

BAB V

PENGUJIAN DAN ANALISIS

Untuk mengetahui apakah sistem bekerja dengan baik dan sesuai dengan perancangan, maka diperlukan serangkaian pengujian. Pengujian yang dilakukan dalam bab ini adalah sebagai berikut:

1. Pengujian *Move Generation*.
2. Pengujian Tingkat Kedalaman Pencarian.
3. Pengujian Kedalaman Pencarian Terhadap Pengambilan Langkah.

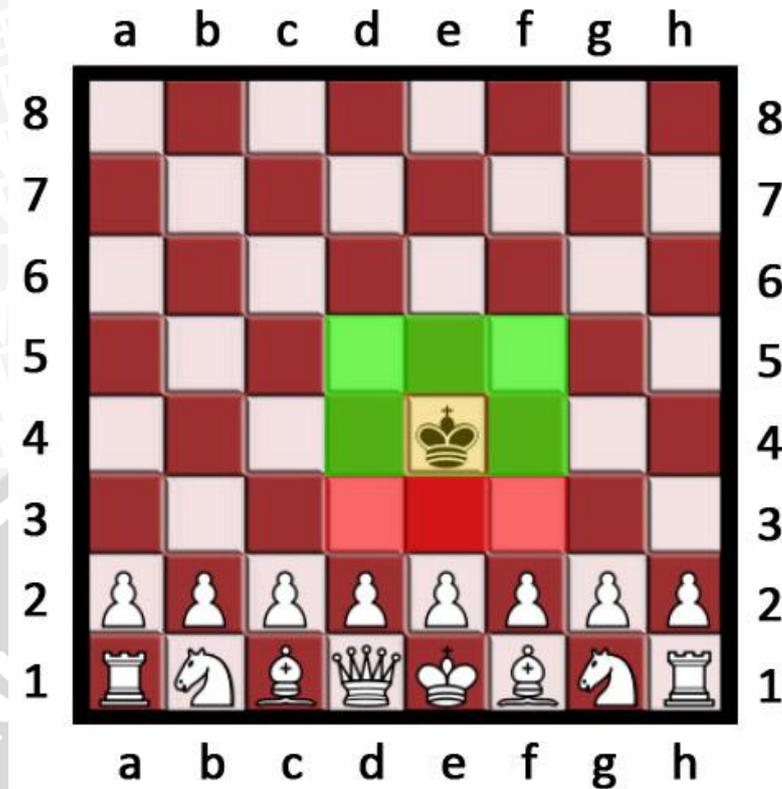
5.1 Pengujian Move Generation

Pengujian ini bertujuan untuk mengecek validasi dari seluruh gerakan apakah sudah sesuai dengan peraturan catur atau belum. Pengujian ini dilakukan dengan metode *piece-to-piece*, seperti telah dijelaskan pada bab sebelumnya.

Pengujian ini disebut juga Proses *Debugging*, yang bertujuan untuk menemukan *bug* (kesalahan) pada sistem, dimana dalam kasus ini adalah ada suatu bidak yang bisa bergerak diluar peraturan catur ataupun tidak bisa bergerak walaupun sudah memenuhi peraturan yang berlaku.

- a. Skenario Buatan untuk setiap Bidak

Yaitu adalah proses manipulasi papan agar seluruh bidak musuh dihapus kecuali bidak yang ingin diuji seperti pada gambar berikut:



Gambar 5.1 *Skenario Buatan*

Skenario diatas tidaklah mungkin terjadi dalam permainan yang sesungguhnya, namun skenario diatas merupakan cara yang paling efektif untuk menguji pergerakan tiap bidak.

Skenario ini diulangi hingga semua bidak selesai diuji, artinya posisi Raja Hitam diatas akan diganti oleh Bidak Hitam yang lain.

b. Memeriksa Respon Gerakan Lawan

Proses ini dilakukan dengan tujuan apakah gerakan yang akan dilakukan komputer sudah memenuhi peraturan atau belum. Dalam pengujian ini kita akan menggerakkan bidak putih untuk melihat berbagai macam respon Bidak yang ingin diuji

Seperti pada gambar pada poin a, Raja Hitam hanya boleh bergerak pada petak yang diwarnai Hijau, sedangkan petak Merah tidak boleh ditempati karena dua alasan berikut:

- ✓ Pergerakan bidak *bukan* Raja pada posisi ini akan mengakibatkan Raja berada dalam posisi skak.
- ✓ Pergerakan bidak Raja akan mengakibatkan Raja itu sendiri terkena skak (bunuh diri).
- ✓ Pergerakan tersebut memang tidak memnuhi peraturan Catur pada umumnya.

c. Analisis Pengujian Move Generation

Dalam skripsi ini pengujian untuk *Move Generation* mencapai nilai keberhasilan 100%, dengan kata lain seluruh bidak sudah bisa bergerak menurut peraturan yang berlaku. Sehingga pada tahap ini aplikasi sudah layak dimainkan oleh dua orang manusia (tanpa komputer).

5.2 Pengujian Tingkat Kedalaman Pencarian Terhadap Waktu Komputasi

Pengujian ini bertujuan untuk melihat seberapa lama proses komputasi untuk setiap kedalaman pencarian. Lalu akan dilihat keefektifannya dalam bermain, karena jika hanya mengandalkan satu algoritma, semakin dalam pencarian tidak berarti kemampuan bermain akan semakin bagus.

Pengujian tingkat kedalaman terhadap bagus-tidaknya suatu gerakan sangatlah sulit dilakukan, karena tidak ada ukuran yang pasti apakah suatu gerakan itu “bagus” atau tidak. Pengujian tingkat kedalaman terhadap hasil akhir permainan juga sulit untuk dilakukan karena banyaknya faktor yang bisa terjadi pada suatu permainan.

Karena itu pengujian pada tahap ini dilakukan berdasarkan jumlah kedalaman pencarian terhadap waktu eksekusi suatu gerakan per giliran komputer.

a. Dasar Pengujian *Searching Depth* terhadap *Execution Time*

Searching Depth disini berarti kedalaman pencarian untuk satu *ply*. Dalam isitlah standar Catur, satu *move* (gerakan) terdiri dari gerakan untuk setiap pemain (Putih dan Hitam), sehingga setelah 20 *moves* dalam satu permainan catur, 40 *plies* telah diselesaikan—20 oleh putih dan 20 oleh hitam.

Pengujian ini dilakukan dengan spesifikasi yang telah disebutkan pada Bab II, dengan satu komputer tambahan dengan processor yang berbeda sebagai perbandingan.

Dalam Algoritma ini, hal yang paling diuji dari sebuah sistem komputer adalah Processor, semakin cepat suatu Processor, semakin cepat kalkulasi untuk setiap langkah sehingga waktu pengambilan langkah sebenarnya juga semakin cepat.

Memory (RAM) hanya berperan kecil dalam algoritma ini, Memory hanya dibutuhkan untuk menjalankan aplikasi dan untuk menyimpan data bidak. Untuk data setiap gerakan, setiap iterasi bertambah, maka nilai Array untuk menyimpan data akan di-reset sehingga Algoritma ini tidak memakan banyak Memory.

b. Hasil Pengujian *Searching Depth* terhadap *Execution Time*

Tabel 5.1 Hasil Pengujian Kedalaman Pencarian

Ply	Execution Time (seconds)		
	Athlon 64 X2	Intel i3-3220	Intel Core i7-2600K
1	0.6	0.2	0
2	2.2	0.7	0
3	3.6	1.3	0.4
4	5.9	1.9	0.6
5	7	2.5	0.8
6	10.2	3.3	1.1
7	13.8	4	1.3
8	17.2	4.9	1.6
9	19	5.9	2.0
10	23	7	2.3

c. Analisis Pengujian *Searching Depth* terhadap *Execution Time*

Pengujian diatas dilakukan pada awal permainan, yaitu seluruh bidak berada pada posisi awal. Hasil pengujian ini akan berubah untuk setiap keadaan papan yang berbeda, karena jumlah gerakan yang mungkin untuk setiap kondisi papan jumlahnya berbeda-beda

Berdasarkan hasil pengujian diatas bisa disimpulkan bahwa:

- ✓ Tingkat kedalaman pencarian berbanding terbalik dengan waktu eksekusi, sehingga semakin dalam tingkat pencarian maka akan semakin lama waktu yang dibutuhkan oleh komputer untuk menghasilkan seluruh gerakan yang ada (*generating all possible moves*).
- ✓ Waktu eksekusi juga sangat dipengaruhi oleh prosesor yang digunakan. Semakin baru (dan canggih) suatu Prosesor maka kemampuan kalkulasi per detiknya juga semakin besar sehingga waktu untuk menghitung seluruh gerakan menjadi semakin cepat.

5.3 Pengujian Kedalaman Pencarian Terhadap Pengambilan Langkah

Pengujian ini bertujuan untuk melihat apakah program sudah bisa bermain selayaknya pemain catur pemula, dikarenakan hanya dengan mengandalkan pencarian maka kemampuan bermain cukup terbatas.

Pengujian ini dilakukan dengan membuat beberapa skenario suatu pertandingan dengan jawaban terbaik yang sudah ditentukan. Selanjutnya kita tinggal melihat apakah gerakan yang akan dilakukan komputer sesuai dengan jawaban terbaik tersebut atau tidak.

- Kasus 1 : **Minimax 1**



Gambar 5.2 Minimax 1

Gerakan Pertama Putih: Kuda di h7 akan memakan Kuda lawan di f6 (skak).

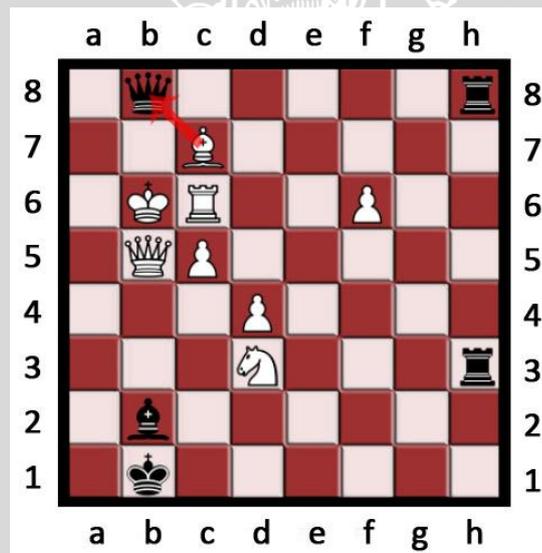
Tujuan: Bagaimana Komputer bisa meminimalisasi kerugian.

Jawaban Paling Tepat untuk Hitam (Komputer):

- Pion di g7 memakan Kuda yang baru saja maju ke f6.
- Ratu Putih di e3 memakan ratu Hitam di e7
- Raja Hitam memakan Ratu Putih di e3
- Benteng Putih di d2 memakan Benteng Hitam di d5.

Langkah diatas adalah langkah terbaik yang bisa diambil Hitam (Komputer), yaitu walaupun Komputer kehilangan Ratu, Benteng, dan Kuda namun masih bisa memakan Ratu dan Kuda lawan. Dalam kondisi yang tidak menguntungkan untuk Hitam langkah inilah yang paling tepat untuk dilakukan.

- Kasus 2 : **Minimax 2**



Gambar 5.3 *Minimax 2*

Gerakan Pertama Putih: Gajah di c7 akan memakan Ratu lawan di b8.

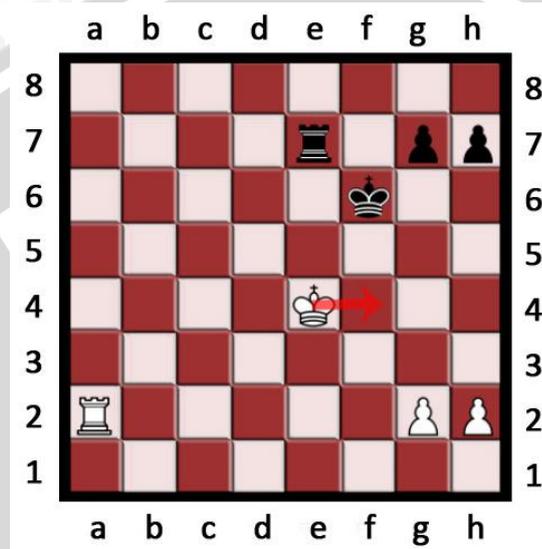
Tujuan: Bagaimana Komputer bisa memperoleh nilai sebesar-besarnya.

Jawaban Paling Tepat untuk Hitam (Komputer):

- Benteng di h8 memakan Gajah di b8 (skak).
- Raja Putih di b6 bergerak ke a5 atau a6 untuk menghindari skak.
- Benteng Hitam di b8 memakan Ratu Putih di b5.
- Raja Putih di a5 atau a6 memakan Benteng Hitam di b5.
- Benteng hitam di h3 memakan Kuda Putih di d3.

Langkah diatas adalah bagaimana Hitam (Komputer) meminimalisasi kerugian, yaitu walaupun Komputer baru saja kehilangan Ratu. Namun Hitam bisa memakan Ratu dan Kuda lawan. Dalam kondisi yang tidak menguntungkan untuk Hitam langkah inilah yang paling tepat untuk dilakukan.

- Kasus 3 : **Pemanfaatan Blunder Lawan**



Gambar 5.4 Pemanfaatan Blunder

Gerakan Pertama Putih: Raja di e4 bergerak ke f4 untuk menghindari skak.

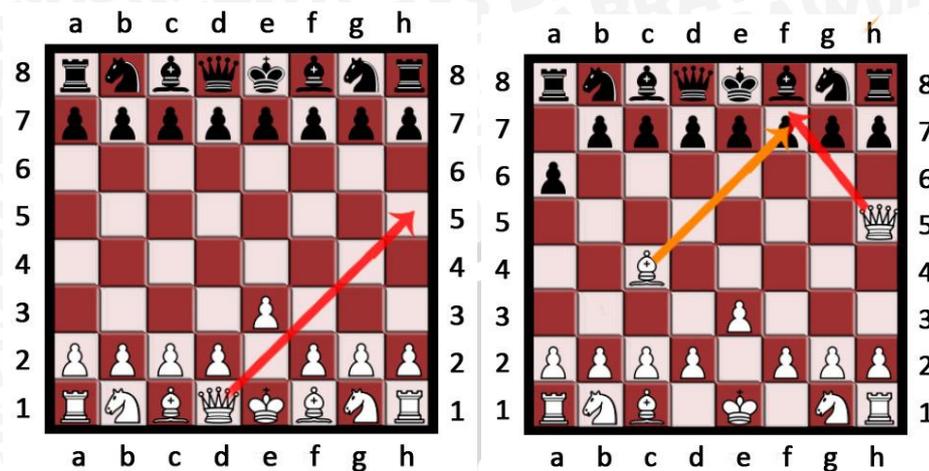
Tujuan: Bagaimana Komputer bisa memanfaatkan blunder lawan.

Jawaban Paling Tepat untuk Hitam (Komputer):

- Pion Hitam di g7 bergerak ke g5 (skak).
- Raja Putih di f4 bergerak ke g4 untuk menghindari skak.
- Benteng Hitam di e7 bergerak ke e4.
- Raja Putih di g4 bergerak ke h5 untuk menghindari skak (inilah blunder yang disengaja).
- Benteng Hitam di e4 bergerak ke h4 (skakmat).

Langkah diatas adalah bagaimana Hitam (Komputer) bisa memanfaatkan blunder yang dilakukan pemain lawan.

- Kasus 4 : Membaca Langkah Lawan



Gambar 5.5 *Queen's Gambit*

Gerakan Pertama Putih: Ratu di d1 bergerak ke h5 untuk melakukan strategi yang dinamakan *Queen's Gambit*, yaitu strategi skakmat paling dasar dan cepat dalam permainan catur.

Tujuan: Bagaimana Komputer bisa menghindari dari skakmat lawan.

Jawaban Paling Tepat untuk Hitam (Komputer):

Queen's Gambit ini bisa dihindari dengan beberapa cara berikut:

- Salah satu dari Pion hitam didepan raja (d7, e7 dan f7) dimajukan
- Menghalangi pergerakan Ratu Putih.
- Menghalangi pergerakan Gajah Putih.

Langkah diatas adalah bagaimana Hitam (Komputer) bisa menghindari skakmat yang akan dilakukan lawan dengan cara melakukan salah satu dari tiga langkah diatas.

a. Hasil Pengujian Kedalaman Pencarian Terhadap Pengambilan Langkah

Tabel 5.2 Hasil Permainan dengan 2 Kedalaman Pencarian

Turn	Kasus 1	Kasus 2	Kasus 3	Kasus 4
2	Pg7-f6 (Benar)	Rh8-b8 (Benar)	Pg7-g5 (Benar)	Pb7-b5 (Salah)
4	Ke8-e7 (Benar)	Rb8-b5 (Benar)	Re7-e4 (Benar)	Pg7-g5 (Salah)
6	-	Rh3-d3 (Benar)	Re4-h4 (Benar)	Pa7-a6 (Salah)

Tabel 5.3 Hasil Permainan dengan 4 Kedalaman Pencarian

Turn	Kasus 1	Kasus 2	Kasus 3	Kasus 4
2	Pg7-f6 (Benar)	Rh8-b8 (Benar)	Pg7-g5 (Benar)	Ph7-h5 (Benar)
4	Ke8-e7 (Benar)	Rb8-b5 (Benar)	Re7-e4 (Benar)	-
6	-	Rh3-d3 (Benar)	Re4-h4 (Benar)	-

Tabel 5.4 Hasil Permainan dengan 10 Kedalaman Pencarian

Turn	Kasus 1	Kasus 2	Kasus 3	Kasus 4
2	Pg7-f6 (Benar)	Rh8-b8 (Benar)	Pg7-g5 (Benar)	Ph7-h5 (Benar)
4	Ke8-e7 (Benar)	Rb8-b5 (Benar)	Re7-e4 (Benar)	-
6	-	Rh3-d3 (Benar)	Re4-h4 (Benar)	-

b. Analisis Hasil Kedalaman Pencarian Terhadap Pengambilan Langkah

Kemampuan pengambilan langkah oleh komputer juga sangat dipengaruhi oleh rating dari lawan dikarenakan algoritma Shannon Type-A ini merupakan prediksi minimax, yaitu memprediksi langkah mana yang kemungkinannya terbesar untuk diambil lawan.

Contohnya saat melawan pemain pemula dengan rating sekitar 800, diasumsikan bahwa pemain dalam rating ini hanya berpikir paling banyak dua langkah kedepan sehingga dengan kedalaman pencarian komputer sebesar 2 akan menghasilkan permainan yang bagus (menang).

Namun jika pemain lawan berpikir tiga langkah kedepan atau lebih (seperti diasumsikan pada pemain dengan rating sekitar 1100) maka prediksi komputer akan sedikit kacau sehingga kadang-kadang akan melakukan blunder karena prediksi *minimax* yang dilakukan komputer salah. Demikian juga sebaliknya, jika lawan hanya mengambil gerakan yang “sederhana” dan komputer melakukan prediksi yang terlalu jauh kedepan maka prediksi akan menjadi sedikit kacau.

Contohnya pada kasus ke-4, dimana tingkat kedalaman pencarian sangat berpengaruh pada hasil pengujian karena strategi skakmat yang dilakukan lawan merupakan pemikiran lebih dari dua langkah kedepan, sehingga jika komputer hanya bermain pada level dua kedalaman pencarian maka hampir bisa dipastikan akan kalah.

