BAB 3
ANALISIS DAN KEBUTUHAN ALGORITMA

3.1. Analisis Masalah

Analisis masalah dilakukan untuk mengetahui masalah – masalah apa saja yang terjadi dalam permainan nine men's morris. Berdasarkan latar belakang masalah dari mengevaluasi jurnal yang buat oleh Kevin Winata yang berjudul "Penerapan Algoritma Alpha-Beta Pruning pada Permainan Nine Men's Morris" yang menggunakan kedalaman terbatas. Kedalaman tersebut ditentukan oleh *user* atau programmer. Semakin besar kedalamannya, maka AI akan semakin kuat dengan kemungkinan menang semakin besar. Namun ketika kedalaman tersebut semakin besar, waktu berpikir AI semakin lama dalam mengambil keputusan.

3.2. Analisis *Game*

Game yang akan dibangun adalah *game* nine men's morris yang bergenre *board game*. *Game* ini dibangun 2D dan dimainkan secara *single player*.

3.2.1. *Gameplay*

Permainan ini terdiri dari tiga fase dan pada ketiganya pemain dapat membentuk mill dan mengambil bidak pemain lawan.

1) Meletakkan Bidak

   Pada fase ini, masing-masing pemain secara bergantian meletakkan bidaknya pada papan. Jika seorang pemain berhasil membentuk mill, pemain tersebut boleh mengambil bidak pemain lawan di papan. Fase ini berakhir ketika seluruh bidak telah diletakkan pada papan permainan atau diambil pemain lain.

2) Menggeserkan Bidak

   Pada fase ini, pemain dapat menggeser bidaknya secara bergiliran sesuai dengan garis yang ada. Kedua pemain dapat saling berlomba untuk membentuk mill atau menghalangi lawan membentuk mill.

3) Memindahkan Bidak

Fase ketiga dimulai ketika seorang pemain hanya tinggal memiliki tiga bidak. Dalam fase ini pemain tersebut dapat memindahkan bidaknya ke posisi manapun di dalam papan.

Permainan berakhir berdasarkan tiga kondisi, yaitu :

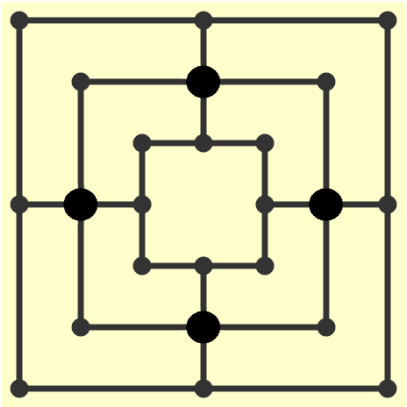
1) Kondisi pertama adalah salah satu pemain tidak bisa menggerakkan bidaknya kemanapun, maka pemain itu dinyatakan kalah.
2) Kondisi kedua adalah jika seorang pemain memiliki kurang dari tiga bidak, maka pemain itu dinyatakan kalah.
3) Kondisi ketiga adalah ketika posisi seluruh bidak terulang, maka permainan dinyatakan seri.

3.3. Analisis Masukan

Pada bagian ini menjelaskan masukan yang diperlukan pada algoritma nine men’s morris yaitu berupa strategi yang digunakan untuk memenangkan permainan Nine Men‟s Morris :

1. Lokasi yang strategis

   Pada fase ini sangat penting untuk meletakkan para bidak di posisi yang strategis (mudah untuk digerakkan, tidak terhalang), dan tidak berfokus untuk membuat mill. Contoh posisi yang strategis adalah titik tengah di kotak tengah, dimana bidak dapat bergerak pada empat arah. Namun meletakkan bidak condong ke salah satu kotak juga tidak direkomendasi, distribusi bidak harus merata. Dapat dilihat lingkaran hitam dari gambar:



Gambar 3.1 Lokasi Strategis

2. Menghalangi pemain lain

   Keunikan permainan ini adalah mill yang dibentuk dapat digunakan berkali-kali. Ketika seorang pemain sudah membentuk sebuah mill, dia bisa memindahkan bidaknya ke tempat lain pada giliran berikutnya, dan mengembalikannya ke tempat semula pada giliran selanjutnya, dan mill akan terbentuk kembali. Maka dia berhak mengambil dua bidak pada tiga giliran tersebut. Kondisi seperti ini harus dicegah oleh pemain lawannya, dengan cara meletakkan bidaknya sendiri di posisi yang ditinggalkan pada giliran kedua. Taktik ini disebut dengan blocking. Atau jika blocking tidak memungkinkan, pemain tersebut juga bisa mencoba membentuk mill sendiri, sehingga pemain lawan harus memilih antara melanjutkan membentuk mill atau melakukan blocking.

3. Double mill

   Double mill dibentuk dengan dua mill yang bersisian sedemikian rupa sehingga pemain dapat menggerakkan bidak dari sebuah mill yang sudah terbentuk ke posisi yang menyebabkan mill lain terbentuk. Double mill adalah kondisi yang sangat menguntungkan bagi seorang pemain, karena dia bisa mengambil bidak lawan pada setiap giliran. Double mill harus menjadi prioritas blocking pemain lawan

Algoritma yang nantinya dibuat akan memanfaatkan strategi yang sudah disebutkan di atas.

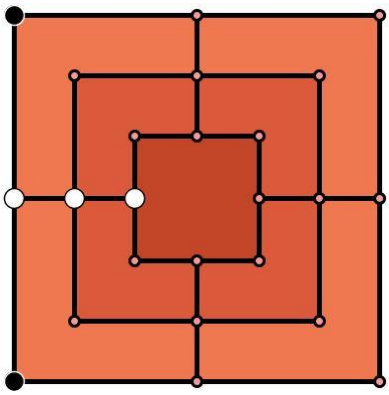
3.3.1. Fungsi Evaluasi pada Nine Men’s Morris

Dalam suatu pemainan yang berbasis giliran (*turn based game*), tugas dari fungsi evaluasi ialah untuk melihat kondisi permainan dan memberikan nilai dari sudut padang satu pemain[9]. Nilai dihitung pada kedua sisi pemain dan fungsi memberikan nilai positif untuk sisi pemain yang sedang melangkah dan nilai negatif untuk pemain yang tidak sedang melangkah. Nilai evaluasi ialah nilai balik yang diberikan oleh fungsi evaluasi.

Adapun fungsi evaluasi pada permainan nine men’s morris yang digunakan :

1. *Closed Morris*

   Merupakan jumlah mill yang baru terbentuk. Cohtoh dari *closed morris* dapat dilihat pada gambar 3.2, dimana bidak putik yang baru membentuk mill.



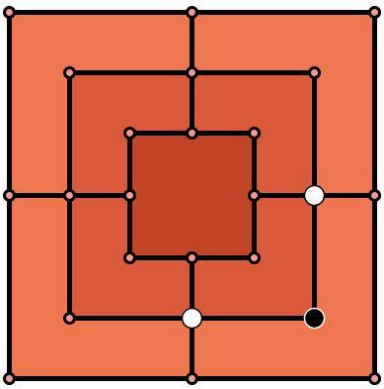
Gambar 3.2 *Closed Morris*

2. *Number of Morrises*

   Merupakan total mill yang ada di papan.

3. *Number of blocked opponent pieces*

   Merupakan jumlah bidak lawan yang terblokir. Dapat dilihat pada gambar 3.2, dimana bidak hitam terblokir ketika giliran putih main.



Gambar 3.3 Bidak Terblokir

4. *Number of pieces*

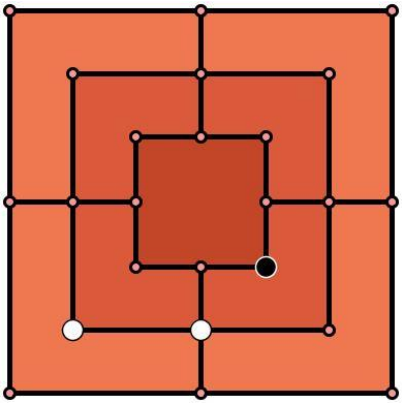
   Perbedaan jumlah bidak pemain dan lawan yang ada di papan.

5. *Opened Morris*

   Jumlah mill yang terbuka, yaitu pada mill yang sudah terbentuk bidaknya dipindahkan ke tempat lain.

6. *Number of 2 pieces configurations*

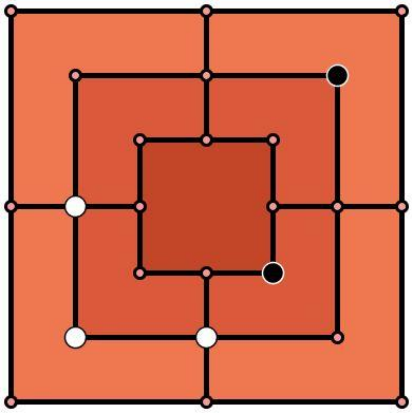
   Merupakan Jumlah 2-pieces configuration, yaitu konfigurasi dua bidak berurutan yang memungkinkan terbentuk mill pada giliran berikutnya. Dapat dilihat pada gambar 3.3, dimana bidak putih dapat terbentuk mill.



Gambar 3.4 Jumlah 2-pieces Configuration

7. *Number of 3 pieces configurations*

   Merupakan Jumlah 3-pieces configuration, yaitu konfigurasi dimana dapat terbentuk mill di dua tempat sehingga lawan tidak bisa melakukan blocking.



Gambar 3.5 Jumlah 3-pieces Configuration

8. *Double Morris*

   Merupakan Jumlah Double Morris.

9. *Winning configurations*

   Konfigurasi unggul, yaitu konfigurasi-konfigurasi tertentu yang berdasarkan pengalaman dapat membawa kemungkinan lebih besar untuk menang.

Karena permainan ini terdiri dari tiga fase yang berbeda dalam cara bermain, maka fungsi evaluasi yang digunakan haruslah terpisah. Berikut fungsi evaluasi setiap fase :

Berdasarkan percobaan yang dilakukan yang dilakukan oleh Petcu dan Holban dimana beberapa program AI nine men's morris dengan algoritma yang sama tapi dengan fungsi penghitung skor yang berbeda. Berikut hasil dari percobaan yang dilakukan oleh Petcu dan Holban dari setiap fase[11] :

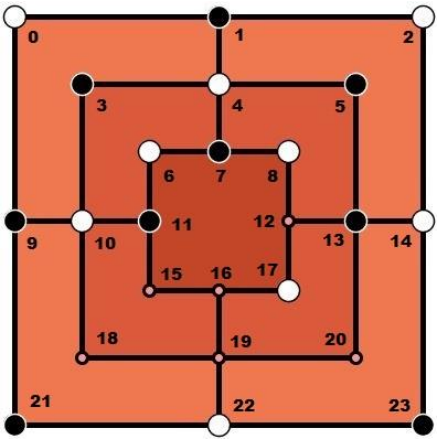
1. Relasi yang diperhitungkan pada fase pertama adalah Meletakkan bidak
   R1. *Clossed Morris*
   R2. *Number of Morrises*
   R3. *Number of blocked opponent pieces*
   R4. *Number of pieces*
   R5. *Number of 2 pieces configurations*
   R6. *Number of 3 pieces configurations*
   Fungsi penghitung skor/nilai evaluasi fase pertama =
   (18*R1)+(26*R2)+(R3)+(6*R4)+(12*R5)+(7*R6)
2. Relasi yang diperhitungkan pada fase kedua adalah Menggeserkan bidak
   R1. *Clossed Morris*
   R2. *Number of Morrises*
   R3. *Number of blocked opponent pieces*
   R4. *Number of pieces*
   R5. *Opened Morris*
   R6. *Double Morris*
   R7. *Winning configuration*
   Fungsi penghitung skor/nilai evaluasi fase kedua =
   (14*R1)+(43*R2)+(10*R3)+(8*R4)+7*R5)+(42*R6)+(1086*R7)
3. Relasi yang diperhitungkan pada fase ketiga adalah Memindahkan bidak
   R1. *Number of 2 pieces configurations*
   R2. *Number of 3 pieces configurations*
   R3. *Closed Morris*
   R4. *Winning configuration*

Fungsi penghitung skor/nilai evaluasi fase ketiga =
(10*R1)+(R2)+(16*R3)+(1190*R4)

3.4. Analisis Algoritma *Negascout*

Pada bagian ini akan dibahas tentang pemecahan permainan Nine Men's Morris menggunakan algoritma negascout. Seperti yang sudah dijelaskan di atas, negascout memerlukan sebuah fungsi penghitung skor untuk mengukur tingkat keuntungan suatu simpul.

Contoh kondisi permainan nine men's morris pada fase kedua yaitu menggeserkan bidak dan membatasi tingkat kedalaman hanya sampai tingkat 3. Berikut ialah gambar kondisi permainan pada nine men's morris dapat dilihat pada gambar 3.6.



Gambar 3.6 Kondisi Fase Kedua Permainan Nine Men's Morris

Dari kondisi gambar 3.6 giliran bermain pada bidak hitam. Berikut suatu pohon permainan yang sesuai dengan segala kondisi yang berlaku mulai dari langkah-langkah yang boleh dilakukan sampai aturan-aturan permainan dapat dilihat pada table 3.1.

Tabel 3.1 Tabel Pohon Permainan Pada Nine Men's Morris

Giliran Hitam	Giliran Putih	Giliran Hitam	Giliran Putih
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 13-20	Node p 8-12	5-13
			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	11-15

		14-13	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

Setelah membangun suatu pohon permainan, maka selanjutnya memberikan nilai evaluasi pada setiap langkah yang terjadi pada pohon permainan. Dapat dilihat table pohon dengan nilai evaluasi pada table 3.2.

Tabel 3.2 Tabel Pohon Dengan Nilai Evaluasi Pada Leaf Node

Giliran Hitam(MAX)	Giliran Putih(MIN)	Giliran Hitam(MAX)	Giliran Putih(MIN)
Simpul a	Simpul b	Simpul e	E1 = 40
			E2 = 40
			E3 = 30
			E4 = 40
		Simpul f	F1 = 40
			F2 = 30
			F3 = 50
			F4 = 50
		Simpul g	G1 = 50
			G2 = 50
			G3 = 50
			G4 = 40
			G5 =50
		Simpul h	H1= 30
			H2 = 30
			H3 = 40
			H4 = 30
			H5 = 50
			H6 = 40
		Simpul i	I1 = 40
			I2 = 60
			I3 = 50
		Simpul j	J1 = 40
			J2 = 30
			J3 = 50
		Simpul k	K1 = 40
			K2 = 30
			K3 = 50

			K4 = 40
			K5 = 40
			K6 = 30
	Simpul c	Simpul l	L1 = 98
			L2 = 50
			L3 = 40
			L4 = 40
			L5 = 40
			L6 = 40
		Simpul m	M1 = 30
			M2 = 50
		Simpul n	N1 = 40
			N2 = 40
			N3 = 50
			N4 = 40
		Simpul o	O1 = 50
			O2 = 50
			O3 = 50
			O4 = 50
			O5 = 50
	Simpul d	Simpul p	P1 = 30
			P2 = 20
			P3 = 30
			P4 = 50
			P5 = 50
		Simpul q	Q1 = 67
			Q2 = 30
			Q3 = 30
			Q4 = 30
			Q5 = 30
			Q6 = 50
			Q7 = 50
		Simpul r	R1 = 30
			R2 = 50
			R3 = 40
		Simpul s	S1 = 50
			S2 = 40
			S3 = 60
			S4 = 60
		Simpul t	T1 = 40
			T2 = 30
			T3 = 50
			T4 = 50
		Simpul u	U1 = 40
			U2 = 30
			U3 = 50
			U4 = 40
			U5 = 40

Berikut adalah proses menghitung nilai evaluasi pada table 3.2 dapat dilihat pada table 3.3.

Tabel 3.3 Tabel Proses Menghitung Nilai Evaluasi

Node	R1	R2	R3	R4	R5	R6	R7	Nilai Akhir
E1	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
E2	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
E3	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
E4	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
F1	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
F2	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
F3	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
F4	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
G1	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
G2	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
G3	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
G4	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
G5	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
H1	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
H2	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
H3	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
H4	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
H5	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
H6	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
I1	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
I2	14*0	43*0	10*6	8*0	7*0	42*0	1086*0	60
I3	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
J1	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
J2	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
J3	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
K1	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
K2	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
K3	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
K4	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
K5	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
K6	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
L1	14*1	43*1	10*4	8*0	7*0	42*0	1086*0	98
L2	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
L3	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
L4	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
L5	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
L6	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
M1	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
M2	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
N1	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
N2	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
N3	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
N4	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
O1	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
O2	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
O3	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
O4	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
O5	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
P1	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
P2	14*0	43*0	10*2	8*0	7*0	42*0	1086*0	20

P3	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
P4	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
P5	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
Q1	14*1	43*1	10*2	8*0	7*0	42*0	1086*0	67
Q2	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
Q3	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
Q4	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
Q5	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
Q6	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
Q7	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
R1	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
R2	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
R3	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
S1	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
S2	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
S3	14*0	43*0	10*6	8*0	7*0	42*0	1086*0	60
S4	14*0	43*0	10*6	8*0	7*0	42*0	1086*0	60
T1	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
T2	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
T3	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
T4	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
U1	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
U2	14*0	43*0	10*3	8*0	7*0	42*0	1086*0	30
U3	14*0	43*0	10*5	8*0	7*0	42*0	1086*0	50
U4	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40
U5	14*0	43*0	10*4	8*0	7*0	42*0	1086*0	40

Berikut adalah langkah – langkah algoritma negascout dengan pohon permainan yang ada di atas dapat dilihat pada table 3.4.

Tabel 3.4 Tabel Analisis Algoritma Negascout

Langkah	Simpul	Alpha	Beta	Catatan
1	a	-∞	∞	Nilai alpha dimulai dengan -∞ dan nilai beta dimulai dengan ∞
2	b	-∞	∞	-
3	e	-∞	∞	-
4	e	40	∞	Dalam tahap ini menelusuri cabang-cabang dari simpul ‘e’ dan mendapat nilai terbaik ialah 40 (karena 40 memiliki nilai lebih besar dari cabang lainnya)
5	b	-∞	40	Pada simpul min, nilai yang diharapkan adalah nilai yang lebih kecil dari 40. Sekarang harus menelusuri cabang-cabang dari simpul f,g,h,I,j dan k.
6	f	-∞	40	-
7	f	50	40	Setelah menelusuri cabang-cabang dari simpul f, nilai yang didapat ialah 50,

				setelah mendapat nilai 50 maka terjadi pemotongan beta karena 50>40
8	b	-∞	40	Nilai tidak diupdate karena 40<50
9	g	-∞	40	-
10	g	50	40	Setelah menelusuri cabang-cabang dari simpul g, nilai yang didapat ialah 50, setelah mendapat nilai 50 maka terjadi pemotongan beta karena 50>40
11	b	-∞	40	Nilai tidak diupdate karena 40<50
12	h	-∞	40	-
13	h	50	40	Setelah menelusuri cabang-cabang dari simpul h, nilai yang didapat ialah 50, setelah mendapat nilai 50 maka terjadi pemotongan beta karena 50>40
14	b	-∞	40	Nilai tidak diupdate karena 40<50
15	i	-∞	40	
16	i	60	40	Setelah menelusuri cabang-cabang dari simpul i, nilai yang didapat ialah 60, setelah mendapat nilai 60 maka terjadi pemotongan beta karena 60>40
17	b	-∞	40	Nilai tidak diupdate karena 40<60
18	j	-∞	40	-
19	j	50	40	Setelah menelusuri cabang-cabang dari simpul i, nilai yang didapat ialah 50
20	b	-∞	40	Nilai tidak diupdate karena 40<50
21	k	-∞	40	-
22	k	50	40	Setelah menelusuri cabang-cabang dari simpul k, nilai yang didapat ialah 50, setelah mendapat nilai 50 maka terjadi pemotongan beta karena 50>40
23	a	40	∞	Nilai 40 telah dinaikan ke akar a. sekarang max telah mengetahui bahwa sekarang nilai terbaiknya ialah 40.
24	c	-∞	∞	-
25	l	-∞	∞	-
26	l	98	∞	Setelah menelusuri cabang-cabang dari simpul l, nilai yang didapat ialah 98
27	c	-∞	98	Update nilai pada simpul c
28	m	-∞	98	-
29	m	50	98	Setelah menelusuri cabang-cabang dari simpul m, nilai yang didapat ialah 50
30	c	-∞	50	Update nilai pada simpul c karena 50<98
31	n	-∞	50	-
32	n	50	50	Setelah menelusuri cabang-cabang dari simpul n, nilai yang didapat ialah 50

33	c	-∞	50	Nilai tidak diupdate karena 50=50
34	o	-∞	50	-
35	o	50	50	Setelah menelusuri cabang-cabang dari simpul o, nilai yang didapat ialah 50
36	c	-∞	50	Nilai tidak diupdate karena 50=50
37	a	50	∞	Update nilai pada simpul a, karena 50>40 (dimana 40 adalah hasil dari simpul b)
38	d	-∞	∞	-
39	p	-∞	∞	-
40	p	50	∞	Setelah menelusuri cabang-cabang dari simpul p, nilai yang didapat ialah 50
41	d	-∞	50	Update nilai pada simpul d
42	q	-∞	50	-
43	q	67	50	Setelah menelusuri cabang-cabang dari simpul q, nilai yang didapat ialah 67, setelah mendapat nilai 67 maka terjadi pemotongan beta karena 67>50
44	d	-∞	50	Nilai tidak diupdate karena 50<67
45	r	-∞	50	-
46	r	50	50	Setelah menelusuri cabang-cabang dari simpul r, nilai yang didapat ialah 50
47	d	-∞	50	Nilai tidak diupdate karena 50=50
48	s	-∞	50	-
49	s	60	50	Setelah menelusuri cabang-cabang dari simpul s, nilai yang didapat ialah 60, setelah mendapat nilai 60 maka terjadi pemotongan beta karena 60>50
50	d	-∞	50	Nilai tidak diupdate karena 50<60
51	t	-∞	50	-
52	t	50	50	Setelah menelusuri cabang-cabang dari simpul t, nilai yang didapat ialah 50
53	d	-∞	50	Nilai tidak diupdate karena 50=50
54	u	-∞	50	-
55	u	50	50	Setelah menelusuri cabang-cabang dari simpul u, nilai yang didapat ialah 50
56	c	-∞	50	Setelah menelusuri cabang-cabang dari simpul c &d nilai yang diperoleh adalah 50. Karena c adalah nilai terbaik pertama. Sehingga pencarian selesai dengan nilai optimum yang diperoleh ialah 50.

3.5. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan sistem non fungsional meliputi analisis perangkat lunak (*software*), analisis perangkat keras (*hardware*), dan analisis pengguna (*user*) sebagai bahan analisis kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.5.1. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan kumpulan perintah-perintah yang diberikan kepada perangkat keras agar saling berinteraksi untuk melakukan suatu tugas. Perangkat lunak yang dibutuhkan untuk membangun aplikasi ini adalah sebagai berikut:

1. Sistem Operasi Windows 7 32 bit
2. NetBeans IDE 7.0
3. Java SE Development Kit 7 Update 45
4. Java SE Runtime Environment 7 Update 45

3.5.2. Analisis Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan untuk mendukung game halma ini adalah sebagai berikut:

1. Prosesor : Pentium 4 > 1GHz atau setara.
2. Memori : 512MB
3. *Hardisk* : 40GB
4. VGA : 128MB
5. *Keyboard* & *Mouse*

3.5.3. Analisis Pengguna

Analisis pengguna digunakan untuk mengetahui spesifikasi pengguna agar dapat memainkan permainan nine men’s morris. Spesifikasi minimal bagi pengguna agar dapat memainkan permainan nine men’s morris dapat dilihat pada table 3.5.

Tabel 3.5 Spesifikasi Pengguna

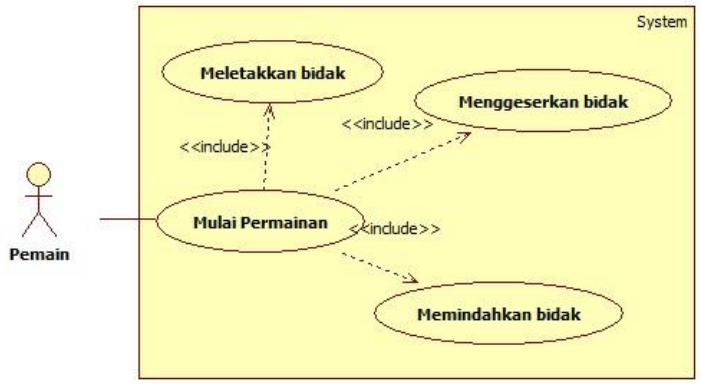
Karakter Fisik Pengguna	
Umur	> 6 Tahun
Gender	Laki-laki dan Perempuan
Disabilities	Tidak buta dan tidak mempunyai cacat fisik seperti cacat tangan
Pengetahuan dan Pengalaman Pengguna	
Computer literacy	Dapat menggunakan dan mengoperasikan Komputer

3.6. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan analisis untuk menggambarkan sistem yang akan dibangun. Analisis kebutuhan fungsional dimodelkan dengan menggunakan UML, tahap-tahap pemodelannya antara lain adalah *use case diagram*, *use case scenario*, *activity diagram*, *class diagram* dan *sequence diagram*.

3.6.1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Berikut adalah *use case diagram* permainan nine men's morris :



Gambar 3.7 *Use Case Diagram* Permainan Nine Men'n Morris

3.6.1.1. Definisi Aktor

Definisi aktor berfungsi untuk menjelaskan aktor yang terdapat pada *use case diagram*. Deskripsi pendefinisian aktor pada permainan nine men's morris dapat dilihat pada tabel 3.6.

Tabel 3.6 Tabel Definisi Aktor Pada Permainan Nine Men's Morris

No	Aktor	Deskripsi
1	Pemain	Orang yang memainkan permainan

3.6.1.2. Definisi *Use Case*

Definisi *use case* berfungsi untuk menjelaskan proses yang terdapat pada *use case diagram*. Deskripsi pendefinisian *use case* pada permainan nine men's morris dapat dilihat pada tabel 3.7.

Tabel 3.7 Tabel Definisi *Use Case* Pada Permainan Nine Men's Morris

No	*Use case*	Deskripsi
1	Mulai permainan	Merupakan proses untuk memulai permainan baru
2	Meletakkan bidak	Merupakan proses untuk meletakkan bidak
3	Menggeserkan bidak	Merupakan proses untuk menggeserkan bidak dimana setelah semua bidak masing-masing pemain telah diletakkan
4	Memindahkan bidak	Merupakan proses untuk memindahkan bidak dimana pemain yang bidaknya hanya tersisa 3 bidak

3.6.1.3. *Use Case Scenario*

Use case scenario merupakan deskripsi alur kejadian yang menjelaskan urutan interaksi antara aktor dengan *use case* tersebut. Berdasarkan *use case* diagram, maka *use case scenario* untuk permainan nine men's morris terdiri dari mulai permainan, meletakkan bidak menggeserkan bidak dan memindahkan bidak. Berikut adalah *use case scenario* pada permainan nine men's morris.

1. *Use Case Scenario* Mulai Permainan

Use case scenario ini menjelaskan interaksi antara aktor pemain dengan *use case* mulai permainan yang dijelaskan pada tabel 3.8.

Tabel 3.8 Tabel *Use Case Scenario* Mulai Permainan

Identifikasi	
Nama	Mulai permainan
Tujuan	Memulai permainan nine men's morris
Deskripsi	Proses untuk memulai permainan baru
Aktor	Pemain
Skenario Utama	

Kondisi awal	Tampil papan permainan
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
a. Memilih mulai permainan	
	b. Menampilkan papan permainan baru
Kondisi Akhir	Menampilkan papan permainan permainan baru

2. *Scenario Use Case* Meletakkan Bidak

Use case scenario ini menjelaskan interaksi antara aktor pemain dengan *use case* meletakkan bidak yang dijelaskan pada tabel 3.9.

Tabel 3.9 Tabel *Use Case Scenario* Meletakkan Bidak

Identifikasi	
Nama	Meletakkan Bidak
Tujuan	Meletakkan bidak pada papan permainan
Deskripsi	Proses untuk memulai permainan nine men's morris
Aktor	Pemain
Skenario Utama	
Kondisi awal	Tampil arena permainan dengan posisi bidak belum diletakan pada papan permainan
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
a. Memilih posisi dimana bidak akan diletakkan	
	b. Uptade papan permainan sesuai bidak yang diletakkan
Kondisi Akhir	Seluruh bidak telah diletakkan pada papan permainan

3. *Scenario Use Case* Menggeserkan Bidak

Use case scenario ini menjelaskan interaksi antara aktor pemain dengan *use case* menggeserkan bidak yang dijelaskan pada tabel 3.10.

Tabel 3.10 Tabel *Use Case Scenario* Menggeserkan bidak

Identifikasi	
Nama	Menggeserkan Bidak
Tujuan	Menggeserkan bidak dari satu posisi ke posisi sebelahnya yang kosong
Deskripsi	Proses untuk menggeserkan bidak

Aktor	Pemain
Skenario Utama	
Kondisi awal	Arena permainan dengan posisi seluruh bidak pada papan permainan
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
a. Pilih bidak yang akan digeserkan	
	b. Tampil kemungkinan langkah bidak
c. Pilih tujuan langkah yang akan dilakukan bidak	
	d. Update papan permainan sesuai dengan pergeseran bidak
Kondisi Akhir	Bidak telah bergeser dan permainan dilanjutkan

4. *Scenario Use Case* Memindahkan bidak

Skenario *use case* ini menjelaskan interaksi antara aktor pemain dengan *use case* memindahkan bidak yang dijelaskan pada tabel 3.11.

Tabel 3.11 Tabel *Use Case Scenario* Memindahkan Bidak

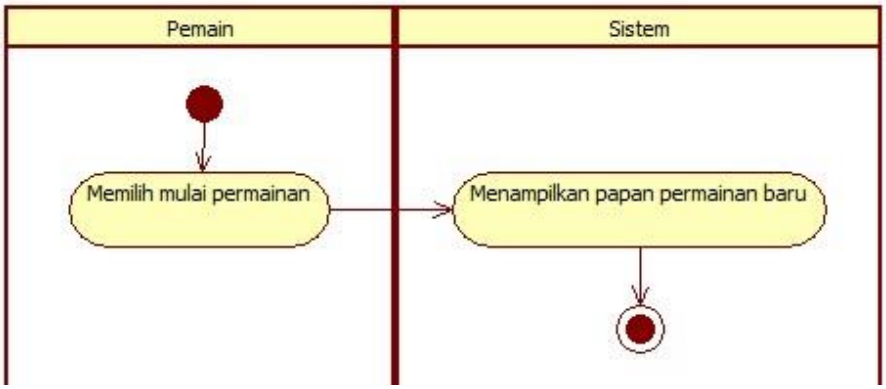
Identifikasi	
Nama	Memindahkan bidak
Tujuan	Memindahkan bidak dari satu posisi ke posisi lain yang kosong
Deskripsi	Proses untuk memindahkan bidak
Aktor	Pemain
Skenario Utama	
Kondisi awal	Bidak pemain hanya tersisa 3 bidak
Aksi Aktor	**Reaksi Sistem**
Skenario Normal	
a. Pilih bidak yang akan dipindahkan	
	b. Tampil kemungkinan langkah bidak
c. Pilih tujuan langkah yang akan dilakukan bidak	
	d. Update papan permainan sesuai dengan perpindahan bidak
Kondisi Akhir	Bidak telah dipindahkan dan permainan dilanjutkan

3.6.2. *Activity Diagram*

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah *activity diagram* dari permainan nine men's morris :

1. *Activity Diagram* Mulai Permainan

Pada proses ini sistem akan menampilkan arena permainan dimana pemain akan memulai permainan baru. *Activity diagram* mulai permainan dapat dilihat pada gambar 3.8.



Gambar 3.8 Activity Diagram Mulai Permainan

2. Activity Diagram Meletakkan Bidak

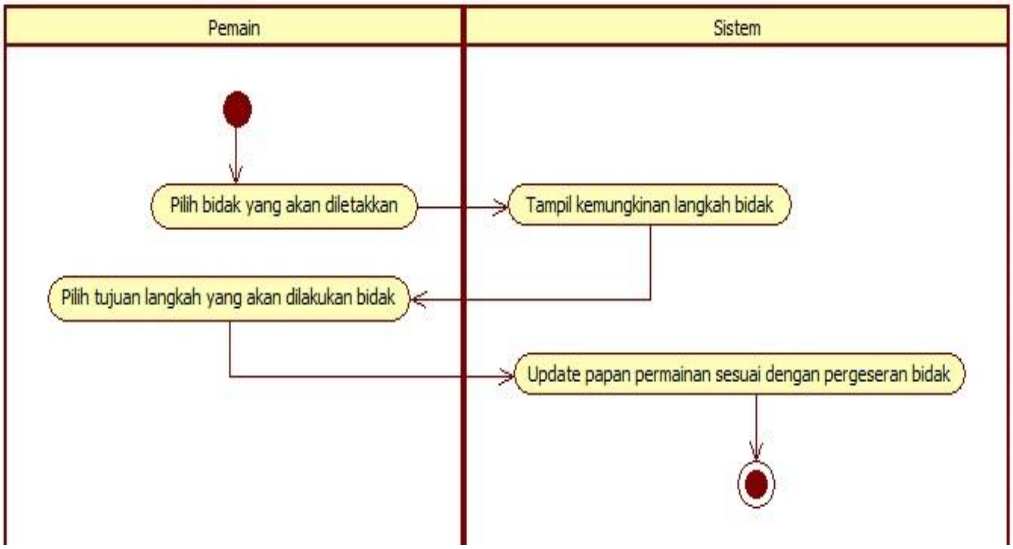
Pada proses ini sistem akan menampilkan arena permainan dimana pemain akan mulai meletakan bidak pada papan. *Activity diagram* meletakkan bidak dapat dilihat pada gambar 3.9.



Gambar 3.9 Activity Diagram Meletakkan Bidak

3. Activity Diagram Menggeserkan Bidak

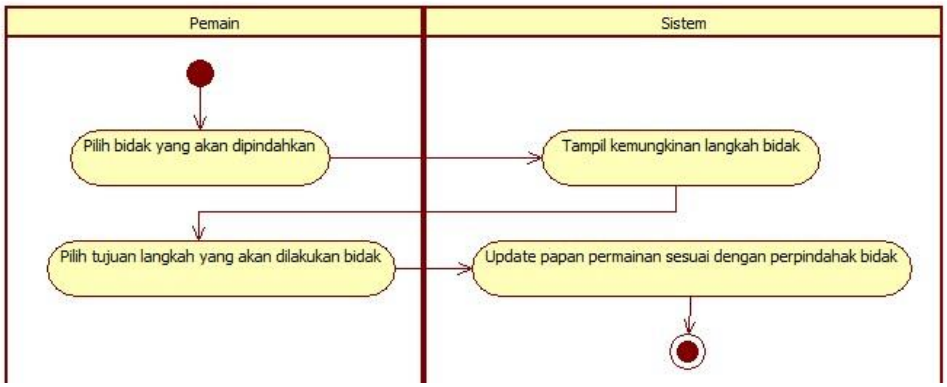
Pada proses ini sistem akan melanjutkan permainan dimana pemain akan mulai menggeserkan bidak dari satu posisi ke posisi sebelahnya. *Activity diagram* menggeserkan bidak dapat dilihat pada gambar 3.10.



Gambar 3.10 Activity Diagram Menggeserkan Bidak

4. Activity Diagram Memindahkan Bidak

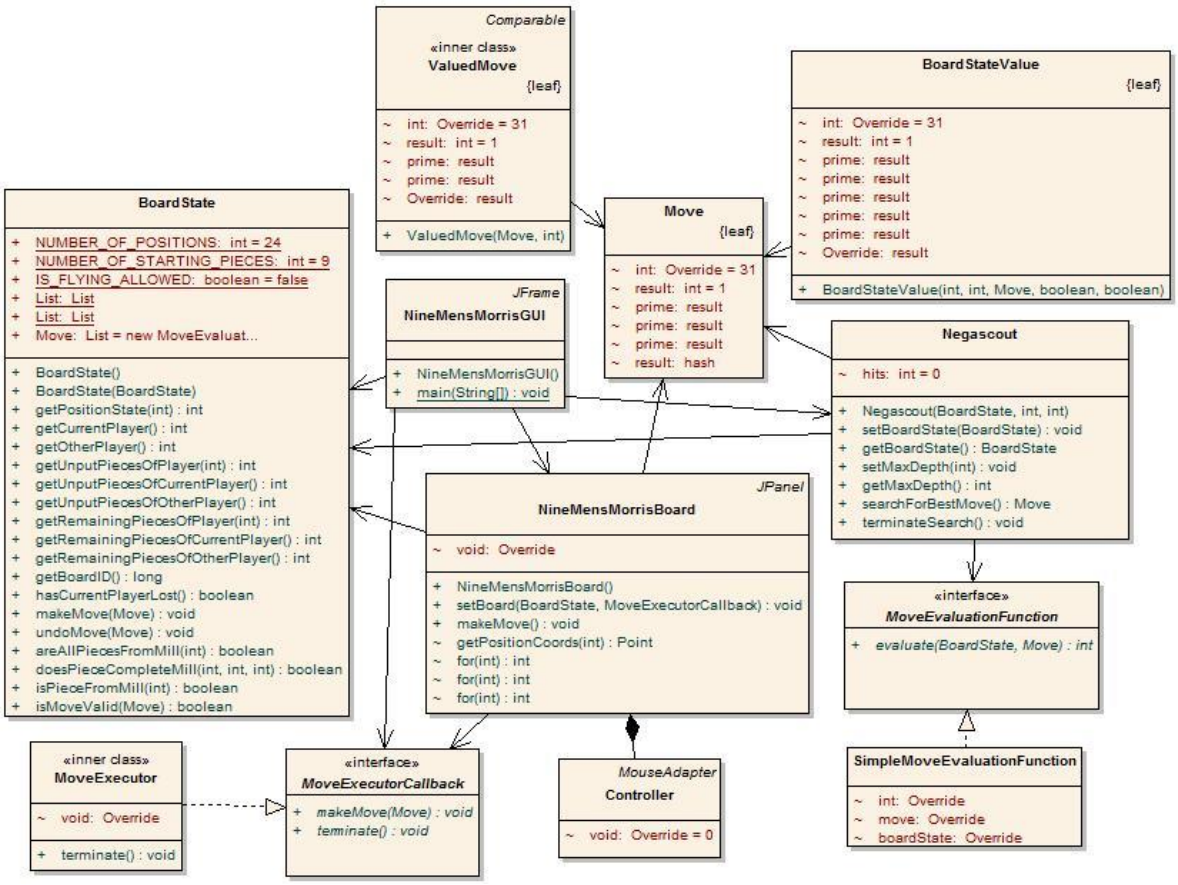
Pada proses ini sistem akan menampilkan permainan dimana pemain akan mulai memindahkan bidak dari satu posisi ke posisi lain. *Activity diagram* memindahkan bidak dapat dilihat pada gambar 3.11.



Gambar 3.11 Activity Diagram Memindahkan bidak

3.6.3. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem [12]. Berikut adalah *class diagram* dari permainan nine men's morris :

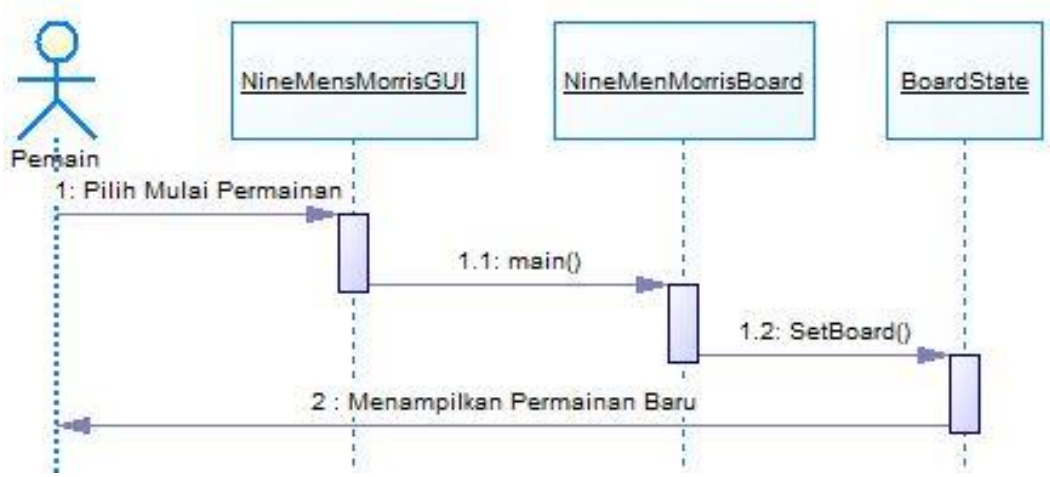


Gambar 3.12 Class Diagram Permainan Nine Men's Morris

3.6.4. *Sequence Diagram*

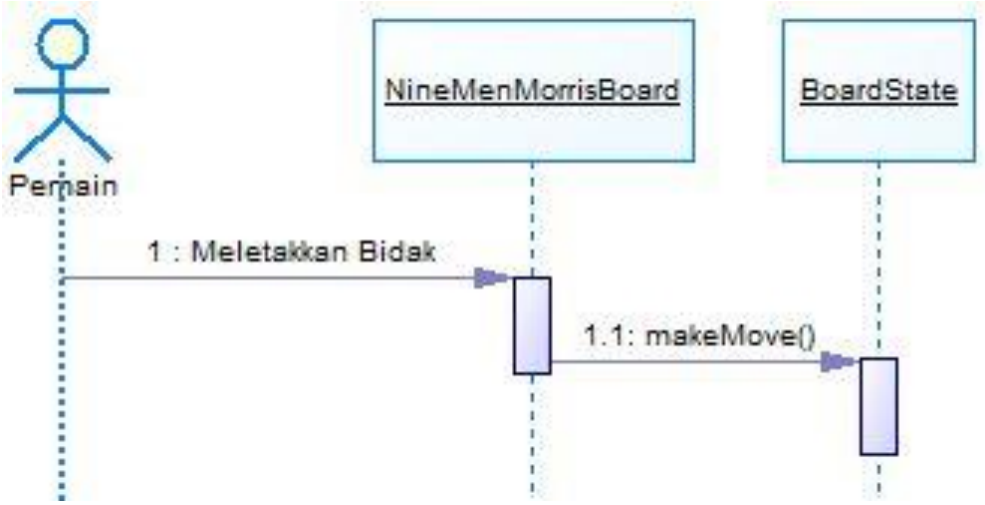
Diagram sequence menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut adalah *sequence diagram* dari permainan nine men's morris :

1. *Sequence Diagram* Mulai Permainan



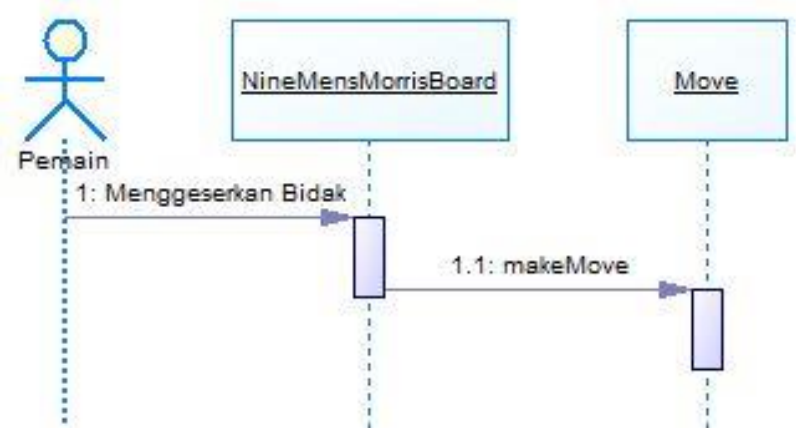
Gambar 3.13 *Sequence Diagram* Mulai Permainan

2. *Sequence Diagram* Meletakkan bidak



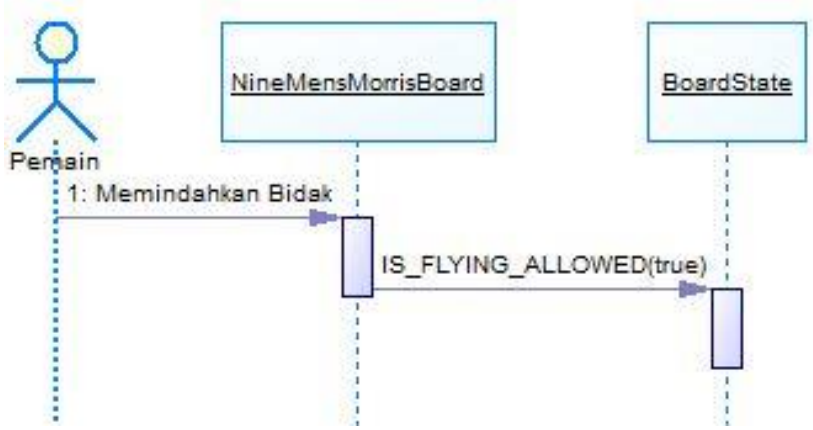
Gambar 3.14 *Sequence Diagram* Meletakkan Bidak

3. *Sequence Diagram* Menggeserkan bidak



Gambar 3.15 *Sequence Diagram* Menggeserkan Bidak

4. *Sequence Diagram* Memindahkan Bidak



Gambar 3.16 *Sequence Diagram* Memindahkan Bidak

3.7. Perancangan Sistem

Perancangan sistem merupakan metodologi pengembangan suatu perangkat lunak yang dilakukan setelah melalui tahapan analisis.

3.7.1. Perancangan Komponen Permainan Nine Men's Morris

Perancangan komponen pada permainan nine men's morris terdiri dari pemain, titik awal dan titik tujuan yang akan dijelaskan sebagai berikut :

1. Pemain

   Permainan ini akan dimainkan oleh dua orang, satu pemain dan satu AI.

2. Papan Nine Men's Morris

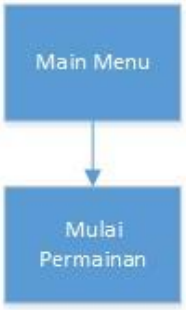
Papan yang digunakan berupa 3 buah persegi berbeda ukuran dengan yang lebih kecil berada di dalam persegi yang lebih besar. Titik tengah sisi sebuah persegi terhubung dengan titik tengah sisi dua persegi lain dengan sebuah garis.

3. Bidak

   Bidak yang berjumlah 9 dengan warna biasanya masing-masing hitam dan putih, diletakkan pada perpotongan antara dua garis.

3.7.2. Perancangan Struktur Menu

Perancangan struktur menu dirancang untuk menyediakan fungsi-fungsi yang akan digunakan dalam *game* Nine Men's Morris yang dapat dilihat pada gambar 3.17.



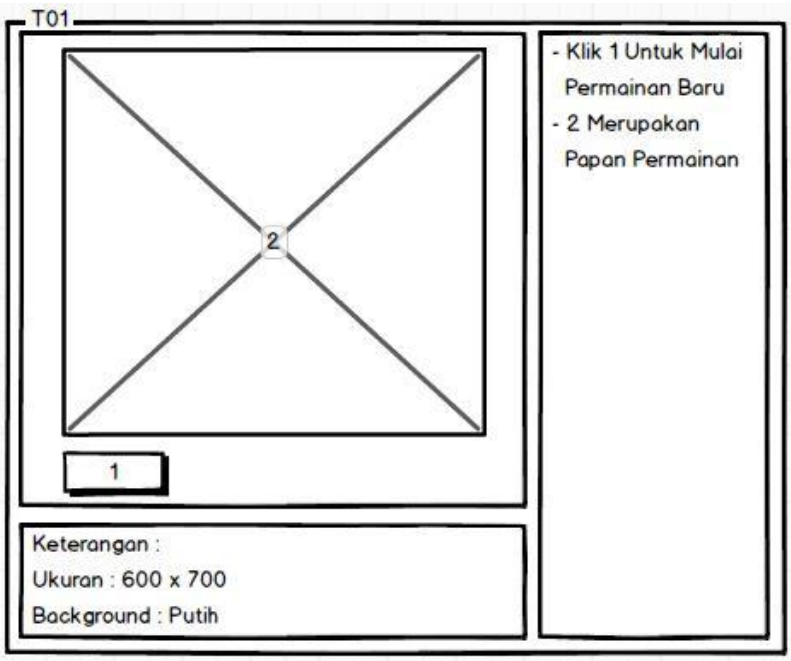
Gambar 3.17 Struktur Menu

3.7.3. Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimple-mentasikan aplikasi serta akan memudahkan pembuatan aplikasi.

1. Tampilan Utama

   Tampilan ini merupakan tampilan pertama saat aplikasi Nine Men's Morris dijalankan. Gambar tampilan utama dapat dilihat pada gambar 3.18.



Gambar 3.18 Tampilan Utama Permainan Nine Men's Morris

3.7.3.1. Perancangan Pesan

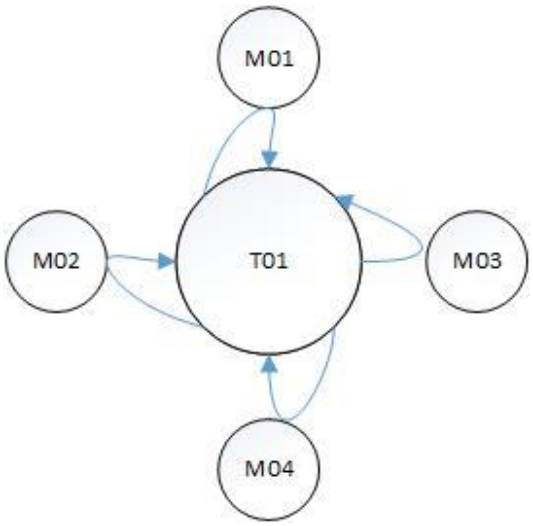
Perancangan antarmuka ini adalah tampilan pesan yang muncul pada aplikasi ini seperti ditunjukan pada Gambar 3.19.



Gambar 3.19 Tampilan Pesan

3.7.3.2. Jaringan Semantik

Perancangan jaringan semantik menggambarkan alur informasi dari setiap antarmuka aplikasi yang dijelaskan pada gambar 3.20.



Gambar 3.20 Jaringan Semantik