

Penambahan Fungsi Evaluasi pada Aplikasi Othello 2007

Tugas Akhir

Disampaikan Sebagai Bagian Dari Persyaratan Kelulusan Diploma 3
Program Studi Teknik Informasi

Oleh :

Normasari Ritonga 11105008

Bani Rizky Telaumbanua 11105009

Samuel Tambunan 11105018



Politeknik Informatika Del
2008

Lembar Pengesahan Tugas Akhir
Politeknik Informatika Del

Penambahan Fungsi Evaluasi pada Aplikasi Othello 2007

Oleh :

Normasari Ritonga 11105008

Bani Rizky Telaumbanua 11105009

Samuel Tambunan 11105018

Dinyatakan memenuhi syarat dan karenanya disetujui dan disahkan sebagai

Laporan Tugas Akhir Diploma 3

Program Studi Teknik Informasi

Pembimbing 1,

Pembimbing 2,

Dr. Ir. G. A. Putri Saptawati, M.Comm.

NIP : 132127666

Ondy Dharma Indra Sukma Tambun, S.T.

NIDN : 0127057901

Prakata

Puji dan syukur kami ucapkan kepada Tuhan Yang Maha Esa atas segala berkat, kasih dan karunia-Nya sehingga Tugas Akhir ini dapat diselesaikan dengan baik. Tugas Akhir dengan judul Penambahan Fitur Algoritma Minimax Alpha-Beta Pruning pada Othello ini disusun untuk memenuhi syarat kelulusan diploma 3 program studi Teknik Informatika Politeknik Informatika Del.

Terimakasih kepada Bapak Ondy Dharma Indra Sukma Tambun S. T. selaku pembimbing yang telah membimbing dan memberikan arahan kepada kami selama pelaksanaan Tugas Akhir ini.

Terimakasih kami ucapkan kepada Ibu Dr. Ir. Inggriani Liem yang telah menggantikan peran Bapak Ondy selama beliau tidak di tempat sebagai pembimbing Tugas Akhir serta Ibu Dr. Ir. G. A. Putri Saptawati, Mcomm., selaku pembimbing dan penguji dari ITB atas dukungan yang diberikan selama pelaksanaan Tugas Akhir. Terimakasih juga kami ucapkan kepada panitia Tugas Akhir, atas bimbingan yang diberikan selama pelaksanaan Tugas Akhir ini.

Kepada segenap staf pengajar Politeknik Informatika Del yang telah mengajar kami selama masa perkuliahan, staf administrasi yang telah membantu dan mendukung perkuliahan serta staf perpustakaan yang membantu pelaksanaan dan penyusunan Tugas Akhir ini. Seluruh pihak yang tidak dapat kami sebutkan satu per satu, yang telah berpartisipasi selama penyusunan Tugas Akhir ini.

Harapan kami, semoga hasil dari pengerjaan Tugas Akhir ini dapat memberikan manfaat.

Sitoluama, 2 September 2008

Normasari Ritonga	11105008
Bani Rizky Totonafu Telaumbanua	11105009
Samuel Tambunan	11105018

Abstrak

Hal yang dilakukan pada Tugas Akhir ini adalah penambahan fungsi evaluasi pada aplikasi Othello 2007 dengan tujuan untuk mempercepat *response time*. Penerapan fungsi evaluasi dilakukan dengan mengacu pada literatur Tugas Akhir.

Fungsi evaluasi yang ditambahkan merupakan implementasi strategi permainan othello di mana fungsi tersebut terdiri dari tiga buah fungsi yaitu *disc position*, *stable disc*, dan *possible moves*. Masing-masing fungsi tersebut menghasilkan nilai setelah melakukan evaluasi terhadap langkah yang akan diambil.

Fungsi evaluasi yang telah ditambahkan pada aplikasi Othello 2008 tersebut diuji dengan menggunakan tujuh data sampel pengujian dan hasilnya dibandingkan dengan aplikasi Othello 2007.

Dari hasil pengujian disimpulkan bahwa penambahan fungsi evaluasi menghasilkan *response time* pada rentang 0 hingga 4 detik, namun masih terdapat beberapa anomali pada setiap sampel. Sementara pada aplikasi Othello 2007 dihasilkan *response time* pada rentang 0 hingga 145 detik. Dengan demikian dapat disimpulkan bahwa penambahan fungsi evaluasi dapat mempercepat *response time* aplikasi Othello 2007.

Kata kunci: **Othello, algoritma Minimax Alpha-Beta Pruning, alpha-pruned, beta-pruned**

DAFTAR ISI

Prakata.....	3
Daftar Kode.....	7
Bab I Pendahuluan	8
1.1 Latar Belakang.....	8
1.2 Rumusan masalah	8
1.3 Tujuan	8
1.4 Lingkup.....	8
1.5 Pendekatan	9
1.6 Sistematika Penyajian	9
Bab II Analisis Aplikasi Othello 2007.....	10
2.1 Deskripsi Permainan Othello	10
2.2 Aturan Permainan Othello	12
2.3 Strategi Permainan Othello	16
2.4 Deskripsi Umum Aplikasi Othello 2007.....	17
2.5 Kelemahan aplikasi	18
2.6 Penerapan Algoritma Minimax Alpha-Beta Pruning.....	18
2.7 Algoritma Minimax Alpha-Beta <i>Pruning</i>	19
2.8 Kesimpulan	20
Bab III Pengembangan Fungsi Evaluasi.....	21
3.1 Mekanisme Umum Fungsi evaluasi.....	21
3.2 Strategi Permainan Othello yang Dikembangkan.....	21
3.3 Fungsi Evaluasi.....	21
3.3.1 Perhitungan Posisi Biji.....	22
3.3.2 Perhitungan Stable Disc	23
3.3.3 Perhitungan Possible Moves	23
Bab IV Implementasi Fungsi Evaluasi	25
Bab V Pengujian Fungsi Evaluasi.....	28
5.1 Fitur Pengujian Response Time	28
5.2 Sampel Pengujian.....	28
5.3 Komputer Pengujian	29
5.4 Pengukuran response time Othello 2007	29
5.5 Pengukuran response time Othello 2008	29
5.6 Perbandingan Grafik Othello 2007 dan Othello 2008	30
Bab VI Kesimpulan dan Saran.....	32
6.1 Kesimpulan	32
6.2 Saran	32
Lampiran A Grafik Perbandingan Othello 2007 dan Othello 2008 Level Easy	33
Lampiran B Grafik Perbandingan Othello 2007 dan Othello 2008 Level Medium	40
Lampiran C Grafik Perbandingan Othello 2007 dan Othello 2008 Level Hard	47
Lampiran D Posisi Papan pada Permainan Othello	53
Daftar Pustaka dan Rujukan.....	54
Daftar Pustaka	54
Rujukan	54

Daftar Gambar

Gambar 1 Keadaan awal permainan [ELA93].....	10
Gambar 2 Notasi pada Othello [BRO05].....	11
Gambar 3 Penamaan Kotak (<i>Square</i>) pada Papan Othello [BRO05]	11
Gambar 4 Biji Hitam Diapit Biji Putih	12
Gambar 5 Kemungkinan Langkah Pertama [ELA93]	13
Gambar 6 Langkah Pertama Pemain Hitam [ELA93]	13
Gambar 7 Biji pada Kotak E5 Berganti Warna [ELA93]	14
Gambar 8 Hitam melangkah pada kotak C6 [ELA93].....	14
Gambar 9 Hasil Langkah Hitam pada C6 [ELA93].....	15
Gambar 10 Akhir Permainan Dengan Satu Kotak Kosong [ELA93]	15
Gambar 11 Tampilan Permainan Othello 2007	17
Gambar 12 Pohon pencarian pada permainan Othello [IAM07]	19
Gambar 13 Penambahan fungsi evaluasi pada Minimax Alpha-Beta <i>Pruning</i>	24
Gambar 14 Perbandingan Level <i>Hard Sample</i> 5	30
Gambar 15 Perbandingan Level <i>Easy Sample</i> 1	33
Gambar 16 Perbandingan Level <i>Easy Sample</i> 2	34
Gambar 17 Perbandingan Level <i>Easy Sample</i> 3	35
Gambar 18 Perbandingan Level <i>Easy Sample</i> 4	36
Gambar 19 Perbandingan Level <i>Easy Sample</i> 5	37
Gambar 20 Perbandingan Level <i>Easy Sample</i> 6	38
Gambar 21 Perbandingan Level <i>Easy Sample</i> 7	39
Gambar 22 Perbandingan Level <i>Medium Sample</i> 1	40
Gambar 23 Perbandingan Level <i>Medium Sample</i> 2	41
Gambar 24 Perbandingan Level <i>Medium Sample</i> 3	42
Gambar 25 Perbandingan Level <i>Medium Sample</i> 4	43
Gambar 26 Perbandingan Level <i>Medium Sample</i> 5	44
Gambar 27 Perbandingan Level <i>Medium Sample</i> 6	45
Gambar 28 Perbandingan Level <i>Medium Sample</i> 7	46
Gambar 29 Perbandingan Level <i>Hard Sample</i> 1	47
Gambar 30 Perbandingan Level <i>Medium Sample</i> 2	48
Gambar 31 Perbandingan Level <i>Medium Sample</i> 3	49
Gambar 32 Perbandingan Level <i>Medium Sample</i> 4	50
Gambar 33 Perbandingan Level <i>Medium Sample</i> 6	51
Gambar 34 Perbandingan Level <i>Medium Sample</i> 7	52
Gambar 35 Posisi Papan pada Pohon Pencarian Permainan Othello.....	53

Daftar Kode

Kode 1 Fungsi <i>Check Stable Disc</i>	25
Kode 2 Fungsi <i>Disc Position</i>	26
Kode 3 Fungsi <i>Possible Moves</i>	26
Kode 4 Pemanggilan Fungsi Evaluasi	27

Lembar Pengesahan Tugas Akhir
Politeknik Informatika Del

Penambahan Fitur pada Aplikasi Othello 2007

Oleh:

Normasari Ritonga	11105008
Bani Rizky Telaumbanua	11105009
Samuel Tambunan	11105018

Dinyatakan memenuhi syarat dan karenanya disetujui dan disahkan sebagai

Laporan Tugas Akhir Diploma 3

Program Studi Teknik Informasi

Pembimbing 1,



Ir. G. A. Putri Saptawati, M.Comm.

NIP : 132127666

Pembimbing 2,



u.b.

Ondy Dharma Indra Sukma Tambun, S.T.

NIDN : 0127057901

Prakata

Puji dan syukur kami ucapkan kepada Tuhan Yang Maha Esa atas segala berkat, kasih dan karunia-Nya sehingga Tugas Akhir ini dapat diselesaikan dengan baik. Tugas Akhir dengan judul Penambahan Fitur Algoritma Minimax Alpha-Beta Pruning pada Othello ini disusun untuk memenuhi syarat kelulusan diploma 3 program studi Teknik Informatika Politeknik Informatika Del.

Terimakasih kepada Bapak Ondy Dharma Indra Sukma Tambun S. T. selaku pembimbing yang telah membimbing dan memberikan arahan kepada kami selama pelaksanaan Tugas Akhir ini.

Terimakasih kami ucapkan kepada Ibu Dr. Ir. Inggriani Liem yang telah menggantikan peran Bapak Ondy selama beliau tidak di tempat sebagai pembimbing Tugas Akhir serta Ibu Dr. Ir. G. A. Putri Saptawati, Mcomm., selaku pembimbing dan penguji dari ITB atas dukungan yang diberikan selama pelaksanaan Tugas Akhir. Terimakasih juga kami ucapkan kepada panitia Tugas Akhir, atas bimbingan yang diberikan selama pelaksanaan Tugas Akhir ini.

Kepada segenap staf pengajar Politeknik Informatika Del yang telah mengajar kami selama masa perkuliahan, staf administrasi yang telah membantu dan mendukung perkuliahan serta staf perpustakaan yang membantu pelaksanaan dan penyusunan Tugas Akhir ini. Seluruh pihak yang tidak dapat kami sebutkan satu per satu, yang telah berpartisipasi selama penyusunan Tugas Akhir ini.

Harapan kami, semoga hasil dari pengerjaan Tugas Akhir ini dapat memberikan manfaat.

Sitoluama, 2 September 2008

Normasari Ritonga	11105008
Bani Rizky Totonafu Telaumbanua	11105009
Samuel Tambunan	11105018

Abstrak

Hal yang dilakukan pada Tugas Akhir ini adalah penambahan fungsi evaluasi pada aplikasi Othello 2007 dengan tujuan untuk mempercepat *response time*. Penerapan fungsi evaluasi dilakukan dengan mengacu pada literatur Tugas Akhir.

Fungsi evaluasi yang ditambahkan merupakan implementasi strategi permainan othello di mana fungsi tersebut terdiri dari tiga buah fungsi yaitu *disc position*, *stable disc*, dan *possible moves*. Masing-masing fungsi tersebut menghasilkan nilai setelah melakukan evaluasi terhadap langkah yang akan diambil.

Fungsi evaluasi yang telah ditambahkan pada aplikasi Othello 2008 tersebut diuji dengan menggunakan tujuh data sampel pengujian dan hasilnya dibandingkan dengan aplikasi Othello 2007.

Dari hasil pengujian disimpulkan bahwa penambahan fungsi evaluasi menghasilkan *response time* pada rentang 0 hingga 4 detik, namun masih terdapat beberapa anomali pada setiap sampel. Sementara pada aplikasi Othello 2007 dihasilkan *response time* pada rentang 0 hingga 145 detik. Dengan demikian dapat disimpulkan bahwa penambahan fungsi evaluasi dapat mempercepat response time aplikasi Othello 2007.

Kata kunci: **Othello, algoritma Minimax Alpha-Beta *Pruning*, alpha-pruned, beta-pruned**

Bab I

Pendahuluan

1.1 Latar Belakang

Pembangunan *game* komputer yang menerapkan dasar *Artificial Intelligence* (AI) dapat dilakukan dengan berbagai cara. Salah satu penerapan yang umum dilakukan yaitu membuat basis kerja *game* tersebut melalui penerapan *game-tree search*. Penerapan tersebut pada *game* komputer dilakukan dengan menggunakan algoritma yang beragam. Sebagai contoh, algoritma Minimax Alpha-Beta *Pruning* telah diterapkan pada aplikasi Othello 2007 [IAM07].

Response time pada Othello 2007 masih lambat yang salah satu penyebabnya adalah banyaknya simpul yang dihasilkan pada pohon permainan. Semakin banyak kemungkinan langkah yang dihasilkan, maka semakin banyak pula jumlah simpul pada pohon permainan.

1.2 Rumusan masalah

Berdasarkan penjelasan di atas, maka rumusan masalah Tugas akhir ini adalah:

1. Bagaimana *response time* aplikasi Othello 2007 dapat dipercepat?
2. Bagaimana cara yang tepat untuk mempercepat *response time*.

1.3 Tujuan

Tujuan Tugas Akhir ini adalah menambah fungsi evaluasi pada aplikasi Othello 2007 dengan maksud mempercepat *response time*.

1.4 Lingkup

Lingkup dari kajian yang akan dilakukan pada Tugas Akhir ini adalah sebagai berikut:

1. Mempelajari aplikasi Othello 2007 khususnya algoritma Minimax Alpha-Beta *Pruning* yang digunakan.
2. Mempelajari strategi permainan Othello dari literatur [BRO05], [ELA93].
3. Mengembangkan cara untuk menghasilkan *response time* pada rentang 0 hingga 4 detik yaitu fungsi evaluasi.
4. Menambahkan fungsi evaluasi pada aplikasi Othello 2007, yang diberi nama aplikasi **Othello 2008**.

1.5 Pendekatan

Pendekatan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah:

1. Mempelajari aplikasi Othello 2007 dengan tujuan untuk memahami bagaimana struktur aplikasi Othello 2007.
2. Mengembangkan fungsi evaluasi yang ditambahkan ke dalam Minimax Alpha-Beta *Pruning*.
3. Menambahkan fungsi evaluasi pada aplikasi Othello 2007.
4. Menguji aplikasi Othello 2008 dan *response time* yang dihasilkan dibandingkan dengan aplikasi Othello 2007.

1.6 Sistematika Penyajian

Penyajian materi bahasan disusun dengan sistematika sebagai berikut:

1. Bab kedua menguraikan analisis terhadap aplikasi Othello 2007, yaitu aplikasi yang di dalamnya ditambahkan fungsi evaluasi.
2. Bab ketiga menguraikan pengembangan fungsi evaluasi.
3. Bab keempat menguraikan implementasi hasil pengembangan fungsi evaluasi terhadap aplikasi Othello 2007.
4. Bab kelima menguraikan pengujian terhadap fungsi evaluasi untuk membandingkan *response time* pada Othello 2007 dan Othello 2008.
5. Bab keenam menguraikan kesimpulan beserta saran-saran pelaksanaan Tugas Akhir.
6. Informasi tambahan untuk melengkapi dokumen ini diuraikan pada lampiran dokumen.

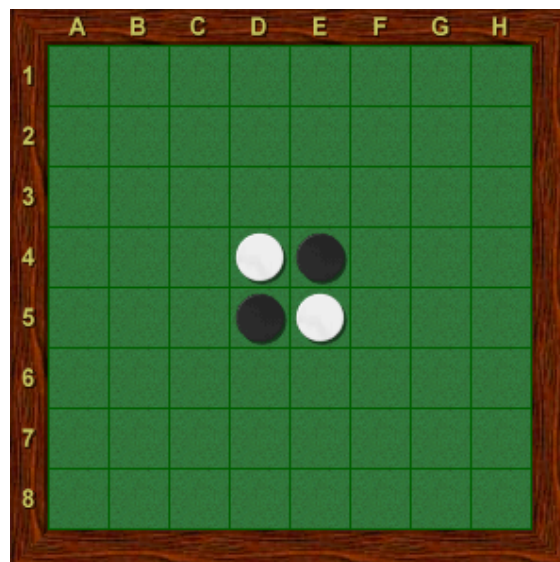
Bab II

Analisis Aplikasi Othello 2007

2.1 Deskripsi Permainan Othello

Othello merupakan permainan strategi [ELA93] yang dimainkan oleh dua pemain Hitam dan Putih. Permainan ini dimainkan pada papan permainan berukuran 8 x 8, yang disebut *Othello Board*. Papan permainan memiliki 64 kotak dengan penamaan untuk sisi horisontal adalah A, B, C, D, D, E, F, G, dan H, sedangkan untuk sisi vertikal adalah 1, 2, 3, 4, 5, 6, 7, dan 8.

Tujuan dari permainan ini adalah [PTC90] untuk memperoleh sebanyak mungkin biji Othello pada akhir permainan sesuai dengan warna yang dipilih sebelum permainan dimulai. Pada awal permainan, dua biji hitam ditempatkan pada kotak E4 dan D5 serta dua biji putih pada kotak D4 dan E5, seperti gambar 1. Permainan akan selalu dimulai dengan posisi biji seperti ini.



Gambar 1 Keadaan awal permainan [ELA93]

Notasi setiap kotak adalah dengan lebih dahulu menyebutkan sisi horisontal diikuti sisi vertikal, misalnya A7, B4, dan H8 (gambar 2). Kedua pemain akan menempatkan 64 biji Othello pada papan permainan, yaitu sebagian biji putih dan sebagian lainnya biji hitam.

	A	B	C	D	E	F	G	H
1	A1	B1	C1	D1	E1	F1	G1	H1
2	A2	B2	C2	D2	E2	F2	G2	H2
3	A3	B3	C3	D3	E3	F3	G3	H3
4	A4	B4	C4	D4	E4	F4	G4	H4
5	A5	B5	C4	D5	E5	F5	G5	H5
6	A6	B6	C6	D6	E6	F6	G6	H6
7	A7	B7	C7	D7	E7	F7	G7	H7
8	A8	B8	C8	D8	E8	F8	G8	H8

Gambar 2 Notasi pada Othello [BRO05]

Seperti yang terlihat pada gambar 3, *corner* merupakan nama yang diasosiasikan kepada kotak yang diberi tanda “*”. *X-square* merupakan nama yang diasosiasikan kepada kotak yang secara diagonal berdekatan dengan kotak *corner* dan diberi tanda huruf X. *C-square* merupakan nama yang diasosiasikan kepada kotak yang secara vertikal dan horisontal berdekatan dengan *corner*, yang ditandai dengan huruf C. *A-square* merupakan kotak yang diberi tanda huruf A pada tepi papan permainan dan yang mengapit *B-Square* yang diberi tanda huruf B.

	A	B	C	D	E	F	G	H
1	*	C	A	B	B	A	C	*
2	C	X	X	C
3	A	A
4	B	B
5	B	B
6	A	A
7	C	X	X	C
8	*	C	A	B	B	A	C	*

Gambar 3 Penamaan Kotak (Square) pada Papan Othello [BRO05]

2.2 Aturan Permainan Othello

Sebelum memulai permainan, masing-masing pemain memperoleh 32 biji dan harus memilih salah satu warna pada sisi biji untuk digunakan selama permainan. Biji Othello (yang mempunyai dua sisi warna) dinyatakan berwarna hitam jika sisi biji hitam berada di atas dan putih jika sisi biji putih yang berada di atas.

Pada permulaan permainan, hitam melangkah terlebih dahulu dan selanjutnya kedua pemain bermain secara bergantian. Setiap gilirannya, pemain harus menaruh biji sesuai dengan warna pemain itu sendiri pada kotak kosong yang terdapat pada papan permainan dan dekat dengan area biji pemain lawan. Pada saat melangkah, biji yang sama warnanya dengan yang ditaruh harus mengapit biji pemain lawan, sehingga langkah tersebut disebut sebagai *legal moves*. Setelah itu, pemain tersebut harus membalik (*flipping*) biji pemain lawan yang diapit sehingga berganti warna menjadi warna bijinya (gambar 4).

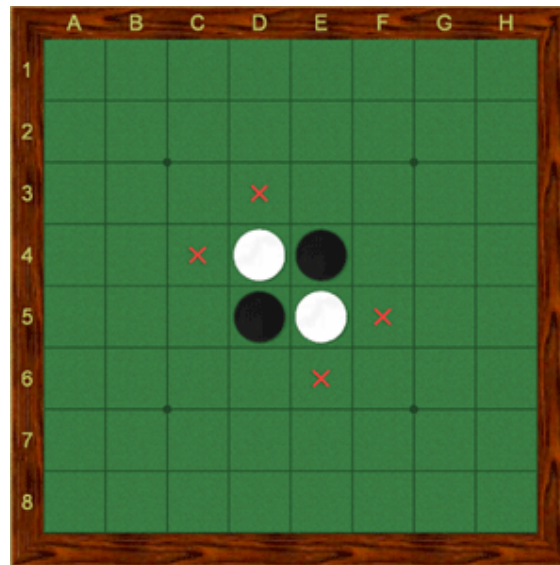


Gambar 4 Biji Hitam Diapit Biji Putih

Setiap biji dapat mengapit satu atau lebih biji lawan dalam satu baris dengan berbagai arah yaitu horisontal, vertikal, atau diagonal. Baris yang dimaksud adalah satu atau lebih biji yang secara lurus saling berkelanjutan atau tanpa terputus dalam satu arah.

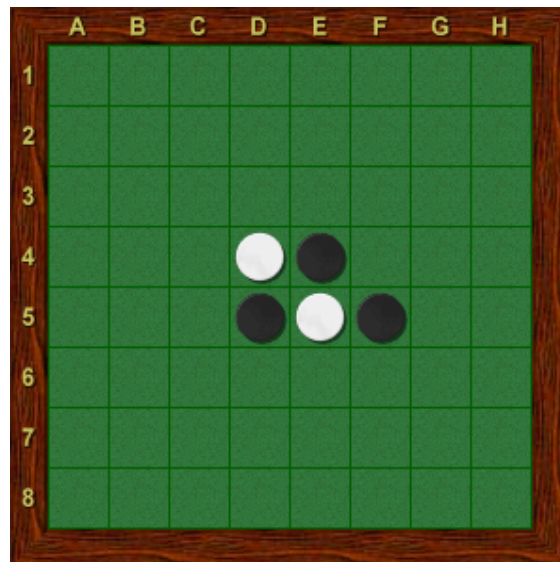
Jika pemain tidak memiliki kesempatan untuk mengapit biji pemain lawan, maka pemain tersebut tidak memiliki satu pun *legal moves*, sehingga langkah selanjutnya dialihkan menjadi giliran pemain lawan. Namun, jika pemain memiliki minimal satu *legal moves*, maka pemain tersebut tidak dapat mengalihkan gilirannya pada pemain lawan, melainkan harus menjalankan langkah tersebut.

Pemain Hitam memiliki empat kotak pilihan sebagai langkah pertama yaitu D3, C4, F5, dan E6 seperti yang terlihat pada gambar 5.

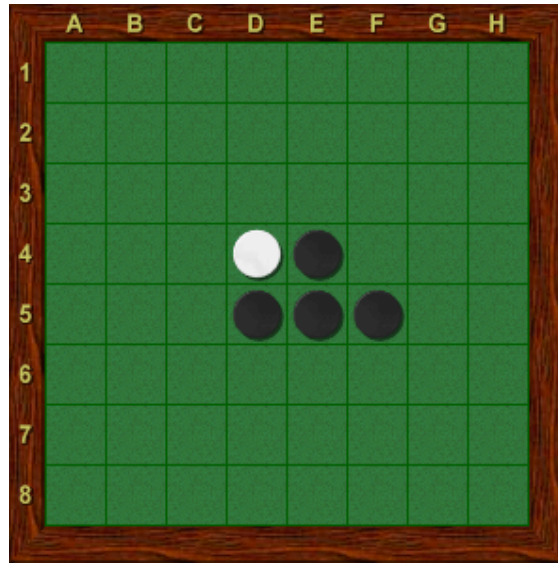


Gambar 5 Kemungkinan Langkah Pertama [ELA93]

Sebagai contoh, pada gambar 6 langkah pertama pemain Hitam adalah F5. Dengan melangkah pada kotak tersebut, maka pemain tersebut mengapit biji pada kotak E5 bersama dengan biji yang telah berada pada papan sebelumnya (D5), dengan demikian pemain tersebut membalikkan biji pada kotak D5 (gambar 7) sehingga menjadi biji berwarna hitam.

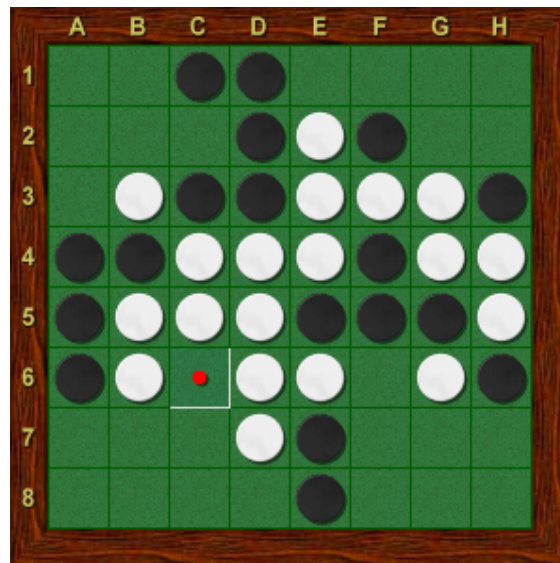


Gambar 6 Langkah Pertama Pemain Hitam [ELA93]

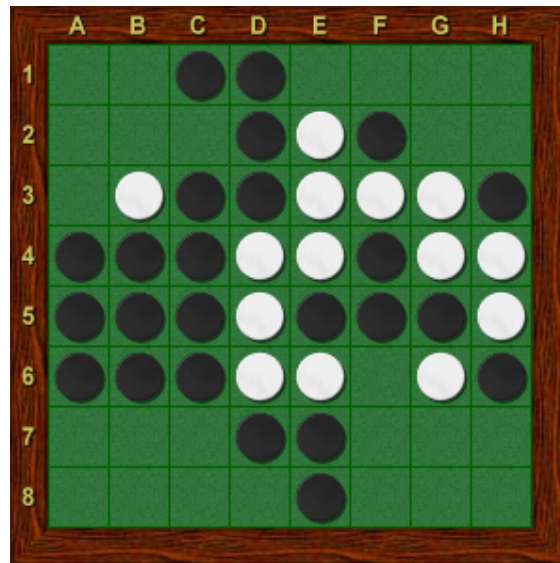


Gambar 7 Biji pada Kotak E5 Berganti Warna [ELA93]

Biji dapat mengapit biji lawan dari 8 arah yaitu arah diagonal sebanyak 4 arah, horisontal 2 arah, dan vertikal 2 arah. Pada gambar 8, biji hitam yang diletakkan pada kotak C6 akan mengapit biji putih yang terdapat pada arah horisontal (B6), vertikal (C4 dan C5), serta diagonal (B5 dan D7) sehingga total biji yang berubah sebanyak 5 biji (gambar 9).

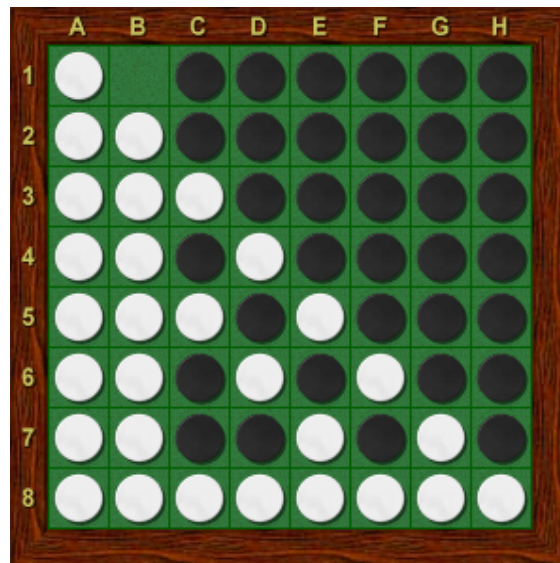


Gambar 8 Hitam melangkah pada kotak C6 [ELA93]



Gambar 9 Hasil Langkah Hitam pada C6 [ELA93]

Biji yang terdapat pada kotak E3, D4, E4, dan D5 tidak ikut berubah menjadi hitam karena biji tersebut tidak diapit oleh biji yang diletakkan oleh pemain Hitam pada kotak C6 (gambar 9). Permainan akan berakhir apabila kedua pemain sama sekali tidak memiliki minimal satu *legal moves*. Umumnya akhir permainan yang seperti ini terjadi apabila seluruh 64 kotak pada papan permainan telah diisi penuh oleh biji-biji kedua pemain. Namun, permainan juga akan berakhir apabila kedua pemain tidak dapat bergerak sama sekali dan masih terdapat kotak yang kosong, misalnya jika semua kotak diisi oleh satu warna saja atau seperti yang terlihat pada gambar 10.



Gambar 10 Akhir Permainan Dengan Satu Kotak Kosong [ELA93]

2.3 Strategi Permainan Othello

Untuk dapat memenangkan permainan Othello, sebaiknya pemain harus mengetahui strategi dasar dalam permainan Othello. Strategi-strategi [BRO05], [ELA93] yang sebaiknya diterapkan adalah sebagai berikut.

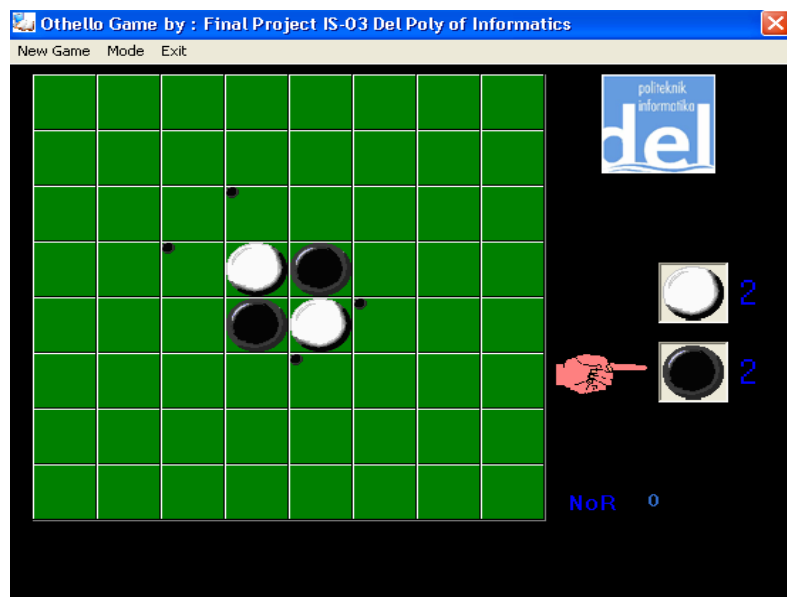
1. *Maximum disc*, maksudnya adalah membalikkan biji pemain sebanyak mungkin pada setiap langkah yang diambil.
2. *Stable Disc*, maksudnya adalah mengambil langkah yang menyebabkan biji pemain menjadi stabil (warna biji pemain tetap hingga akhir permainan) karena tidak dapat dibalikkan oleh pemain lawan.
3. *Wedges*, maksudnya adalah mengambil langkah yang menyebabkan biji yang baru ditaruh menjadi terdesak (*wedged*), karena tidak dapat dibalikkan oleh lawan pada langkah berikutnya, sehingga pada langkah berikutnya pemain tersebut dapat meletakkan biji pada posisi yang menguntungkan. Salah satu posisi yang menguntungkan yang bisa diambil adalah *corner*.
4. *Mobility*, maksudnya adalah mengambil langkah yang menyebabkan pemain lawan hanya memiliki langkah sesedikit mungkin, terutama jika langkah yang mungkin tersebut merupakan langkah yang merugikan. Contohnya, pemain lawan hanya memiliki satu kemungkinan langkah dan langkah tersebut berada pada kotak *C-Square* atau *X-Square*.
5. *Frontiers*, maksudnya adalah menghindari langkah pada kotak yang dekat dengan banyak kotak kosong. Semakin banyak kotak kosong yang berdekatan dengan biji salah satu pemain, maka semakin banyak pula kemungkinan langkah yang dimiliki pemain lawan. Demikian sebaliknya.
6. *Tempo, waiting moves*, maksudnya adalah menunggu lawan untuk terlebih dahulu melangkah pada kotak di mana biji yang ditaruh tersebut dekat dengan banyak kotak kosong. Pemain tersebut menunggu dengan cara melangkah pada posisi/daerah lain.
7. *Creeping along edges*, maksudnya adalah membiarkan lawan menguasai daerah tengah pada papan Othello dengan cara bermain pada bagian tepi papan. Dengan demikian lawan akan memiliki kemungkinan langkah yang sedikit karena tidak dapat membalikkan biji pemain pada bagian tepi papan. Namun, keadaan ini juga dapat merugikan pada saat pemain memiliki sedikit kemungkinan langkah dan terpaksa bermain pada kotak yang dekat dengan banyak kotak kosong.

8. *Parity*, maksudnya adalah pemain merebut keadaan sebagai pemain yang terakhir melangkah pada akhir permainan. Keadaan tersebut merupakan keuntungan bagi pemain karena biji yang dibalikkan merupakan biji yang pasti stabil karena akhir permainan dan sebagai penentu siapa pemenang dari permainan.
9. *Unbalanced edges*, maksudnya adalah ketidakseimbangan posisi biji pada papan. Misalnya pemain putih memiliki banyak biji pada dua tepi yang berdekatan. Putih mengambil inisiatif untuk terlebih dahulu merebut salah satu *corner* pada dua tepi tersebut. Hal ini menyebabkan putih memiliki *stable disc* pada salah satu tepi. Namun, biji putih yang terdapat pada tepi yang lain yang dekat dengan *corner*. Dengan menempatkan biji pada tepi tersebut, hitam dapat menguasai tepi tersebut dengan mengambil yang tersisa dan membalikkan banyak biji putih pada tepi tersebut. Dengan demikian hitam juga memiliki *stable disc* pada tepi tersebut, yang sebelumnya dikuasai oleh putih.

2.4 Deskripsi Umum Aplikasi Othello 2007

Aplikasi Othello 2007 [IAM07] merupakan aplikasi yang dimainkan oleh dua pemain, yaitu komputer dan manusia. Komputer (*Player 1*) diasosiasikan dengan biji warna putih dan manusia (*Player 2*) diasosiasikan dengan biji warna hitam. Tujuan utama dari permainan adalah membalikkan biji lawan sesuai dengan warna biji pemain sebanyak-banyaknya.

Berikut ini merupakan gambar dari permainan othello 2007 ketika pertama kali dijalankan.



Gambar 11 Tampilan Permainan Othello 2007

Program permainan Othello 2007 memiliki tiga menu, yaitu *New Game*, *Mode*, dan *Exit*. Ketika menu *New Game* di klik, akan ditampilkan tiga submenu yang berupa level permainan pada aplikasi Othello 2007. Submenu tersebut adalah:

1. Level *Easy* merupakan permainan dengan tingkat kesulitan paling rendah dalam permainan Othello 2007. Level ini memiliki kedalaman pohon sebanyak 3 kedalaman.
2. Level *Medium* adalah level dengan tingkat kesulitan menengah yang memiliki kedalaman pohon sebanyak 5 kedalaman.
3. Level *Hard* yaitu level yang memiliki tingkat kesulitan paling tinggi dengan kedalaman pohon sebanyak 7 kedalaman.

Menu Mode merepresentasikan jumlah kotak kosong yang dipilih oleh pemain apabila pemain tidak ingin bermain mulai dari awal permainan. *Menu Exit* dipilih ketika pemain ingin keluar dari permainan.

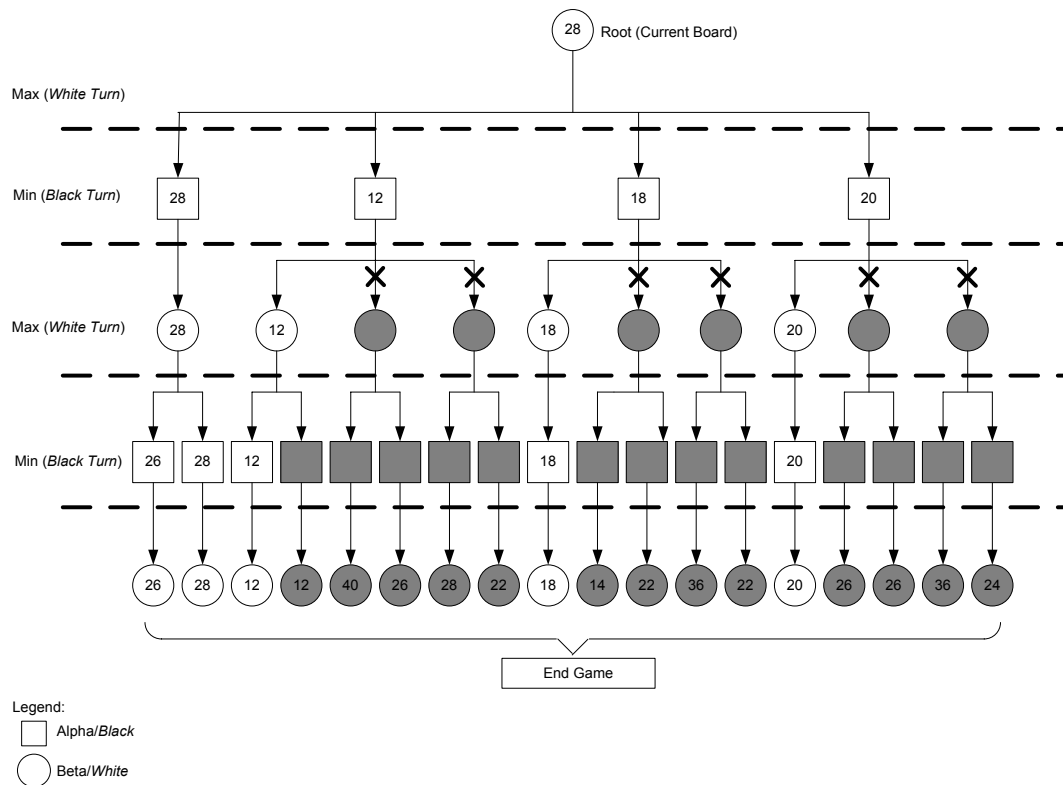
2.5 Kelemahan aplikasi

Kelemahan dari aplikasi Othello 2007 adalah lambatnya permainan di tengah permainan yang diakibatkan oleh banyaknya cabang yang dicari di setiap simpulnya. Banyaknya pencarian ini mengakibatkan *response time* menjadi lambat di saat menjalankan aplikasi.

2.6 Penerapan Algoritma Minimax Alpha-Beta Pruning

Algoritma minimax alpha-beta *pruning* adalah algoritma yang digunakan untuk melakukan pencarian terhadap simpul yang diterapkan pada aplikasi Othello 2007. Algoritma digunakan untuk mengambil keputusan terbaik, yaitu berupa langkah yang tepat bagi komputer, sehingga komputer tidak mudah dikalahkan oleh lawan. Prinsip kerja algoritma ini adalah dengan melakukan pemangkasan terhadap cabang *tree* pada saat pencarian dan pengambilan langkah yang tepat. Pemangkasan dilakukan apabila cabang *tree* tersebut tidak memiliki nilai yang lebih baik daripada cabang yang telah diperiksa sebelumnya [IAM07].

Berdasarkan contoh pada gambar 12, nilai 28 merupakan langkah terbaik dalam menentukan langkah selanjutnya. Sedangkan langkah yang disilang merupakan langkah yang diabaikan. Penjelasan minimax Alpha-Beta *Pruning* dapat dibaca pada dokumen rujukan [IAM07]. Gambar tersebut mengilustrasikan pohon pencarian pada permainan Othello yang posisi papannya dapat dilihat pada lampiran D.



Gambar 12 Pohon pencarian pada permainan Othello [IAM07]

2.7 Algoritma Minimax Alpha-Beta Pruning

Meskipun algoritma *basic* Minimax merupakan algoritma yang mampu memberikan solusi terbaik untuk pengambilan keputusan, namun *response time* yang lama merupakan kelemahan dari algoritma ini [IAM07]. Keterbatasan ini disebabkan algoritma Minimax memiliki *tree* yang sangat dalam, dan hal ini membutuhkan *response time* yang lama untuk memeriksa setiap cabang *tree*. Algoritma Minimax Alpha-Beta Pruning merupakan prosedur untuk mengurangi jumlah perhitungan dan pencarian pada metoda Minimax [MAB05]. Pengurangan perhitungan dan pencarian dilakukan pada simpul yang tidak mungkin mempengaruhi nilai yang sudah ada pada simpul induknya. Pencarian pada Alpha-Beta Pruning bekerja dengan cara *depth-first*.

Algoritma Minimax Alpha-Beta Pruning menggunakan dua buah variabel untuk menyimpan nilai, yaitu *alpha* dan *beta*. Nilai *alpha* merupakan nilai inisial atau sementara yang diasosiasikan pada simpul Max. Karena simpul Max mengambil nilai maksimum dari simpul anak, maka nilai *alpha* tidak akan pernah dapat berkurang, tetapi hanya dapat bertambah. Nilai *beta* juga merupakan nilai inisial atau sementara yang diasosiasikan pada simpul Min. Karena simpul Min mengambil nilai minimum dari simpul anak, maka nilai

beta tidak akan pernah dapat bertambah, tetapi hanya dapat berkurang. Sebagai contoh, simpul Max memiliki nilai $\alpha = 6$, dengan demikian pencarian dapat mengabaikan nilai dari simpul anak, yaitu nilai dari keturunan simpul Min yang memiliki nilai *score* yang lebih kecil atau sama dengan 6. Jika nilai Max lebih besar atau sama dengan Min, maka nilai Min diabaikan dan tidak perlu dilakukan pencarian lebih jauh pada simpul Min. Hal ini disebut dengan beta *pruned*. Sebaliknya, jika Min lebih kecil atau sama dengan Max, maka nilai Max diabaikan. Kondisi seperti ini disebut alpha *pruned*.

2.8 Kesimpulan

Aplikasi Othello 2007 telah menggunakan algoritma minimax Alpha-Beta *pruning*. Namun pada saat dijalankan, *response time* dari aplikasi belum menunjukkan performansi yang optimal. Penyebabnya adalah semakin banyak jumlah simpul yang dijalani, *response time* akan menjadi semakin lambat. Sebaliknya, semakin sedikit jumlah simpul yang dijalani, *response time* menjadi semakin cepat.

Bab III

Pengembangan Fungsi Evaluasi

Kecepatan *response time* dalam permainan Othello sangat diperlukan. Yang dimaksud dengan kecepatan dalam permainan ini adalah bagaimana komputer dapat memberikan *response* yang cepat dalam permainan. Tetapi, selain cepat dalam memberikan *response*, komputer juga mampu memilih langkah terbaik, sehingga tidak mudah dikalahkan oleh lawan.

3.1 Mekanisme Umum Fungsi evaluasi

Fungsi evaluasi yang ditambahkan pada aplikasi Othello 2007 bertujuan untuk menghasilkan *response time* lebih cepat dengan cara menambah kemampuan Minimax Alpha Beta *Pruning* untuk memotong simpul tertentu. Fungsi evaluasi dirancang untuk menerapkan strategi yang dijelaskan pada bab 3.2.

3.2 Strategi Permainan Othello yang Dikembangkan

Strategi permainan Othello yang dikembangkan sebagai fungsi evaluasi diambil dari literatur Tugas Akhir. Strategi-strategi tersebut adalah:

1. *Corners and Stable Disc* [BRO05], merupakan strategi dasar. Strategi ini bertujuan untuk menguasai *corner* lebih dahulu dibanding lawan dan mengambil langkah yang menyebabkan biji pemain menjadi stabil (warna biji pemain tetap hingga akhir permainan) karena tidak dapat diabolikkan oleh pemain lawan dan menguasai *corner* lebih dahulu dibanding lawan. Dengan merebut *corner*, maka kemungkinan menghasilkan *stable disc* semakin besar.
2. *Mobility* [ELA93], maksudnya adalah mengambil langkah yang menyebabkan pemain lawan hanya memiliki sesedikit mungkin langkah, terutama jika langkah yang mungkin tersebut merupakan langkah yang merugikan.

3.3 Fungsi Evaluasi

Fungsi evaluasi yang dikembangkan ada sebanyak tiga fungsi yaitu posisi biji, *stable disc*, dan *possible moves*. Masing-masing fungsi tersebut menghasilkan nilai evaluasi. Ketiga fungsi evaluasi tersebut kemudian digabungkan dengan cara menambahkan ketiga hasil perhitungan masing-masing fungsi evaluasi (perhitungan posisi biji + perhitungan *stable disc* + perhitungan *possible moves*).

3.3.1 Perhitungan Posisi Biji

Pada penjelasan sebelumnya disebutkan bahwa posisi biji terbagi atas empat daerah. Masing-masing daerah diberi nilai. Tujuan pemberian nilai pada masing-masing posisi biji adalah untuk memberikan keputusan yang mutlak pada nilai yang dihasilkan oleh Minimax Alpha Beta *Pruning*. Berikut penjelasan mengenai daerah-daerah yang ditentukan pada posisi biji.

a. Pemberian Nilai *Corner*

Nilai *corner* sebesar 260 diperoleh dari 65 (lebih besar dari 64) dikalikan dengan 4 (jumlah *corner*). Nilai -65 dan 65 merupakan rentang nilai alpha dan beta yang dipilih karena nilai ini tidak mungkin dihasilkan dalam aplikasi Othello. Nilai maksimum dan minimum yang dapat dihasilkan dalam aplikasi Othello adalah -64 dan 64.

b. Pemberian Nilai *C-Square*

Berdasarkan rentang nilai alpha dan beta yang telah dijelaskan sebelumnya, diberikan nilai terhadap *C-Square* sebesar -65 (lebih kecil dari -64). Nilai ini diperoleh dari nilai minimum alpha dan beta. Alasan diberikannya nilai minimum pada *C-Square* adalah karena *C-Square* merupakan posisi yang merugikan bagi pemain. Ketika pemain memilih melangkah ke *C-Square*, akan membuka peluang yang besar bagi lawan untuk merebut *corner* dan menguasai permainan.

c. Pemberian Nilai *X-Square*

Nilai -130 pada *X-Square* diperoleh dari nilai minimum alpha dan beta (-65) dikalikan dengan 2. *X-Square* merupakan posisi yang lebih merugikan bagi pemain jika dibandingkan dengan *C-Square*. Inilah yang menjadi alasan mengapa *X-Square* diberi nilai lebih rendah dari *C-Square*.

d. Pemberian Nilai *A-Square* dan *B-Square*

Nilai 32 pada *A-Square* dan *B-Square* diperoleh dari hasil perkalian jumlah kotak *A-Square* dan *B-Square* sebanyak 16 dikali dengan 2. *A-Square* dan *B-Square* diberi nilai 32 karena posisi tersebut merupakan posisi yang stabil jika dipilih oleh pemain. Posisi ini tidak memberikan pengaruh yang besar bagi lawan untuk menguasai permainan jika pemain memilihnya.

3.3.2 Perhitungan Stable Disc

Seperti pada posisi biji, *stable disc* juga diimplementasikan dengan memberikan nilai pada status dari sebuah biji, yaitu *stable disc* atau tidak. Pemberian nilai pada *stable disc* adalah sebagai berikut.

1. Biji yang tidak dapat diambil alih oleh lawan (*Stable Disc*) diberi nilai 260.
2. Biji yang masih dapat diambil alih oleh lawan (*Non-Stable Disc*) diberi nilai 0.

Nilai 260 pada *stable disc* diperoleh dari nilai yang digunakan pada perhitungan posisi biji dengan tujuan ketiga fungsi evaluasi menggunakan nilai yang sama. Sedangkan nilai 0 digunakan karena biji yang tidak stabil tidak memberikan pengaruh yang besar pada permainan (kondisi normal).

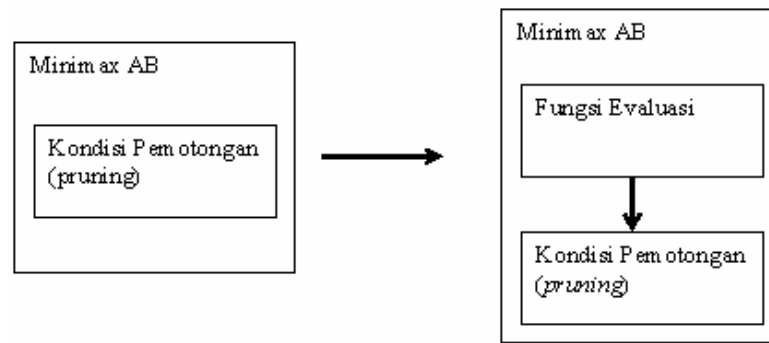
3.3.3 Perhitungan Possible Moves

Pemain hanya dapat melangkah jika memiliki kemungkinan langkah (*legal moves*). Semakin banyak *legal moves* yang dimiliki oleh seorang pemain, maka semakin besar pula peluang untuk menguasai permainan. Keuntungan akan semakin besar apabila pemain lawan tidak memiliki *legal moves* sama sekali.

Perhitungan jumlah langkah yang tersedia diimplementasikan dengan memberikan nilai pada beberapa kondisi tertentu dengan rincian sebagai berikut.

1. Jumlah langkah yang tersedia sebanyak 0 atau 1 langkah diberi nilai 260.
2. Jumlah langkah yang tersedia sebanyak 2 hingga 4 langkah diberi nilai 0.
3. Jumlah langkah yang tersedia lebih besar dari 5 diberi nilai -32.

Nilai 260 pada *possible moves* diperoleh dari nilai yang digunakan pada perhitungan posisi biji. Nilai 0 digunakan karena biji yang tidak stabil tidak memberikan pengaruh yang besar pada permainan (kondisi normal). Nilai -32 diperoleh dari nilai *A-Square* dan *B-Square* yang telah dinegatifkan. Alasannya adalah jumlah *possible moves* tersebut memiliki pengaruh yang cukup besar pada permainan.



Gambar 13 Penambahan fungsi evaluasi pada Minimax Alpha-Beta *Pruning*

Pemanggilan fungsi evaluasi dilakukan pada algoritma minimax Alpha-Beta *Pruning* sebelum dilakukan pemeriksaan terhadap kondisi *pruning*, seperti ditunjukkan pada gambar 14. Setiap langkah yang diambil telah terlebih dahulu dievaluasi oleh fungsi evaluasi kemudian dilakukan pemeriksaan terhadap kondisi *pruning*. Penambahan fungsi evaluasi tersebut memungkinkan terjadinya *pruning* yang lebih awal/banyak. Dengan demikian kemungkinan langkah yang dievaluasi menjadi lebih sedikit dan waktu evaluasi menjadi lebih cepat.

Bab IV

Implementasi Fungsi Evaluasi

Fungsi evaluasi pada permainan Othello terdiri atas tiga fungsi. Fungsi-fungsi tersebut adalah posisi biji (*disc position*), jumlah biji yang tidak dapat diambil alih oleh lawan (*stable disc*), dan jumlah langkah yang mungkin untuk dipilih (*possible moves*).

Penambahan fitur fungsi evaluasi pada permainan Othello 2007 dilakukan dengan cara menggabungkan ketiga fungsi yang telah dijelaskan sebelumnya. Penggabungan fungsi evaluasi dilakukan dengan cara memanggil masing-masing fungsi dan kemudian nilai dari setiap fungsi disimpan oleh tiga variabel yang berbeda. Ketiga nilai variabel tersebut ditambahkan dan disimpan kedalam sebuah variabel lain.

Variabel yang menyimpan nilai total penambahan dari pemanggilan fungsi evaluasi merupakan akhir dari penggunaan fungsi evaluasi. Hasil tersebut kemudian digunakan untuk memberikan perubahan nilai pada alpha dan beta yang mempengaruhi pengambilan langkah terbaik. Perubahan nilai alpha dilakukan dengan mengganti nilai alpha menjadi nilai total fungsi evaluasi. Sedangkan perubahan nilai beta dilakukan dengan mengganti nilai beta menjadi negatif nilai total fungsi evaluasi.

```
Private Function CheckIfStable(tempCheckGrid() As Integer, X As Integer,
Y As Integer) As Integer
    CheckIfStable = 0
    GridStableDisc(X, Y) = WHITE
    If EvaluateStableDisc(tempCheckGrid(), (X), (Y)) Then
        CheckIfStable = 260
    Else
        CheckIfStable = 0
    End If
    GridStableDisc(X, Y) = 0
End Function
```

Kode 1 Fungsi Check Stable Disc

Kode 1 di atas merupakan implementasi *stable disc* pada aplikasi Othello 2008. Fungsi *CheckIfStableDisc* memeriksa biji yang diletakkan pada setiap langkah. Jika langkah menghasilkan *stable disc*, maka fungsi menghasilkan nilai 260. Sedangkan jika langkah tidak menghasilkan *stable disc*, maka fungsi menghasilkan nilai 0.

Diberikan sebuah fungsi yang dapat menentukan jenis posisi seperti *corner*, *C-Square*, *X-Square*, *A-Square*, dan *B-Square* yang dapat dilihat pada Kode 2.

```

Private Function DiscPosition(TEMP_ROW As Integer, TEMP_COL As Integer)
As Integer
    Dim Value As Integer, cellnum As Integer

    TEMP_ROW = TEMP_ROW + 1
    TEMP_COL = TEMP_COL + 1
    cellnum = (TEMP_COL - 1) * 8 + TEMP_ROW

    If IsCorner(cellnum) Then
        Value = 260
    ElseIf IsCSquare(cellnum) Then
        Value = -65
    ElseIf IsXSquare(cellnum) Then
        Value = -130
    ElseIf IsABSquare(cellnum) Then
        Value = 32
    Else
        Value = 0
    End If

    Evaluate = Value
End Function

```

Kode 2 Fungsi *Disc Position*

Kode 3 merupakan implementasi *possible moves* pada aplikasi Ohtello 2008. Fungsi *PossibleMoves* memeriksa dan menghitung jumlah *legal moves* yang dihasilkan oleh pemain pada saat memilih langkah. Fungsi tersebut menghasilkan nilai sesuai dengan jumlah *legal moves*. Semakin sedikit jumlah *legal moves*, semakin besar nilai yang dikembalikan. Sebaliknya, semakin banyak jumlah *legal moves*, semakin kecil nilai yang dikembalikan.

```

Function PossibleMoves(gridTemp() As Integer) As Integer
Dim row As Integer, col As Integer 'gridTemp = papan permainan Othello
Dim AbleMovesNum As Integer 'AbleMovesNum = kemungkinan langkah
AbleMovesNum = 0
For CheckRow = 0 To MaxRows - 1
    For CheckCol = 0 To MaxCols - 1
        If gridTemp(CheckRow, CheckCol) = 3 Then
            AbleMovesNum = AbleMovesNum + 1
        End If
    Next CheckCol
Next CheckRow
If AbleMovesNum = 0 Or AbleMovesNum = 1 Then
    PossibleMoves = 260
ElseIf AbleMovesNum >= 2 And AbleMovesNum <= 4 Then
    PossibleMoves = 0
ElseIf AbleMovesNum >= 5 Then
    PossibleMoves = -32
End If
End Function

```

Kode 3 Fungsi *Possible Moves*

Fungsi evaluasi yang dikembangkan diletakkan didalam fungsi minimax Alpha-Beta *Pruning*. Kode 4 dipanggil ketika papan Othello 2008 diklik oleh pemain hitam (manusia). Fungsi evaluasi akan mengeksekusi fungsi *Check Stable Disc*, *Disc Position*,

dan *Possible Moves*. Setelah nilai dari tiap fungsi diperoleh, dilakukan penjumlahan nilai dari masing-masing fungsi yang. Total nilai tersebut ditampung pada variabel `totalEval` dan selanjutnya digunakan untuk mengubah nilai alpha dan beta.

```
'3 fungsi evaluasi dipanggil terpisah
evalSD = CheckIfStable(tempArray(), row, col)
evalPM = PossibleMoves(tempArray())
evalDP = DiscPosition((row), (col))

'penjumlahan fungsi evaluasi
totalEval = evalDP + evalPM + evalSD

If (val < beta) Then
beta = val: minimaxAB = beta
If (totalEval > beta) Then beta = -totalEval: minimaxAB = beta
Else: minimaxAB = val
End If
```

Kode 4 Pemanggilan Fungsi Evaluasi

Bab V

Pengujian Fungsi Evaluasi

Pengujian fungsi evaluasi dilakukan untuk mendapatkan *response time* yang dihasilkan oleh Othello 2007 yang kemudian dibandingkan dengan Othello 2008. Hasil pengujian tersebut digunakan untuk pencapaian tujuan Tugas Akhir ini.

5.1 Fitur Pengujian Response Time

Untuk melakukan pengujian terhadap *response time* pada aplikasi Othello 2007 dan Othello 2008, sebelumnya ditambahkan sebuah variabel yang dapat menampung data waktu yang diperlukan komputer dalam permainan. Data waktu tersebut diperoleh dengan memanfaatkan fungsi `GetTickCount`.

5.2 Sampel Pengujian

Untuk melakukan pengujian, terlebih dahulu ditentukan beberapa sampel yang akan digunakan untuk menguji aplikasi Othello 2007 dan Othello 2008. Sampel yang digunakan ada sebanyak tujuh sampel. Tujuh sampel tersebut dijadikan acuan untuk membandingkan *response time* pada aplikasi Othello 2007 dan Othello 2008. Karakteristik pada masing-masing sampel yaitu:

1. Sampel 1: Hitam menang, merupakan sampel yang memilih langkah dengan cara menghadang posisi hitam (*player 2*). Pada sampel ini *possible move*, posisi biji, dan *stable disc* tidak terlalu diperhatikan.
2. Sampel 2: Putih menang, merupakan sampel yang memilih langkah untuk mengepung biji lawan. Hal ini dimaksudkan untuk membentuk *stable disc* pada aplikasi Othello.
3. Sampel 3: Putih menang, merupakan sampel yang lebih memilih posisi biji dibandingkan *possible move* dan *stable disc*. Hal ini dimaksudkan untuk menempatkan biji ke kotak yang memiliki nilai terbesar.
4. Sampel 4: Putih menang, merupakan sampel yang memilih langkah untuk memperkecil kemenangan lawan. Strateginya adalah dengan menempatkan biji ke kotak yang paling merugikan bagi lawan sehingga lawan hanya memiliki satu langkah untuk penentuan langkah berikutnya.
5. Sampel 5: Putih menang, merupakan sampel yang memilih langkah dengan cara mengepung langkah lawan, memprioritaskan langkah menuju *corner* dan menyiasati lawan untuk memiliki langkah yang merugikan dirinya sendiri.

6. Sampel 6: Putih menang, merupakan sampel yang memilih langkah yang mementingkan posisi biji dan *stable disc*. Hal ini dimaksudkan agar lawan memilih langkah yang dapat merugikan dirinya sendiri.
7. Sampel 7: Putih menang, merupakan sampel yang memilih langkah untuk menyiasati lawan untuk melakukan langkah *pass*.

5.3 Komputer Pengujian

Dalam melakukan pengujian, penguji harus menggunakan dua komputer. Hal ini dimaksudkan karena beberapa proses menggunakan memori yang sama dan apabila satu proses dieksekusi maka yang proses yang lainnya harus menunggu. Oleh sebab itu, pada saat pengujian harus menggunakan dua komputer di mana spesifikasi masing-masing komputer adalah sebagai berikut:

1. Prosesor Pentium IV
2. Harddisk 30 GB
3. Memory 512 MB

5.4 Pengukuran *response time* Othello 2007

Pengukuran *response time* pada Othello 2007 dilakukan dengan menambahkan fungsi untuk menghitung *response time* pada saat *player 2* (hitam) melangkah atau *pass* (setelah tombol *pass* ditekan).

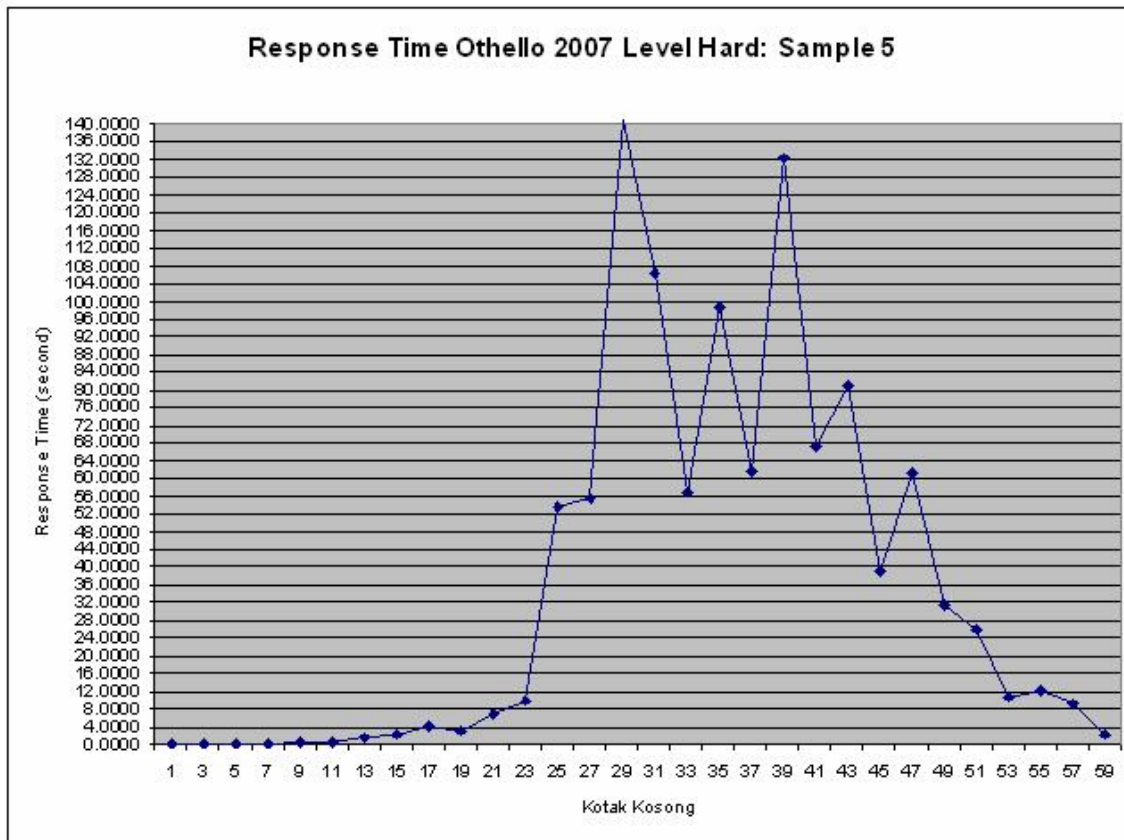
1. *Response time* yang dihasilkan ditampilkan dalam sebuah *list*, bersamaan dengan langkah-langkah yang diambil.
2. *List* yang dihasilkan dipindahkan ke dalam **MS. Excel**.
3. Data yang terdapat di MS. Excel dipindahkan ke **MS. Word**.

5.5 Pengukuran *response time* Othello 2008

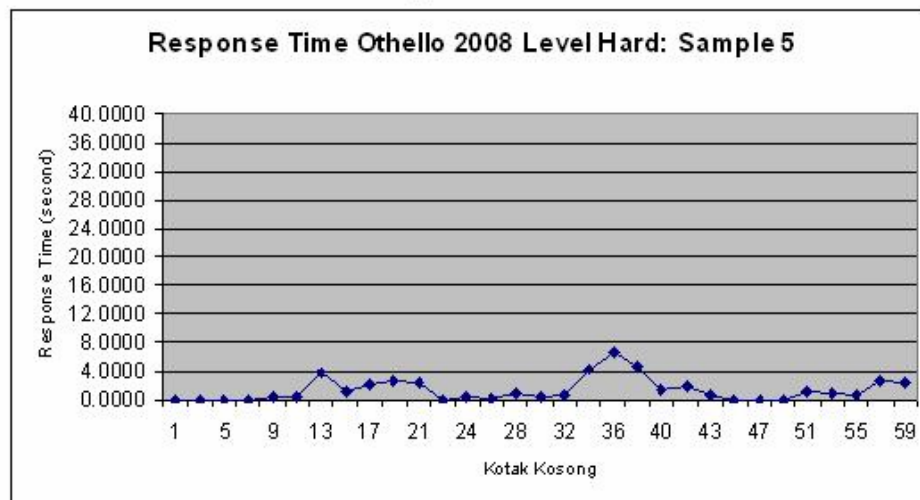
Pengukuran *response time* pada aplikasi Othello 2008 dilakukan sama seperti pada aplikasi Othello 2007, yaitu dengan menambahkan variabel untuk menyimpan *response time*. Langkah yang diambil dalam mengarahkan biji Othello pada saat pengukuran *response time* sama dengan langkah biji yang diambil pada saat melakukan pengukuran *response time* terhadap Othello 2007. Hasil pengukuran *response time* untuk setiap sampel dicatat ke dalam tabel kemudian dari tabel tersebut dihasilkan grafik yang menunjukkan *response time* dari Othello 2008.

5.6 Perbandingan Grafik Othello 2007 dan Othello 2008

Perbandingan *response time* pada Othello 2007 dan Othello 2008 memiliki rentang waktu yang sangat signifikan. Hal ini terlihat dari bentuk grafik antara kedua aplikasi Othello tersebut. Gambar 14 merupakan grafik perbandingan antara Othello 2007 dan Othello 2008 untuk sampel kelima.



(a)



(b)

Gambar 14 Perbandingan Level Hard Sample 5

Gambar 14a merupakan grafik dari aplikasi Othello 2007 level *hard* untuk sampel 5. Pada grafik ditunjukkan *response time* yang lambat dari awal permainan hingga berakhirnya permainan. Pada awal permainan *response time* masih cepat, namun pada saat pertengahan *response time* menjadi lambat. Rentang waktu untuk othello 2007 adalah 0.016 hingga 140.578 detik.

Gambar 14b merupakan grafik dari Othello 2008 level *hard* untuk sampel ketujuh. Pada grafik ditunjukkan *response time* yang cepat di awal hingga akhir permainan. Rentang *response time* aplikasi Othello 2008 yaitu antara 0.0006 hingga 6.703 detik.

Dengan ditambahkannya fungsi, langkah yang dijalankan komputer menjadi lebih baik dibandingkan aplikasi Othello 2007. Grafik tentang sampel yang lainnya dimuat pada lampiran A (Grafik perbandingan Othello 2007 dan Othello 2008 level *easy*), lampiran B (Grafik perbandingan Othello 2007 dan Othello 2008 level *medium*), dan lampiran C (Grafik perbandingan Othello 2007 dan Othello 2008 level *hard*).

Pada tabel 1 ditunjukkan data *number of recursive* dan *response time* yang diambil dari tujuh sampel data yang dijadikan sebagai acuan pengujian.

Tabel 1 Perbandingan Response Time dan Number of Recursive Othello 2007 dan Othello 2008

Level	Sampel	Number of Recursive		Response Time	
		Othello 2007	Othello 2008	Othello 2007	Othello 2008
Easy	1	1-271	1-280	0.015-0.063	0.012-0.07
	2	1-90	7-313	0.015-0.094	0.012-0.1
	3	1-95	1-444	0.015-0.094	0.012-0.141
	4	2-46	17-242	0.016-0.078	0.012-0.08
	5	1-86	1-331	0.015-0.094	0.012-0.08
	6	1-91	1-470	0.015-0.093	0.012-0.12
	7	1-93	1-556	0.015-0.125	0.012-0.15
Medium	1	1-83	14-2549	0.031-1.156	0.031-1.156
	2	1-86	1-8874	0.015-2.422	0.015-2.422
	3	1-93	1-8875	0.031-1.578	0.031-1.578
	4	1-86	1-4725	0.015-0.984	0.015-0.984
	5	1-95	1-6845	0.015-2.641	0.015-2.641
	6	1-95	1-3698	0.016-2.156	0.016-2.516
	7	1-97	1-11771	0.015-1.578	0.015-1.578
Hard	1	1-150312	4-70131	0.047-32.891	0.006-19.515
	2	1-250742	1-85840	0.015-49.078	0.001-22.25
	3	1-76704	1-39565	0.016-20.297	0.0012-10.922
	4	1-119777	5-81723	0.015-33.953	0.006-6.703
	5	1-763198	1-24275	0.016-140.578	0.006-6.703
	6	1-240687	1-48563	0.016-47.859	0.006-6.703
	7	1-218640	63-12752	0.016-45.234	0.031-3.953

Bab VI

Kesimpulan dan Saran

6.1 Kesimpulan

Dari pelaksanaan tugas akhir dapat disimpulkan beberapa hal sebagai berikut:

1. Penambahan fungsi evaluasi dapat mempercepat *response time* pada aplikasi Othello 2008.
2. Penambahan fungsi evaluasi dapat menghasilkan *response time* pada rentang 0 hingga 4 detik, namun masih terdapat anomali pada kondisi tertentu.
3. Fungsi evaluasi yang ditambahkan merupakan penerapan dari strategi permainan Othello yang diacu dari literatur Tugas Akhir.
4. Pengukuran *response time* pada aplikasi Othello 2007 dan Othello 2008 harus dilakukan dengan menggunakan dua komputer yang berbeda dengan spesifikasi yang sama. Pengukuran juga dapat dilakukan pada satu komputer dengan cara aplikasi harus dijalankan secara bergantian.
5. Fungsi evaluasi memungkinkan pemotongan yang lebih banyak.
6. Pemberian nilai yang diterapkan pada fungsi evaluasi semula diperoleh dari *disc position*, kemudian diterapkan pada *stable disc* dan *possible moves*.

6.2 Saran

Dari hasil pelaksanaan tugas akhir ini dapat dikemukakan beberapa saran, diantaranya:

1. Selama pengerjaan Tugas Akhir ini, analisis terhadap pohon yang dihasilkan oleh aplikasi Othello 2007 dengan menggunakan algoritma Minimax Alpha-Beta *Pruning* tidak maksimal, sehingga pemahaman terhadap pohon permainan masih kurang. Sehingga disarankan untuk pengembangan selanjutnya untuk lebih dahulu melakukan analisis yang baik sebelum dilakukan implementasi.
2. Pengujian aplikasi sebaiknya mempertimbangkan spesifikasi komputer yang digunakan dan hal-hal yang dapat mempengaruhi performansi aplikasi.