IJCCS, Vol. 1 No. 2, June, 2007: 105 - 114

Artificial Intelligence on Computer Based Chess Game: An Implementation of Alpha-Beta-Cutoff Search Method

Albert V. Dian Sano[1] dan Retantyo Wardoyo[2]
[1]Computer Science Study Program, Graduate Program of Gadjah Mada University, Yogyakarta
[2]Computer Science Study Program, Gadjah Mada University, Yogyakarta
email: [2]rw@ugm.ac.id

Abstract

A chess program usually consists of three main parts, that is, a move generator to generate all legal moves, an evaluation function to evaluate each move, and a search function to select the best move. The search function is the core of thinking process. The goal of this research is to implement the alpha beta cutoff as a search method. This method is derived from minimax search method and is more optimal than the minimax search method.

In minimax, all nodes is searched and compared one by one to get the best value. On the other hand, the alpha beta cutof methd only searches nodes which make contribution to the previous value and cuts off nodes which are not useful. It means that the alpha beta method will not search and compare all nodes. The new node will be better than the previous one and replace the old value with the new one. This will make the alpha beta method requires smaller search time.

The proposed method is tested by doing a series of matches between humans and a computer. The results show that the computer has ability to think well and performs a good artificial intelligence though it is very open to be modified and more optimized.

Keywords: *move generator function, evaluation function, search function, minimax, alpha beta cutoff*

1. Pengantar

Dalam AI (*Artificial Intelligence*), game playing merupakan salah satu peletak dasar dalam penciptaan kecerdasan buatan pada generasi awal AI. Sejak semula beberapa pakar AI sangat tertarik dengan karakteristik game, karena game playing merupakan salah satu bentuk simulasi kehidupan nyata, mampu memberian tantangan dan stimulasi intelektual. Banyak game dalam AI juga merupakan contoh problem pencarian yang sangat kompleks dimana proses pencarian harus menerapkan metode heuristic. Permainan catur misalnya mempunyai node $\approx 35^{100}$ dan pada go mempunyai node $\approx 250^{200}$ dalam pohon pencarian [1]. Dalam proses pencarian seperti itu tentu harus diterapkan proses pencarian heuristic untuk membatasi proses pencarian. Dalam penelitian ini, penulis akan mengambil sampel permainan catur untuk mengimplementasikan proses pencarian dengan metode apha-beta cutoff.

Metode pencarian alpha-beta cutoff pada dasarnya adalah bentuk optimasi dari metode pencarian minimax. Alpha mengacu pada pemain pertama sedangkan beta mengacu pada pemain kedua. Sedangkan cutoff mengacu pada pemangkasan node-node yang tidak memberikan andil peningkatan nilai dalam proses pencarian. Dengan pemangkasan tersebut akan didapatkan efisiensi yang lebih baik dalam proses pencarian.

Kode algoritma pencarian tersebut secara garis besar adalah sebagai berikut [2]:

B pada mulanya diset sebagai skor awal.

```
evaluasimin(u, B) // u is a min
node
{
  alpha = +infinity;
  if u = leaf return the score;
  else
```

```
    for all children v of u
    {
      Val = evaluatemax(v, B);
      alpha = Min{alpha, Val};
      if alpha <= beta then exit
      loop;
    }
    return alpha;
}

evaluasimax(u, B)    // u is max
node
{
  alpha = -infinity;
  if u = leaf return the score;
  else
      for all children v of u
      {
       Val = evaluasimin(v, B);
       alpha = Min{alpha, Val};
       if alpha >= beta then
       exit loop;
      }
  return alpha;
}
```

2. Cara Penelitian

Langkah-langkah dalam penelitian ini secara garis besar adalah sebagai berikut:

1. Studi pustaka dengan bahan dari buku-buku maupun dari internet yang berkaitan dengan teknik dasar pencarian dalam AI terutama mengenai pencarian Alpha-Beta Cutoff.
2. Merancang sistem yang akan dibangun, dalam ini menggunakan software pemodelan UML.
3. Pembuatan program dengan menggunakan bahasa pemrograman Java.
4. Pengujian / Evaluasi. Pengujian disini meliputi dua hal, yang pertama adalah menguji apakah langkah-langkah sudah sesuai dengan aturan permainan catur. Yang kedua adalah pengujian terhadap kecerdasan komputer dalam bermain catur, dalam hal ini melawan manusia.

2.1. Perancangan Sistem

Dalam mendesain sistem untuk permainan catur ini pada dasarnya bisa diklasifikasi menjadi dua bagian, yaitu perancangan pada sisi non-AI dan pada sisi AI. Perancangan sisi non-AI adalah bagaimana mendesain penyajian sarana dan aturan yang dibutuhkan dalam permainan catur, seperti penyajian papan catur, buah tiap pemain, posisi awal permainan, langkah-langkah masing-masing buah catur, aturan-aturan yang berlaku pada permainan catur, aturan melangkah tiap-tiap buah dan bagaimana aturan menangkap buah lawan, rokade, promosi, dan sebagainya.

Sedangkan perancangan sisi AI adalah mendesain untuk peng-implementasian mengenai bagaimana metode berpikir komputer. Kedua bagian tersebut akan disatukan dalam sistem permainan catur ini.

Inti dari perancangan ini meliputi perancangan penyajian papan catur, penyajian buah catur, penyajian langkah-langkah tiap-tiap buah, penyajian proses berpkir komputer, dan semuanya akan disajikan dengan user interface yang user-friendly.

2.2. Penyajian Papan Catur

Papan catur terdiri dari 8 x 8 kotak. Dalam perancangan ini akan digunakan array dua dimensi [8][8]. Masing-masing kotak akan merupakan indeks array tersebut dan akan divisualisasikan dalam bentuk koordinat catur, yaitu sumbu y berupa angka 1,2,3,4,5,6,7,dan 8 ke arah atas, sedangkan sumbu x berupa huruf A,B,C,D,E,F,G,dan H ke arah kanan.

2.3. Penyajian Buah Catur

Buah Catur dalam permainan catur ada 32 buah, namun pada dasarnya jenis buah catur hanya ada enam jenis, yaitu Raja, Menteri, Benteng, Gajah, Kuda, dan Bidak. Tiap-tiap jenis buah catur bisa disajikan dalam sebuah konstanta integer misalnya BIDAK = 0, KUDA = 1, GAJAH = 2, BENTENG = 3, MENTERI = 4, RAJA = 5.

2.4. Penyajian Langkah-langkah Buah Catur

Seperti telah disebutkan di atas, meskipun buah catur ada 32 buah, namun pada dasarnya jenis buah catur hanya ada enam jenis, yaitu Raja, Menteri, Benteng, Gajah, Kuda, dan Bidak. Ketika akan melangkah hal pertama yang harus dilakukan adalah menghasilkan daftar langkah dan melakukan inspeksi terhadap semua langkah yang legal, dimana semua langkah yang legal ini disimpan dalam vector. Vector adalah fasilitas semacam array dinamis yang disediakan dalam bahasa pemrograman java. Index pada vector bisa bertambah secara otomatis bila elemennya bertambah, sehingga tidak perlu dikawatirkan terjadi index overflow. Masing-masing jenis buah tersebut mempunyai aturan sendiri dalam melangkah dan menagkap buah lawan. Untuk menghasilkan langkah yang legal maka langkah tiap-tiap buah catur tersebut harus menaati aturan yang berlaku. Untuk menyajikan aturan melangkah masing-masing buah tersebut maka tiap-tiap aturan langkah buah catur tersebut akan disimpan dalam metode atau operasi sendiri-sendiri.

Selain harus mengikuti aturan melangkah, tiap-tiap buah catur ketika melangkah harus tetap berada di dalam papan. Karena itu, dalam setiap langkah harus mengalami pengecekan kondisi: if ((indeks_x <= 7) && (indeks_x>=0) && (indeks_y <= 7) && (indeks_y>=0))

Selain pengecekan kondisi di atas, masih ada beberapa pengecekan tambahan untuk langkah bidak dan raja. Untuk langkah bidak, pertama kali boleh melangkah maju satu kotak atau dua kotak. Ini khusus hanya untuk langkah pertama kali saja. Karena itu kondisi seperti ini harus dicek terlebih dahulu, apakah bidak pernah melangkah sebelumnya atau belum. Untuk langkah raja juga mengalami pengcekan ketika melakukan rokade. Rokade merupakan salah satu langkah rumit karena ada beberapa syarat yang harus terpenuhi, yaitu:

1. Kotak diantara raja dan benteng harus kosong dan tidak dalam keadaan terancam.
2. Raja tidak boleh dalam keadaan terancam baik sebelum maupun sesudah rokade.

Semua kondisi di atas harus dicek secara eksplisit untuk menghasilkan langkah yang legal. Kondisi lain yang harus terpenuhi adalah bahwa semua langkah yang membuat raja menjadi terancam tidak diperbolehkan. Ini merupakan inspeksi tahap kedua.

Jadi setelah daftar langkah legal dibuat, beberapa langkah yang tidak memenuhi kondisi yang kedua harus dihilangkan. Pengecekan ini bisa dilakukan dengan memeriksa semua kotak yang berpotensi menyerang kotak dimana raja berada.

2.5. Proses Bepikir Komputer

Bagian yang paling menarik pada program catur adalah kemampuannya untuk berpikir, dan ini merupakan proses komputasi yang cukup rumit. Bagian ini merupakan bagian AI dimana metode alpha-beta cutoff akan diterapkan. Proses berpikir dimulai dengan membuat struktur pohon pencarian dan membuat penulusuran, kemudian memberi nilai supaya bisa dipertimbangkan langkah mana yang terbaik.

Pada proses berpikir tahap pertama komputer akan memilih secara acak langkah legal yang dihasilkan dari daftar langkah yang ada. Sehingga komputer nampaknya akan melangkah secara asal dan tanpa tujuan yang jelas. Proses tahap kedua yaitu komputer akan membuat daftar langkah yang menangkap buah lawan. Prioritas penangkapan buah lawan adalah berdasarkan skor buah yang ditangkap. Skor masing-masing buah disini adalah berdasarkan perhitungan bahwa bidak bernilai 100, kuda bernilai 300, gadjah 325, benteng 500, menteri 900, raja 10000. Pemberian skor disini sangat terbuka untuk mengalami modifikasi. Pemberian skor yang ideal dengan perbandingan yang akurat antara masing-masing buah akan meningkatkan kecerdasan komputer. Konsekwensi dari proses ini adalah bahwa komputer akan selalu berusaha menangkap buah lawan sedapat mungkin, tanpa peduli dengan resikonya. Bila tidak ada buah lawan yang bisa didapatkan

maka proses berpikir akan seperti pada proses sebelumnya dan tanpa tujuan. Proses berikutnya adalah bahwa komputer memulai proses pencarian dengan dua perulangan berkalang. Perulangan yang pertama akan memeriksa semua kemungkinan langkah pertama, mencatat skor buah yang dimakan, kemudian membuat langkah, dan mengupdate posisi papan. Perulangan untuk kalang yang di dalam akan mencari semua langkah yang dihasilkan oleh lawan, mencatat skor buah yang dimakannya dan mengambil skor yang tertinggi. Nilai pada langkah pertama merupakan skor nilai buah yang dimakan pada langkah pertama dikurangi dengan skor nilai buah yang diambil pada langkah kedua.

2.6. Penyajian Kelas

Pada level implementasi akan digunakan bahasa pemrograman java, karena itu dalam perancangan ini akan menerapkan desain berbasis kelas. Dalam perancangan ini direncanakan terdiri dari satu interface (kelas yang hanya berisi definisi konstanta-konstanta) dan tujuh kelas. Interface dan kelas-kelas tersebut adalah sebagai berikut, Interface DaftarKonstanta, Kelas Koordinat, Kelas BuahCatur, Kelas Langkah, Kelas PapanCatur, Kelas VisualPapan, Kelas KomponenPapan, Kelas Catur.

Dalam desain sistem permainan catur ini digunakan pemodelan dengan menggunakan UML (Unified Modelling Language). Diagram yang digunakan cukup dengan menggunakan diagram kelas karena sistem yang dibangun bukanlah sistem yang besar dan kompleks sehingga kesederhanaan dalam pemodelan akan tetap dipertahankan.

Pada sub-bab berikut akan mendeskripsikan hal-hal pokok mengenai isi dan diagram kelas dari masing-masing kelas di atas.

2.6.1. Interface DaftarKonstanta

Inti dari interface ini adalah menyajikan buah-buah catur dan bobot nilai masing-masing buah tersebut. Buah-buah catur tersebut disajikan dalam bentuk konstanta integer dan bobot nilai masing-masing buah dalam bentuk array integer. Meskipun dalam permainan catur terdiri dari 32 buah catur, namun pada dasarnya jenis buah-buah catur tersebut hanya ada enam saja, yaitu Bidak, Kuda, Gajah, Benteng, Menteri, dan Raja. Tiap-tiap buah tersebut disajikan dalam bentuk konstanta integer, secara berturut-turut masing-masing adalah 0, 1, 2, 3, 4, dan 5.

Penyajian berikutnya adalah pemberian bobot nilai masing-masing buah dalam bentuk array integer, misalnya untuk bidak nilainya 100, untuk kuda 300, gajah 325, benteng 500, menteri 900, sedangkan raja nilainya harus besar sekali misalnya 100000 (Marsland).

2.6.2. Kelas Koordinat

Kelas ini bertujuan untuk menyajikan kotak-kotak permainan catur dalam bentuk koordinat x dan y. Kotak-kotak tersebut direncanakan dalam bentuk array dua dimensi 8 x 8, dan masing-masing kotak akan disajikan dalam bentuk koordinat x dan y.

2.6.3. Kelas BuahCatur

Inti dari kelas ini adalah menyajikan buah-buah catur dalam bentuk visual (image). Meskipun buah-buah catur sudah disajikan dalam bentuk integer pada interface DaftarKonstanta, namun perlu juga disajikan dalam bentuk visual sebagai kelengkapan user interface permainan. Selain itu pemberian nilai masing-masing buah catur untuk tiap pemain juga didetailkan dalam kelas ini, misalnya untuk putih diberi nilai positif dan hitam diberi nilai negatif.

2.6.4. Kelas Langkah

Tujuan utama dari kelas ini adalah memindahkan buah catur dari kotak asal ke kotak tujuan. Selain itu ada beberapa metode tambahan dalam kelas ini yang berfungsi untuk membatalkan langkah, mencetak koordinat langkah dalam bentuk koordinat papan catur, dan mengecek keadaan remis yang diakibatkan oleh perulangan langkah yang sama sebanyak tiga kali.

2.6.5. Kelas PapanCatur

Kelas ini merupakan jantung dari semua kelas yang digunakan untuk mengembangkan sistem ini. Banyak hal penting akan disajikan dalam kelas ini, salah

satunya adalah posisi awal permainan catur. Selain itu, aturan melangkah dan aturan menangkap buah lawan juga disajikan dalam kelas ini. Masing-masing buah catur akan diimplementasikan dalam satu metode. Khusus untuk bidak dan raja masing-masing disajikan dalam dua metode. Ini karena untuk bidak dan raja mempunyai sedikit perkecualian. Untuk bidak ada perbedaan antara langkah biasa dan langkah untuk menangkap buah lawan. Sedangkan untuk raja ada tambahan khusus untuk melakukan langkah. Metode penting lain yang akan disajikan dalam kelas ini adalah metode untuk mencek keadaan akhir, yaitu skakmat dan remis. Perkiraan nilai untuk pengambilan langkah juga akan disajikan dalam metode di kelas ini. Bagian yang paling menarik dalam kelas ini adalah penyajian metode alfa-beta. Metode inilah yang merupakan implementasi dari proses berpikir komputer dalam mengambil keputusan untuk melangkah.

2.6.6. Kelas VisualPapan

Inti dari kelas ini adalah membuat visualisasi dari papan catur dengan 64 kotak di dalamnya dan event handler untuk melakukan pemilihan buah untuk melangkah.

2.6.7. Kelas KomponenPapan

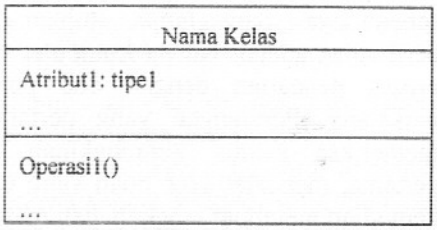
Kelas ini berfungsi untuk mengelola tampilan user interface dan tidak berkaitan dengan sistem kecerdasan buatan permainan catur ini.

2.6.8. Kelas Catur

Kelas ini merupakan kelas utama dimana konstruktornya akan memanggil kelas KomponenPapan di atas. Dalam kelas ini juga diberikan nilai default level proses berpikir komputer, yaitu tiga.

2.6.9. Diagram Kelas UML

Simbol untuk menggambarkan diagram kelas adalah segi empat yang terbagi menjadi tiga bagian. Yang pertama mendefinisikan nama kelas, yang kedua mendefinisikan atribut, properti, atau data, yang ketiga mendefinisikan serangkaian operasi. Secara singkat bentuk diagram kelas adalah seperti Gambar 1.



Gambar 1. Bentuk diagram kelas

Berikutnya akan disajikan diagram kelas uml dan relasi antar kelas yang digunakan. Beberapa simbol relasi yang digunakan dalam desain ini adalah sebagai berikut:

1.  Simbol dengan garis putus-putus dengan ujung segitiga putih. Simbol ini biasa disebut dengan realization yang berfungsi untuk merelasikan antara interface dan kelas.
2.  Simbol dengan garis lurus dengan ujung panah terbuka. Simbol ini biasa disebut dengan association. Kelas yang terasosiasi akan menginstans objek kelas yang lain.
3.  Simbol dengan garis putus-putus dengan ujung panah terbuka. Simbol ini biasa disebut dengan dependency. Kelas yang satu akan mempunyai ketergantungan terhadap kelas yang lain.

Gambar 2 adalah gambar master diagram kelas dan relasi antar kelas tersebut.

3. Implementasi dan Hasil

3.1 Implementasi

Implementasi sebuah program catur pada dasarnya berisi tiga hal pokok: adanya suatu fungsi yang menghasilkan langkah, adanya suatu fungsi yang memberikan evaluasi dengan memberikan skor pada tiap-tiap buah catur, dan adanya suatu fungsi yang bertugas sebagai mesin pencari langkah terbaik. Fungsi ini yang mendasari bagaimana sebenarnya komputer berpikir untuk mendapatkan langkah terbaiknya.

Meskipun pada dasarnya hanya terdiri dari tiga hal pokok, namun pada level implementasi, sebagai konsekuensi untuk menghasilkan suatu produk, akan diperlukan banyak hal yang perlu ditambahkan, misalnya

pembentukan user inteface yang *user-friendly.*

3.1.1. Penghasil Langkah

Langkah masing-masing buah disajikan dalam sebuah metode atau operasi, kecuali untuk bidak dan raja. Semua buah catur mempunyai aturan melangkah dan aturan menangkap buah lawan sama, kecuali bidak, sehingga untuk bidak perlu dibuatkan dua metode, masing-masing untuk melangkah dan untuk menangkap buah lawan. Sementara raja mempunyai perkecualian untuk langkah rokade. Semua langkah akan ditampung dalam bentuk vektor.

3.1.1.1. Langkah Bidak

Untuk menyajikan langkah bidak dibuat dua metode yaitu, metode LangkahBidak yang mengatur langkah bidak dan metode LangkahBidakMakan yang mengatur langkah bidak untuk menangkap lawan.

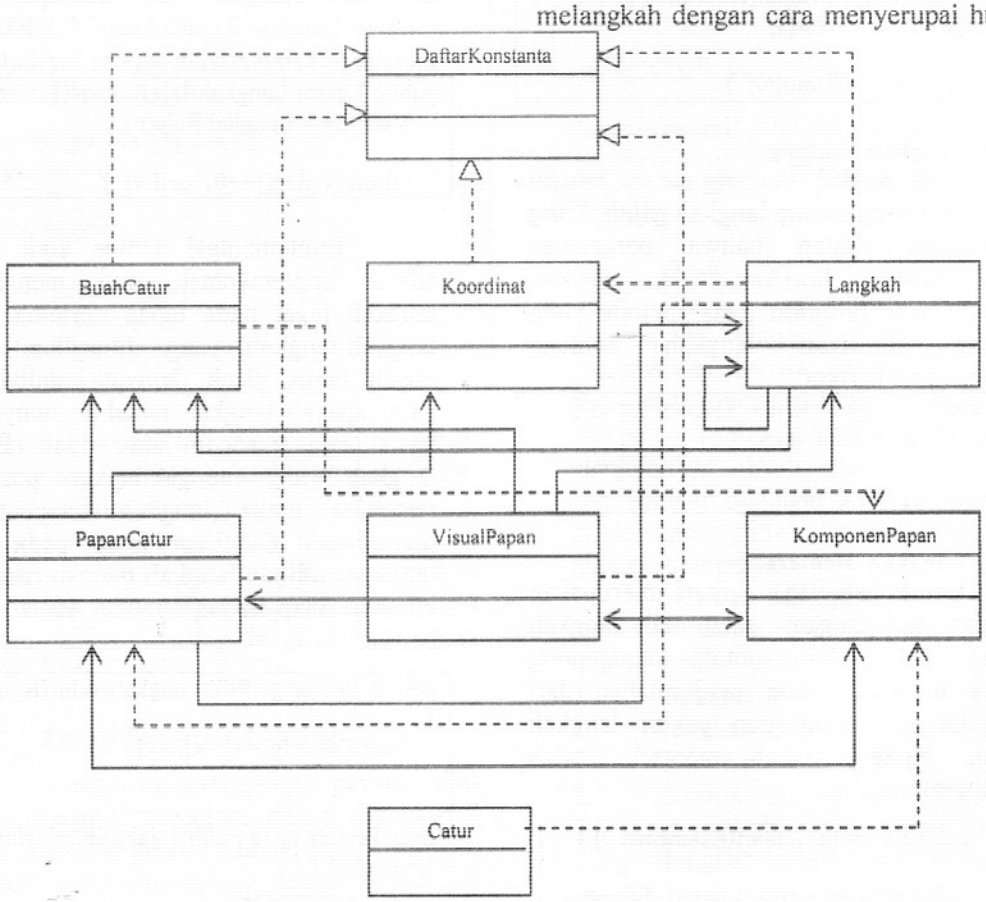
Berikut adalah cuplikan kode dari dua metode tersebut:

```
public Vector LangkahBidak(Koordinat c) {
  Vector v = new Vector();
  ...
  return (v.size()==0)?null:v; }
```

```
public Vector LangkahBidakMakan(Koordinat c)
{
  Vector v = new Vector();
  ...
  return (v.size()==0)?null:v; }
```

3.1.1.2. Langkah Kuda

Langkah kuda merupakan salah satu langkah yang unik. Buah ini merupakan satu-satunya buah yang boleh melangkah dengan cara melompati buah lain. Selain itu buah ini melangkah dengan cara menyerupai huruf L.



Gambar 2. Diagram kelas dan relasi antar kelas

Berikut adalah cuplikan kode dari metode yang mengatur langkah kuda:

```
Public Vector LangkahKuda(Koordinat c) {
  Vector v = new Vector();
  ...
  return (v.size()==0)?null:v; }
```

3.1.1.3. Langkah Gajah

Untuk langkah gajah tidak ada sesuatu yang istimewa. Gajah hanya boleh melangkah serong baik ke kiri maupun ke kanan. Batasan yang ada pada kode ini hanyalah pengecekan bahwa gajah harus tetap berada di dalam papan dan pergeseran langkah hanya berhenti bila di depannya ada buah lain (tidak boleh melompati). C uplikan metode dari kode langkah gajah adalah sebagai berikut:

```
public Vector LangkahGajah(Koordinat c) {
  Vector v = new Vector();
  ...
  return (v.size()==0)?null:v; }
```

3.1.1.4. Langkah Benteng

Pada langkah benteng secara prinsip adalah sama dengan cara langkah gajah. Yang membedakan adalah bahwa pergeseran langkah benteng horizontal dan vertikal. Cuplikan kode program pada metode yang mengimplementasikan langkah benteng adalah sebagai berikut:

```
public Vector LangkahBenteng(Koordinat c) {
  Vector v = new Vector();
  ...
  return (v.size()==0)?null:v; }
```

3.1.1.5. Langkah Menteri

Untuk langkah menteri merupakan gabungan dari langkah gajah dan langkah benteng. Jadi vektor untuk menampung langkah menteri cukup penjumlahan dari vektor langkah gajah dan vektor langkah benteng. Kode langkah menteri adalah sebagai berikut:

```
public Vector LangkahMenteri(Koordinat c) {
  return
jumLangkah(LangkahGajah(c),LangkahBenten
g(c)); }
```

3.1.1.6. Langkah Raja

Raja juga mempunyai dua kategori aturan untuk melangkah seperti halnya pada bidak. Aturan yang pertama adalah bahwa raja boleh melangkah ke segala arah. Aturan ini diimplementasikan dalam metode LangkahRaja.

Cuplikan kode tersebut bisa dilihat seperti berikut:

```
public Vector LangkahRaja(Koordinat c) {
  Vector v = new Vector();
  ...
  return (v.size()==0)?null:v; }
```

Selain itu, raja juga boleh melangkah rokade. Persyaratan untuk melakukan rokade sudah dijelaskan pada bab sebelumnya. Rokade merupakan langkah yang rumit karena banyaknya kondisi yang harus dipenuhi. Langkah ini disimpan dalam metode LangkahRajaRokade. Cuplikan kode metode tersebut adalah seperti berikut:

```
public Vector LangkahRajaRokade(Koordinat c) {
  Vector v = LangkahRaja(c);
  ...
  return (v.size()==0)?null:v; }
```

Implementasi semua kode di atas adalah implementasi untuk menghasilkan langkah legal pada tahap pertama. Namun langkah-langkah yang dihasilkan tersebut masih harus dicek dengan kondisi kedua, yaitu apakah langkah tersebut menyebabkan posisi raja terancam atau tidak. Bila ada langkah yang mengakibatkan posisi raja terancam maka langkah tersebut harus dieliminasi. Cuplikan kode pada metode untuk pemilihan langkah dengan pengecekan kondisi tahap kedua tersebut adalah sebagai berikut:

```
public Vector getPilihLangkahAkhir(Koordinat c)
{
  BuahCatur fg = f[c.x][c.y];
  ...

if(ujiTerserang(fg.putih?rajaPutih:rajaHitam,
!fg.putih)) {
      v.remove(k);
      k--;
    }
    ...
  return v; }
```

Semua kode pada metode-metode di atas adalah kode yang menghasilkan semua kemungkinan langkah yang legal.

3.1.2. Fungsi Evaluasi

Fungsi ini bertugas menganalisa nilai yang didapatkan dari suatu langkah yang diambil. Nilai tertinggi yang didapatkan akan dianggap sebagai langkah terbaik. Fungsi ini diambil dari perhitungan nilai pada masing-masing buah dan berapa nilai buah yang bisa ditangkap oleh lawan. Fungsi ini belum tentu merupakan perhitungan yang terbaik. Sejauh ini belum pernah ada fungsi yang bisa dipastikan menghasilkan perhitungan terbaik. Jadi fungsi disini masih sangat terbuka untuk mengalami modifikasi. Cuplikan kode dari metode yang mengimplementasikan fungsi ini adalah sebagai berikut:

```
public int getSkor() {
  int skor = 0;
      ...
  return skor; }

 public int getNilaiMakan() {
  int skorm = 0;
      ...
  return -skorm; }

public int gettotalNilai() {
  int dc = getSkor();
  int dac = getNilaiMakan();
  return dc*1000+dac; }
```

Fungsi getSkor() bermaksud untuk menghitung skor total masing-masing buah. Fungsi getNilaiMakan() dimaksudkan untuk mengambil skor buah yang bisa ditangkap. Sedangkan fungsi gettotalNilai() merupakan perhitungan yang digunakan sebagai fungsi evaluasi akhir terhadap langkah yang akan dipilih.

3.1.3. Fungsi Pencarian Langkah

Fungsi ini merupakan jantung dari sistem berpikir komputer dalam mengambil keputusan langkah mana yang terbaik. Metode yang diterapkan dalam fungsi ini adalah metode pencarian dengan menggunakan alpha-beta cutoff. Inti dalam penyusunan kode fungsi pencarian ini adalah adanya perulangan yang menggunakan fungsi rekursif. Fungsi rekursif pada dasarnya harus memenuhi dua syarat pokok yaitu adanya bagian yang menyebabkan fungsi tersebut berhenti (terminating point) dan adanya bagian yang berulang.

Sedangkan inti dari metode alpha-beta pada kode ini adalah adanya perulangan dalam mencari node-node namun dengan kondisi bahwa perulangan akan berhenti bila dijumpai node yang tidak lebih baik dari node sebelumnya. Culikan kode metode yang mengimplementasikan fungsi pencarian ini adalah seperti di bawah ini:

```
public int alfaBeta(boolean putih, int alpha, int
beta, int level) {
  if(level==0)
    return gettotalNilai();
  ...
        int est = -alfaBeta(!putih, -beta, -alpha,
level-1);
  ...
  return best; }
```

Fungsi rekursif ditandai dengan kode berikut:

```
        int est = -alfaBeta(!putih, -beta, -alpha,
level-1);
```

3.2 Hasil

Untuk pengujian implementasi kecerdasan buatan pada aplikasi ini diasumsikan bahwa permainan melibatkan dua pemain, pemain pertama adalah manusia dan pemain kedua adalah komputer. Manusia menjalankan buah putih dan komputer menjalankan buah hitam. Pengujian terutama difokuskan pada langkah-langkah yang diambil komputer.

Inti dari pengujian kecerdasan buatan pada aplikasi permainan catur di sini meliputi dua hal pokok, yaitu:

1. Komputer mampu mengambil langkah-langkah yang sesuai dengan aturan permainan catur.
2. Komputer mampu berpikir dan mengambil keputusan mengenai langkah terbaik yang harus dipilihnya sesuai dengan metode alpha-beta cutoff. Pengujian ini merupakan inti pada penelitian ini.

Pengujian akan dilakukan dengan sederetan pertandingan antara manusia melawan komputer dan disertai dengan analisa pada beberapa langkah yang diambil komputer.

Dari semua langkah yang sudah dicoba, semuanya berhasil memenuhi aturan permainan catur. Percobaan pertandingan sudah dicoba beberapa kali dengan hasil bahwa semua langkah berhasil dijalankan dengan baik tanpa ada yang mengalami kesalahan.

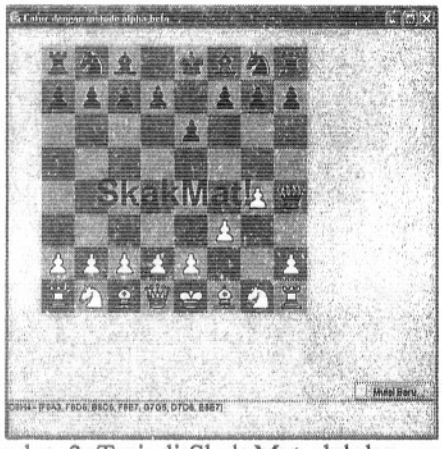
Pengujian berikutnya adalah menguji apakah komputer mampu berpikir dan mengambil langkah terbaik dalam permainan catur tersebut. Jadi komputer tidak akan melangkah secara sembarangan.

3.2.1. Pengujian Pencarian Langkah Menggunakan Metode Alpha-Beta Cutoff

Pengujian pada tahap ini bertujuan untuk membuktikan bahwa komputer melangkah berdasarkan hasil proses berpikirnya (pencarian metoda alpha-beta cut-off). Percobaan pertandingan berikut akan menggunakan tiga level kedalaman berpikir. Kedalaman level tiga ini dipilih karena merupakan level yang ideal untuk digunakan sebagai pengujian ini sebagai konsekwensi kompromi antara kecepatan proses berpikir dan kecerdasan yang dihasilkan oleh proses berpikir tersebut. Bila level kedalaman terlalu rendah, misalnya satu atau dua, komputer akan berpikir dengan cepat, namun sebagai konsekwensinya hasil berpikirnya tidak terlalu baik. Sebaliknya bila level kedalaman berpikirnya terlalu tinggi, misalnya 4, 5, dan seterusnya proses berpikir komputer akan semakin baik namun sebagai konsekwensinya akan memakan waktu yang terlalu lama. Hasil capture gambar dan analisa berikut berdasarkan salah satu percobaan pertandingan yang paling simpel pada kedalaman pencarian dengan level tiga.

3.2.2. Analisis Hasil

Pada pertandingan di atas sengaja dijalankan bidak putih f2-f3 dan g2-g4 untuk menguji apakah komputer mengetahui langkah terbaiknya atau tidak.



Gambar 3. Terjadi Skak Mat oleh komputer dalam empat langkah

Hasilnya, pada langkah keempat komputer mampu menemukan langkah terbaiknya dengan mengakhiri pertandingan dengan skak mat.

Serangkaian pertandingan sudah dilakukan dan komputer mampu menunjukkan kemampuan berpikirnya dengan baik dan tidak melangkah secara sembarangan terutama di permainan tengah. Bahkan beberapa langkah yang dianalisa menunjukkan langkah yang diambil komputer diluar dugaan dan melangkah dengan sangat cerdas.

Pada pengujian tahap kedua tersebut, yaitu kemampuan komputer untuk berpikir dalam mengambil langkah, secara umum berhasil dengan baik. Sebagian besar langkah bisa dikategorikan sebagai langkah yang cerdas. Ada sebagian kecil langkah yang bisa dikategorikan sebagai langkah lemah. Analisa yang bisa ditarik dari langkah seperti ini adalah bahwa level proses pencarian komputer tidak mampu menjangkau problem-problem pada situasi tertentu. Secara singkat dan kasar bisa dikatakan bahwa level kedalaman pencarian habis atau kurang.

4. Kesimpulan dan Saran

4.1 Kesimpulan

Beberapa kesimpulan yang dapat diambil dari pembuatan aplikasi permainan catur dengan mengimplementasikan metode berpikir alpha-beta cutoff adalah sebagai berikut:

1. Alpha beta cutoff berhasil diimplementasikan dengan baik. Aplikasi yang dibuat mampu berpikir dengan baik terutama pada permainan awal dan tengah. Bahkan tidak jarang komputer memenangkan pertandingan dalam jumlah langkah yang tidak terlalu banyak.
2. Langkah-langkah yang diambil oleh komputer mampu memenuhi aturan-aturan permainan catur dengan sangat baik.
3. Fungsi evaluasi masih sangat terbuka untuk dimodifikasi karena belum pernah ditemukan fungsi evaluasi terbaik. Dari berbagai percobaan yang dilakukan, fungsi evaluasi sangat berpengaruh terhadap kecerdasan komputer.

4.2 Saran - saran

Saran-saran ditujukan untuk meningkatkan kemampuan aplikasi ini bagi penelitian sejenis selanjutnya adalah sebagai berikut:

1. Metode alpha-beta cutoff masih bisa dikombinasikan dengan metode yang lain untuk meningkatkan kecerdasan proses berpikir, misalnya dikombinasikan dengan metode best first search.
2. Bisa dimungkinkan untuk melakukan penelitian yang menjadikan level proses pencarian bersifat dinamis.
3. Masih terbuka kemungkinan untuk menambahkan database langkah untuk permainan awal dan permainan akhir. Penggunaan database mampu meningkatkan kecepatan proses pencarian.
4. Fungsi evaluasi masih terbuka untuk diteliti supaya ditemukan fungsi yang terbaik.

Daftar Pustaka

[1] K. Korb, *Lecture Notes CSE2309/3309 Artificial Intelligence*, School of Computer Science and Software Eng, Monash University.

[2] Class Notes, McGill University, 1997.