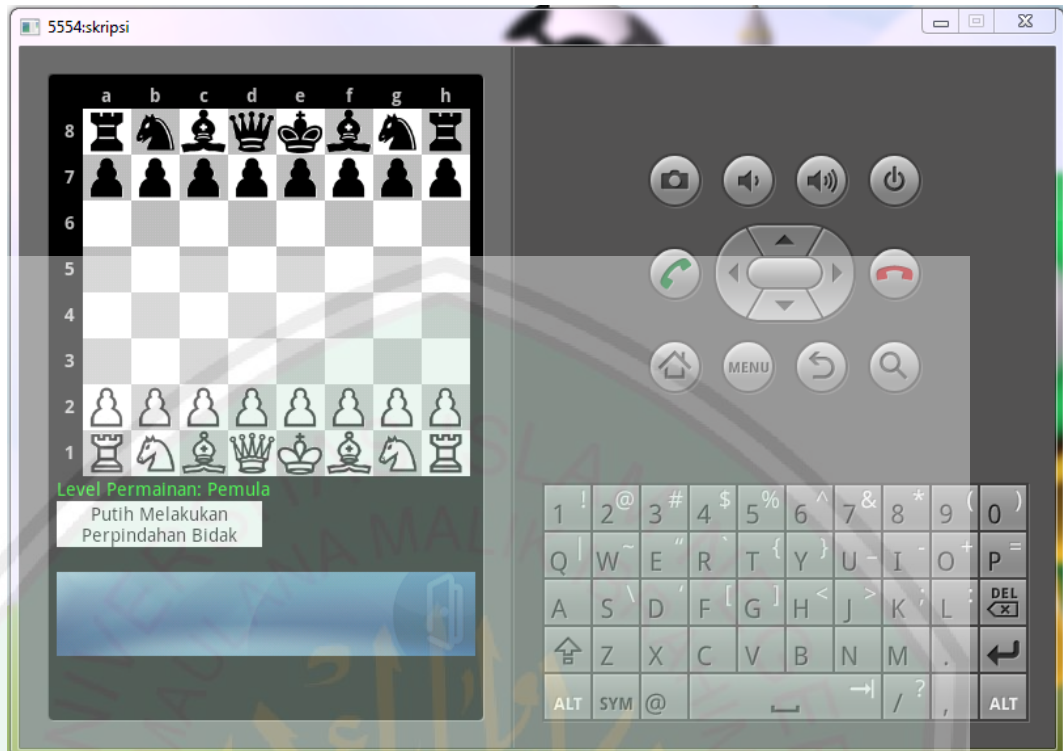
Berikut ini potongan source code java tentang

```
protected void onResume() {
    super.onResume();
    Music.play(this, R.raw.main);}
protected void onPause() {
    super.onPause();
    Music.stop(this);}
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.about);
    Button mAwal = (Button)findViewById(R.id.about);
    mAwal.setOnClickListener(new View.OnClickListener() {
        public void onClick(View arg0) {
            finish();}
    });}
private Button mRead;}
```

#### e. Tampilan Layar *Form* Mulai Permainan

Dengan memilih memulai permainan baru maka akan tampil bidak-bidak catur pada posisinya masing-masing. Di bagian bawah papan catur terdapat penunjuk giliran langkah untuk pemain.

Berikut adalah tampilan layar *form* mulai permainan :

Gambar 4.5 : Tampilan Layar *Form* Mulai Permainan

Berikut ini potongan source code xml Mulai Permainan

```

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="10dip"
    android:gravity="center"
    android:id="@+id/widget60"
    android:orientation="vertical"
    xmlns:admobsdk="http://schemas.android.com/apk/res/tegu
h.skripsi.catur">
    <TableLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:orientation="vertical"
    android:stretchColumns="*" >
        <TableRow>
            <TextView android:layout_column="1"
android:text="a" android:textStyle="bold"
                android:padding="3dip"
android:gravity="center"/>
            <TextView android:text="b"
android:padding="3dip" android:textStyle="bold"
android:gravity="center"/>
            <TextView android:text="c"

```

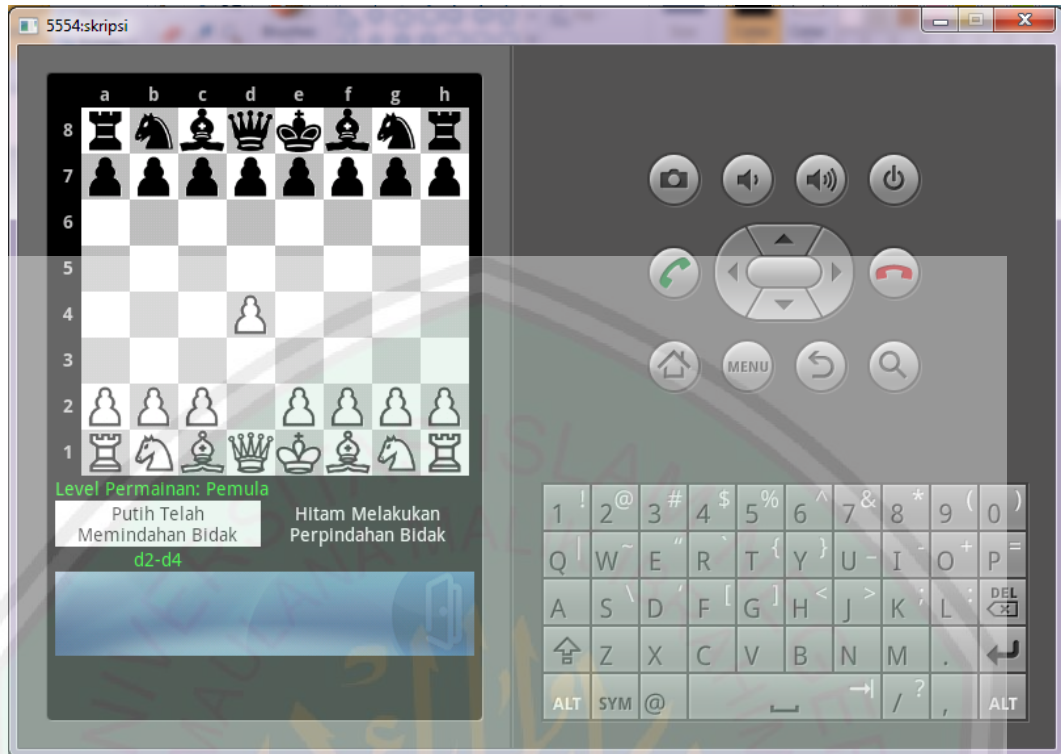
Berikut ini potongan source code java mulai permainan

```
private void init() {
    chessView = new ChessBoard(this);
    status = chessView.status; setMaxTime(5000); }
protected void onStart() {super.onStart(); }
protected void onPause() {super.onStop();
    stop(); }
public void onConfigurationChanged(Configuration
newConfig) {super.onConfigurationChanged(newConfig);
    if (newConfig.orientation ==
Configuration.ORIENTATION_LANDSCAPE
        || newConfig.orientation ==
Configuration.ORIENTATION_PORTRAIT
        || newConfig.orientation ==
Configuration.ORIENTATION_SQUARE
        || newConfig.orientation ==
Configuration.ORIENTATION_UNDEFINED) {
    } }
public void stop() {
    searcher.stopThinking(); }
public void playNewGame() {
    stop();
    computerSide = DARK;
    guessedMove = null;
    searcher = new Search();
    board = new Board();
    chessView.setupBoard(); }
```

#### f. Tampilan Layar Pemain Memindahkan Bidak

Dalam permainan catur, setiap pemain melangkah (memindahkan bidak) secara bergantian. Pemain dapat melangkah dengan memilih salah satu bidak yang akan dipindahkan. Pemain mengarahkan bidak bidak tersebut ke posisi yang diinginkan dengan mengklik bidak dan menggerakkan *mouse device*. Kemudian pemain harus mengklik lagi pada posisi tujuan bidak yang dipindahkan. Setelah langkah berhasil dilakukan, maka komputer akan menambahkan langkah tersebut ke dalam *history* langkah.

Berikut tampilan layar *form* pemain melangkah:



Gambar 4.6 : Tampilan Layar Pemain Putih Memindahkan Bidak



Gambar 4.7 : Tampilan Layar Pemain Hitam (Komputer) Memindahkan Bidak

Berikut ini potongan source code java Memindahkan bidak

```

private void setMove(Move m) {
    curMove = m;
    handler.post(doMakeMove);}
private final Runnable doMakeMove = new Runnable() {
    public void run() {
        makeMove(curMove); }};
private void switchMoveMarkers(boolean b) {
    whiteToMove = b;
    handler.post(doSwitchMoveMarkers);
    showStatus("");}
private final Runnable doSwitchMoveMarkers = new Runnable() {
    public void run() {
        chessView.switchMoveMarkers(whiteToMove);
    }};
private void showMove(String move, boolean b) {
    moveStr = move;
    whiteMoved = b;
    handler.post(doShowMove); }
private final Runnable doShowMove = new Runnable() {
    public void run() {
        chessView.showMove(moveStr, whiteMoved);
    }};
public void pieceChange(MoveBase mo) {
    MoveBase move = mo;
    if (move == null)
        return;
    int promote = 0;
    int to = move.getTo();
    int from = move.getFrom();
    if (((to < 8) && (board.side == LIGHT)) || ((to
    > 55) && (board.side == DARK)))
        && (board.getPiece(from) == PAWN)) {
        promote =
chessView.promotionDialog(board.side == LIGHT); }
    boolean found = false;
    Collection validMoves = board.gen();
    Iterator i = validMoves.iterator();
    Move m = null;
    while (i.hasNext()) {
        m = (Move) i.next();
        if (m.getFrom() == from && m.getTo() == to
    && m.promote == promote) {
            found = true;
            break; }}
    if (!found || !board.makeMove(m)) {
        showStatus("Illegal move");
        chessView.setHighlight(mo.getFromRow(),
mo.getFromCol(), false);
        chessView.setMoving(false);
    } else {
        setMove(m);
        showMove(m.toString(), board.side == DARK);
        switchMoveMarkers(board.side == LIGHT);
    }
}

```

### g. Tampilan Layar Akhir Permainan

Permainan berakhir jika salah satu pemain dinyatakan menang. Kondisi menang terjadi ketika salah satu pemain sedang dalam keadaan diskak dan pemain tersebut tidak dapat membuat langkah untuk dapat melepaskan diri dari kondisi tersebut. Jika kondisi seperti ini terjadi, maka pemain dianggap menang.

Berikut tampilan layar akhir permainan :



Gambar 4.8 : Tampilan Layar Akhir Permainan



Berikut ini potongan source code java Akhir permainan

```

private boolean isResult() {
    Collection validMoves = board.gen();
    Iterator i = validMoves.iterator();
    boolean found = false;
    while (i.hasNext()) {
        if (board.makeMove((Move) i.next())) {
            board.takeBack();
            found = true;
            break; }}
    String message = null;
    if (!found) {
        if (board.inCheck(board.side)) {
            if (board.side == LIGHT)
                message = "0 - 1 Black mates";
            else
                message = "1 - 0 White mates";
        } else
            message = "0 - 0 Stalemate";
    } else if (board.reps() == 3)
        message = "1/2 - 1/2 Draw by repetition";
    else if (board.fifty >= 100)
        message = "1/2 - 1/2 Draw by fifty move rule";
    if (message != null) {showStatus(message);
        searcher.stopThinking();
        setChoice4NewGame();
        return true; }
    if (board.inCheck(board.side)) showStatus("Check!");
    return false; }
private void choice4NewGame() {
    AlertDialog.Builder alert1 = new
    AlertDialog.Builder(this).setTitle("Permainan
    Selesai").setMessage("Apakah anda ingin memulai permainan
    baru?").setPositiveButton("Ya",
    new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int whichButton) {
        playNewGame();}
    }).setNegativeButton("Tidak", new
    DialogInterface.OnClickListener() {public void
    onClick(DialogInterface dialog, int whichButton) {}});
    alert1.show();}

```

## 4.2 Pengujian Program

Untuk melakukan pengujian program, kita perlu beberapa hal yang harus dilakukan secara bertahap. Mulai dari penginstalan *software* yang dibutuhkan, serta cara menjalankan program dengan baik.

### a. Penginstalan *Software* yang Dibutuhkan

Sebelum melakukan pengujian program maka kita perlu menginstal beberapa *software* yang dibutuhkan. *software* yang dibutuhkan antara lain:

1. Microsoft Windows Xp/Vista/7 sebagai sistem operasi.
2. Eclipse Indigo Version: 3.7.1 sebagai editor yang digunakan untuk menulis bahasa program JAVA.
3. JDK 1.7.0\_02, JRE 7
4. *Handphone* dengan OS Android *Gingerbread*

Setelah *software* yang dibutuhkan sudah ada, yang perlu dilakukan adalah mengcompile program dan menjalankannya di emulator android. Apabila sudah selesai dijalankan maka installer program android berada pada folder bin di dalam project program dengan ekstensi *catur.apk* dan bisa langsung diinstal dengan *handphone* yang bersistem operasi android *gingerbread*

### b. Cara Menjalankan Program Catur

Klik icon program aplikasi permainan catur, pada saat pertama kali aplikasi dijalankan, maka akan muncul *form* menu. Untuk memulai permainan *user* bisa memilih mulai permainan baru dengan konfigurasi *default*. Untuk memindahkan bidak, *user* hanya perlu mengklik bidak yang ingin dipindahkan dan mengklik kotak tujuan dari bidak tersebut. Permainan akan berakhir ketika salah satu pemain berada dalam kondisi skakmat atau ketika *user* menutup aplikasi permainan.

**c. Uji Coba Desain Interface, Kemudahan Penggunaan, Hiburan**

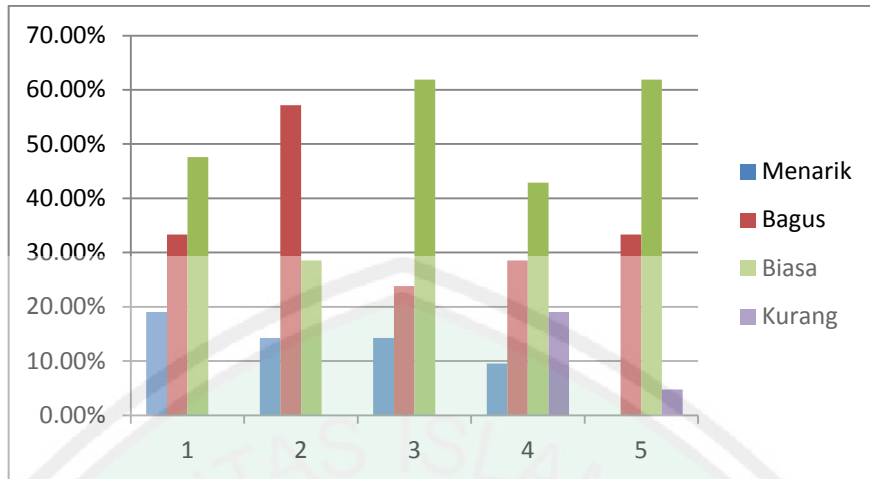
Uji coba dilakukan dengan membagikan angket kepada 21 orang responden. Dan aplikasi dijalankan pada smartphone samsung galaxy young yang mempunyai sistem operasi android 2.3.6 (Gingerbread). Dalam pengujian desain interface responden di mintai pendapat mengenai tampilan atau desain, kemudahan dalam penggunaan, dan hiburan dari aplikasi permainan catur.

Kriteria yang digunakan pada bobot pengujian desain interface adalah sangat menarik, menarik, biasa, dan kurang. Kemudian kriteria yang digunakan pada bobot pengujian kemudahan penggunaan adalah sangat mudah, mudah, susah, dan sangat susah. Dan kriteria yang digunakan pada bobot pengujian hiburan adalah sangat terhibur, terhibur, biasa, dan bosan.

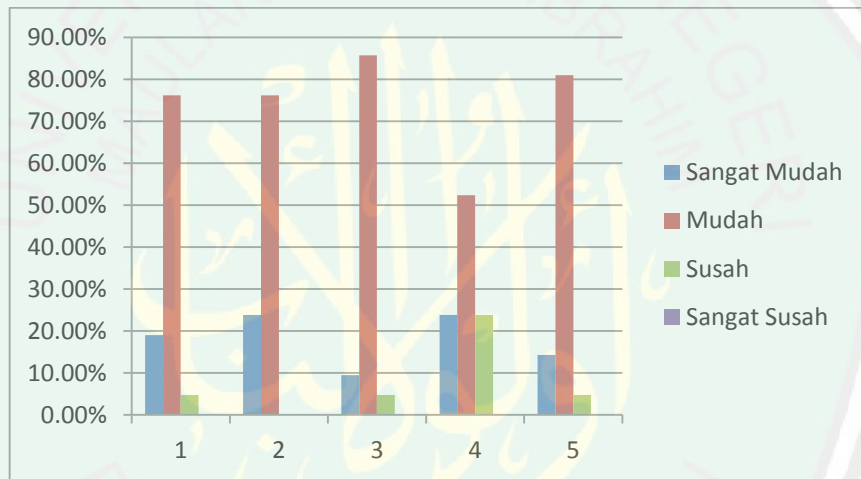
Berdasarkan data yang diperoleh untuk atribut pada item A.1 Desain interface jumlah responden yang mengatakan menarik 11,43%, bagus 35,42%, biasa 48,57%, dan kurang 4,76%. Kemudian berdasarkan data yang diperoleh untuk atribut pada item A.2 Kemudahan Penggunaan jumlah responden yang mengatakan sangat mudah 18,09%, mudah 74,28%, susah 7, 62%, dan sangat susah 0%.dan berdasarkan data yang diperoleh untuk atribut pada item A.3 Hiburan jumlah responden yang mengatakan sangat terhibur 14, 28%, terhibur 64,28%, biasa 21,43%, bosan 0%. Kemudian pada atribut item A.4 Level Permainan masuk dalam uji coba sistem. Tabel 4.1 merupakan tabel hasil uji coba quisioner game catur.

Tabel 4.1 Hasil Uji Coba Desain interface, kemudahan penggunaan, hiburan

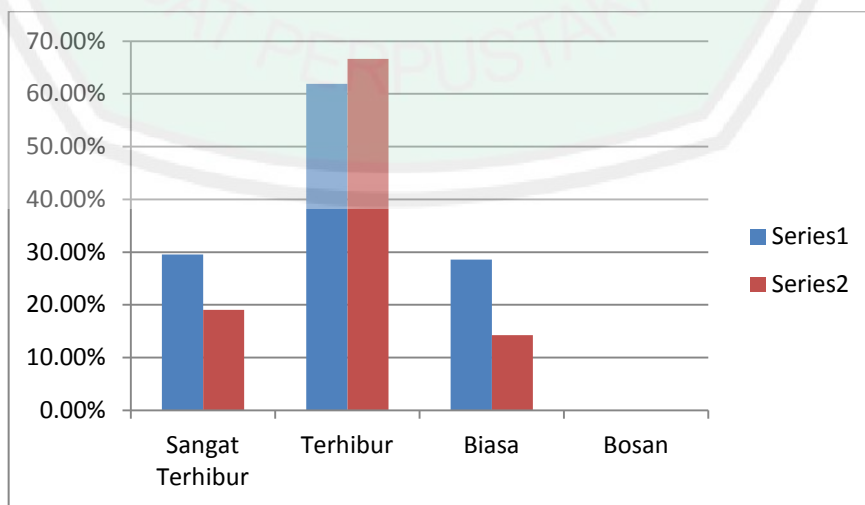
GAME CATUR		TINGKAT KEPUASAN			
No.	ATRIBUT/PERTANYAAN				
<b>A.1</b>	<b>Desain Interface</b>	<b>Menarik</b>	<b>Bagus</b>	<b>Biasa</b>	<b>Kurang</b>
1.	Bagaimanakah keseluruhan tampilan dari game catur	19,05%	33,33%	47,62%	0%
2.	Bagaimanakah tampilan dari menu game catur	14,28%	57,14%	28,57%	0%
3.	Bagaimanakah tampilan papan permainan dari game catur	14,28%	23,81%	61,90%	0%
4.	Bagaimanakah tampilan bidak – bidak yang ada pada permainan catur	9,52%	28,57%	42,86%	19,05%
5.	Bagaimanakah tampilan menu cara bermain dalam permainan catur	0%	33,33%	61,90%	4,76%
<b>A.2</b>	<b>Kemudahan Penggunaan (Easy Of Use)</b>	<b>Sangat Mudah</b>	<b>Mudah</b>	<b>Susah</b>	<b>Sangat Susah</b>
6.	Kemudahan dalam menggunakan game catur secara keseluruhan	19,05%	76,19%	4,76%	0%
7.	Kemudahan dalam mengakses menu game catur	23,81%	76,19%	0%	0%
8.	Mudah dalam memindahkan bidak dipapan permainan catur	9,52%	85,71%	4,76%	0%
9.	Mudah dalam memahami pola permainan catur	23,81%	52,38%	23,81%	0%
10.	Memudahkan dengan adanya menu cara bermain	14,28%	80,95%	4,76%	0%
<b>A.3</b>	<b>Hiburan (Entertainment)</b>	<b>Sangat Terhibur</b>	<b>Terhibur</b>	<b>Biasa</b>	<b>Bosan</b>
11.	Merasa terhibur dengan bermain game catur	29,52%	61,90%	28,57%	0%
12.	Dengan adanya musik sebagai pengiring saat bermain game catur	19,05%	66,66%	14,28%	0%
<b>A.4</b>	<b>Level Permainan</b>	<b>Pemula</b>	<b>Menengah</b>	<b>Lanjut</b>	<b>Master</b>
13.	Lama komputer dalam memindahkan bidak pada level? (kedalaman pencarian algoritma)	<input type="checkbox"/> 4 detik <input type="checkbox"/> 5 detik	<input type="checkbox"/> 9 detik <input type="checkbox"/> 10 detik	<input type="checkbox"/> 29 detik <input type="checkbox"/> 30 detik	<input type="checkbox"/> 59 detik <input type="checkbox"/> 60 detik



Gambar 4.9 Grafik Dari Pertanyaan A.1



Gambar 4.10 Grafik Pertanyaan A.2



Gambar 4.11 Grafik Pertanyaan A.3

#### d. Uji Coba Sistem

Uji coba dilakukan pada smartphone samsung galaxy young yang mempunyai sistem operasi android 2.3.6(Gingerbread). Dalam permainan catur ini terdapat 4 level yaitu : level pemula, menengah, lanjut, master. Uji coba dilakukan untuk membandingkan kecepatan jalannya aplikasi setiap level saat komputer memindahkan bidak dengan uji coba 10x pengujian dengan langkah yang berbeda. Ujicoba dilakukan dengan menjalankan 10 langkah awal dalam permainan catur, satuan angka dalam hasil permainan adalah detik. Hasilnya adalah sebagai berikut :

Tabel 4.2 Hasil Uji Coba Sistem

no	Level	Kecepatan Komputer Memindahkan Bidak										
		Percobaan ke										
		1	2	3	4	5	6	7	8	9	10	
1	Level Pemula	Langkah 1	4	4	4	4	4	5	4	5	5	5
		Langkah 2	5	4	5	4	4	5	5	5	5	5
		Langkah 3	5	4	5	5	5	5	5	4	4	5
		Langkah 4	4	5	5	5	5	4	5	5	4	4
		Langkah 5	5	5	4	4	5	4	4	5	4	5
		Langkah 6	4	5	5	5	5	4	5	5	4	4
		Langkah 7	4	4	5	5	5	4	5	5	4	5
		Langkah 8	4	5	4	4	4	4	4	5	5	5
		Langkah 9	5	4	5	5	5	5	5	5	5	4
		Langkah 10	4	5	5	4	5	5	5	5	4	5
		4 - 5 detik										
2	Level Menengah	Percobaan Ke										
		1	2	3	4	5	6	7	8	9	10	
		Langkah 1	9	8	9	8	9	8	8	8	8	8
		Langkah 2	8	8	10	9	10	8	9	8	8	9
		Langkah 3	8	8	8	8	8	9	8	8	8	9
		Langkah 4	8	9	8	9	8	9	8	9	8	9
		Langkah 5	9	10	9	10	9	10	9	10	9	10
		Langkah 6	10	9	8	9	8	9	8	9	8	9
		Langkah 7	10	9	8	9	8	9	8	9	8	10
		Langkah 8	9	10	9	10	9	10	9	10	9	9
		Langkah 9	9	10	9	10	9	10	9	10	9	10
Langkah 10	9	10	9	10	9	10	9	10	9	9		

		8 - 10 detik										
3	Level Lanjut	Percobaan Ke										
		1	2	3	4	5	6	7	8	9	10	
		Langkah 1	30	29	30	30	29	29	29	29	30	30
		Langkah 2	30	30	29	30	29	29	29	30	29	29
		Langkah 3	29	30	29	29	29	30	29	29	29	30
		Langkah 4	29	29	29	30	30	30	29	30	30	30
		Langkah 5	29	30	29	30	29	29	29	30	29	29
		Langkah 6	30	30	29	29	29	30	29	29	29	30
		Langkah 7	30	29	29	30	30	30	29	30	30	30
		Langkah 8	29	30	29	30	29	29	29	30	29	29
		Langkah 9	30	30	29	29	29	30	29	29	29	30
Langkah 10	30	29	29	30	30	30	29	30	30	30		
		29 - 30 detik										
4	Level Master	Percobaan Ke										
		1	2	3	4	5	6	7	8	9	10	
		Langkah 1	60	59	60	60	59	60	60	59	59	60
		Langkah 2	59	60	60	60	59	60	60	59	60	60
		Langkah 3	60	59	59	60	59	60	60	59	60	60
		Langkah 4	60	59	60	60	59	60	60	59	59	60
		Langkah 5	59	59	60	59	59	59	59	60	59	60
		Langkah 6	60	59	60	59	60	60	59	59	59	60
		Langkah 7	59	59	59	59	60	60	59	59	59	59
		Langkah 8	60	59	60	59	59	60	59	59	60	60
		Langkah 9	60	60	60	60	60	59	59	59	59	59
Langkah 10	60	59	60	59	60	60	60	59	60	60		
		59 - 60 detik										

### e. Implementasi Metode *Forward Chaining dan Pruning*

Implementasi ini merupakan implementasi metode yang digunakan untuk membuat aplikasi ini, yaitu Metode *Forward Chaining dan Pruning*.

Berikut ini merupakan source code implementasi dari metode *forward chaining dan pruning* pada permainan catur.

```
int search(int alpha, int beta, int depth) throws
StopSearchingException {
    if (depth == 0)
        return quiesce(alpha, beta);
    if (beta - alpha != 1) {
        pvLength[ply] = ply;
    }
}
```

```

if ((ply > 0) && (board.reps() > 0))
    return 0;
if (ply >= MAX_PLY - 1)
    return board.eval();
boolean check = board.inCheck(board.side);
if (check)
    ++depth;
List validMoves = board.gen();
if (followPV)
    sortPV(validMoves);

Collections.sort(validMoves);
boolean foundMove = false;
Iterator i = validMoves.iterator();
int a = alpha;
int b = beta;
boolean first = true;
while (i.hasNext()) {
    if (!board.makeMove(m))
        continue;
    ++ply;
    ++nodes;
    if ((nodes & 1023) == 0)
        checkup();
    foundMove = true;
    int x = -search(-b, -a, depth - 1);
    boolean betterMove = false;
    if ((x > a) && (x < beta) && (!first)) {
        a = -search(-beta, -x, depth - 1);
        if (a >= x)
            betterMove = true; }
    board.takeBack();--ply;
    if (x > a) {
        a = x;
        betterMove = true; }
    if (betterMove) {
        board.history[m.getFrom()][m.getTo()] += depth;
        if (x >= beta)
            return beta;
        pv[ply][ply] = m;
        for (int j = ply + 1; j < pvLength[ply + 1]; ++j)
            pv[ply][j] = pv[ply + 1][j];
        pvLength[ply] = pvLength[ply + 1]; }
    b = a + 1;
    first = false; }
if (!foundMove) {
    if (check)
        return -10000 + ply;
    else
        return 0; }
if (board.fifty >= 100)
    return 0;
return a; }

```



### 4.3 Pembahasan

Dari data hasil uji coba diatas dapat diketahui bahwa ada beberapa faktor yang mempengaruhi optimasi penggunaan aplikasi ini, diantaranya adalah:

- 1) Aplikasi ini dapat dijalankan dengan mudah karena mempunyai tampilan yang cukup jelas.
- 2) Konfigurasi permainan yang dapat diubah – ubah sesuai dengan keinginan *user*, membuat *user* tidak cepat merasa bosan.
- 3) Terdapat fasilitas daftar *history* yang menunjukkan langkah yang telah dilakukan oleh pemain putih dan pemain hitam.
- 4) Waktu berpikir pemain komputer pada level pemula antara 4 – 5 detik, level menengah antara 8 – 10 detik, lanjut antara 29 – 30 detik, dan master antara 59 – 60 detik. Pada uji coba sistem ini waktu berpikir pemain komputer mempengaruhi pencarian langkah terbaik pada permainan catur. Sehingga semakin tinggi level yang digunakan dalam permainan, maka semakin banyak pula waktu permainan komputer yang dibutuhkan untuk mencari langkah terbaik.
- 5) Adanya fasilitas *cara bermain* yang berisi tentang penjelasan dasar permainan catur yang dapat digunakan oleh pengguna awam untuk bermain catur.

Dalam game catur ini juga dapat dimanfaatkan sebagai media pembelajaran yang lebih inovatif dan interkatif karena dengan bermain catur pemain dapat merangsang perhatian dan konsentrasi, penilaian dan pengendalian, imajinasi dan antisipasi memori, keinginan untuk menang, kesabaran dan pengendalian diri, semangat dalam membuat keputusan dan

keberanian, logika matematika, berpikir analitik dan sintetik, kreativitas, kecerdasan, studi metode organisasi.

وَأَتَّبِعْ فِي مَآءِ اتِّلَكَ اللَّهُ الدَّارَ الْآخِرَةَ ۖ وَلَا تَنْسَ نَصِيبَكَ مِنَ الدُّنْيَا ۗ  
وَأَحْسِنَ كَمَا أَحْسَنَ اللَّهُ إِلَيْكَ ۖ وَلَا تَبْغِ الْفَسَادَ فِي الْأَرْضِ ۗ إِنَّ اللَّهَ لَا يُحِبُّ  
الْمُفْسِدِينَ ﴿٧٧﴾

Artinya : “Dan carilah pada apa yang telah dianugerahkan Allah kepadamu (kebahagiaan) negeri akhirat, dan janganlah kamu melupakan bahagianmu dari (kenikmatan) duniawi dan berbuat baiklah (kepada orang lain) sebagaimana Allah telah berbuat baik, kepadamu, dan janganlah kamu berbuat kerusakan di (muka) bumi. Sesungguhnya Allah tidak menyukai orang-orang yang berbuat kerusakan.(Q.S. Al Qoshosh : 77).

Sesuai dengan Q.S. Al Qoshosh : 77 dapat dipahami bahwa manusia dituntut untuk selalu berusaha menemukan teknologi dan cara – cara yang bisa memudahkan dan mendukung usaha – usahanya dalam rangka mendapatkan hikmah dari Allah SWT, tak terkecuali dalam rangka pembelajaran. Salah satu teknologi yang harus terus berkembang pesat dan dianggap bisa memberikan pelajaran yang menyenangkan adalah dengan melibatkan *game*. Gabungan antara interaktivitas dengan multimedia yang banyak dimanfaatkan dalam *game* telah terbukti menjadi sesuatu yang sangat adiktif. Potensi seperti ini akan sangat berguna jika dimanfaatkan secara positif, salah satunya adalah *game* yang menggabungkan unsur pembelajaran (*education*) dengan hiburan (*entertainment*).

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil analisa yang dilakukan terhadap permasalahan dan aplikasi yang dikembangkan, maka dapat ditarik kesimpulan bahwa algoritma *forward chaining dan pruning* dapat di implementasikan pada permainan catur sehingga komputer memiliki kemampuan dalam mencari langkah terbaik dengan waktu berpikir pemain komputer pada level pemula antara 4 – 5 detik, level menengah antara 8 – 10 detik, lanjut antara 29 – 30 detik, dan master antara 59 – 60. Semakin tinggi level yang digunakan dalam permainan, maka semakin banyak pula waktu permainan komputer yang dibutuhkan untuk mencari langkah terbaik.

#### 5.2 Saran

Aplikasi catur yang dibuat masih bisa dikembangkan menjadi aplikasi catur yang lebih baik lagi dengan cara :

1. Aplikasi permainan catur dapat dimainkan *multiplayer via bluetooth* atau *online*.
2. Dapat ditambahkan fitur *voice command* dimana pengguna dapat memindahkan bidak dengan menggunakan suara.
3. Algoritma yang digunakan masih bisa dikembangkan dengan algoritma yang lebih baik dimasa mendatang.

## DAFTAR PUSTAKA

- Abt, Clark C. 1970. *Serious Games*. New York: Viking Press.
- Prayudi. 2008. Definisi Computer Game.  
<http://cgrg.wordpress.com/2008/03/12/definisi-computer-game>  
diakses tanggal 12 Januari 2012
- Desiani, Anita, dan Arhami, Muhammad., 2005, *Konsep Kecerdasan Buatan* ,  
penerbit andi, yogyakarta.
- Durkin, John. “*Expert Systems Design and Development*”, Prentice- Hall, 1994.
- Merritt, Dennis. “*Building Expert Systems in Prolog*”, Springer, 1989.
- Wilson, Edward O. “*Consilience: The Unity Of Knowledge*”, Knopf, 1998.
- Gregorius Agung, 2005. *Teknik Membuat Game Dengan Flash Mx 2004*,  
Yogyakarta : Elex Media Komputindo.
- Nugroho, Adi. 2005. *Analisis Dan Perancangan Sistem Informasi Dengan  
Metodologi Beroirentasi Objek*. Bandung : Informatika Bandung.
- Wibawanto, Wandah, 2006, *Bermain Membuat Game Dengan Macromedia Flash*,  
Yogyakarta: Andi Offset.
- Jasson. 2009. *Role Playing Game (RPG) Maker*. Yogyakarta: Andi Offset.
- Gamedevid. 2007. Pengantar Artificial Intelligence dalam Game.  
[http://wiki.gamedevid.org/wiki/Pengantar\\_Artificial\\_Intelligence\\_dalam\\_Game](http://wiki.gamedevid.org/wiki/Pengantar_Artificial_Intelligence_dalam_Game)  
diakses tanggal 12 Januari 2012.
- Administrator. 2011. Cara Mudah Belajar Metode Forward Chaining.  
[http://contohprogramsourcecodemetodealgoritma.com/2011/10/07/cara-  
mudah-belajar-metode-forward-chaining-runut-maju-contoh-program-source-  
code/](http://contohprogramsourcecodemetodealgoritma.com/2011/10/07/cara-mudah-belajar-metode-forward-chaining-runut-maju-contoh-program-source-code/) diakses tanggal 13 Januari 2012
- Administrator. 2011. *Kenalan Dengan OS Android*.  
<http://edutechnolife.com/kenalan-dengan-os-andoid-yuk/>  
diakses tanggal 13 Januari 2012
- Retno Ayu. 2010. *Pengertian Game*.  
[http://retno-ayu-sp.blogspot.com/2010/03/pengertian-game\\_23.html](http://retno-ayu-sp.blogspot.com/2010/03/pengertian-game_23.html)  
diakses 13 Januari 2012
- Burd, Barry, 2005. *Beginning Programming With Java For Dummies*, Second  
Edition, Indianapolis, Indiana : Wiley Publishing, Inc.

Holder, Wayne, 1998. *Java Game Programming*, USA : IDG Books Worldwide, Inc.

Laramee, Francois-Dominic, 2001. *Chess Programming*.  
<http://www.gamedev.net/reference/programming/features/chess1/>

Lowe, Doug, 2005. *Java All In One Desk Reference For Dummies*, Indianapolis, Indiana, Wiley Publishing, Inc.

Siswanto, 2005. *Kecerdasan Tiruan*, Cetakan Pertama, Yogyakarta : Graha Ilmu.

T.Heineman, George, 2009. *Algorithms In A Nutshell*, USA : O'Reilly Media, Inc.

Arroffie, Ahmad Syah, 2011. <http://www.koniriau.or.id/cabor/catur.html>

Diakses tanggal 27 Januari 2013



Tanggal Pengisian Kuesioner	:	
Nama Responden	:	
Jenis Kelamin	:	
Pendidikan	:	
Umur	:	

No.	GAME CATUR ATRIBUT/PERTANYAAN	TINGKAT KEPUASAN			
		Menarik	Bagus	Biasa	Kurang
<b>A.1</b>	<b>Desain Interface</b>				
1.	Bagaimanakah keseluruhan tampilan dari game catur				
2.	Bagaimanakah tampilan dari menu game catur				
3.	Bagaimanakah tampilan papan permainan dari game catur				
4.	Bagaimanakah tampilan bidak – bidak yang ada pada permainan catur				
5.	Bagaimanakah tampilan menu cara bermain dalam permainan catur				
<b>A.2</b>	<b>Kemudahan Penggunaan (Easy Of Use)</b>	<b>Sangat Mudah</b>	<b>Mudah</b>	<b>Susah</b>	<b>Sangat Susah</b>
6.	Kemudahan dalam menggunakan game catur secara keseluruhan				
7.	Kemudahan dalam mengakses menu game catur				
8.	Mudah dalam memindahkan bidak dipapan permainan catur				
9.	Mudah dalam memahami pola permainan catur				
10.	Memudahkan dengan adanya menu cara bermain				
<b>A.3</b>	<b>Hiburan (Entertainment)</b>	<b>Sangat Terhibur</b>	<b>Terhibur</b>	<b>Biasa</b>	<b>Bosan</b>
11.	Merasa terhibur dengan bermain game catur				
12.	Dengan adanya musik sebagai pengiring saat bermain game catur				
<b>A.4</b>	<b>Level Permainan</b>	<b>Pemula</b>	<b>Menengah</b>	<b>Lanjut</b>	<b>Master</b>
13.	Lama komputer dalam memindahkan bidak pada level? (kedalaman pencarian algoritma)	<input type="checkbox"/> 4 detik <input type="checkbox"/> 5 detik	<input type="checkbox"/> 9 detik <input type="checkbox"/> 10 detik	<input type="checkbox"/> 29 detik <input type="checkbox"/> 30 detik	<input type="checkbox"/> 59 detik <input type="checkbox"/> 60 detik

Note: No 1 – 12 beri tanda centang (√) pada salah satu kolom  
No 13 beri tanda centang salah satu pilihan waktu pada setiap kolom

Responden

TTD