BAB 4
IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi yang dibangun. Tahapan ini dilakukan setelah analisis dan perancangan selesai dilakukan dan selanjutnya akan diimplementasikan kedalam bahasa pemograman. Setelah implementasi maka dilakukan pengujian terhadap aplikasi. Aplikasi yang telah dibangun akan diimplementasikan untuk mengetahui apakah sistem tersebut dapat berjalan sesuai dengan tujuannya atau tidak.

4.1. Implementasi

Implementasi merupakan terjemahan perancangan yang berdasarkan hasil analisis pada bab sebelumnya ke dalam bahasa pemprograman yang dapat dimengerti oleh komputer. Bahasa pemprograman yang digunakan adalah java.

4.1.1. Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk menjalankan permainan nine men's morris tidak harus yang memiliki spesifikasi tinggi. Spesifikasi minimum yang dapat digunakan untuk menjalankan permainan nine men's morris adalah sebagai berikut :

1. Prosessor dengan kecepatan 1,80 GHz
2. RAM berkapasitas 1 GB
3. VGA berkapasitas 512 MB
4. Harddisk dengan kapasitas memori kosong 32 GB
5. Monitor Standar
6. Mouse Standar
7. Keyboard Standar

4.1.2. Implementasi Perangkat Lunak

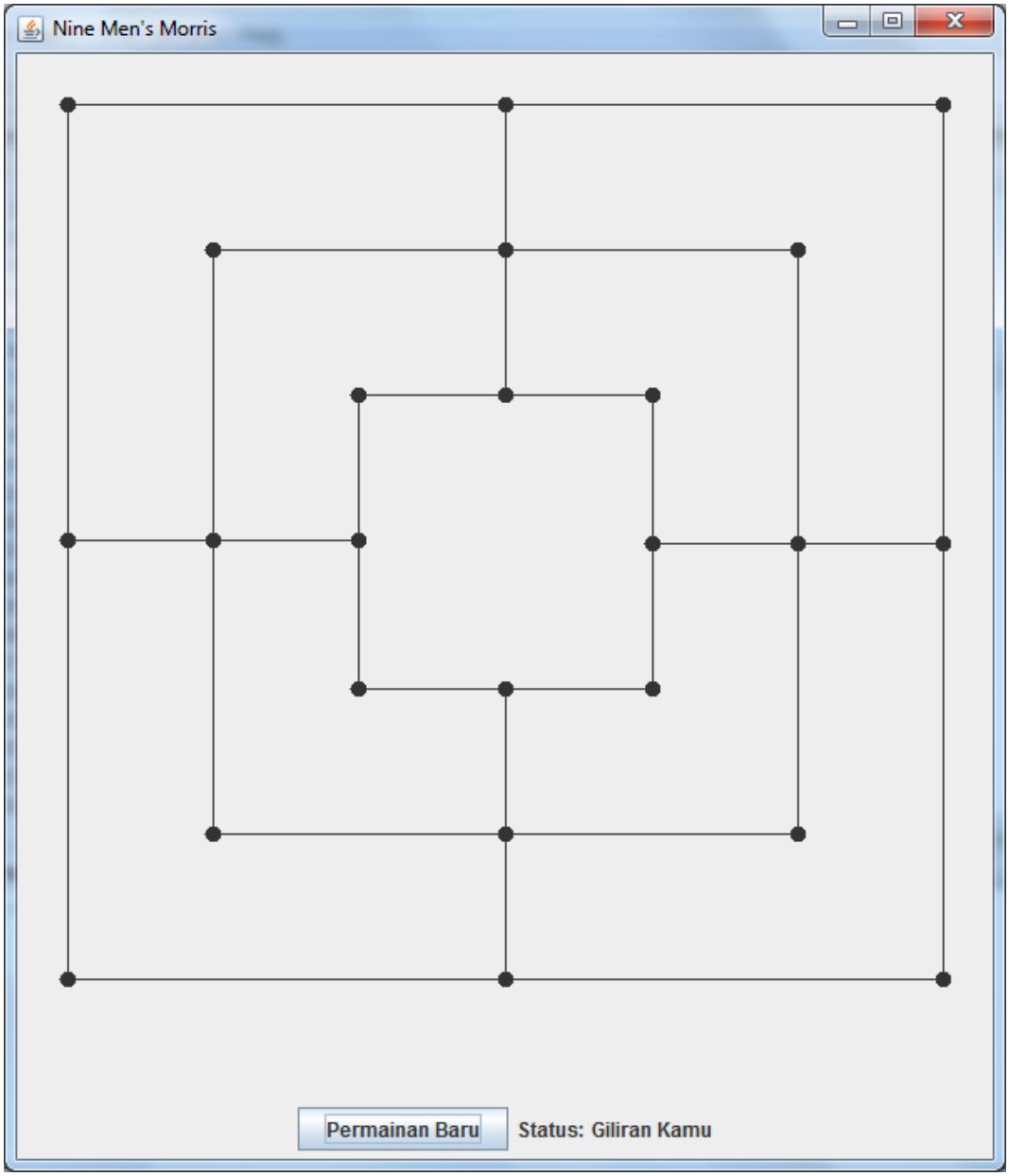
Perangkat lunak yang digunakan untuk mengimplementasikan dan enguji permainan nine men's morris adalah sebagai berikut :

1. Windows 7 Ultimate sebagai sistem operasi

2. Netbeans 7.0 sebagai perangkat lunak pembangunan

4.1.3. Implementasi Antarmuka

Untuk memperjelas bentuk dari implementasi antarmuka berikut tampilan antarmuka form utama dari program yang telah dibuat.



Gambar 4.1 Tampilan Utama Permainan Nine Men's Morris

4.2. Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang dilakukan untuk menemukan kekurangan atau kesalahan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat telah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak. Adapun metode pengujian yang digunakan pada perangkat lunak ini adalah metode pengujian *black box* dan *white box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak, sedangkan *wihite box* pengujian dengan melihat ke dalam modul untuk meneliti kode-kode program yang ada, dan menganalisis apakah ada kesalahan atau tidak.

4.2.1. Rencana Pengujian

Rencana pengujian yang akan dilakukan pada permainan nine men's morris dapat dilihat pada tabel 4.1.

Tabel 4.1 Rencana Pengujian

No.	Komponen yang diuji	Skenario pengujian	Jenis pengujian
1.	Menu Utama	Memilih Permainan Baru	*Black Box*
2.	Mengontrol Permainan	Meletakkan bidak	*Black Box*
		Menggeserkan bidak	*Black Box*
		Memindahkan bidak	*Black Box*

4.2.2. Pengujian *Black Box*

Berikut adalah hasil pengujian perangkat lunak yang sudah diimplementasikan menggunakan metode *Black Box* berdasarkan Tabel 4.1 Rencana pengujian yang telah dibuat sebelumnya :

Tabel 4.2 Hasil Pengujian *Black Box*

No.	Komponen yang diuji	Skenario pengujian	Hasil yang diharapkan	Hasil pengujian
1.	Menu Utama	Memilih Permainan Baru	Memulai permainan baru	[√] Berhasil Tidak Berhasil
2.	Mengontrol Permainan	Meletakkan bidak	Bidak terletakkan pada papan permainan	[√] Berhasil Tidak Berhasil

		Menggeserkan bidak	Bidak bergeser dari posisi awal ke posisi tujuan	[√] Berhasil Tidak Berhasil
		Memindahkan bidak	Bidak berpindah dari posisi awal ke posisi tujuan	[√] Berhasil Tidak Berhasil

4.2.3. Pengujian *White Box*

Pengujian *white box* merupakan metode perancangan *test case* yang menggunakan struktur control dari perancangan metode dalam mendapatkan test case. Adapun metode yang digunakan dalam pengujian *white box* ini adalah metode *basis path*. Metode *basis path* mengijinkan pendesain kasus uji untuk mendapatkan perkiraan lojik yang kompleks dari desain metode dan menggunakan perkiraan ini untuk mendefenisikan aliran eksekusi.

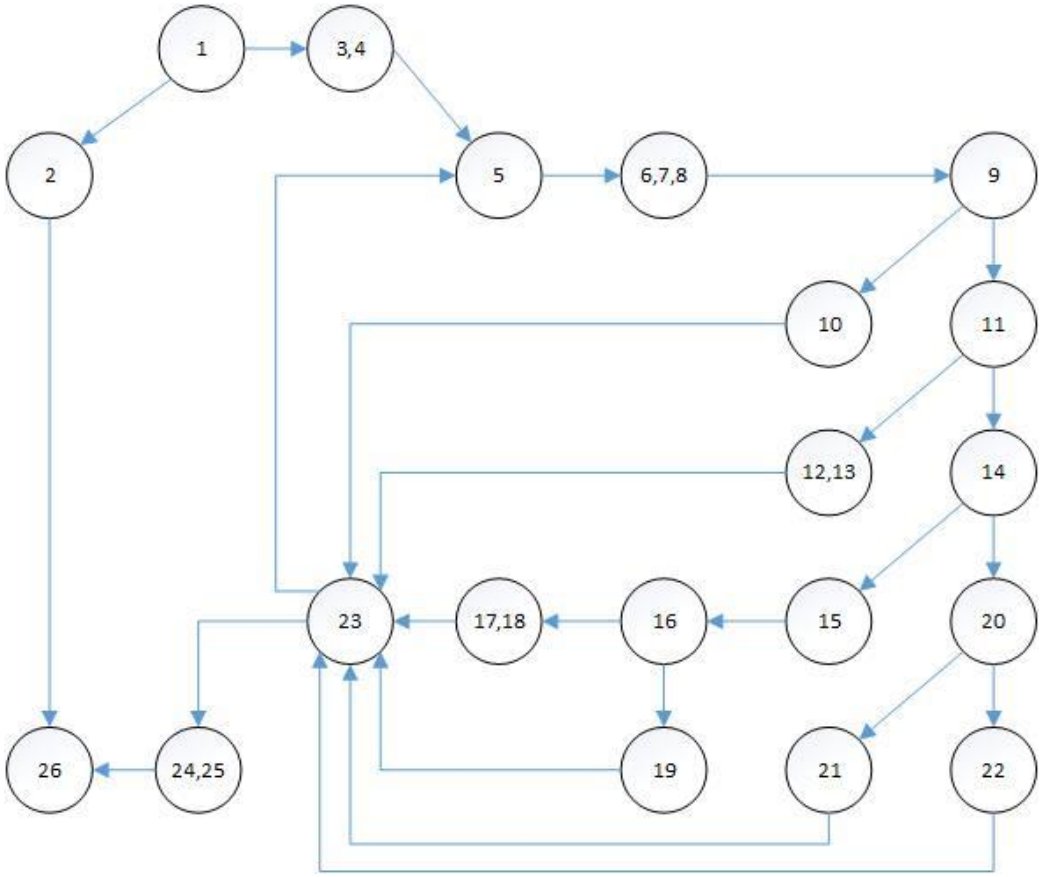
Pengujian proses algoritma negascout merupakan pengujian dari pengimplementasian algoritma negascout menggunakan pengujian white box.

Tabel 4.3 Pseudocode Algoritma Negascout

No	Pseudocode
1	`if (remainingDepth == 0) {`
2	`return evaluateCurrentBoard();`
3	`} else { Move nodeBestMove = null;`
4	`int nodeBestValue = -INFINITY;`
5	`for (Move move : validMoves) {`
6	`currentBoard.makeMove(move);`
7	`int value = -negascoutSearch(-beta, -alpha, currentDepth + 1, remainingDepth -1);`
8	`currentBoard.undoMove(move);`
9	`  if (Math.abs(value) == END_SEARCH) {`
10	`  return END_SEARCH;`
11	`} if (value > nodeBestValue) {`
12	`  nodeBestValue = value;`
13	`  nodeBestMove = move;`
14	`} if (value > alpha) {`
15	`  alpha = value;`
16	`  if (currentDepth == 0) {`
17	`  currentBestMove = move;`
18	`  currentBestMoveValue = alpha;`
19	`  }`

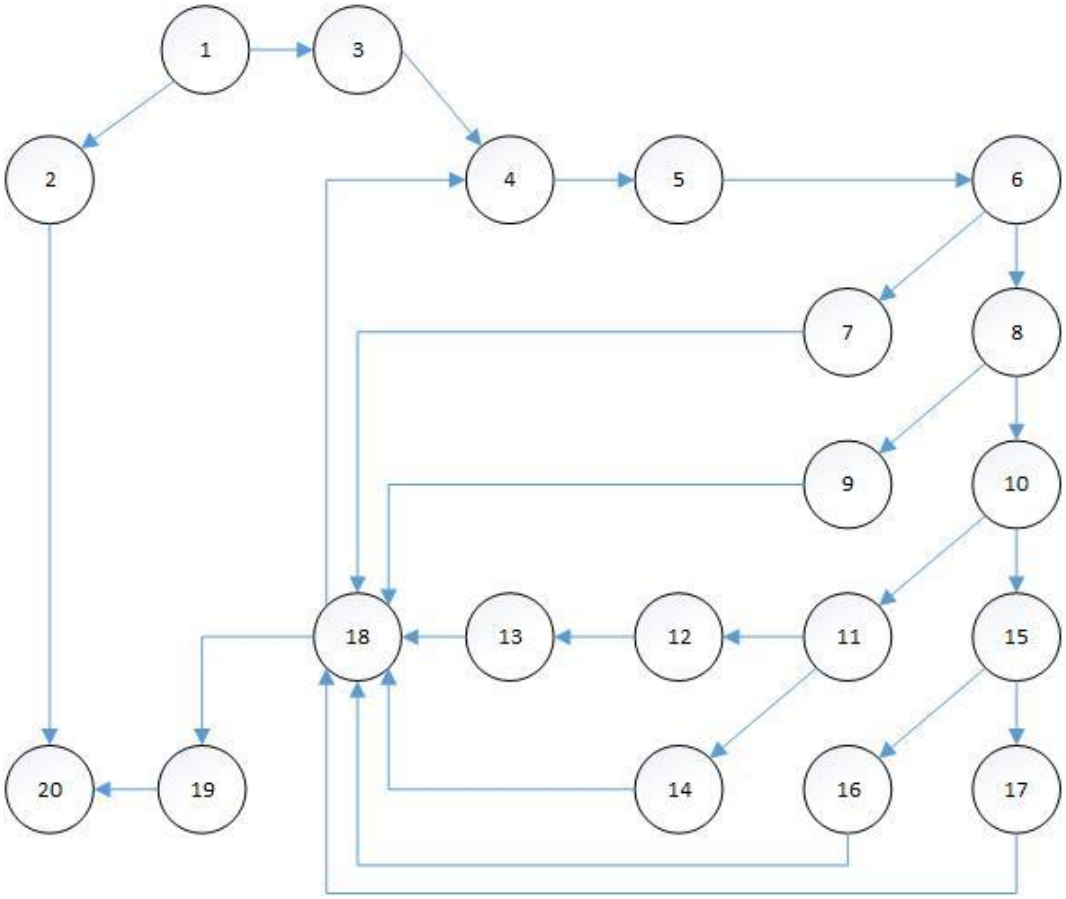
20	} if (alpha >= beta) {
21	break;
22	}
23	}
24	transpositionTable.put(currentBoard.getBoardID(), new BoardStateValue(nodeBestValue, remainingDepth, nodeBestMove, alpha >= beta, nodeBestValue < alpha));
25	return nodeBestValue;
26	}

a. *Flow Graph*



Gambar 4.2 Flowgraph Algoritma Negascout

Atau disederhanakan berdasarkan kondisi atau simpul dapat dilihat pada gambar 4.3.



Gambar 4.3 Penyederhanaan Flowgraph Algoritma Negascout

Keterangan :

= Menggambarkan kondisi

= Menggambarkan aksi

a. *Cylomatic Complexity* V(G)

V(G) = E – N + 2

V(G) = 25 – 20 + 2

V(G) = 7

Keterangan :

E = Jumlah edge pada grafik alir

N = Jumlah node pada grafik alir

b. *Independent Path*

Path 1 = 1-2-26

Path 2 = 1-3- 4-5-6-7-8-9-10-23-24-25-26

Path 3 = 1-3- 4-5-6-7-8-9-11-12-13-23-24-25-26

Path 4 = 1-3- 4-5-6-7-8-9-11-14-15-16-17-18-23-24-25-26

Path 5 = 1-3- 4-5-6-7-8-9-11-14-15-16-19-23-24-25-26

Path 6 = 1-3- 4-5-6-7-8-9-11-14-20-21-23-24-25-26

Path 7 = 1-3- 4-5-6-7-8-9-11-14-20-22-23-24-25-26

c. *Graph Matrix*

Tabel 4.4 *Graph Matrix*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	H
1		1	1																		1
2																				1	
3				1																	
4					1																
5						1															
6							1	1													1
7																		1			
8									1	1											1
9																		1			
10											1				1						1
11												1		1							1
12													1								
13																		1			
14																		1			
15																1	1				1
16																		1			
17																		1			
18																			1		
19																				1	
20																					
SUM(E)+1																					7

Keterangan:

1. Baris dan kolom merepresentasikan simpul (*node*)
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul.

Kesimpulan yang didapatkan berdasarkan hasil pengujian dihasilkan nilai Cyclomatic Complexity yang sama yaitu 7, maka dapat disimpulkan bahwa pada proses algoritma negascout pada permainan nine men's morris berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

Hasil Uji Algoritma Negascout pada permainan nine men's morris dapat dilihat pada tabel 4.5.

Tabel 4.5 Hasil Uji Algoritma Negascout

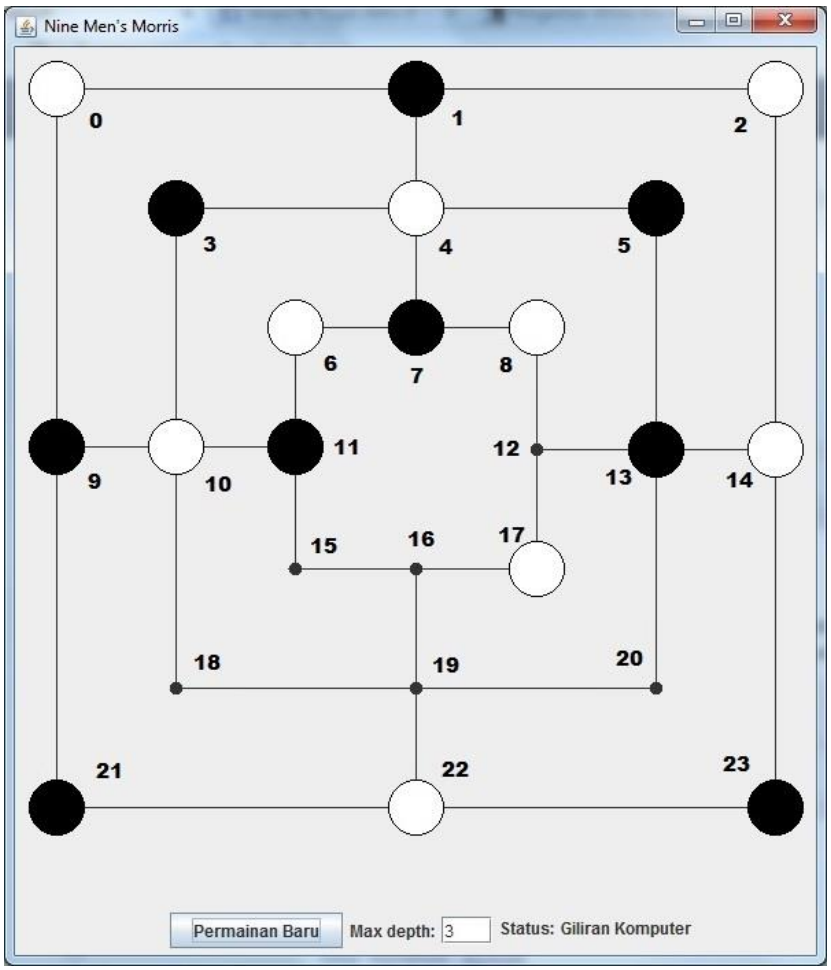
No	Kasus Uji	Yang diharapkan	Keterangan
1	remainingDepth = 0	Mengembalikan nilai evaluasi sebelumnya	[√] Berhasil Tidak Berhasil
2	• remainingDepth!=0 • move!=ValidMove • Math.abs(value)=END_SEARCH	Pencarian selesai. Mengembalikan nodeBestValue	[√] Berhasil Tidak Berhasil
3	• remainingDepth!=0 • move!=ValidMove • value>noneBestValue	nodeBestValue=value nodeBestMove=move	[√] Berhasil Tidak Berhasil
4	• remainingDepth!=0 • move!=ValidMove • value>alpha • currentDepth!=0	Alpha=value	[√] Berhasil Tidak Berhasil
5	• remainingDepth!=0 • move!=ValidMove • value>alpha • currentDepth==0	Alpha=value currentBestMove=move currentBestMoveValue=alpha	[√] Berhasil Tidak Berhasil
6	• remainingDepth!=0 • move!=ValidMove • alpha>=beta	Perulangan berhenti karena nilai alpha >= beta	[√] Berhasil Tidak Berhasil
7	• remainingDepth!=0 • move=ValidMove	Perulangan berjalan terus sampai move=ValidMove	[√] Berhasil Tidak Berhasil

Dari hasil pengujian yang dilakukan, dapat disimpulkan bahwa sistem sudah berjalan seperti yang diharapkan dan secara fungsional sudah dapat menghasilkan keluaran yang diharapkan.

4.2.4. Pengujian Algoritma Negascout Pada Nine Men's Morris

Pengujian dilakukan untuk membuktikan bahwa algoritma negascout telah bekerja sesuai rancangan. Pengujian dilakukan dengan melakukan uji pada 1 buah sampel, kemudian dilakukan pendataan secara manual.

Berikut merupakan sampel posisi papan dapat dilihat pada gambar 4.4. Pada sampel ini AI bermain sebagai bidak hitam dan mendapat giliran melangkah.



Gambar 4.4 Sampel Posisi Permainan Nine Men's Morris

Pada sampel ini AI memilih untuk melangkah 13-12 (menggeserkan bidak dari posisi 13 menuju posisi 12). Untuk menguji apakah langkah yang dipilih AI sesuai dengan yang telah dirancang, maka akan dilakukan pengujian menggunakan tabel yang berisi aliran perubahan nilai yang terjadi.

Tabel 4.5 Pohon Permainan Untuk Sampel

Tingkat			
0	1	2	3
Node a	Node b 11-15	Node e 6-11	7-6
			13-12
			13-20
			15-16
		Node f 8-12	7-8
			13-20
			15-11
			15-16
		Node g 10-11	3-10
			9-10
			13-12
			13-20
			15-16
		Node h 10-18	3-10
			9-10
			13-12
			13-20
			15-11
			15-16
		Node i 17-12	13-20
			15-11
			15-16
		Node j 17-16	13-12
			13-20
			15-11
		Node k 22-19	13-12
			13-20
			15-11
			15-16
			21-22
			23-22
	Node c 13-12	Node l 10-18	3-10
			5-13
			9-10
			11-10
			11-15
			12-13
		Node m 14-13	11-15
			23-14
		Node n 17-16	5-13
			11-15
			12-13
			12-17
		Node o 22-19	5-13
			11-15
			12-13
			21-22
			23-22
	Node d	Node p	5-13

	13-20	8-12	7-8
			11-15
			20-13
			20-19
		Node q 10-18	3-10
			5-13
			9-10
			11-10
			11-15
			20-13
			20-19
		Node r 14-13	11-15
			20-19
			23-14
		Node s 17-12	5-13
			11-15
			20-13
			20-19
		Node t 17-16	5-13
			11-15
			20-13
			20-19
		Node u 22-19	5-13
			11-15
			20-13
			21-22
			23-22

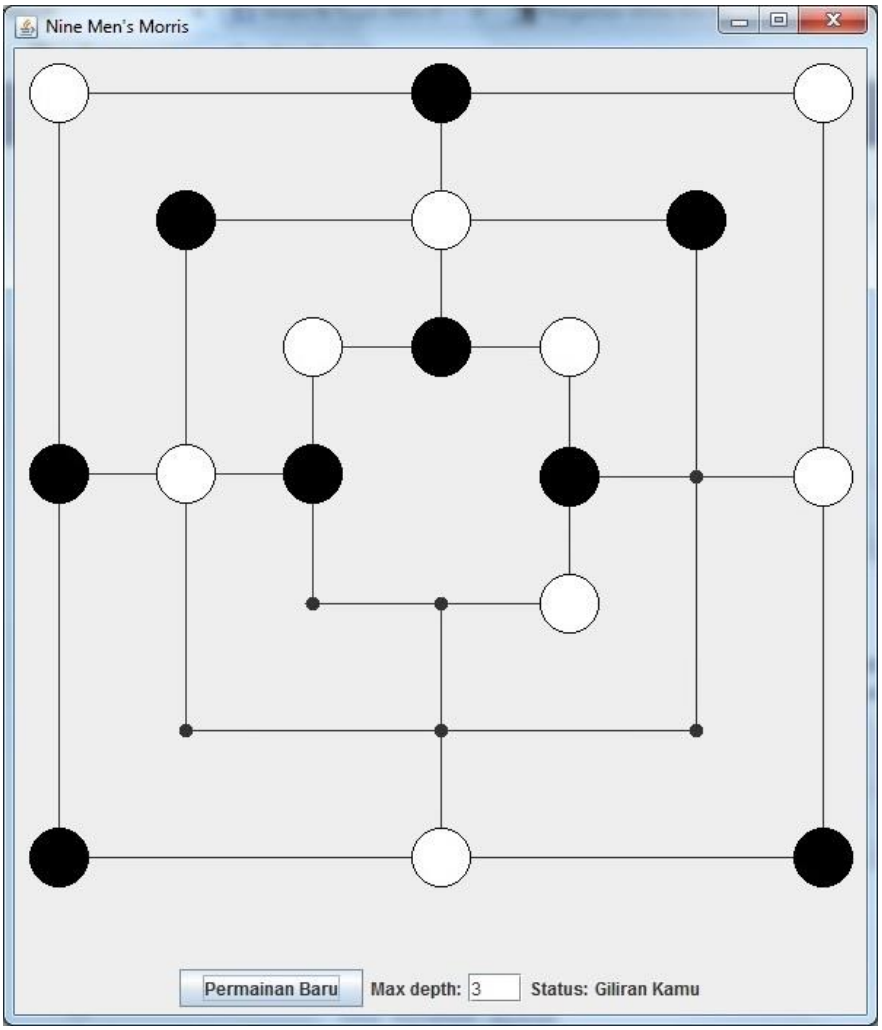
Tabel 4.5 Proses Negascout Untuk Sampel

Tingkat			
0	1	2	3
	40	40	40
			40
			30
			40
		50	40
			30
			50
			50
		50	50
			50
			50
			40
			50
		50	30
			30
			40
			30
			50
			40
		60	40
			60

			50
		50	40
			30
			50
		50	40
			30
			50
			40
			40
			30
	50	98	98
			50
			40
			40
			40
			40
		50	30
			50
		50	40
			40
			50
			40
		50	50
			50
			50
			50
			50
	50	50	30
			20
			30
			50
			50
		67	67
			30
			30
			30
			30
			50
			50
		50	30
			50
			40
		60	50
			40
			60
			60
		50	40
			30
			50
			50
		50	40
			30
			50

			40
			40

Tabel 4.5 telah menunjukkan bahwa langkah yang dipilih AI telah sesuai dengan yang ditampilkan pada tabel. Karena agen akan mendapatkan nilai 50 jika AI memilih langkah 13-12. Hasil setelah AI melangkah dapat dilihat pada gambar 4.6.



Gambar 4.6 Hasil Sampel Pengujian