BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Masalah

Analisis masalah menjelaskan proses identifikasi masalah serta evaluasi mengenai *game* sejenis dan *game* yang akan dibangun.

Berdasarkan pada bab I yang telah diuraikan, yang menjadi titik permasalahan dalam penelitian ini adalah sebagai berikut :

1. Membangun media alternatif pembelajaran yang menyampaikan materi sejarah walisongo yang dapat membantu guru. Dimana dalam peraturan Menteri Pendidikan Nasional No. 24 tahun 2006, mulai tahun pelajaran 2006/2007 diberlakukan Kurikulum Tingkat Satuan Pendidikan (KTSP). Pada perubahan kurikulum tersebut guru diberikan kebebasan dalam menyampaikan materi pembelajaran dengan lebih kreatif. Salah satu cara penyampaian materi pembelajaran dengan lebih kreatif adalah dengan menggunakan media pembelajaran yang lebih bervariasi.
2. Membangun media pembelajaran yang memberikan nuansa menyenangkan pada siswa-siswi dengan menyajikan animasi karena menurut hasil survey lembaga riset dan penerbitan komputer yaitu *Computer Technology Research* (CRT) menyatakan bahwa orang hanya mampu mengingat 20% dari yang dilihat dan 30% dari yang didengar. Tetapi orang mengingat 50% dari yang dilihat dan didengar dan 80% dari yang dilihat, didengar dan dilakukan sekaligus.[17]

3.1.1 Pengenalan *Game* Sejenis

Game dengan jenis *gameplay* Petualangan sudah tidak asing lagi dalam dunia *game,* perkembangannya tidak terlalu drastis namun jenis ini tetap saja menarik untuk dimainkan, banyak *game* yang memakai sistem *gameplay* Petualangan ini, beberapa diantaranya yaitu *Metal Slug, Zombies* dan *Zombie Horde.*

Game petualangan yang memadukan dengan pembelajaran juga tidak terlalu drastis. Namun ada beberapa *game* edukasi berjenis petualangan yang sudah disebarluaskan, seperti Petualangan Berburu Bintang Hijaiyah (edukasi islam), Merapi Boy Perjalanan Bocah Merapi Menghindari Wedhus Gembel (edukasi kepedulian sosial), Perjalanan Kartini (edukasi sejarah), dan Sejarah Kemerdekaan Indonesia (edukasi sejarah)

Dalam pembahasan, penulis akan membahas *game* dengan jenis *gameplay* petualangan. Dimana *Game* yang menjadi acuan disini ialah *game Zombie Horde* yang serupa dengan *gameplay* yang akan di bangun oleh penulis.

3.1.1.1 Pengenalan *Game Zombie Horde*

Seperti yang sudah dijelaskan bahwa game sejenis yang menjadi acuan adalah game zombie horde. Berikut ini penjelasan mengenai *game Zombie Horde.*

1. Storyline *Game Zombie Horde*

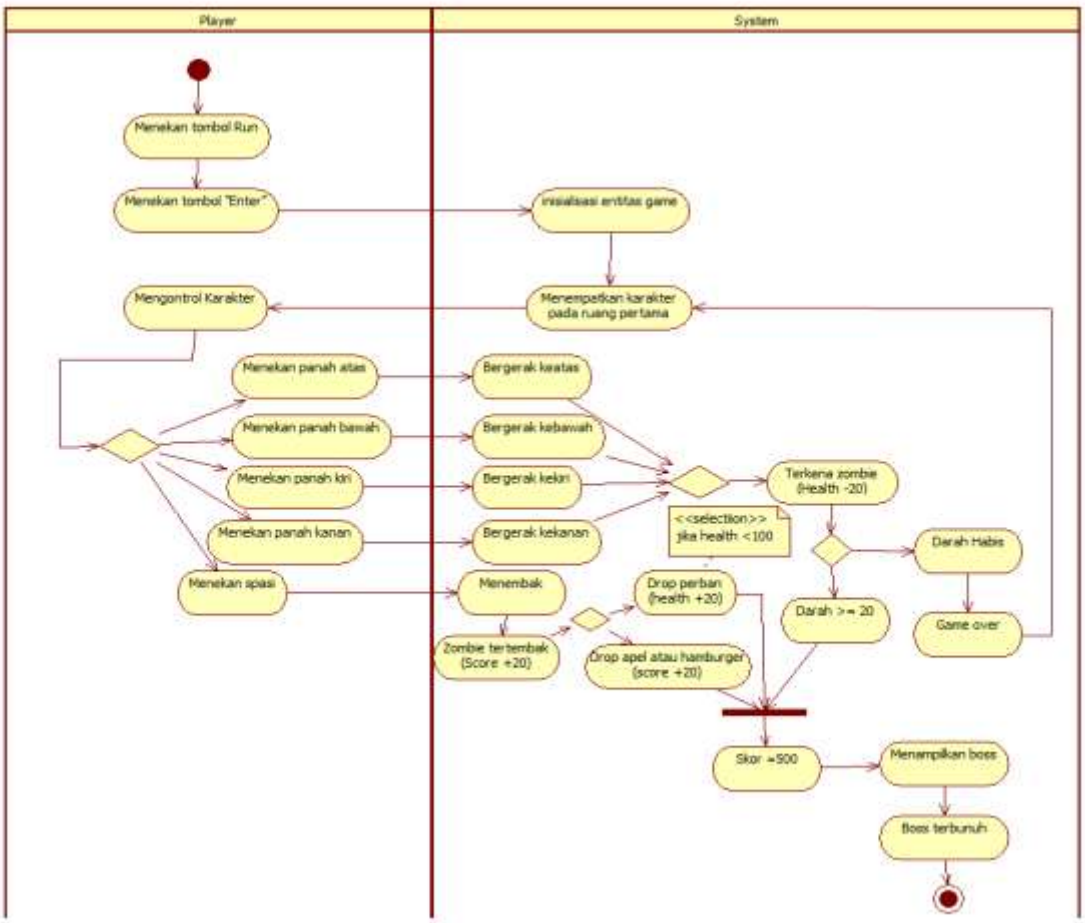
Pada tahun 2020, dunia telah diambil alih oleh penyakit jahat yang mengubah orang menjadi zombie dan membuat manusia terlihat sama. Karakter pemain adalah salah satu korban selamat yang tersisa dan mencoba untuk

melindungi keluarganya. Tugas karakter adalah untuk mengalahkan bos zombie, namun harus melewati anak buah atau pengikutnya terlebih dahulu.

2. Gameplay *Zombie Horde*

Karakter memulai permainan dengan Healtbar 100, Jika setiap karakter terkena zombie akan kehilangan 20 poin health. Karakter harus membunuh zombie hingga mencapai skor 500. Pada setiap zombie yang terbunuh maka akan drop perban, apel, atau hamburger. Disetiap karakter mengambil perban maka point health bertambah 20 hingga mencapai 100%. Sedangakan untuk apel akan dan hamburger akan menambah score tambahan 20. Kemudian karakter akan menghadapi bos. Bos pada *game* zombie horde ini jauh lebih sulit untuk dikalahkan yang berbeda dengan zombie pengikutnya karena memiliki darah 4 kali lipat dari zombie pengikutnya.

Untuk melihat lebih jelas *gameplay* Zombie Horde, di bawah ini merupakan *activity diagram* permainan pada Zombie Horde.



Gambar 3.1 Activity Diagram Permainan pada Zombie Horde

3. Kelemahan *Zombie Horde*

Pada *Game Zombie Horde* terdapat beberapa kelemahan diantaranya sebagai berikut :

a. Hanya menonjolkan sisi permainan saja tanpa melibatkan sisi edukasi.
b. Pemain tidak merasakan perubahan pada ruang 1 dengan yang lainnya karena background sama.
c. Musuh hanya bergerak statis secara horizontal dari arah kiri ke kanan. Begitu pula dari bos musuh.

4. Konten-Konten *Game* Pada *Game Zombie Horde*

Konten dalam *game* merupakan faktor penting dalam menjaga pemain agar tidak cepat merasa bosan dan tetap terlibat dalam dunia *game* [Takatsuki, Y. 2007]. Berikut ini adalah konten-konten yang terdapat pada *game zombie horde* dapat dilihat dari tabel berikut.

Tabel 3.1 Konten-konten pada *game Zombie Horde*[20].

No	Nama	Gambar	Penjelasan
1.	Contra		Contra merupakan nama dari karakter utama pada *game* Zombie Horde yang sekaligus karakter yang dikendalikan oleh pemain.
2.	Zombie		Zombie merupakan karakter *enemy* (musuh) dari karakter utama yang bergerak horizontal dengan kecepatan konstan dan berjalan statis dari arah kiri ke kanan.
3.	Bos pertama		Bos Zombie ini merupakan karakter *enemy* (musuh) yang harus dikalahkan oleh Contra yang memiliki darah lebih dari Zombie pengikutnya..
4.	*Health*		Health merupakan item yang dapat memulihkan health karakter utama. Setiap mendapatkan health. Health

			Karakter utama bertambah 20, hingga seterusnya hingga mencapai health maximal 100.
5.	*Bullet*		Peluru merupakan item yang keluar dari senjata Contra untuk membunuh zombie.
6.	Apel		Apel merupakan i*tem* dari *drop* zombie yang dapat menambah score pemain.
7.	Hamburger		Apel merupakan i*tem* dari *drop* zombie yang dapat menambah score pemain.
8.	Manusia		Manusia karakter yang akan diselamatkan oleh Contra.

5. Tampilan Utama *Game* Zombie Horde



Gambar 3.2 Tampilan *Game Zombie Horde*

3.1.2 *Game* Edukasi Sejarah Walisongo

Game yang akan dibangun berjudul Sejarah Walisongo yang bertipe *Adventure* (Petualangan). *Game* ini dibangun dengan mengaplikasikan teknologi sebagai media alternatif pembelajaran untuk menyajikan materi sejarah walisongo secara *interaktif* dan *edukatif* yang ditujukan untuk siswa-siswi kelas V dan VI Sekolah Dasar maupun Madrasah Ibtidaiyah dengan tampilan grafis 2D (2 Dimensi). Berikut ini fitur-fitur dalam *game* sejarah walisongo ini :

1. Sistem *single player* dengan
2. Tampilan grafis 2D dan berbasis desktop
3. Jenis *game* bertipe petualangan yang disajikan secara *interaktif* dan *edukatif*.
4. Target pengguna adalah kelas V dan VI sekolah dasar.
5. mengangkat tema sejarah walisongo, dimana materi yang disajikan diambil dari materi pembelajaran Ilmu Pengetahuan Sosial (IPS) kelas V Sekolah Dasar dan Sejarah Kebudayaan Islam (SKI) kelas VI Madrasah Ibtidaiyah.
6. *Game* Sejarah Walisongo ini mengambil metode bermain sambil belajar karena dalam *game* disajikan materi-materi mengenai sejarah walisongo.

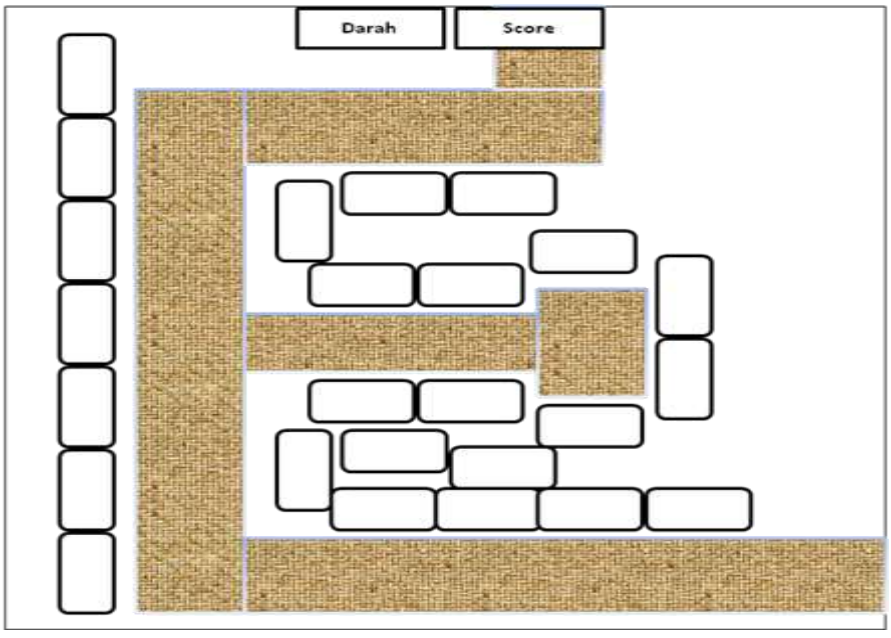
3.1.2.1 Storyline

Game ini berlatar di sebuah pulau, yang dikisahkan bahwa akhlak umat manusia di masa depan hancur karena kurangnya contoh-contoh yang memberikan nilai dan norma agama islam. Untuk itu karakter *player* kembali ke masa lalu demi mengetahui sejarah walisongo dan menyelamatkan foto para tokoh walisongo yang dijadikan bukti bahwa terdapat contoh yang baik untuk umat

manusia di masa depan yang berasal dari tanah jawa. Namun perjalanan *player* untuk mengetahui sejarah walisongo dan mendapatkan foto tokoh walisongo tidak mudah karena dihalangi oleh Zombie yang menyesatkan umat manusia di masa depan. Maka tugas pemain dalam *game* ini adalah mengumpulkan semua foto walisongo, dan mengalahkan 3 (tiga) bos zombie.

3.1.2.2 Storyboard

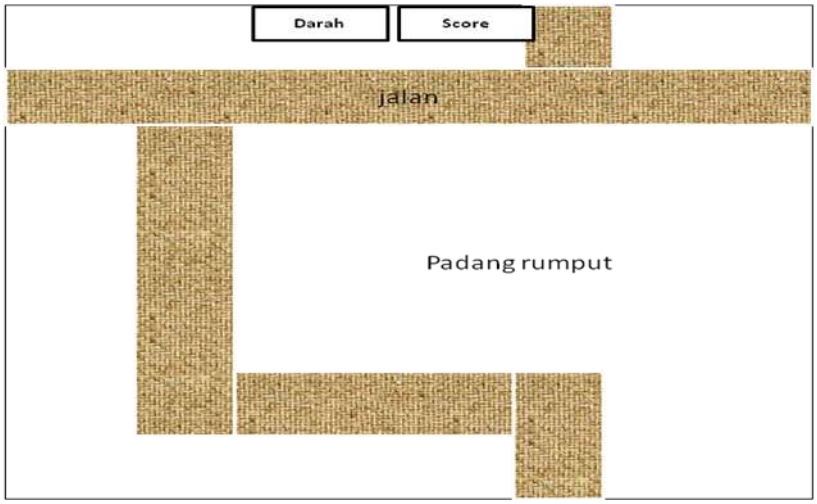
Storyboard adalah sketsa gambar yang disusun berurutan sesuai naskah, dengan storyboard kita dapat menyampaikan ide cerita kepada orang lain dengan lebih mudah, karena dapat menggiring khayalan seseorang mengikuti gambar-gambar yang tersaji, sehingga menghasilkan persepsi yang sama pada ide cerita. Berikut merupakan storyboard dari *game* edukasi sejarah walisongo :



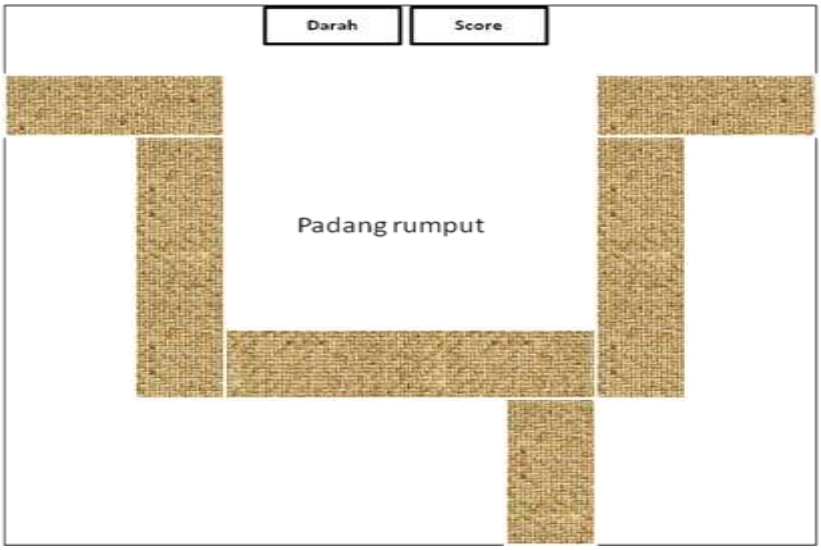
Gambar 3.3 *Storyboard* Tempat 1



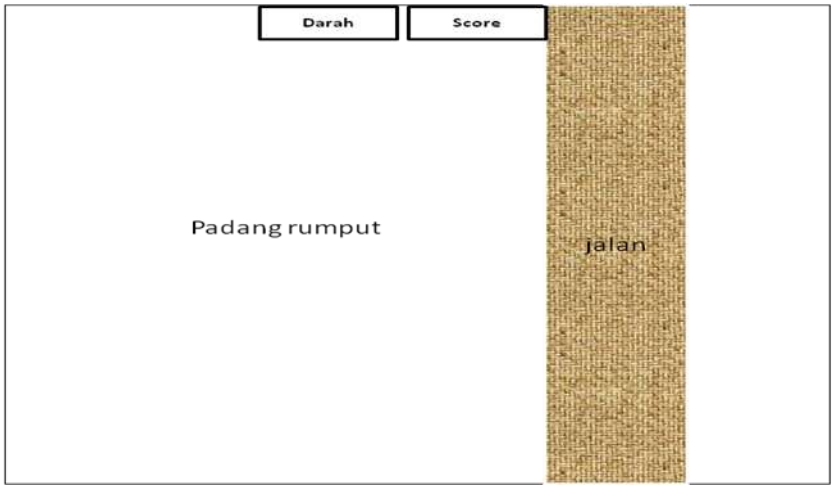
Gambar 3.4 *Storyboard* Tempat 2



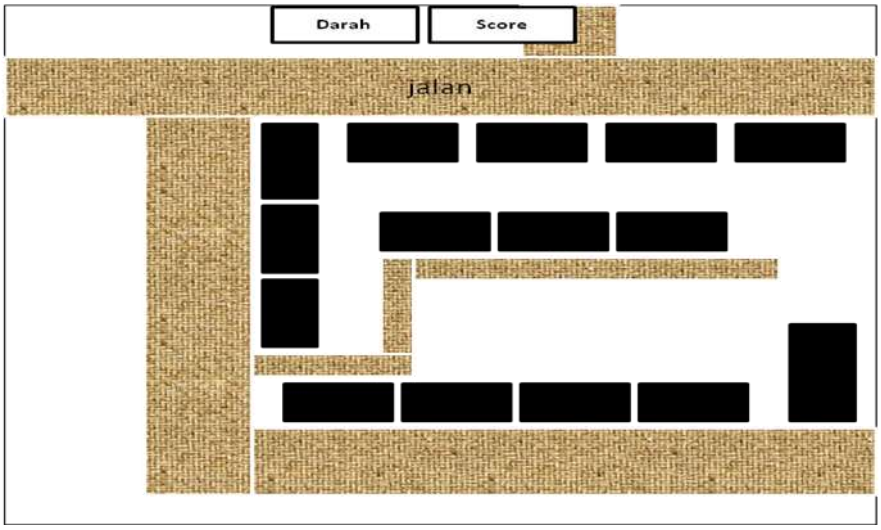
Gambar 3.5 *Storyboard* Tempat 3



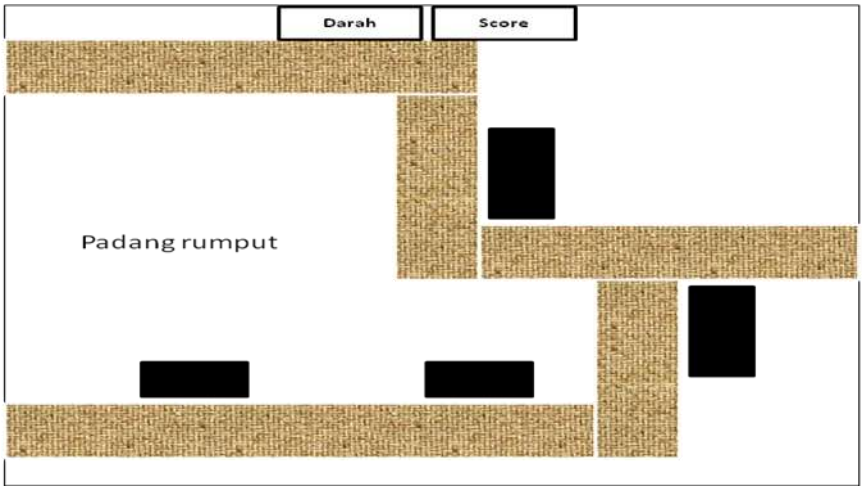
Gambar 3.6 *Storyboard* Tempat 4



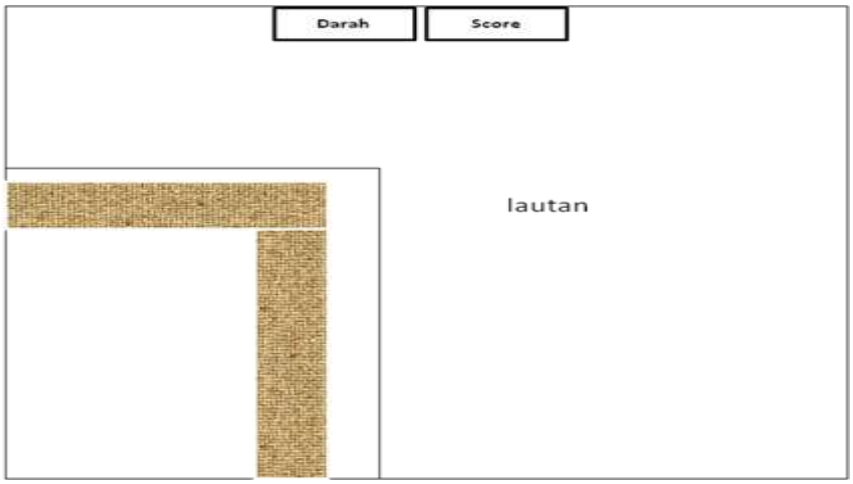
Gambar 3.7 *Storyboard* Tempat 5



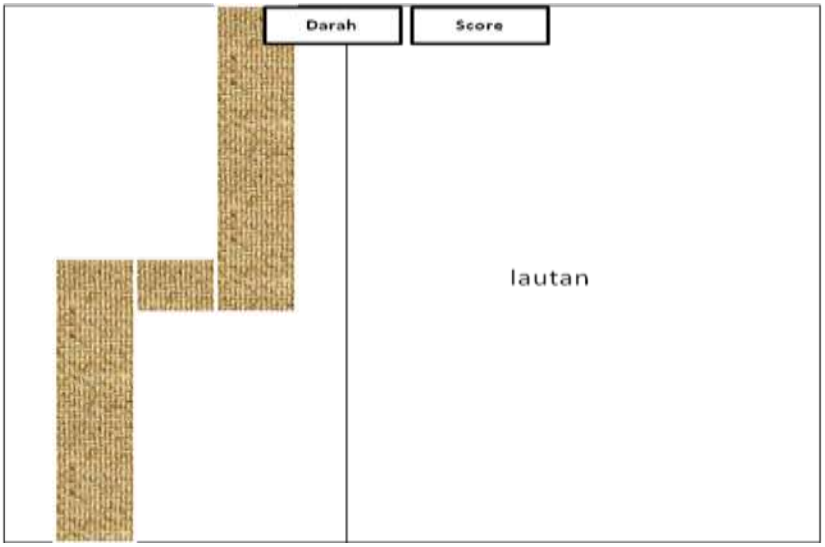
Gambar 3.8 *Storyboard* Tempat 6



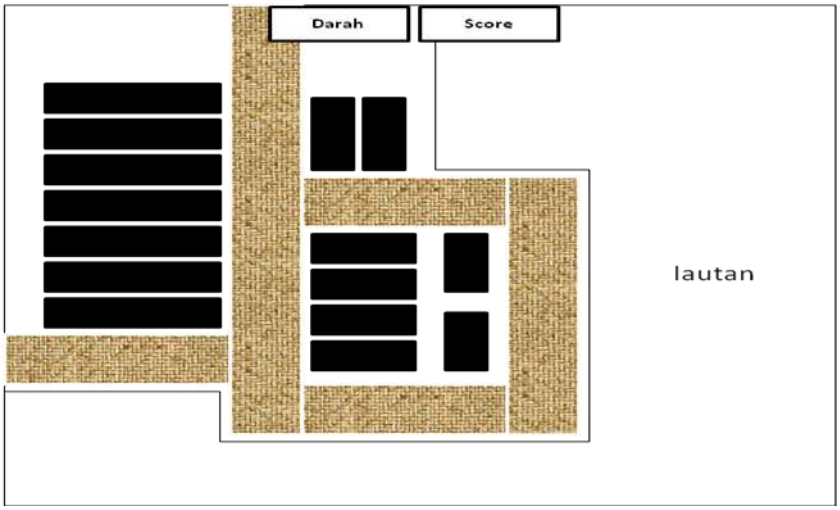
Gambar 3.9 *Storyboard* Tempat 7



Gambar 3.10 *Storyboard* Tempat 8



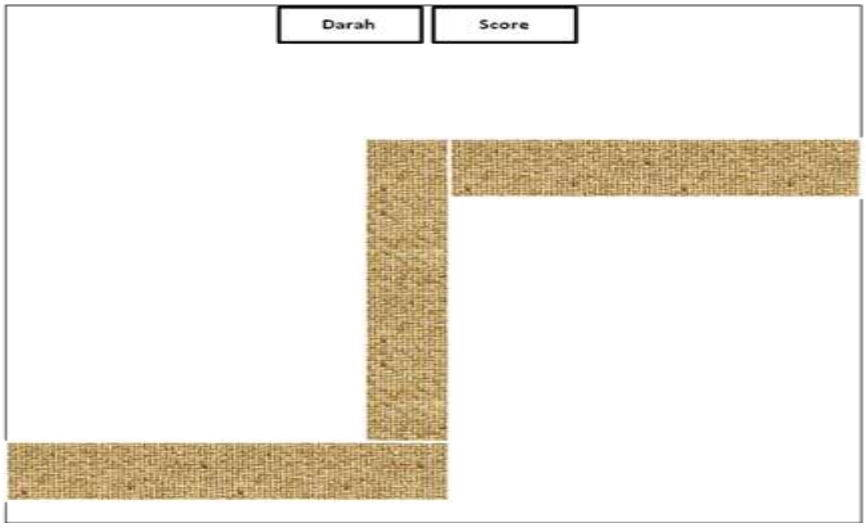
Gambar 3.11 *Storyboard* Tempat 9



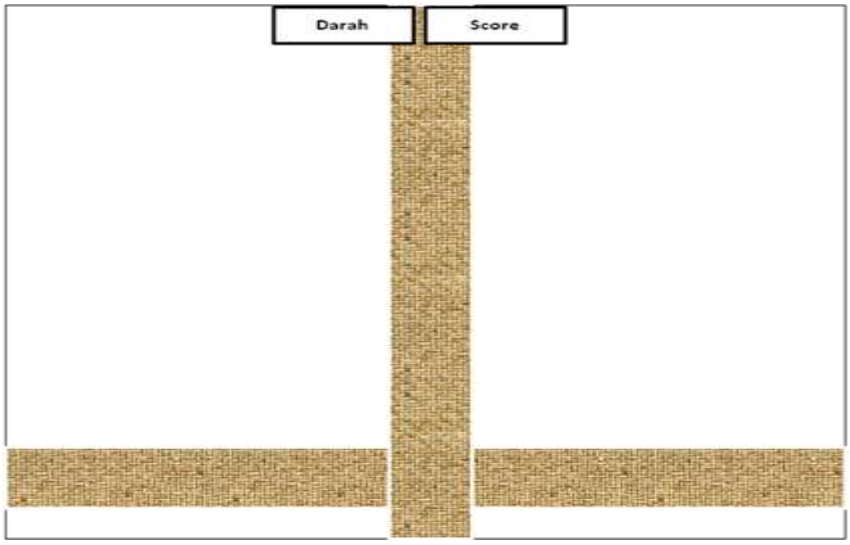
Gambar 3.12 *Storyboard* Tempat 10



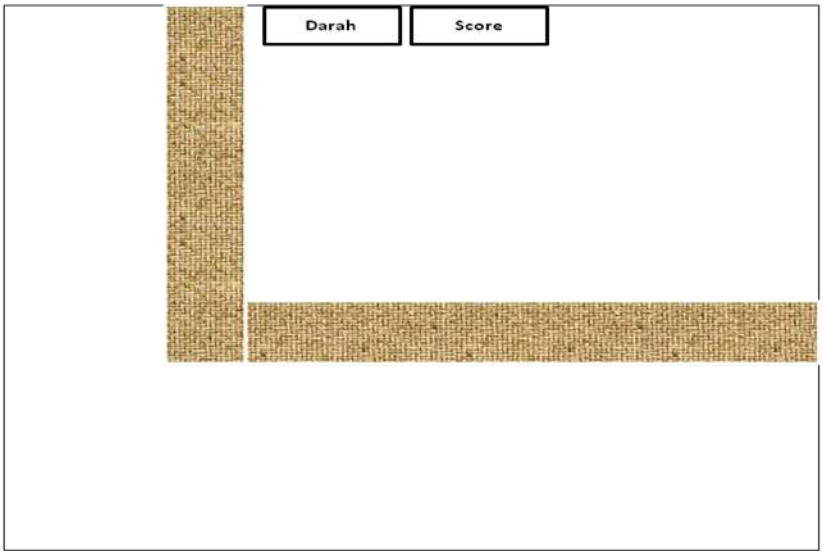
Gambar 3.13 *Storyboard* Tempat 11



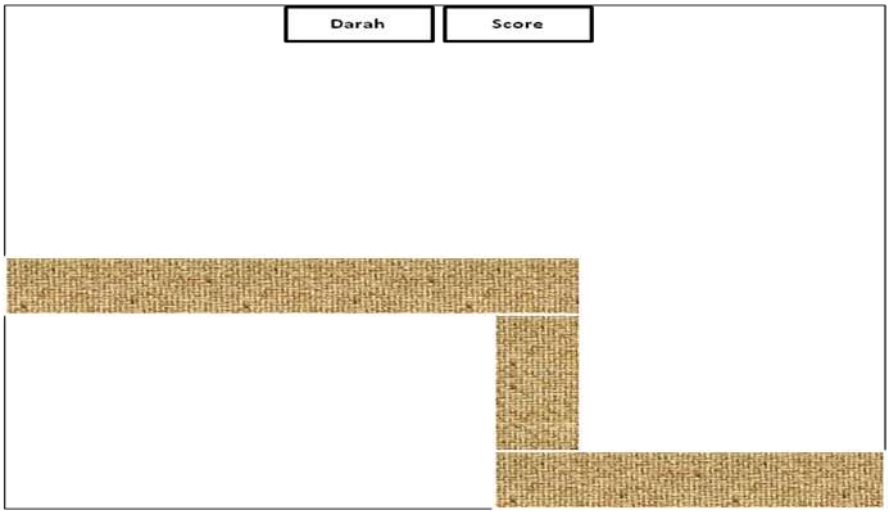
Gambar 3.14 *Storyboard* Tempat 12



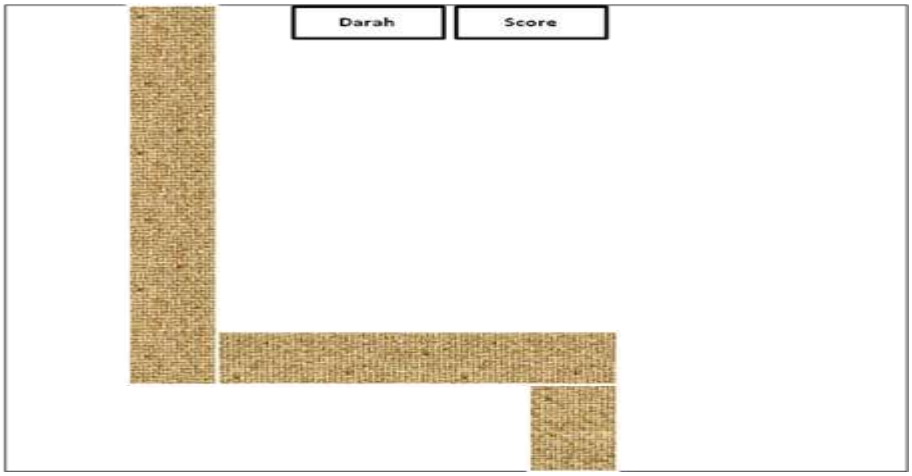
Gambar 3.15 *Storyboard* Tempat 13



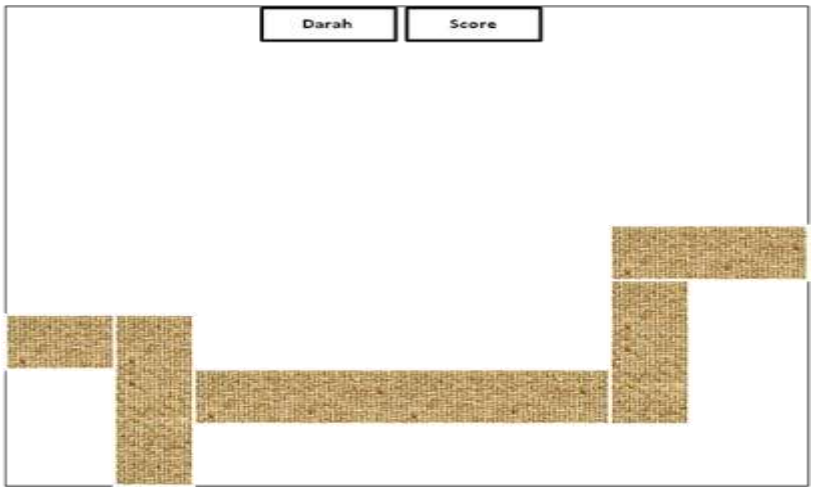
Gambar 3.16 *Storyboard* Tempat 14



Gambar 3.17 *Storyboard* Tempat 15



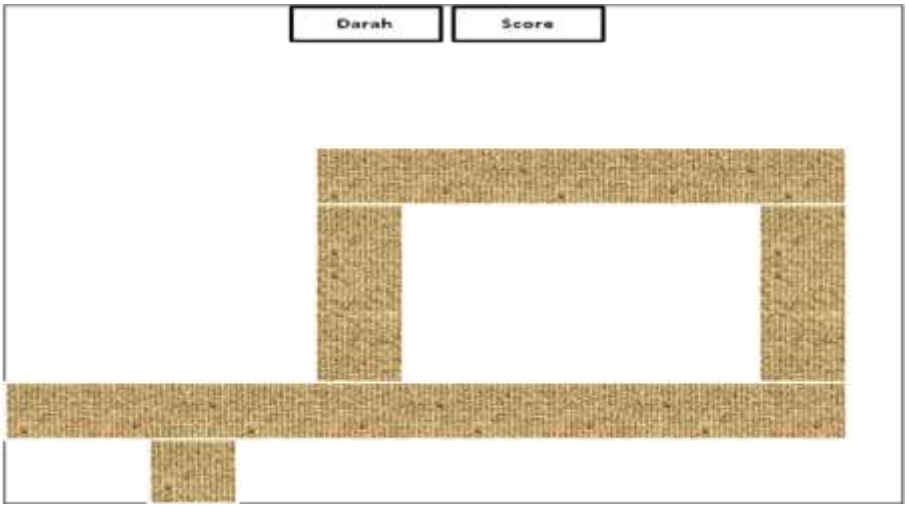
Gambar 3.18 *Storyboard* Tempat 16



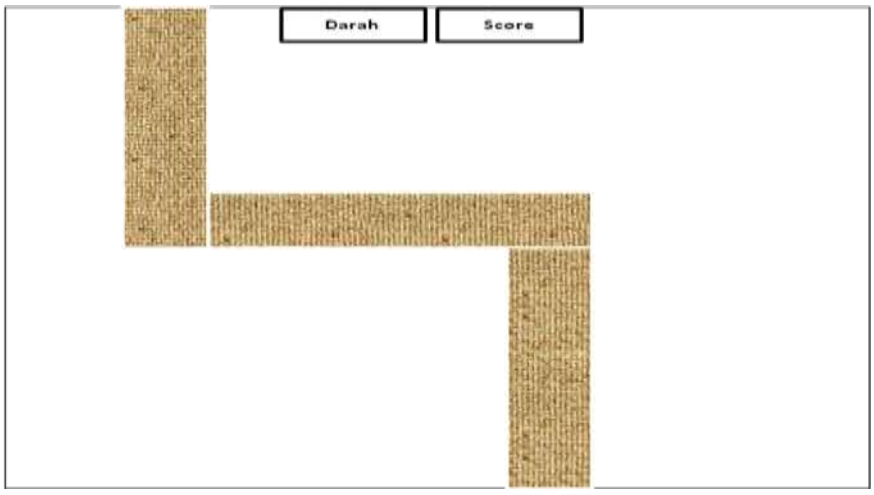
Gambar 3.19 *Storyboard* Tempat 17



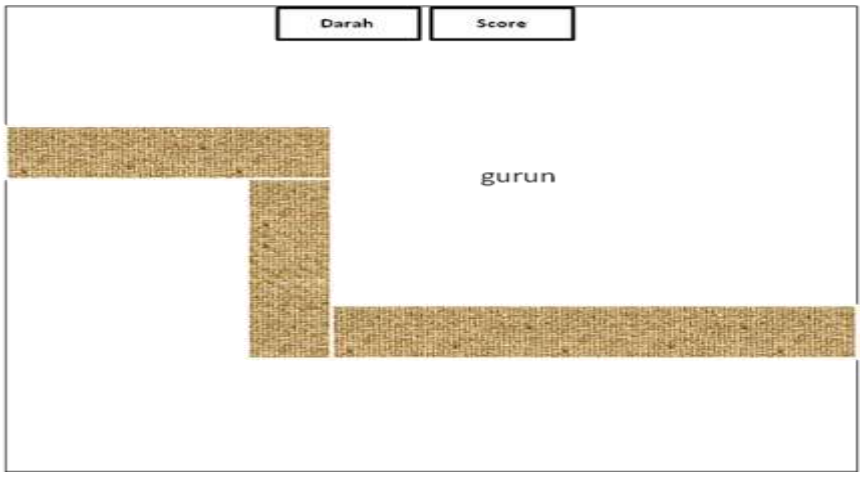
Gambar 3.20 *Storyboard* Tempat 18



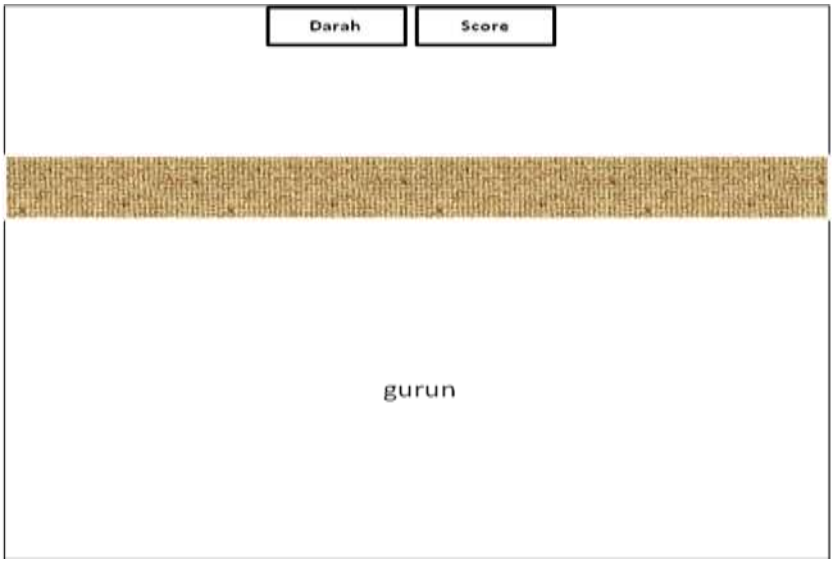
Gambar 3.21 *Storyboard* Tempat 19



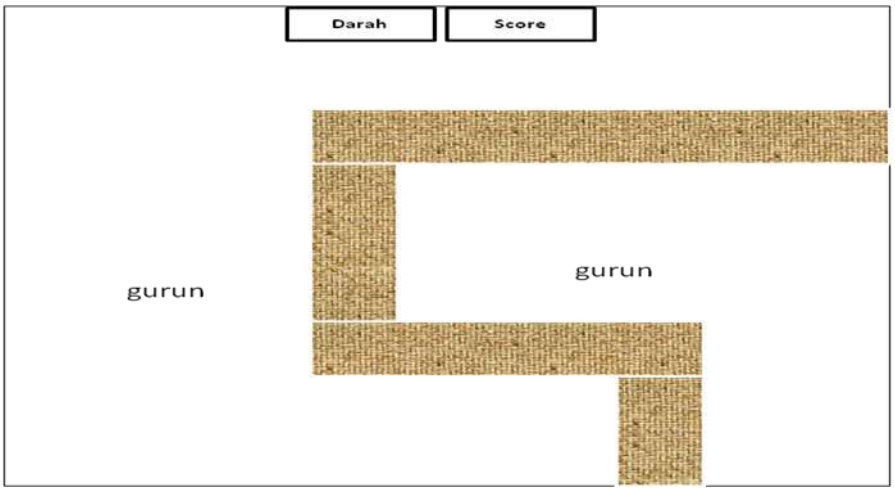
Gambar 3.22 *Storyboard* Tempat 20



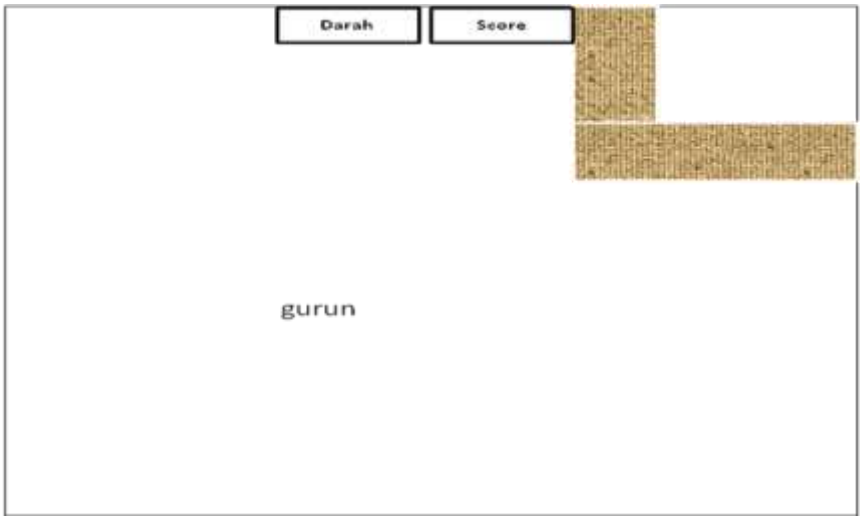
Gambar 3.23 *Storyboard* Tempat 21



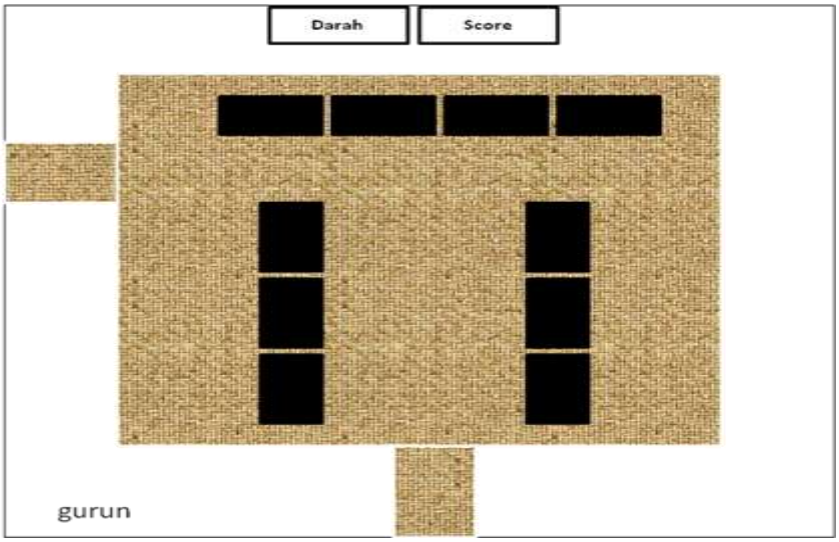
Gambar 3.24 *Storyboard* Tempat 22



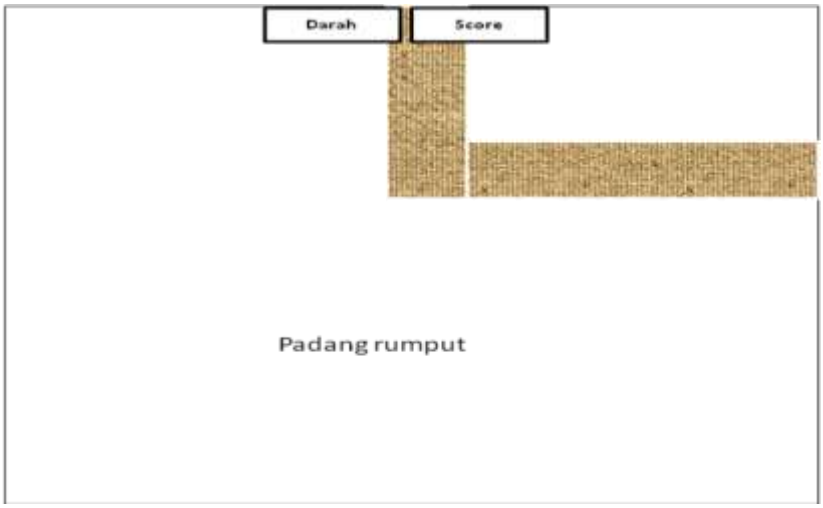
Gambar 3.25 *Storyboard* Tempat 23



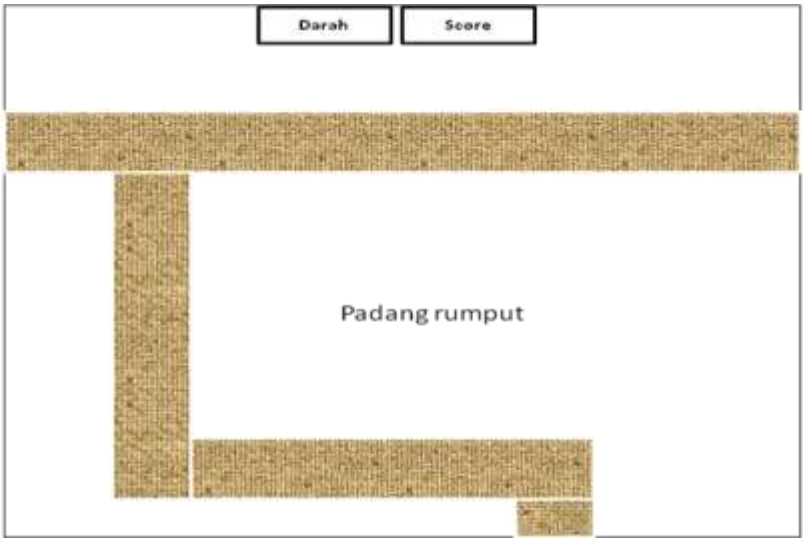
Gambar 3.26 *Storyboard* Tempat 24



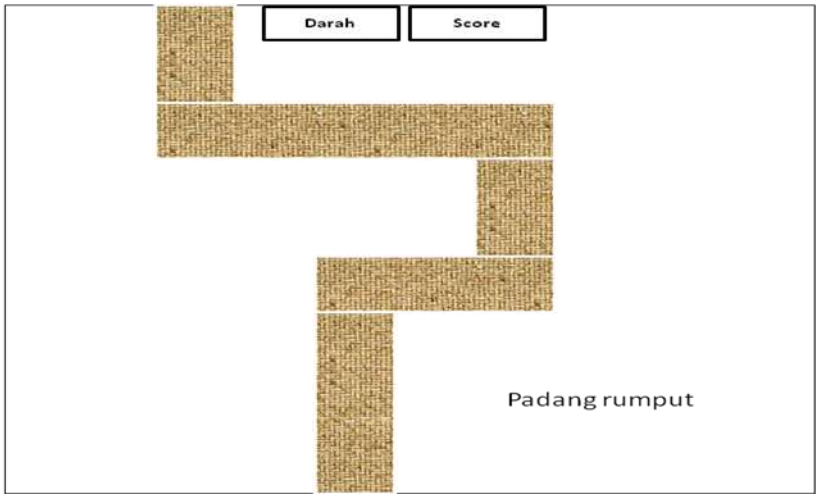
Gambar 3.27 *Storyboard* Tempat 25



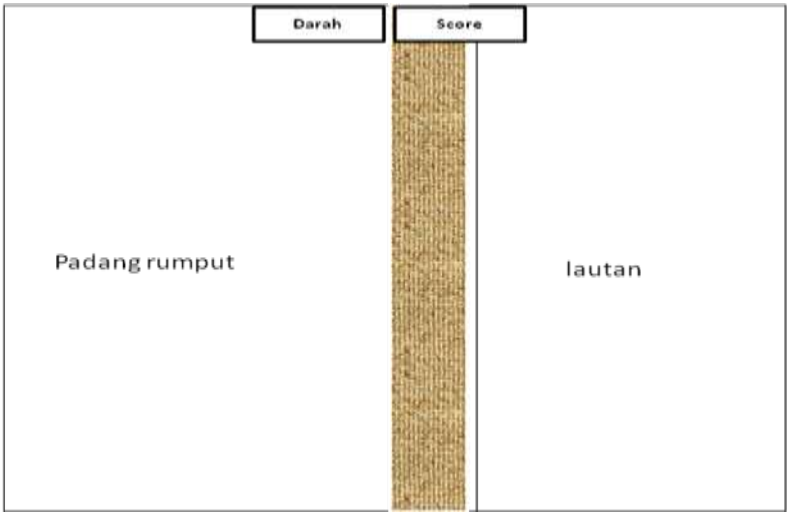
Gambar 3.28 *Storyboard* Tempat 26



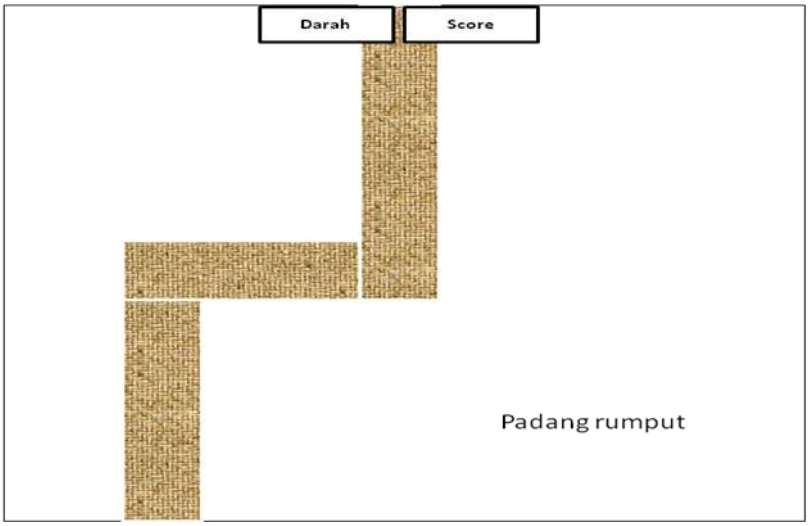
Gambar 3.29 *Storyboard* Tempat 27



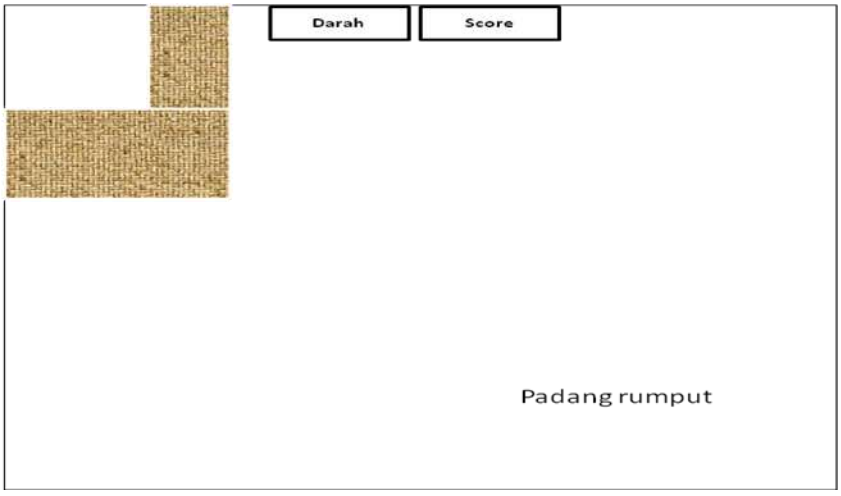
Gambar 3.30 *Storyboard* Tempat 28



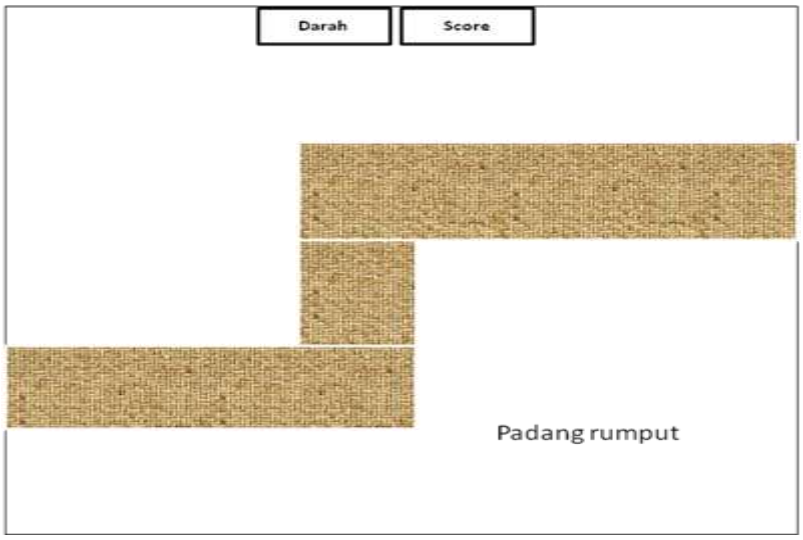
Gambar 3.31 *Storyboard* Tempat 29



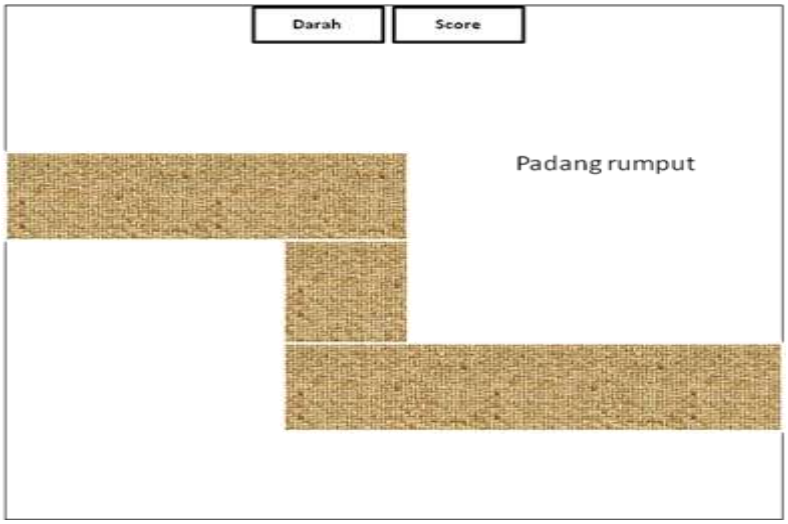
Gambar 3.32 *Storyboard* Tempat 30



Gambar 3.33 *Storyboard* Tempat 31



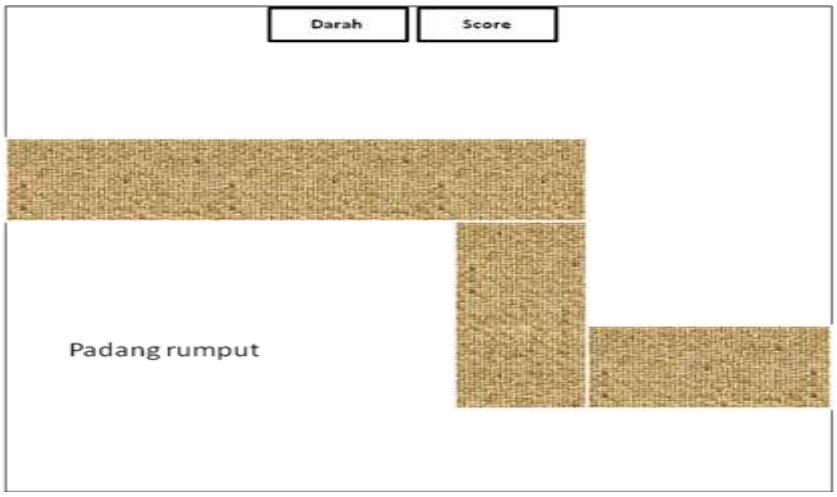
Gambar 3.34 *Storyboard* Tempat 32



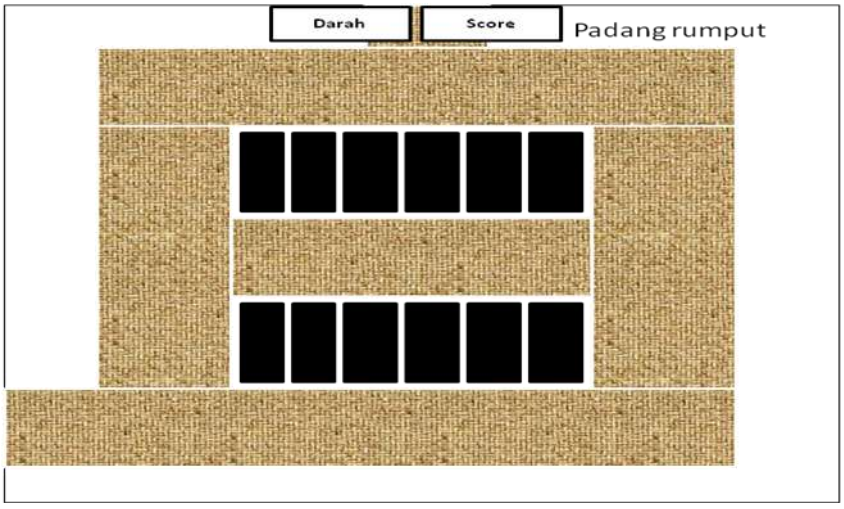
Gambar 3.35 *Storyboard* Tempat 33



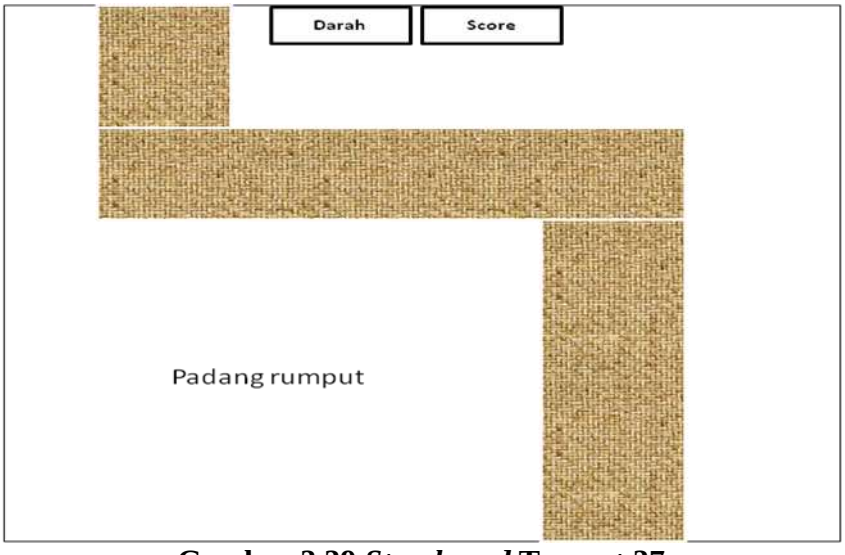
Gambar 3.36 *Storyboard* Tempat 34



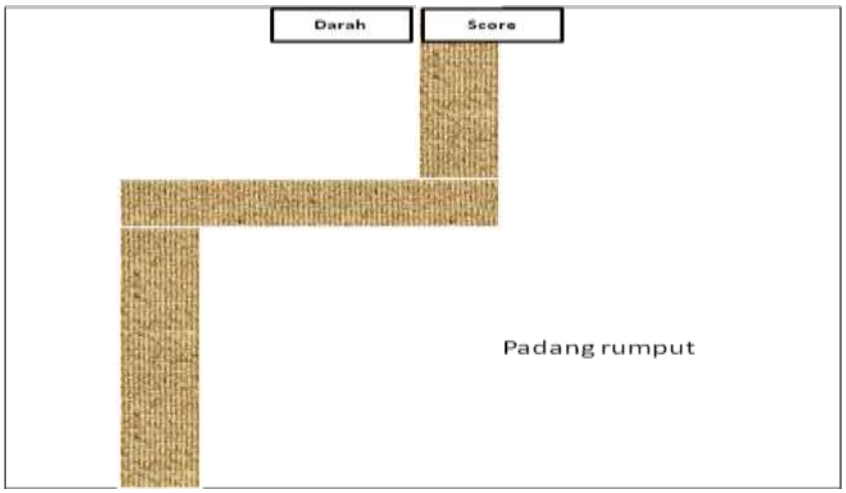
Gambar 3.37 *Storyboard* Tempat 35



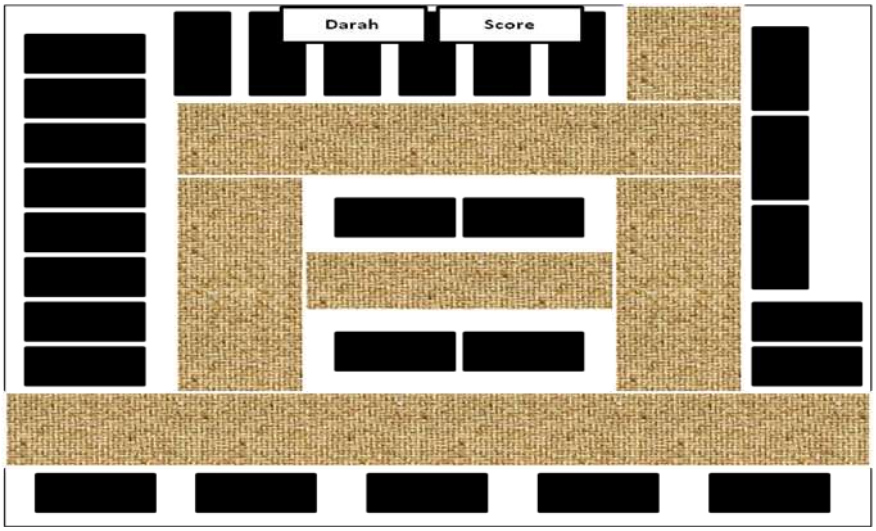
Gambar 3.38 *Storyboard* Tempat 36



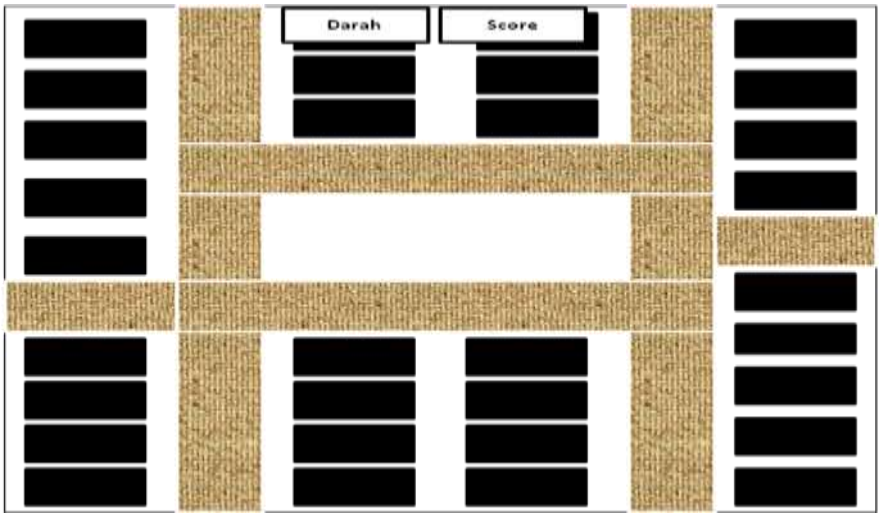
Gambar 3.39 *Storyboard* Tempat 37



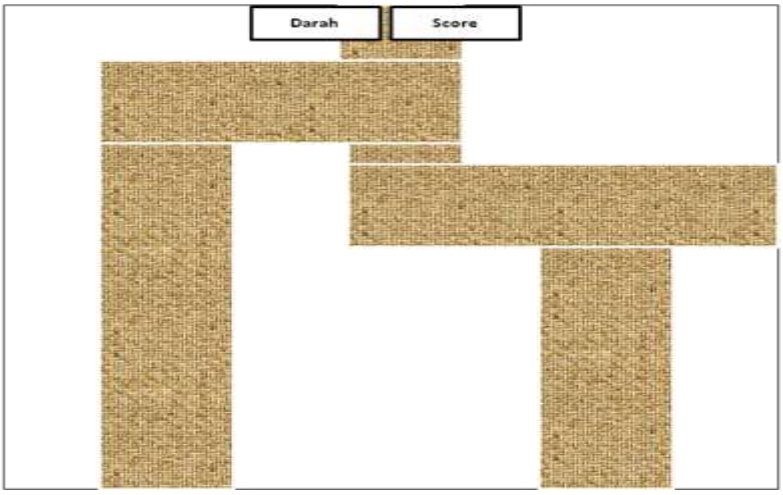
Gambar 3.40 *Storyboard* Tempat 38



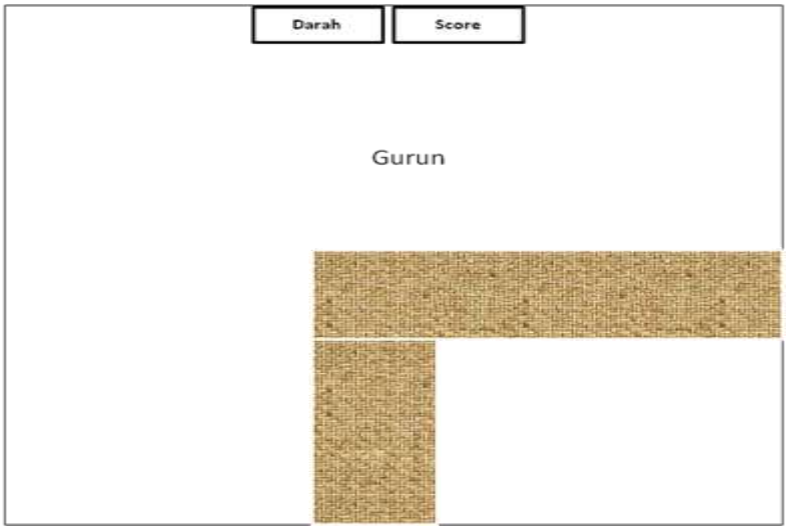
Gambar 3.41 *Storyboard* Tempat 39



Gambar 3.42 *Storyboard* Tempat 40



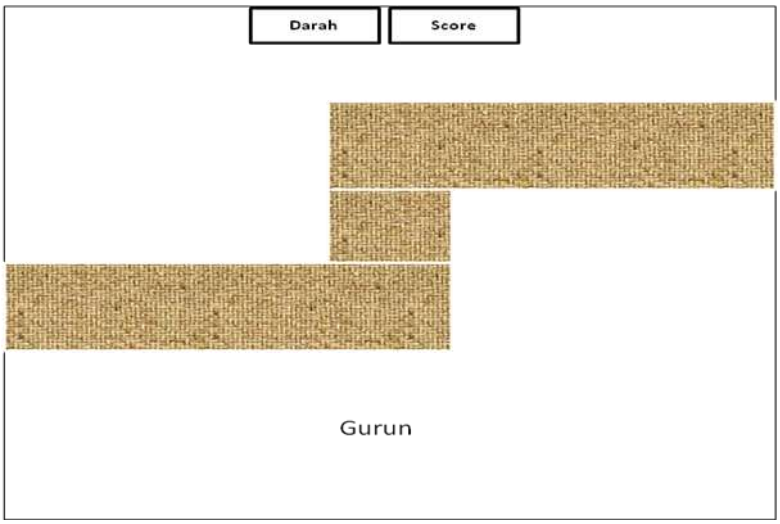
Gambar 3.43 *Storyboard* Tempat 41



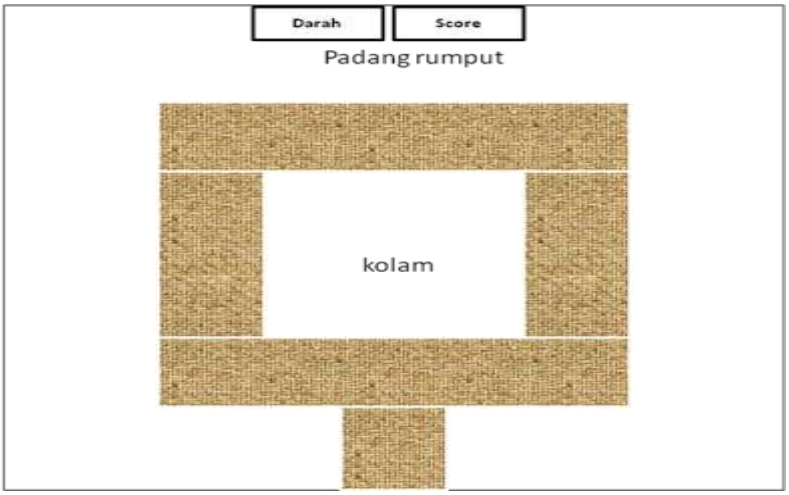
Gambar 3.44 *Storyboard* Tempat 42



Gambar 3.45 *Storyboard* Tempat 43



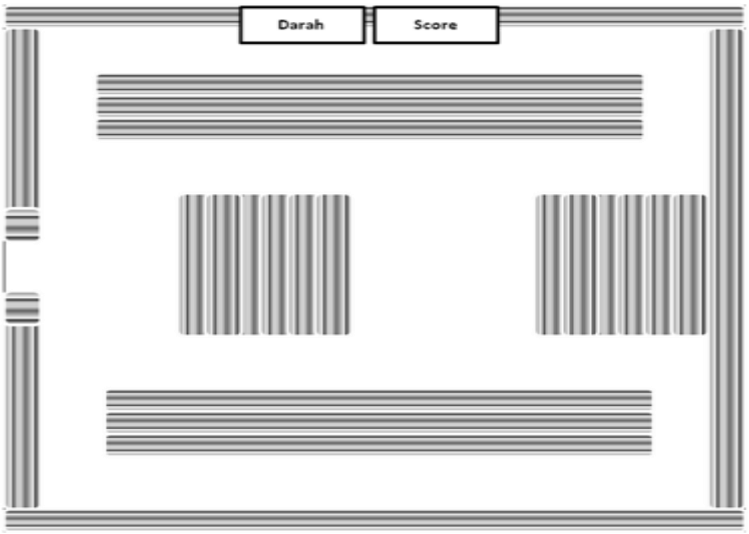
Gambar 3.46 *Storyboard* Tempat 44



Gambar 3.47 *Storyboard* Tempat 45



Gambar 3.48 *Storyboard* Tempat 46

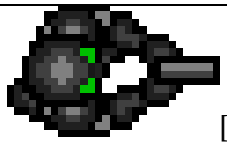
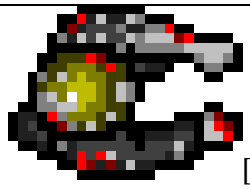


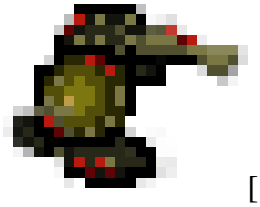
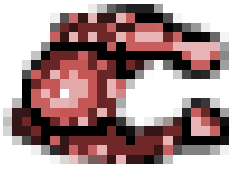
Gambar 3.49 *Storyboard* Tempat 47

3.1.2.3 Konten-konten pada *Game* Edukasi Sejarah Walisongo

Konten *game* merupakan faktor penting dalam menjaga pemain agar tidak cepat merasa bosan dan tetap terlibat dalam dunia *game*. Berikut adalah konten-konten yang akan digunakan dalam pembangunan aplikasi *game* edukasi sejarah walisongo dapat dilihat pada Tabel 3.2.

Tabel 3.2 Konten-konten pada *game* Edukasi Sejarah Walisongo

No.	Nama	Gambar	Penjelasan
1	Player	 [15]	Player merupakan karakter utama dalam *game* edukasi sejarah walisongo yang di kendalikan oleh pemain *game*.
2	Zombie 1	 [15]	Zombie merupakan karakter musuh yang akan menghalangi player dalam menjalankan

			misinya.
3	Zombie 2	 [15]	Zombie merupakan karakter musuh yang akan menghalangi player dalam menjalankan misinya.
4	Zombie 3	 [15]	Zombie merupakan karakter musuh yang akan menghalangi player dalam menjalankan misinya.
5	Senjata 1		Senjata yang digunakan oleh player untuk mengalahkan musuhnya.
6	Senjata 2		Senjata yang digunakan oleh player untuk mengalahkan musuhnya.
7	Senjata 3		Senjata yang digunakan oleh player untuk mengalahkan musuhnya.
8	Senjata 4		Senjata yang digunakan oleh player untuk mengalahkan musuhnya.
9	Senjata 5		Senjata yang digunakan oleh player untuk mengalahkan

			musuhnya.
10	Perban	 [15]	Perban untuk mengobati luka dari player sebanyak 20% darah/perban
11	Amunisi	 [15]	Amunisi untuk mengisi senjata player
12	Sunan Ampel		Foto Sunan yang akan diselamatkan oleh player
13	Sunan Bonang		Foto Sunan yang akan diselamatkan oleh player
14	Sunan Drajat		Foto Sunan yang akan diselamatkan oleh player
15	Sunan Giri		Foto Sunan yang akan diselamatkan oleh player
16	Sunan Gresik		Foto Sunan yang akan diselamatkan oleh player
17	Sunan Gunung Jati		Foto Sunan yang akan diselamatkan oleh player

18	Sunan Kalijaga		Foto Sunan yang akan diselamatkan oleh player
19	Sunan Kudus		Foto Sunan yang akan diselamatkan oleh player
20	Sunan Muria		Foto Sunan yang akan diselamatkan oleh player
21	Pohon		Pohon merupakan obstacle atau hambatan.

3.1.2.4 Analisis *Level*

Game sejarah walisongo ini terdiri dari 47 tempat, dimana setiap level dibagi bukan berdasarkan tingkat kesulitan, tetapi berdasarkan materi yang disampaikan. Adapun materi yang disampaikan dapat di lihat dari tabel 3.3

Tabel 3.3 Konten-konten pada *game* Edukasi Sejarah Walisongo

No	Tempat	Materi
1.	Tempat 1	Sejarah lahirnya sunan Maulana Malik Ibrahim.
2.	Tempat 2	Silsilah sunan Maulana Malik Ibrahim.
3.	Tempat 3	Sejarah penyebaran agama islam oleh sunan Maulana Malik Ibrahim.

4.	Tempat 4	Sejarah penyebaran agama islam oleh sunan Maulana Malik Ibrahim.
5	Tempat 5	Kisah mashyur sunan maulana malik Ibrahim
6	Tempat 6	Sejarah lahirnya sunan Ampel
7	Tempat 7	Silsilah sunan Ampel
8	Tempat 8	Sejarah penyebaran agama islam oleh sunan Ampel
9	Tempat 9	Sejarah penyebaran agama islam oleh sunan Ampel
10	Tempat 10	Keturunan Sunan Ampel
11	Tempat 11	Sejarah lahirnya sunan Bonang
12	Tempat 12	Silsilah sunan Bonang
13	Tempat 13	Sejarah penyebaran agama islam oleh sunan Bonang
14	Tempat 14	Sejarah penyebaran agama islam oleh sunan Bonang
15	Tempat 15	Karya seni sunan Bonang
16	Tempat 16	Sejarah lahirnya sunan Drajat
17	Tempat 17	Silsilah sunan Drajat
18	Tempat 18	Sejarah penyebaran agama islam oleh sunan Drajat
19	Tempat 19	Sejarah penyebaran agama islam oleh sunan Drajat
20	Tempat 20	Makna filosofi ke tujuh sap tangga dari tataran komplek makam sunan Drajat
21	Tempat 21	Sejarah lahirnya sunan Kalijaga

22	Tempat 22	Silsilah sunan Kalijaga
23	Tempat 23	Sejarah penyebaran agama islam oleh sunan Kalijaga
24	Tempat 24	Sejarah penyebaran agama islam oleh sunan Kalijaga
25	Tempat 25	Salah satu karya seni sunan kalijaga
26	Tempat 26	Sejarah lahirnya sunan Kudus
27	Tempat 27	Silsilah sunan Kudus
28	Tempat 28	Sejarah penyebaran agama islam oleh sunan Kudus
29	Tempat 29	Sejarah penyebaran agama islam oleh sunan Kudus
30	Tempat 30	Sunan Kudus menjadi seorang senopati atau panglima perang
31	Tempat 31	Sejarah lahirnya sunan Muria
32	Tempat 32	Silsilah sunan Muria
33	Tempat 33	Sejarah penyebaran agama islam oleh sunan Muria
34	Tempat 34	Sejarah penyebaran agama islam oleh sunan Muria
35	Tempat 35	Kesaktian sunan Muria
36	Tempat 36	Sejarah lahirnya sunan Giri
37	Tempat 37	Silsilah sunan Giri
38	Tempat 38	Sejarah penyebaran agama islam oleh sunan Giri
39	Tempat 39	Sejarah penyebaran agama islam oleh sunan Giri
40	Tempat 40	Perkawinan Sunan Giri

41	Tempat 41	Sejarah lahirnya sunan Gunung Jati
42	Tempat 42	Silsilah sunan Gunung Jati
43	Tempat 43	Sejarah penyebaran agama islam oleh sunan Gunung Jati
44	Tempat 44	Sejarah penyebaran agama islam oleh sunan Gunung Jati
45	Tempat 45	Keturunan Sunan Gunung Jati
46	Tempat 46	Peran dakwah sunan Gunung Jati
47	Tempat 47	Sifat-sifat berdakwah yang dimiliki para tokoh walisongo

3.1.2.5 *Gameplay*

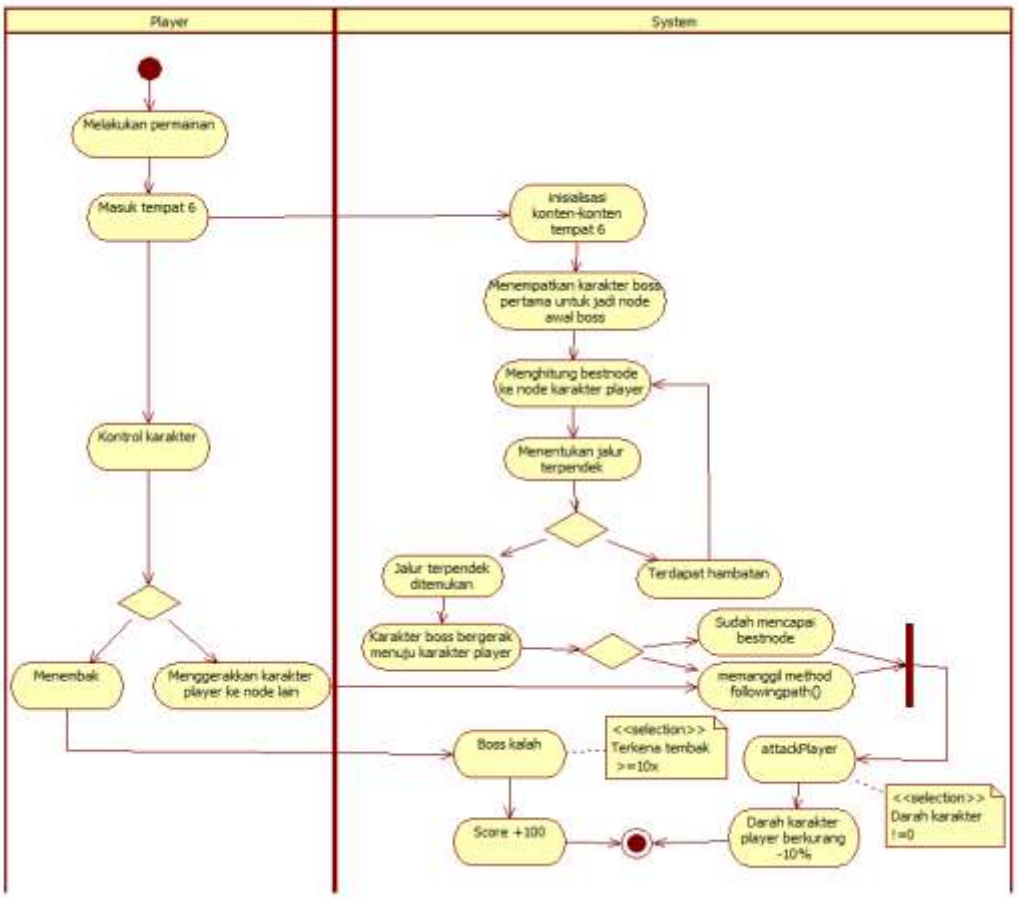
Misi dalam *game* edukasi sejarah walisongo ini adalah mendapatkan Sembilan foto dari tokoh walisongo, dan mengalahkan tiga bos zombie atau musuh dalam *game*. dimana karakter pemain memulai permainan dengan darah 100% dan score 0. Score dapat bertambah apabila pemain mendapatkan foto dan mengalahkan zombie.

Tipe *game* berupa Petualangan dimana karakter pemain dapat digerakan oleh pemain untuk bergerak ke kiri, bergerak ke, bergerak ke atas, bergerak ke bawah, dan menembak.

3.1.2.6 Analisis Karakter Bos Pada *Game* Edukasi Sejarah Walisongo

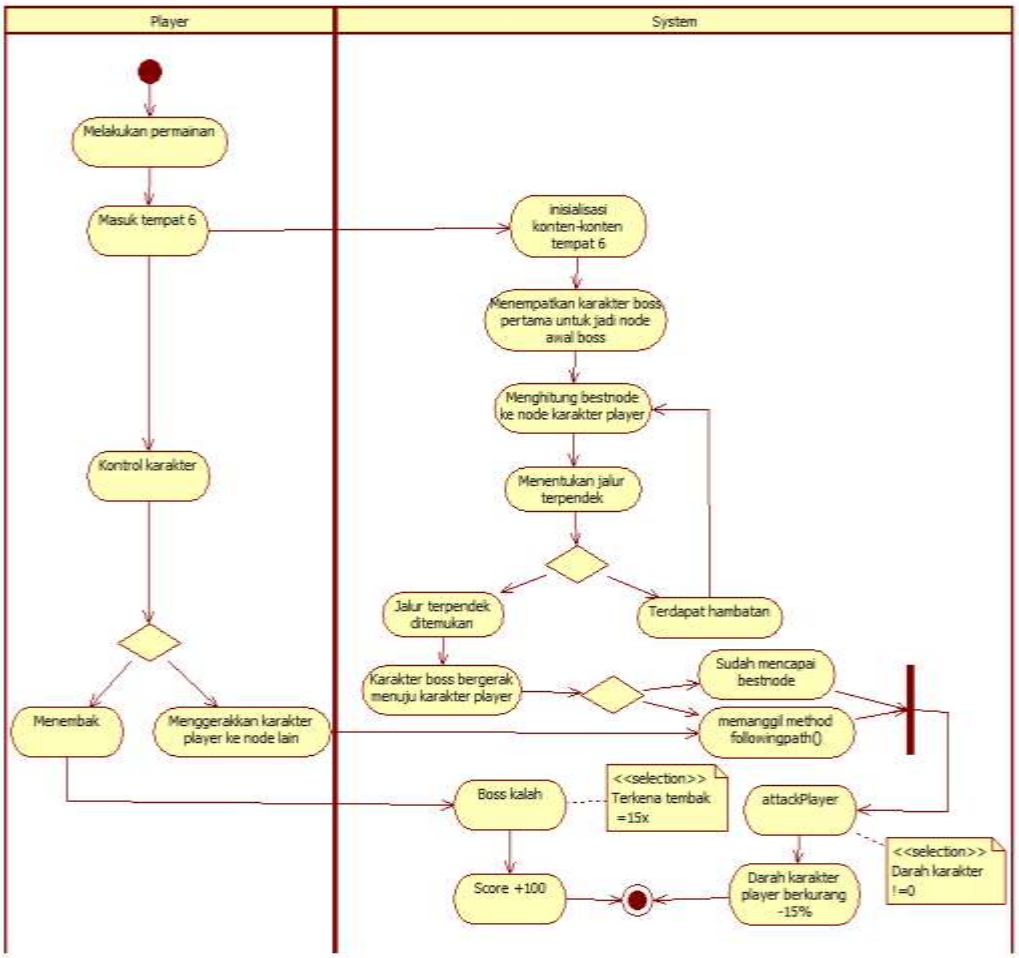
Tingkat kesulitan pada *game* edukasi sejarah walisongo ini adalah pada bos, dimana terdapat 3(tiga) bos yang harus dikalahkan oleh pemain untuk menyelesaikan misi. Berikut penjelasannya:

1. Bos pertama yang berada ditempat ke 6 (enam) berbeda dengan zombie biasa. Selain dari ukurannya lebih besar, bos pertama ini dapat mengecek seluruh node pada tempat keberadaanya untuk menuju ke karakter pemain. Darah bos pertama ini dua kali lebih besar dari anak buahnya dan apabila karakter pemain terkena pukul oleh bos akan berkurang dua kali lipat dari zombie biasa. Untuk lebih jelasnya dapat dilihat pada *activity diagram* bos pertama.



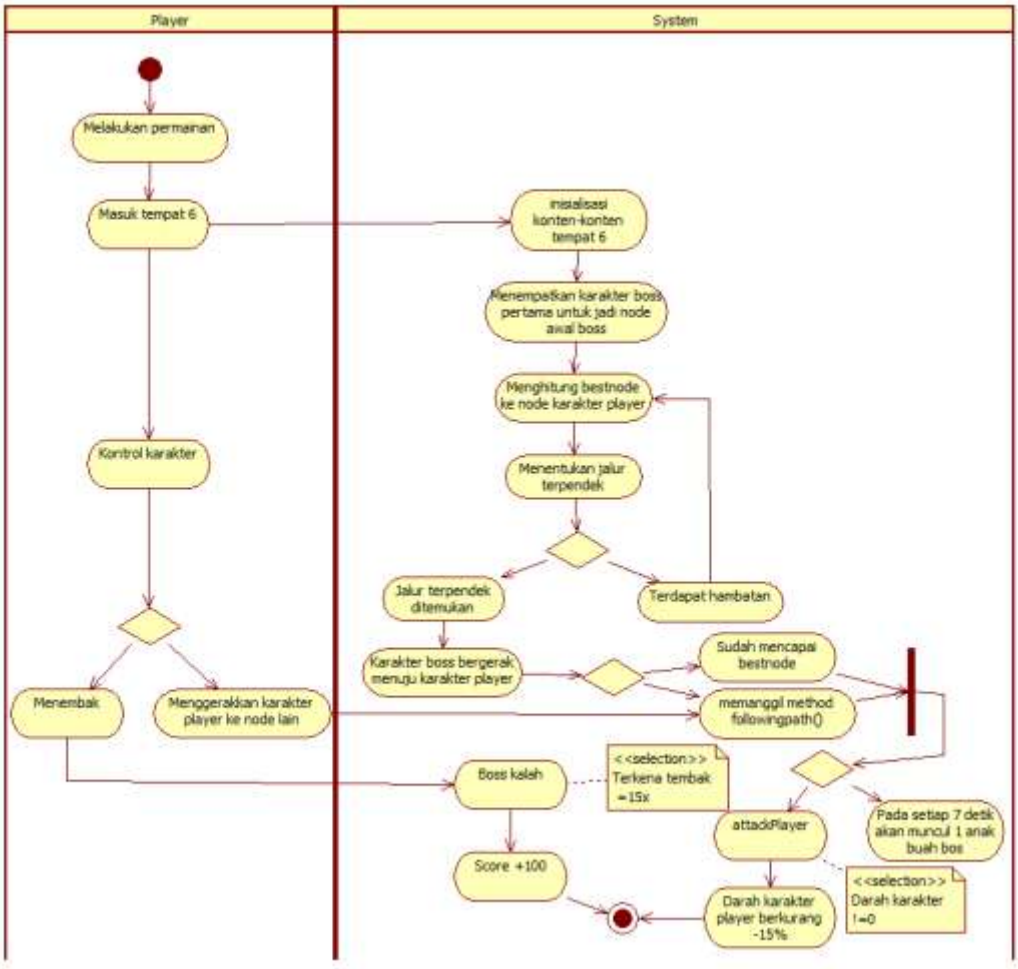
Gambar 3.50 *Activity Diagram* Bos Pertama

2. Bos kedua berada ditempat 25 (dua puluh lima). Pada bos ke dua ini sama dengan bos pertama yang dapat mengecek seluruh node pada tempat keberadaanya untuk menuju ke karakter pemain. Perbedaannya dengan bos pertama adalah darah bos ke dua lebih besar dari bos pertama dan apabila karakter pemain terkena pukul oleh bos kedua akan berkurang dua kali lipat dari bos pertama. Untuk lebih jelasnya dapat dilihat pada *activity diagram* bos kedua.



Gambar 3.51 *Activity Diagram* Bos Kedua

3. Bos ketiga berada di tempat 47 (empat puluh tujuh). Pada bos ke dua ini sama dengan bos pertama yang dapat mengecek seluruh node pada tempat keberadaanya untuk menuju ke karakter pemain. Perbedaannya dengan bos pertama adalah darah bos ke dua lebih besar dari bos pertama dan apabila karakter pemain terkena pukul oleh bos kedua akan berkurang dua kali lipat dari bos pertama. Untuk lebih jelasnya dapat dilihat pada *activity diagram* bos ketiga.



Gambar 3.52 *Activity Diagram* Bos Ketiga

3.1.1.7 Analisis Algoritma A* (A Star) pada *Game* Edukasi Sejarah Walisongo.

Terdapat beberapa hal yang perlu didefinisikan terlebih dahulu dalam kasus *game pathfinding* dengan penerapan algoritma A* *(A Star)*. Adapun istilah-istilah yang akan dibahas yaitu *path, open list, closed list,* nilai f, g dan n.

Algoritma A* menggunakan dua senarai yaitu *OPEN* dan *CLOSED*. *OPEN* adalah senarai (*list*) yang digunakan untuk menyimpan simpul-simpul yang pernah dibangkitkan dan nilai heuristiknya telah dihitung tetapi belum terpilih sebagai simpul terbaik (*best node*) dengan kata lain, *OPEN* berisi simpul-simpul masih memiliki peluang untuk terpilih sebagai simpul terbaik, sedangkan *CLOSED* adalah senarai untuk menyimpan simpul-simpul yang sudah pernah dibangkitkan dan sudah pernah terpilih sebagai simpul terbaik. Artinya, *CLOSED* berisi simpul-simpul yang tidak mungkin terpilih sebagai simpul terbaik (peluang untuk terpilih sudah tertutup).[9]

Fungsi f sebagai estimasi fungsi evaluasi terhadap *node* n, dapat dituliskan :

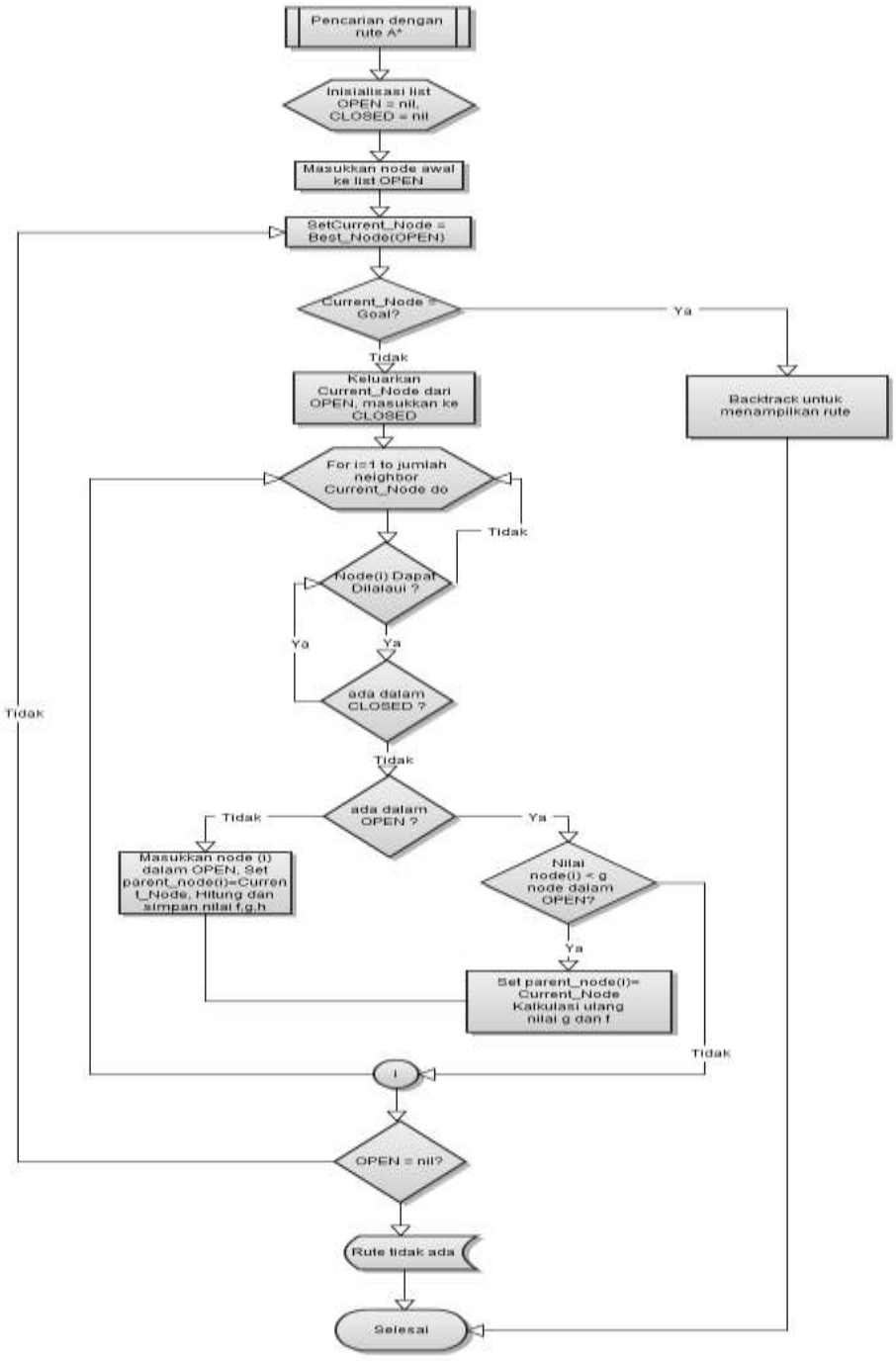
$$f(n) = g(n) + h(n)$$

dengan :

$f(n)$ = fungsi evaluasi (jumlah $g(n)$ dengan $h(n)$)

$g(n)$ = biaya *(cost)* yang dikeluarkan dari keadaan awal sampai keadaan n

$h(n)$ = estimasi biaya untuk sampai pada suatu tujuan mulai dari n



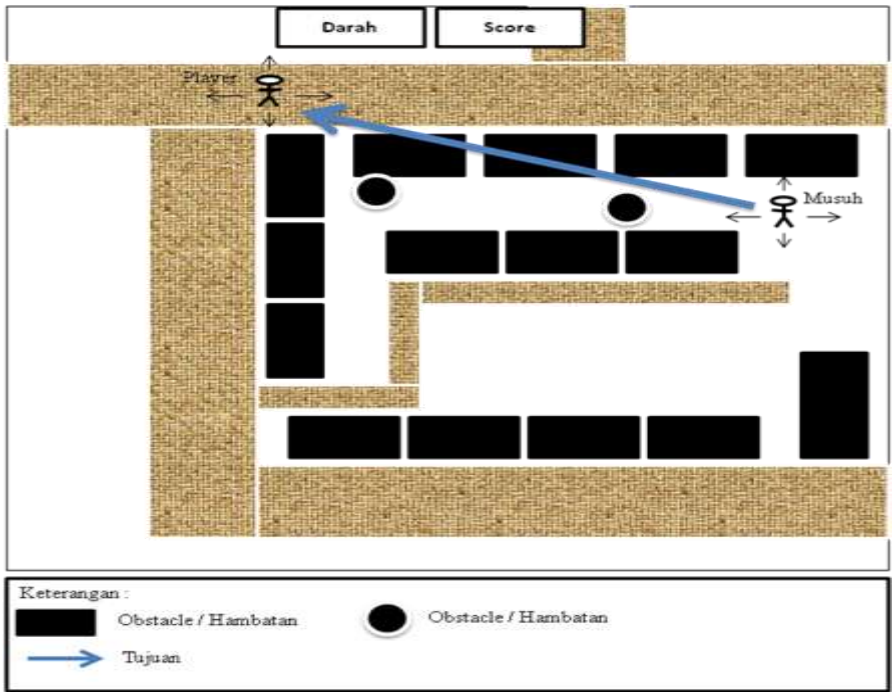
Gambar 3.53 Flowchart A* (A Star)

Pergerakan diagonal pada map diperbolehkan maka digunakan fungsi *Heuristic Non-Manhattan Distance*. Manhattan Distance adalah fungsi *heuristic* standar untuk algoritma A*, yang digunakan pada aplikasi yang memiliki empat arah gerakkan.

Rumus perhitugan fungsi *heuristic manhattan distance :*

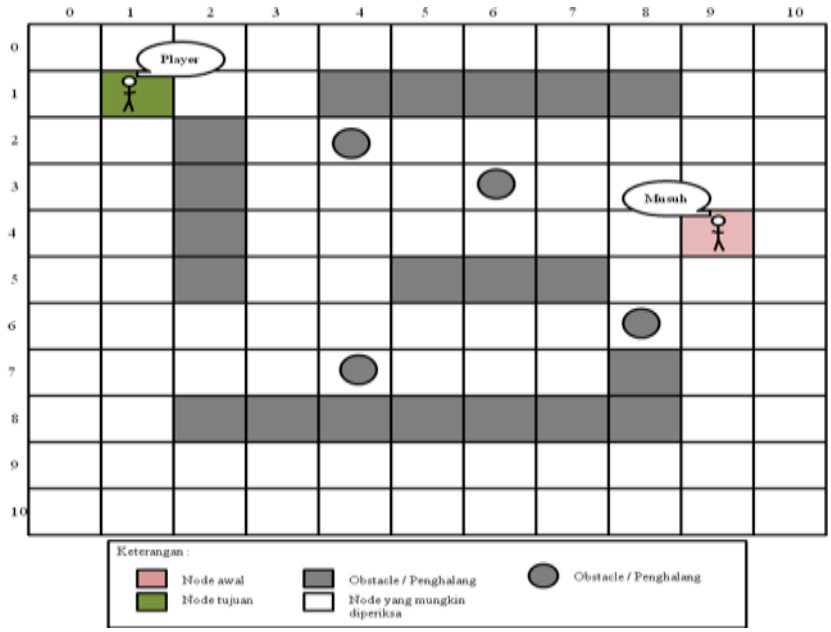
```
h_diagonal(n) = -(abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(n) = (abs(n.x-goal.x)+(abs(n.y-goal.y))
h(n) =(h_diagonal(n) + (h_orthogonal(n) – (2*h_diagonal(n)))
```

Dalam kondisi jarak yang ditempuh, terdapat hambatan atau rintangan.sehingga untuk mencari solusi terbaik. Setiap pergerakan yang dilakukan disimpan dalam suatu *list. List* ini akan digunakan untuk melakukan pengecekan untuk menemukan solusi dan langkah terpendek. Untuk lebih jelasnya dapat dilihat pada gambar berikut.



Gambar 3.54 Contoh Kodisi Awal Pencarian

Contoh kondisi awal pencarian ini, menggunakan storyboard pada tempat ke 6 (enam) dimana pada *game edukasi* sejarah walisongo yang dapat mengecek seluruh node pada tempat tersebut adalah bos zombie. untuk lebih mudah memahami proses pencarian karakter musuh (zombie) ke karakter player (pemain). Berikut ini perhitungan pada *game* edukasi sejarah walisogo, yang diberi bobot terkecil yaitu 1 (satu).



Gambar 3.55 Contoh Kondisi Awal Pencarian

Posisi node awal = Ax:9, Ay:4

Posisi node tujuan = goal x:1,goal y:1

Langkah ke satu

n(8,3) : g(8,3) = 1	
h_orthogonal(n)	= (abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(8,3)	= (abs(8-1)+abs(3-1)))
	= (abs(7)+abs(2))
	= 9
h_diagonal(n)	= -(abs(n.x-goal.x)+abs(n.y-goal.y))
h_diagonal(8,3)	= -(abs(8-1)+abs(3-1))

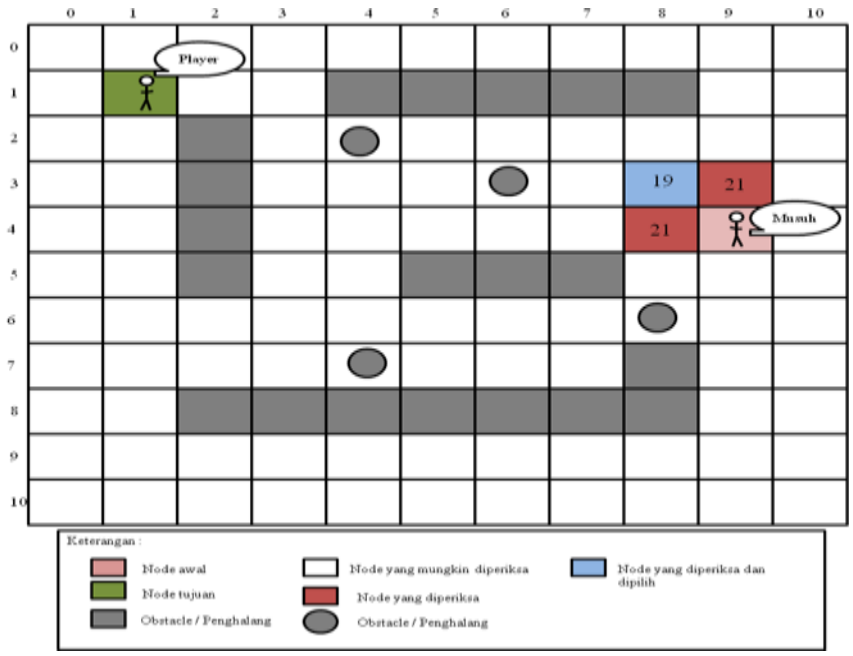
= -(abs(7)+abs(2))
= -9

h(n) = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(8,3) = (-9)+(9-(2*(-9)))
= -9+27
= 18
f(8,3) = g(8,3)+h(8,3)
=1+18
= 19

N(8,4):g(8,4) = 1
h_orthogonal(n) = (abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(8,4) = (abs(8-1)+abs(4-1))
= (abs(7)+abs(3))
= 10
h_diagonal(n) = -(abs(n.x-goal.x), abs(n.y-goal.y))
h_diagonal(8,4) = -(abs(8-1)+abs(4-1))
= -(abs(7)+abs(3))
= -10
h(n) = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(8,4) = (-10)+(10-(2*(-10)))
= -10+30
= 20
f(8,4) = g (8,4)+h(8,4)
= 1+20
= 21

n(9,3) : g(9,3) = 1
h_orthogonal(n) = (abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(9,3) = (abs(9-1)+abs(3-1))
= (abs(8)+abs(2))
= 10
h_orthogonal(n) = -(abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(9,3) = -(abs(9-1)+abs(3-1))
= -(abs(8)+abs(2))
= -10
h(n) = h_diagonal(n) + (h_orthogonal(n)-(2*h_diagonal(n)))
h(9,3) = (-10) + (10-(2*(-10)))
= -10+30

	= 20
f(9,3)	= g(9,3)+h(9,3)
	=1+20
	= 21



Gambar 3.56 Langkah Pertama Pencarian Karakter

Pada gambar 3.52 terdapat tiga simpul yag mungkin menjadi *BestNode* yaitu (8,4), (9,3), dan (8,3) dengan biaya terkecil yaitu f(n)=19. Sehingga dari ketiga node yang mungkin diperiksa, maka node yang dipilih adalah (8,3) dengan biaya terkecil.

Langkah ke dua

n(7,2) : g(7,2) = 1		
	h_orthogonal(n)	= (abs(n.x-goal.x)+abs(n.y-goal.y))
	h_orthogonal(7,2)	= (abs(7-1)+abs(2-1)))
		= (abs(6)+abs(1))
		= 7
	h_diagonal(n)	= -(abs(n.x-goal.x)+abs(n.y-goal.y))
	h_diagonal(7,2)	= -(abs(7-1)+abs(2-1))
		= -(abs(6)+abs(1))
		= -7

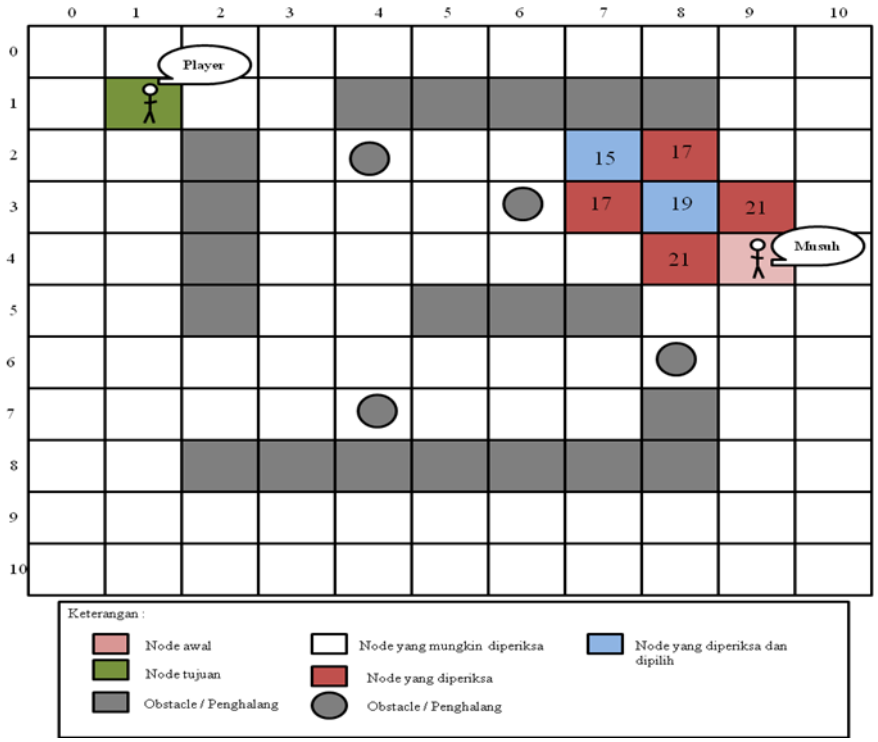
h(n) = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(7,2) = (-7)+(7-(2*(-7)))
= -7+21
= 14
f(7,2) = g(7,2)+h(7,2)
=1+14
= 15

N(7,3):g(7,3) = 1
h_orthogonal(n) = (abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(7,3) = (abs(7-1)+abs(3-1))
= (abs(6)+abs(2))
= 8
h_diagonal(n) = -(abs(n.x-goal.x), abs(n.y-goal.y))
h_diagonal(7,3) = -(abs(7-1)+abs(3-1))
= -(abs(6)+abs(2))
= -8
h(n) = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(7,3) = (-8)+(8-(2*(-8)))
= -8+24
= 16
f(7,3) = g (7,3)+h(7,3)
= 1+16
= 17

n(8,2) : g(8,2) = 1
h_orthogonal(n) = (abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(8,2) = (abs(8-1)+abs(2-1))
= (abs(7)+abs(1))
= 8
h_orthogonal(n) = -(abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(8,2) = -(abs(8-1)+abs(2-1))
= -(abs(7)+abs(1))
= -8
h(n) = h_diagonal(n) + (h_orthogonal(n)-(2*h_diagonal(n)))
h(8,2) = (-8) + (8-(2*(-8)))
= -8+24
= 16
f(8,2) = g(8,2)+h(8,2)

=1+16

= 17



Gambar 3.57 Langkah Kedua Pencarian Karakter

Pada gambar 3.53 terdapat tiga simpul yag mungkin menjadi *BestNode* yaitu (7,2), (8,2), (7,3), (8,3), (9,3), dan (8,4) dengan biaya terkecil yaitu f(n)=15. Sehingga dari ketiga node yang diperiksa, maka node yang dipilih adalah (7,2) dengan biaya terkecil.

Langkah ketiga

n(6,2) : g(6,2) = 1

h_orthogonal(n) = (abs(n.x-goal.x)+abs(n.y-goal.y))

h_orthogonal(6,2) = (abs(6-1)+abs(2-1)))

= (abs(5)+abs(1))

= 6

h_diagonal(n) = -(abs(n.x-goal.x)+abs(n.y-goal.y))

h_diagonal(6,2) = -(abs(6-1)+abs(2-1))

= -(abs(5)+abs(1))

= -6

h(n) = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))

h(6,2) = (-6)+(6-(2*(-6)))

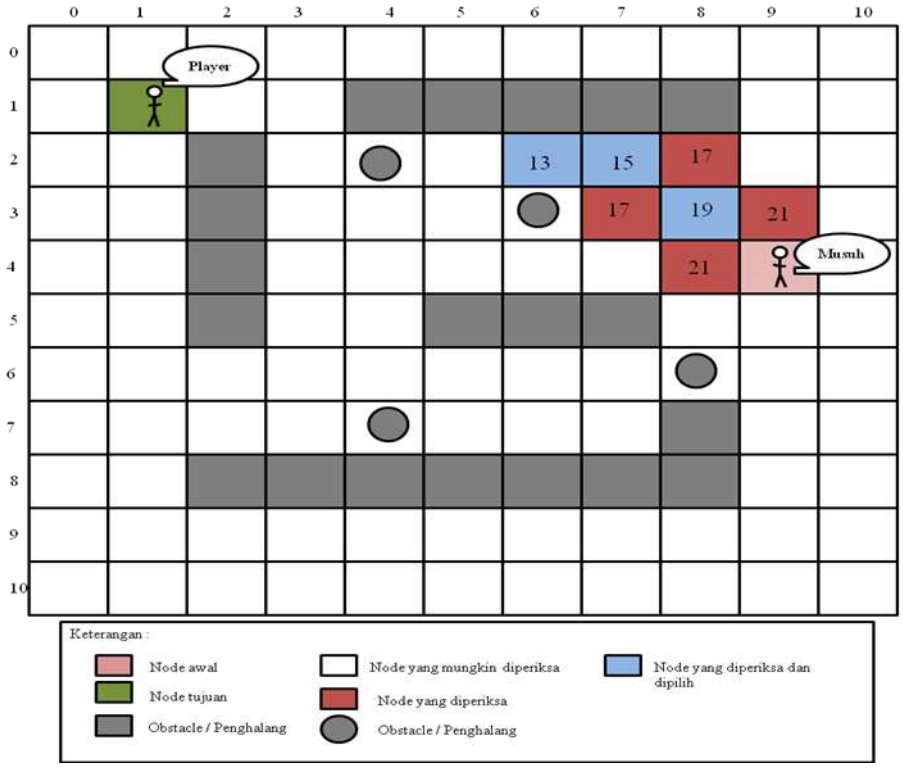
= -6+18

= 12

f(6,2) = g(6,2)+h(6,2)

=1+12

= 13



Gambar 3.58 Langkah Ketiga Pencarian Karakter

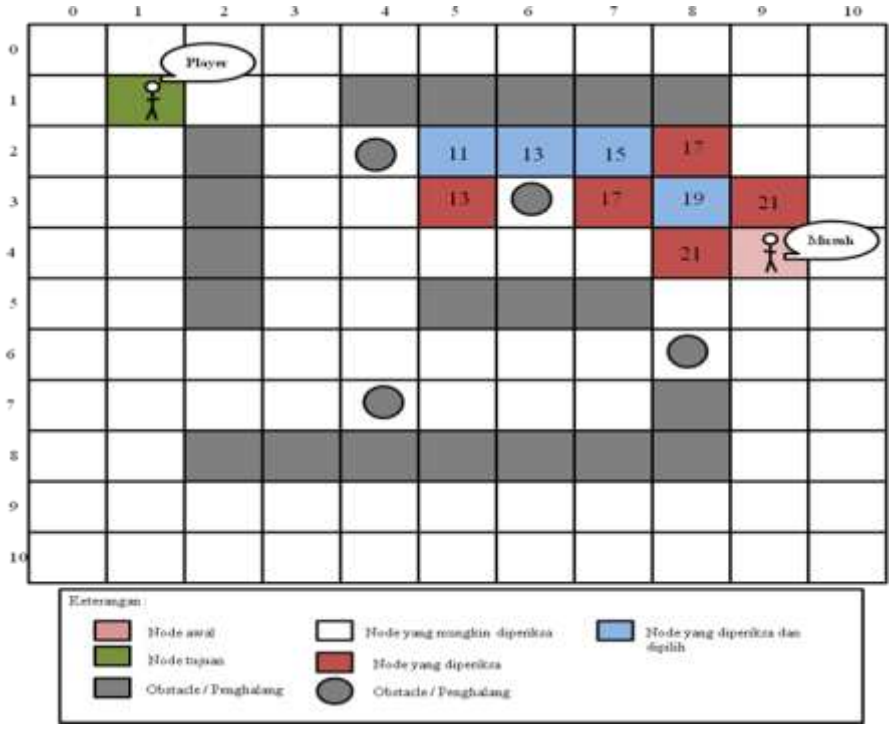
Pada gambar 3.54 terdapat tiga simpul yag mungkin menjadi *BestNode* yaitu (6,2), (7,2), (8,2), (7,3), (8,3), (9,3), dan (8,4) dengan biaya terkecil yaitu f(n)=13. Sehingga dari ketiga node yang mungkin diperiksa, maka node yang dipilih adalah (6,2) dengan biaya terkecil.

Langkah keempat

```
n(5,2) : g(5,2) = 1
        h_orthogonal(n)     = (abs(n.x-goal.x)+abs(n.y-goal.y))
        h_orthogonal(5,2)   = (abs(5-1)+abs(2-1)))
                            = (abs(4)+abs(1))
                            = 5
        h_diagonal(n)       = -(abs(n.x-goal.x)+abs(n.y-goal.y))
        h_diagonal(5,2)     = -(abs(5-1)+abs(2-1))
                            = -(abs(4)+abs(1))
                            = -5
h(n)    = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(5,2)  = (-5)+(5-(2*(-5)))
        = -5+15
        = 10
f(5,2)  = g(5,2)+h(5,2)
        =1+10
        = 11
```

```
n(5,3) : g(5,3) = 1
        h_orthogonal(n)     = (abs(n.x-goal.x)+abs(n.y-goal.y))
        h_orthogonal(5,3)   = (abs(5-1)+abs(3-1)))
                            = (abs(4)+abs(2))
                            = 6
        h_diagonal(n)       = -(abs(n.x-goal.x)+abs(n.y-goal.y))
        h_diagonal(5,3)     = -(abs(5-1)+abs(3-1))
                            = -(abs(5)+abs(2))
                            = -6
h(n)    = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(5,3)  = (-6)+(6-(2*(-6)))
        = -6+18
        = 12
```

f(5,3) = g(5,3)+h(5,3)
=1+12
= 13



Gambar 3.59 Langkah Keempat Pencarian Karakter

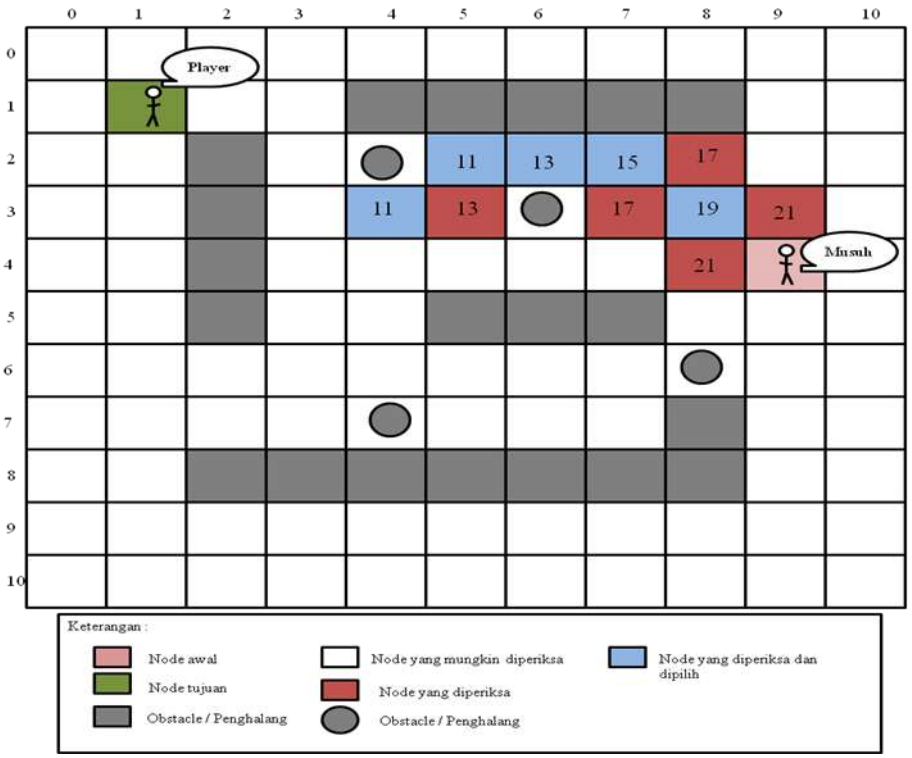
Pada gambar 3.55 terdapat tiga simpul yag mungkin menjadi *BestNode* yaitu (5,2), (6,2) , (7,2) , (8,2) , (5,3) , (7,3) , (8,3) , (9,3) , dan (8,4) dengan biaya terkecil yaitu f(n)=11. Sehingga dari ketiga node yang mungkin diperiksa, maka node yang dipilih adalah (5,2) dengan biaya terkecil.

Langkah kelima

n(4,3) : g(4,3) = 1

h_orthogonal(n) = (abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(4,3) = (abs(4-1)+abs(3-1)))
= (abs(3)+abs(2))
= 5
h_diagonal(n) = -(abs(n.x-goal.x)+abs(n.y-goal.y))
h_diagonal(4,3) = -(abs(4-1)+abs(3-1))
= -(abs(3)+abs(2))
= -5

h(n) = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(4,3) = (-5)+(5-(2*(-5)))
= -5+15
= 10
f(4,3) = g(4,3)+h(4,3)
=1+10
= 11



Gambar 3.60 Langkah Kelima Pencarian Karakter

Pada gambar 3.56 terdapat tiga simpul yag mungkin menjadi *BestNode* yaitu (5,2), (6,2) , (7,2) , (8,2) , (4,3) , (5,3) , (7,3) , (8,3) , (9,3) , dan (8,4) dengan biaya terkecil yaitu f(n)=11. Sehingga dari ketiga node yang mungkin diperiksa, maka node yang dipilih adalah (4,3) dengan biaya terkecil.

Langkah keenam

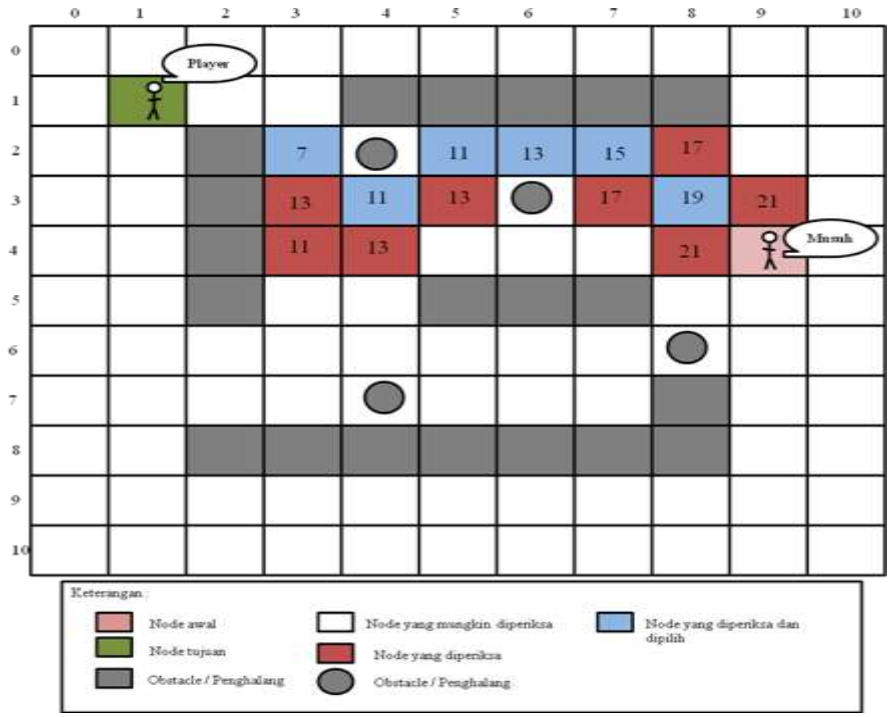
```
n(4,4) : g(4,4) = 1
        h_orthogonal(n)      = (abs(n.x-goal.x)+abs(n.y-goal.y))
        h_orthogonal(4,4)    = (abs(4-1)+abs(4-1)))
                             = (abs(3)+abs(3))
                             = 6
        h_diagonal(n)        = -(abs(n.x-goal.x)+abs(n.y-goal.y))
        h_diagonal(4,4)      = -(abs(4-1)+abs(4-1))
                             = -(abs(3)+abs(3))
                             = -6
h(n)    = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(4,4)  = (-6)+(6-(2*(-6))
        = -6+18
        = 12
f(4,4)  = g(4,4)+h(4,4)
        =1+12
        = 13
```

```
n(3,4) : g(3,4) = 1
        h_orthogonal(n)      = (abs(n.x-goal.x)+abs(n.y-goal.y))
        h_orthogonal(3,4)    = (abs(3-1)+abs(4-1)))
                             = (abs(2)+abs(3))
                             = 5
        h_diagonal(n)        = -(abs(n.x-goal.x)+abs(n.y-goal.y))
        h_diagonal(3,4)      = -(abs(3-1)+abs(4-1))
                             = -(abs(2)+abs(3))
                             = -5
h(n)    = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(3,4)  = (-5)+(5-(2*(-5))
        = -5+15
        = 10
```

f(3,4) = g(3,4)+h(3,4)
=1+10
= 11

n(3,3) : g(3,3) = 1
h_orthogonal(n) = (abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(3,3) = (abs(3-1)+abs(3-1)))
= (abs(2)+abs(2))
= 4
h_diagonal(n) = -(abs(n.x-goal.x)+abs(n.y-goal.y))
h_diagonal(3,3) = -(abs(3-1)+abs(3-1))
= -(abs(2)+abs(2))
= -4
h(n) = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(3,3) = (-4)+(4-(2*(-4))
= -4+16
= 12
f(3,3) = g(3,3)+h(3,3)
=1+12
= 13

n(3,2) : g(3,2) = 1
h_orthogonal(n) = (abs(n.x-goal.x)+abs(n.y-goal.y))
h_orthogonal(3,2) = (abs(3-1)+abs(2-1)))
= (abs(2)+abs(1))
= 3
h_diagonal(n) = -(abs(n.x-goal.x)+abs(n.y-goal.y))
h_diagonal(3,2) = -(abs(3-1)+abs(2-1))
= -(abs(2)+abs(1))
= -3
h(n) = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(3,2) = (-3)+(3-(2*(-3))
= -3+9
= 6
f(3,2) = g(3,2)+h(3,2)
=1+6
= 7



Gambar 3.61 Langkah Keenam Pencarian Karakter

Pada gambar 3.57 terdapat tiga simpul yag mungkin menjadi *BestNode* yaitu (3,2), (5,2), (6,2) , (7,2) , (8,2) , (3,3), (4,3) , (5,3) , (7,3) , (8,3) , (9,3) , (3,4), (4,4), dan (8,4) dengan biaya terkecil yaitu f(n)=9. Sehingga node yang mungkin diperiksa, maka node yang dipilih adalah (3,2) dengan biaya terkecil.

Langkah ketujuh

```
n(3,1) : g(3,1) = 1
        h_orthogonal(n)      = (abs(n.x-goal.x)+abs(n.y-goal.y))
        h_orthogonal(3,1)    = (abs(3-1)+abs(1-1)))
                             = (abs(2)+abs(0))
                             = 2
        h_diagonal(n)        = -(abs(n.x-goal.x)+abs(n.y-goal.y))
        h_diagonal(3,1)      = -(abs(3-1)+abs(1-1))
                             = -(abs(2)+abs(0))
                             = -2
h(n)    = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))
h(3,1)  = (-2)+(2-(2*(-2))
```

= -2+6
= 4
f(3,1) = g(3,1)+h(3,1)
=1+4
= 5

n(2,1) : g(2,1) = 1

h_orthogonal(n) = (abs(n.x-goal.x)+abs(n.y-goal.y))

h_orthogonal(2,1) = (abs(2-1)+abs(1-1)))

= (abs(1)+abs(0))

= 1

h_diagonal(n) = -(abs(n.x-goal.x)+abs(n.y-goal.y))

h_diagonal(2,1) = -(abs(2-1)+abs(1-1))

= -(abs(1)+abs(0))

= -1

h(n) = h_diagonal(n)+(h_orthogonal(n)-(2*h_diagonal(n)))

h(2,1) = (-1)+(1-(2*(-1))

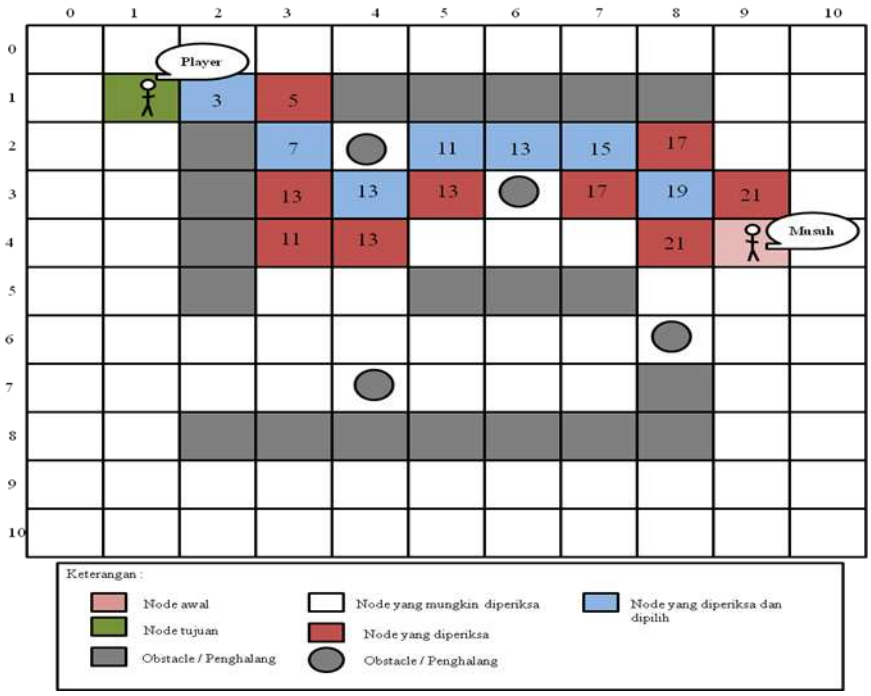
= -1+3

= 2

f(2,1) = g(2,1)+h(2,1)

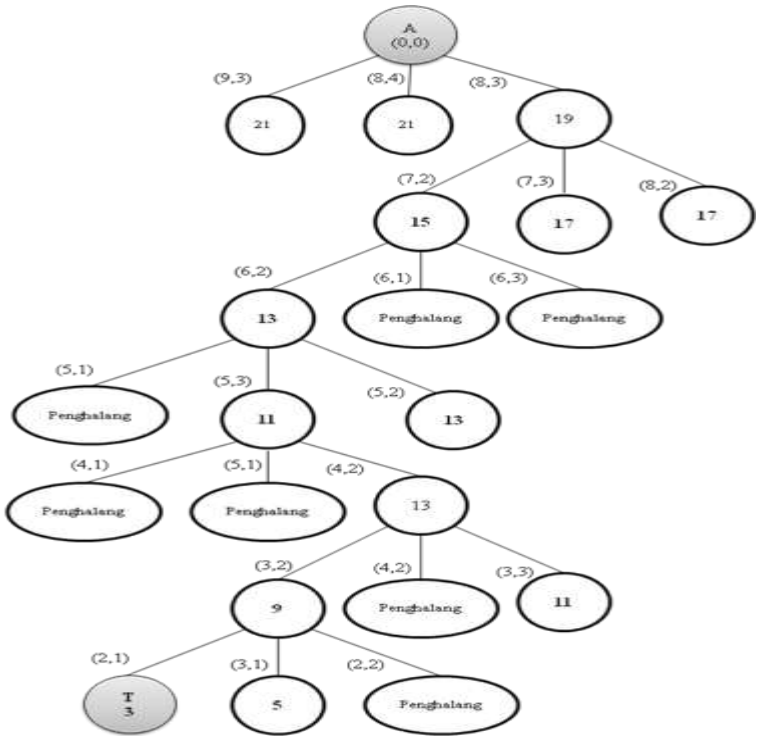
=1+2

=3



Gambar 3.62 Langkah Ketujuh Pencarian Karakter

Pada gambar 3.58 terdapat tiga simpul yag mungkin menjadi *BestNode* yaitu (2,1), (3,1), (3,2), (5,2), (6,2) , (7,2) , (8,2) , (3,3), (4,3) , (5,3) , (7,3) , (8,3) , (9,3), (3,4), (4,4), dan (8,4) dengan biaya terkecil yaitu f(n)=3. Sehingga dari ketiga node yang mungkin diperiksa, maka node yang dipilih adalah (2,1) dengan biaya terkecil dan menjadi node tujuan. Untuk lebih jelasnya dapat dilihat dari pohon pencarian sebagai 3.59.



Gambar 3.63 Pohon pencarian Algoritma A Star

3.2 Analisis Kebutuhan Non-Fungsional

Analisis non-fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Pada analisis kebutuhan sistem non fungsional ini dijelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis pengguna.

3.2.1 Analisis Kebutuhan Perangkat Lunak

Game edukasi sejarah walisongo ini memiliki kebutuhan perangkat lunak sebagai berikut :

1. Semua sistem operasi
2. JDK di 6 atau lebih

Dari hasil pengamatan perangkat lunak, terdapat beberapa hal penting yang harus diperhatikan agar sistem yang dirancang dapat berjalan dengan baik dan sesuai dengan yang diharapkan. Pengguna diharapkan dapat memenuhi kebutuhan perangkat lunak yang telah direkomendasikan agar aplikasi dapat berjalan dengan baik.

3.2.2 Analisis Kebutuhan Perangkat Keras

Agar aplikasi dapat berjalan dengan baik, maka dibutuhkan perangkat keras yang sesuai dengan kebutuhan aplikasi. Spesifikasi minimum perangkat keras yang dibutuhkan agar *game* edukasi sejarah walisongo ini berjalan adalah sebagai berikut:

1. Prosesor dengan kecepatan 1.8 Ghz
2. Memori 256 Mb
3. Harddisk 20 GB
4. VGA card 256 Mb
5. Monitor
6. Mouse dan Keyboard

3.2.3 Analisis Pengguna

Analisis pengguna g*ame* edukasi sejarah walisongo adalah sebagai berikut :

1. Target utama ditujukan untuk siswa-siswi kelas V SDN Karya Bhakti dan kelas VI MI Niblomusibyan Sekolah Dasar
2. Mengetahui dan mengerti komputer sehingga dapat menggunakan aplikasi yang akan dibangun.
3. Pernah memainkan *game* dengan *genre* aksi *shooting* dan petualangan.
4. Mampu membaca dan mengerti bahasa Indonesia.

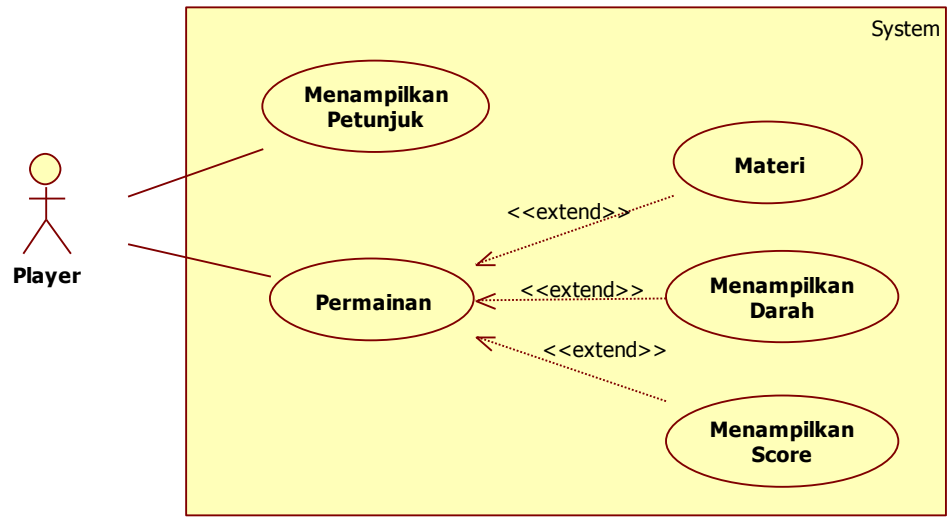
3.3 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik sesuai kebutuhan.

3.3.1 *Use Case Diagram*

Use Case atau *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 informasi yang akan dibuat [6].

Berikut ini adalah perancangan dari proses-proses yang terdapat pada *game edukasi sejarah walisongo*, yang digambarkan dengan *Use Case* Diagram yang dapat dilihat pada gambar dibawah ini



Gambar 3.64 *Use Case Diagram game* edukasi sejarah walisongo

Use Case terdiri dari 3 bagian yaitu definisi *actor*, definisi *Use Case* dan *Use Case* Skenario.

1. Definisi *Actor*

Definisi *Actor* berfungsi untuk menjelaskan *Actor* yang terdapat pada *Use Case Diagram*. Definisi *Actor* diterangkan pada Tabel 3.3.

Tabel 3.4 Definisi *Actor*.

No.	***Actor***	**Deksipsi**
1.	Pemain	Orang yang memainkan *game*

2. Definisi *Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram*. Definisi *Use Case* diterangkan pada Tabel 3.4

Tabel 3.5 Definisi *Use Case.*

No	***Use Case***	**Deskripsi**
1.	Menampilkan petunjuk	Proses untuk memulai permainan
2.	Permainan	Proses melakukan permainan
3.	Materi	Proses penyampaian materi sejarah walisongo
4.	Menampilkan darah	Proses untuk menghitung dan menampilkan darah
5.	Menampilkan score	Proses untuk menghitung dan menampilkan score

3.3.2 Skenario Use Case dan Activity Diagram

Skenario pada diagram *Use Case* berisi mengenai alur kejadian yang di dalamnya menggambarkan alur penggunaan sistem dimana setiap skenario digambarkan dari sudut pandang aktor atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara. Sedangkan *Activity Diagram* atau diagram aktivitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.[6]

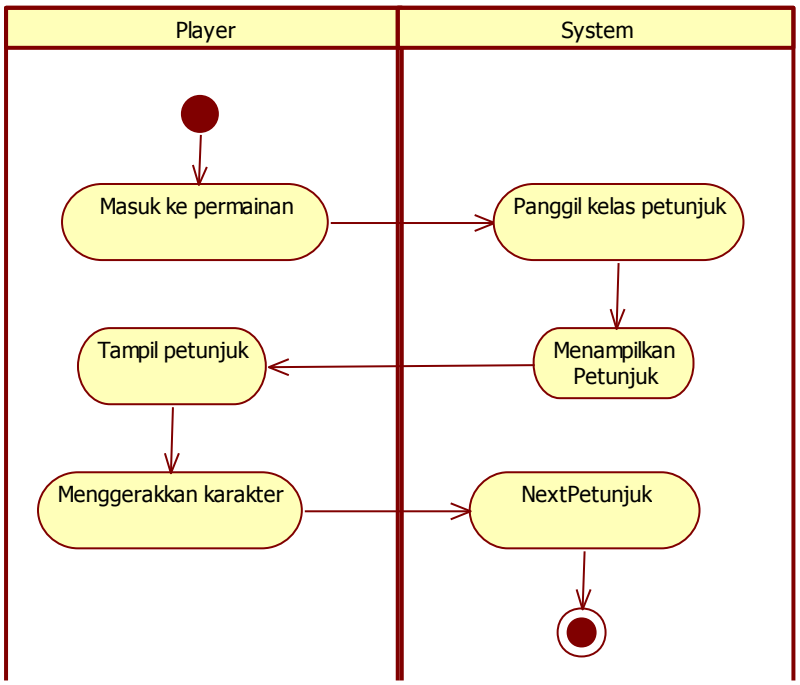
1. Skenario *Use Case* Menampilkan Petunjuk

Tabel 3.6 Skenario *Use Case* Petunjuk

Identifikasi		
Nomor	2	
Nama	Petunjuk	
Tujuan	Proses untuk menampilkan petunjuk cara bermain *game* edukasi sejarah walisongo	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di ruang petunjuk	
Aksi Aktor		Reaksi Sistem
1. Pemain berada di ruang petunjuk		
		2. Panggil kelas petunjuk pada kelas Worlds
		3. Menampilkan petunjuk permainan
4. Tampil petunjuk		
5. Menggerakkan karakter		
		6. Menampilkan petunjuk selanjutya
Kondisi akhir	Menampilkan petunjuk cara bermain	

1.1 Activity Diagram Petunjuk

Activity diagram petunjuk dapat dilihat pada gambar di bawah.



Gambar 3.65 Activity Diagram Petunjuk

2. Skenario *Use Case* Permainan

Tabel 3.7 Skenario *Use Case* permainan

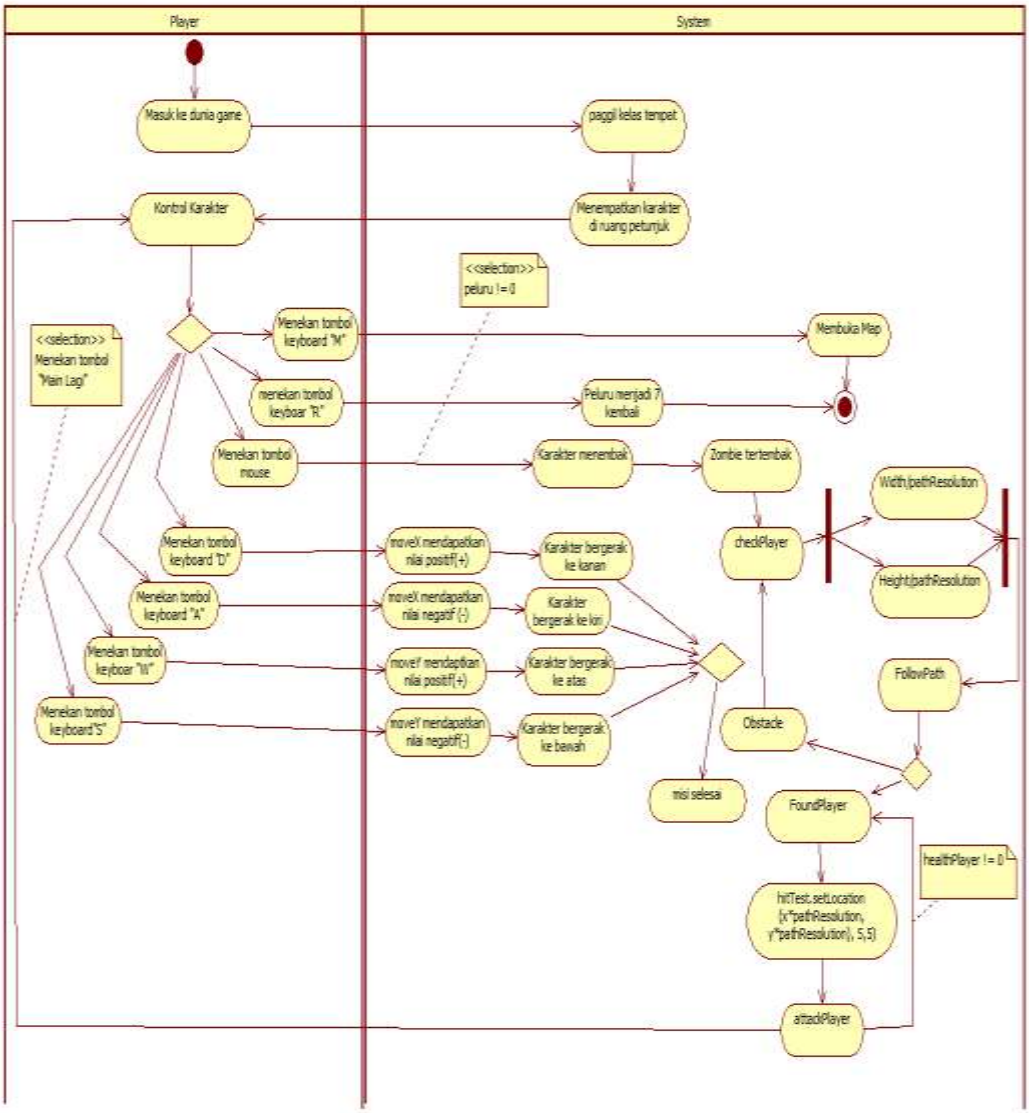
Identifikasi	
Nomor	3
Nama	Permainan
Tujuan	Melakukan permainan berupa *game* bonang dan petualangan
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain melakukan permainan
Aksi Aktor	Reaksi Sistem
1. Pemain masuk ke dunia	

game	
	2. Panggil kelas tempat
	3. Menempatkan karakter pada tempat petunjuk
4. Kontrol karakter	
5. Pemain menekan tombol *keyboard* "A"	
	6. Panggil method gerakkanplayer() yang berarti nilai moveX negatif(-). Sehingga karakter player bergerak ke kiri
7. Pemain menekan tombol *keyboard* "D"	
	8. Panggil method gerakkanplayer() yang berarti nilai moveX positif(+).Sehingga karakter player bergerak ke kanan
9. Pemain menekan tombol *keyboard* "W"	
	10. Panggil method gerakkanplayer() yang berarti nilai moveY positif(+). Sehingga karakter player bergerak ke atas
11. Pemain menekan tombol *keyboard* "S"	
	12. Panggil method gerakkanplayer() yang berarti nilai moveY positif(-). Sehingga karakter player bergerak ke bawah.
13. Pemain menekan tombol *mouse*	
	14. Karakter menembak
	15. Zombie megecek lokasi pemain
	16. Mencari bestNode ke node player
	17. Panggil kelas followPath()

	18. Mengecek apakah node yag dilalui obstacle. Apabila obstacle maka akan dipaggil *method* checkPlayer(), dan apabila pemain ketemu akan memanggil *method* foundPlayer()
	19. Node zombie = -1 node player akan dipanggil *method* attackPlayer.
	20. Selama health player != 0 maka zombie akan mengulang memanggil *method* attackPlayer.
21. Pemain menekan tombol *keyboard* "R"	
	22. Panggil method reload() untuk mengisi peluru menjadi 7 kembali.
23. Pemain menekan tombol *keyboard* "M"	
	24. Panggil kelas map untuk menampilkan posisi pemain berada dan misi yang harus dilakukan.
Kondisi akhir	Pemain melakukan permainan.

2.1 *Activity Diagram Permainan*

Activity diagram pada *game* edukasi sejarah walisongo dapat dilihat dari gambar berikut



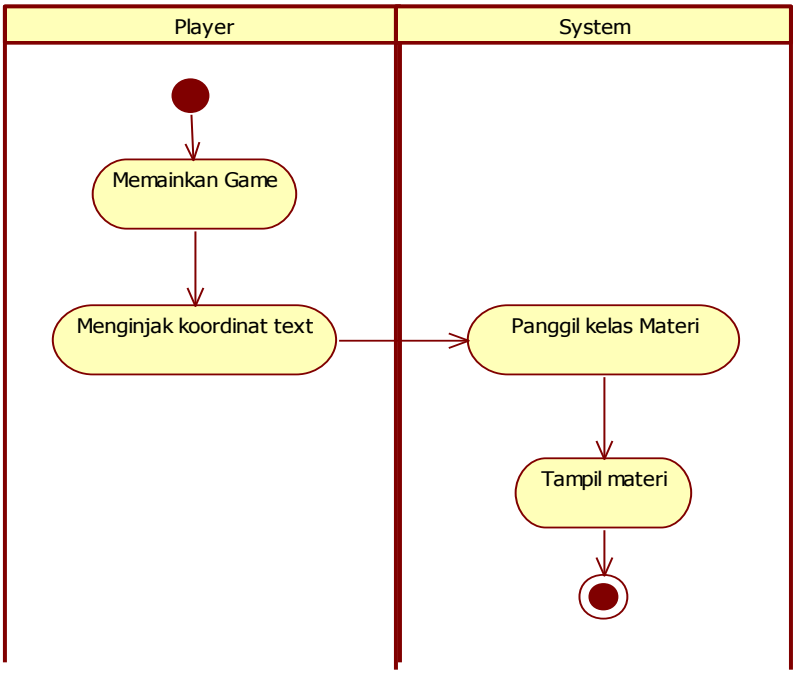
Gambar 3.66 *Activity Diagram* Permainan

3. Skenario *Use Case* Materi

Tabel 3.8 Skenario Materi

Identifikasi		
Nomor	4	
Nama	Materi	
Tujuan	Proses untuk menampilkan materi sejarah walisongo	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di ruangan selain petunjuk	
Aksi Aktor		Reaksi Sistem
1. Pemain berada di menginjak koordinat object text		
		2. Panggil kelas materi
Kondisi akhir	Menampilkan materi-materi dari sejarah walisongo	

3.1 Activity Diagram Menampilkan Materi



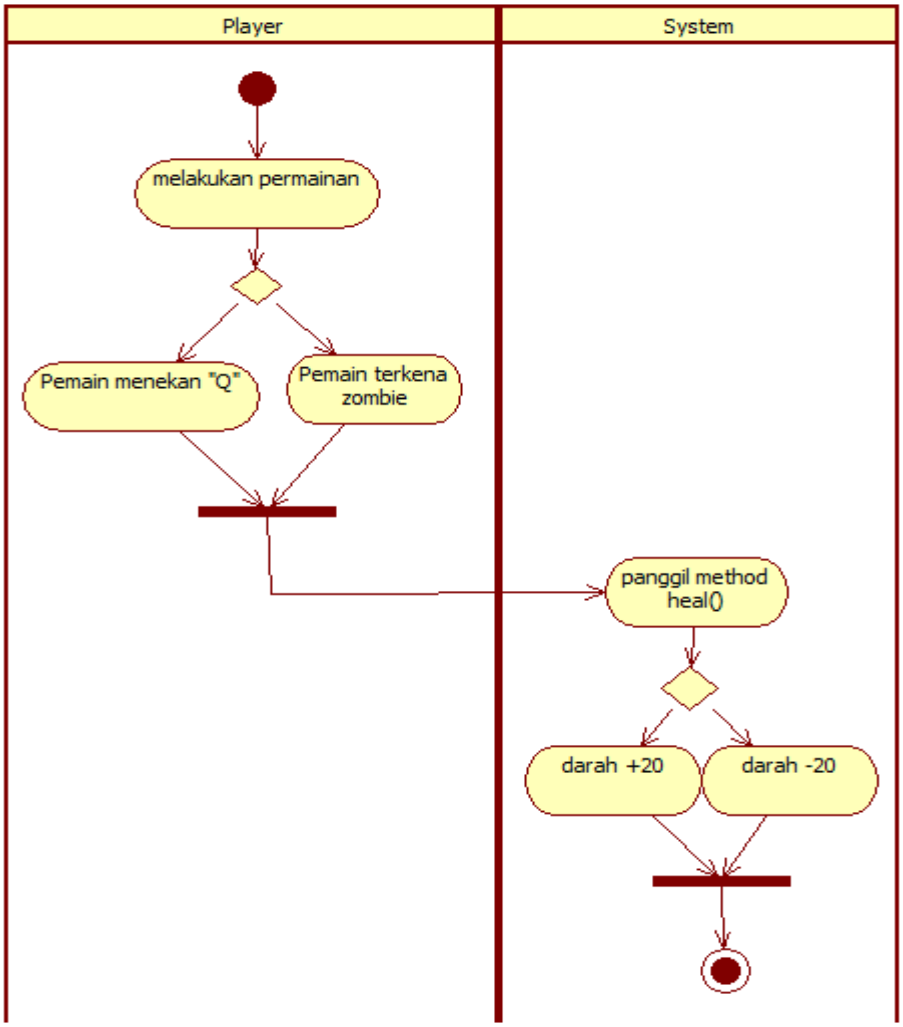
Gambar 3.67 *Activity Diagram* Materi

4. Skenario *UseCase* Menampilkan Darah

Tabel 3.9 Skenario Menampilkan Darah

Identifikasi		
Nomor	4	
Nama	Menampilkan darah	
Tujuan	Proses untuk menampilkan darah	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di ruangan selain petunjuk	
Aksi Aktor		Reaksi Sistem
1. Pemain melakukan permainan		
		2. Memanggil *method* heal()
		3. jika pemain terkena musuh maka darah akan berkurang 20
3.Pemain menekan tombol *keyboard* "Q"		4. panggil *method* heal()
		5. akan menambahkan darah +20
Kondisi akhir	Menampilkan materi-materi dari sejarah walisongo	

4.1 Activity Diagram Menampilkan Darah



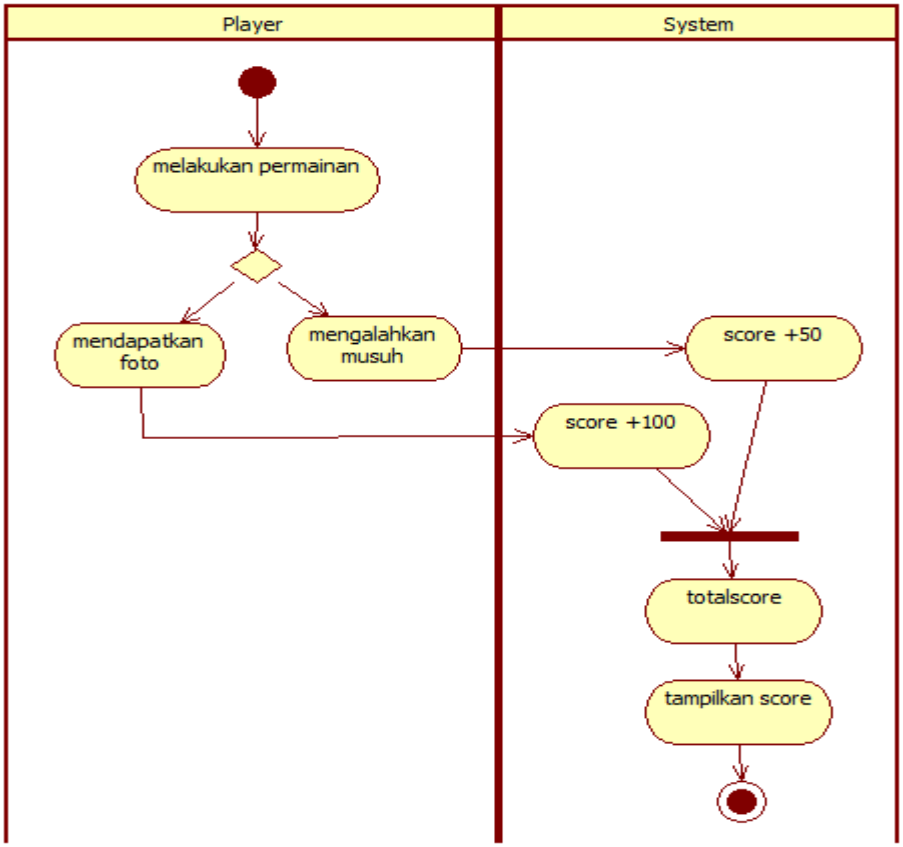
Gambar 3.68 *Activity Diagram* Menampilkan Darah

5. Skenario *UseCase* Menampilkan Score

Tabel 3.10 Skenario Menampilkan Score

Identifikasi		
Nomor	4	
Nama	Menampilkan score	
Tujuan	Proses untuk menampilkan darah	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di ruangan selain petunjuk	
Aksi Aktor		Reaksi Sistem
1. Pemain mendapatkan foto walisongo		
		2. score +100
3. Pemain mengalahkan musuh		
		4. score +50
		5. Panggil *method* totalscore()
		6. Tampilkan score
Kondisi akhir	Menampilkan materi-materi dari sejarah walisongo	

5.1 Activity Diagram Menampilkan Score

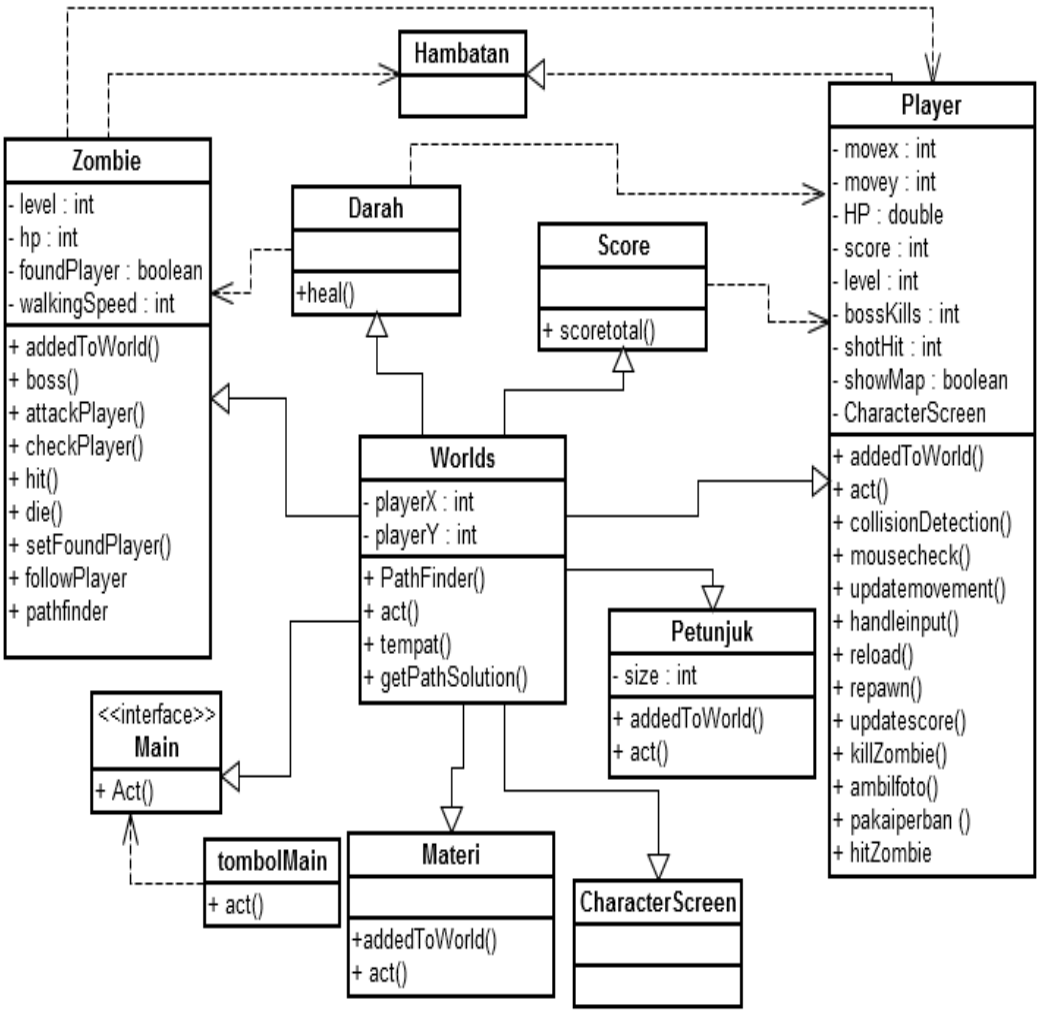


Gambar 3.69 *Activity Diagram* Menampilkan Score

3.3.3 *Class Diagram*

Diagram kelas atau atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem [6].

Berikut ini adalah perancangan struktur sistem yang terdapat pada *game* edukasi sejarah walisongo, yang digambarkan dengan class diagram yang dapat dilihat dibawah.

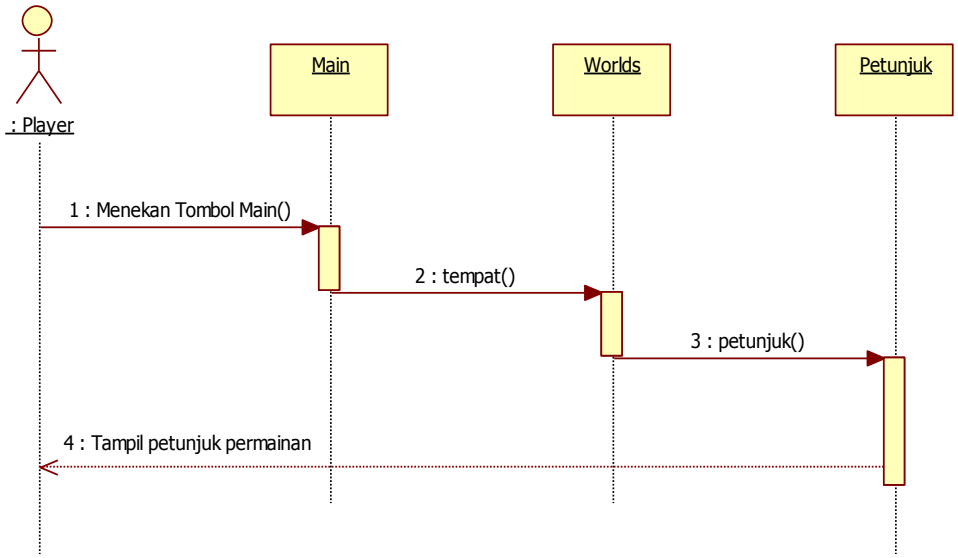


Gambar 3.70 *Class Diagram Game* Walisongo

3.3.4 *Sequence Diagram*

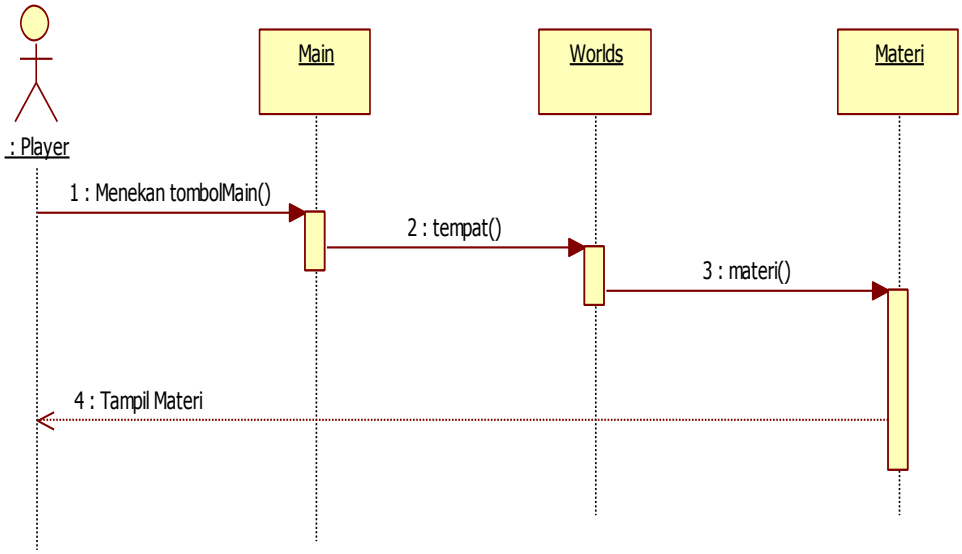
Diagram sequence atau *sequence diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek [6]. Berikut ini adalah gambar sequence diagram pada *game* edukasi sejarah walisongo.

1. Sequence Diagram Petunjuk



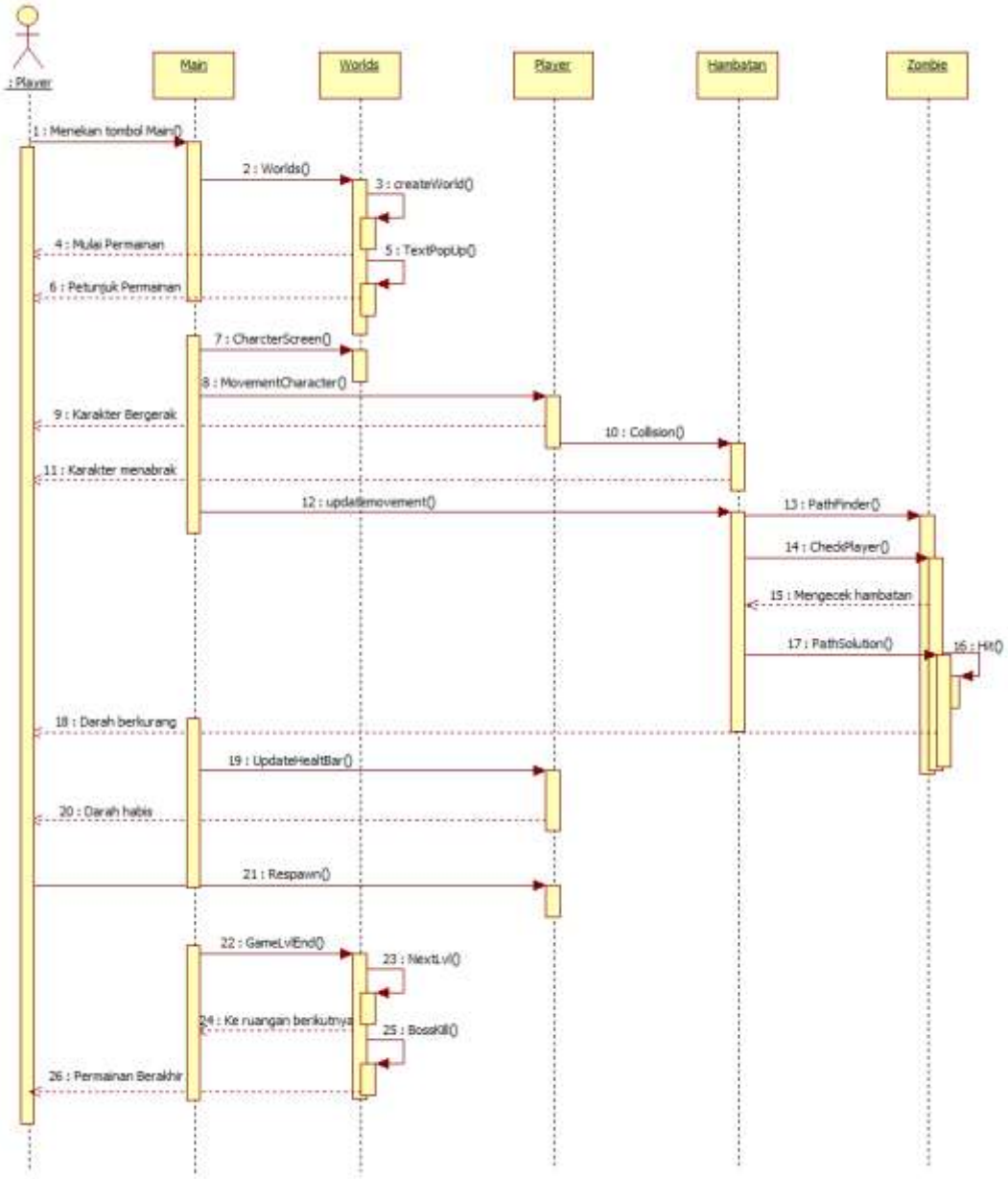
Gambar 3.71 Sequence Diagram Petunjuk

2. Sequence Diagram Materi



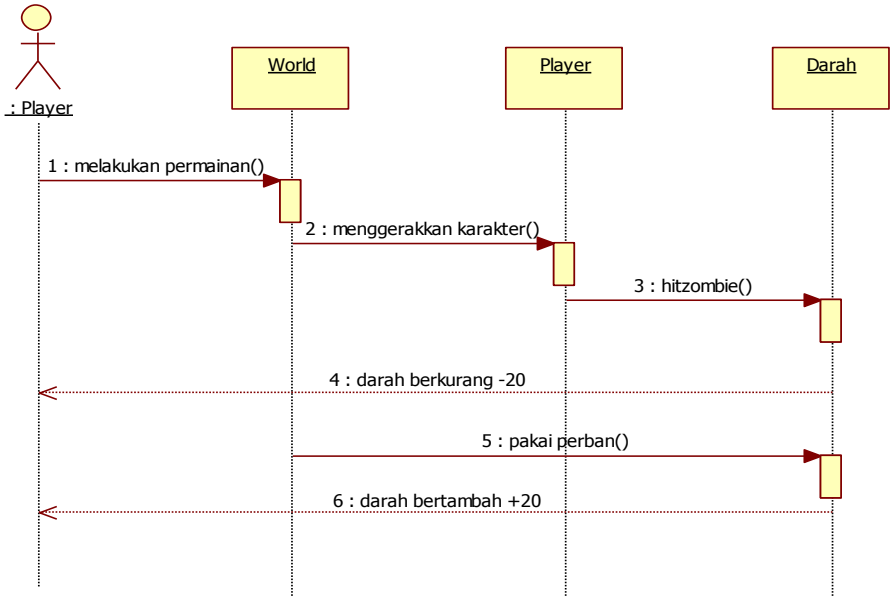
Gambar 3.72 Sequence Diagram Materi

3. Sequence Diagram Permainan



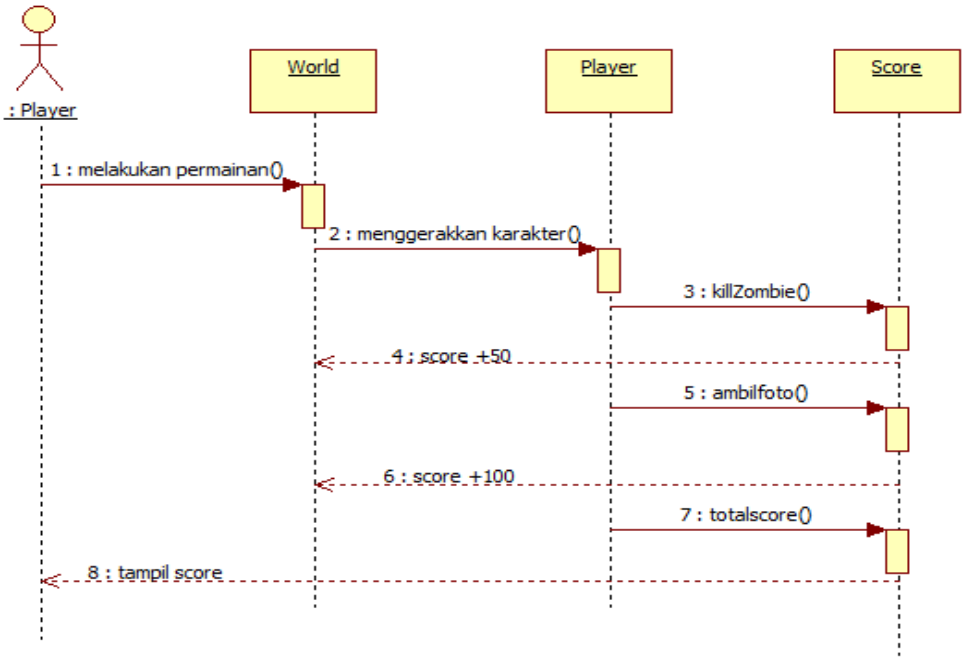
Gambar 3.73 *Sequence Diagram* Permainan

5. Sequence Diagram Menampilkan Darah



Gambar 3.74 Sequence Diagram Menampilkan Darah

6. Sequence Diagram Menampilkan Score



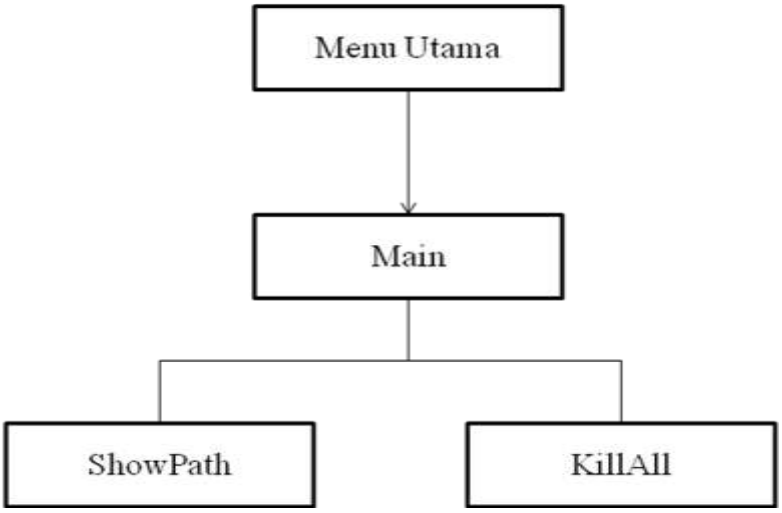
Gambar 3.75 Sequence Diagram Menampilkan Score

3.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak merupakan kegiantan untuk merancang aplikasi yang akan dibangun, tahapannya dimulai dari perancangan struktur menu antarmuka dan jaringan semantik

3.4.1Struktur Menu

Struktur menu bertujuan untuk melihat susunan menu apa saja yang terdapat dalam aplikasi tersebut. Berikut struktur menu dari aplikasi *game* tersebut:



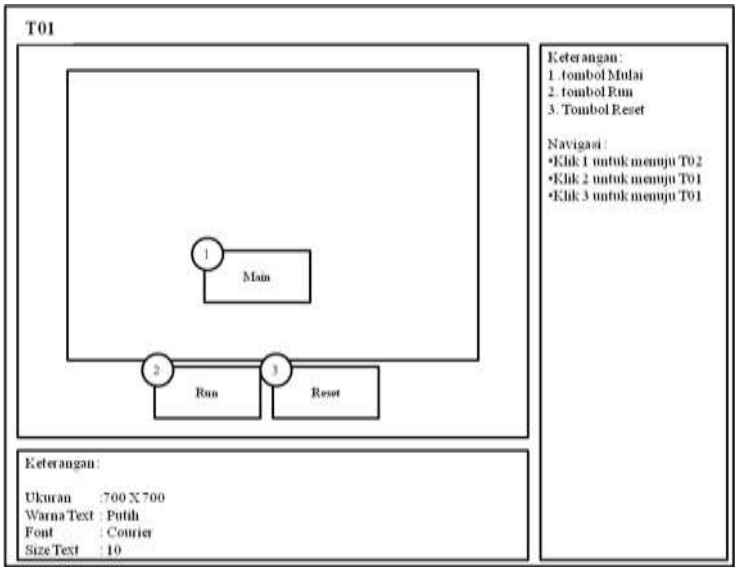
Gambar 3.76 Struktur Menu

3.4.2 Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi. [6]

1. Perancangan Antarmuka Menu

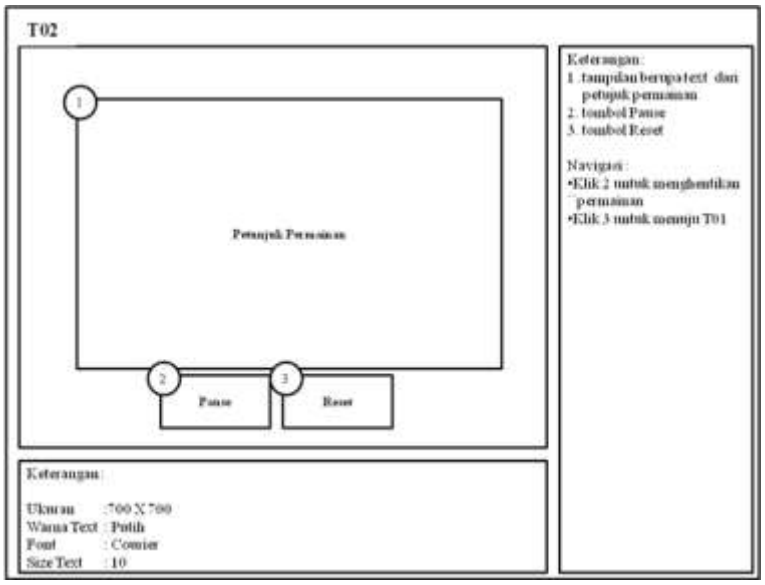
Berikut ini adalah perancangan antarmuka menu yang diberi identitas T01.



Gambar 3.77 Perancangan Antar Muka Menu

2. Perancangan Antarmuka Petunjuk

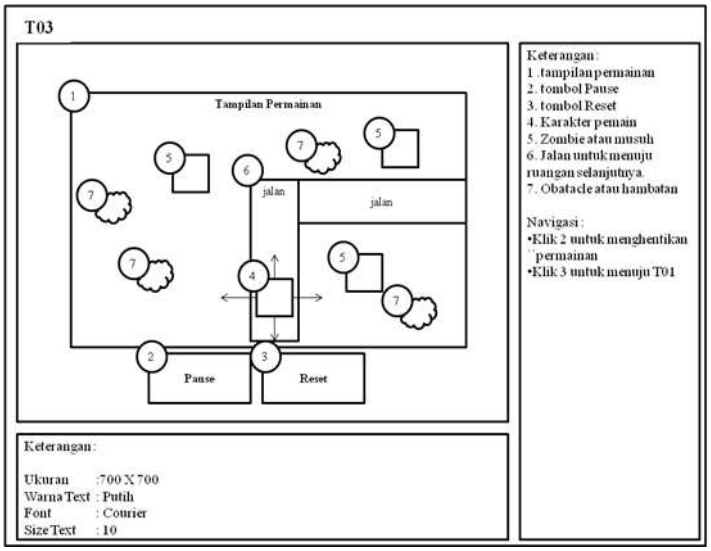
Berikut perancangan antarmuka petunjuk dimana pada bagian ini terdapat petunjuk tentang cara-cara dalam memainkan *game* sejarah walisongo ini. Perancangan antarmuka petunjuk diberi identitas T02.



Gambar 3.78 Perancangan Antar Muka Petunjuk

3. Perancangan Antarmuka Permainan

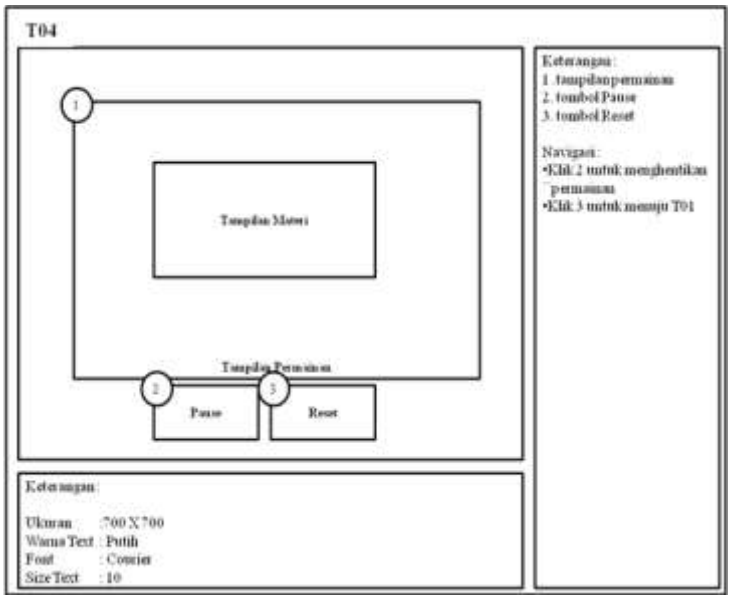
Berikut perancangan antarmuka permainan dimana pada bagian ini terdapat beberapa konten *game* diantaranya, karakter pemain, karakter musuh, hambatan, dan jalan untuk menuju ruangan selanjutnya. Perancangan antarmuka permainan diberi identitas T03.



Gambar 3.79 Perancangan Antar Muka Permainan

4. Perancangan Antarmuka Materi

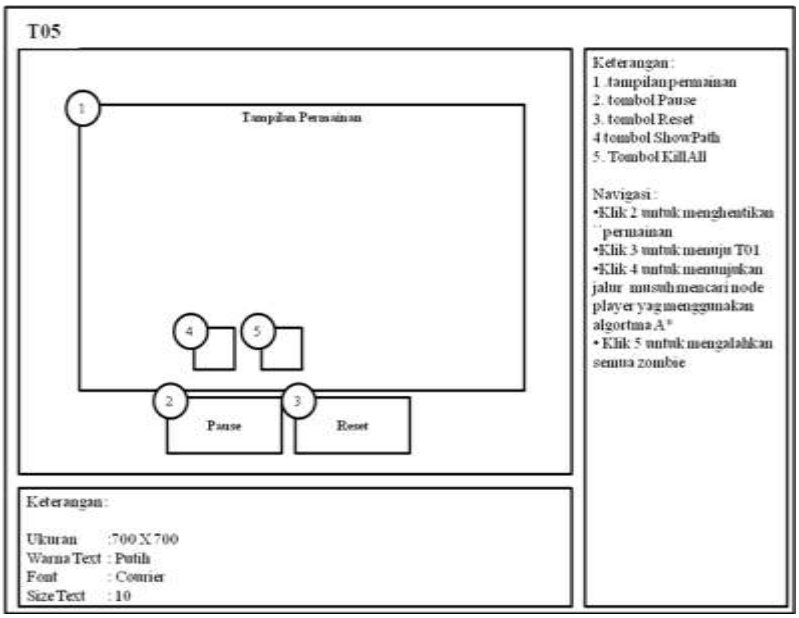
Berikut perancangan antarmuka materi dimana pada bagian ini terdapat penyampaian materi sejarah walisongo. Perancangan antarmuka materi diberi identitas T04.



Gambar 3.80 Perancangan Antar Muka Materi

5. Perancangan Antarmuka Menu Option

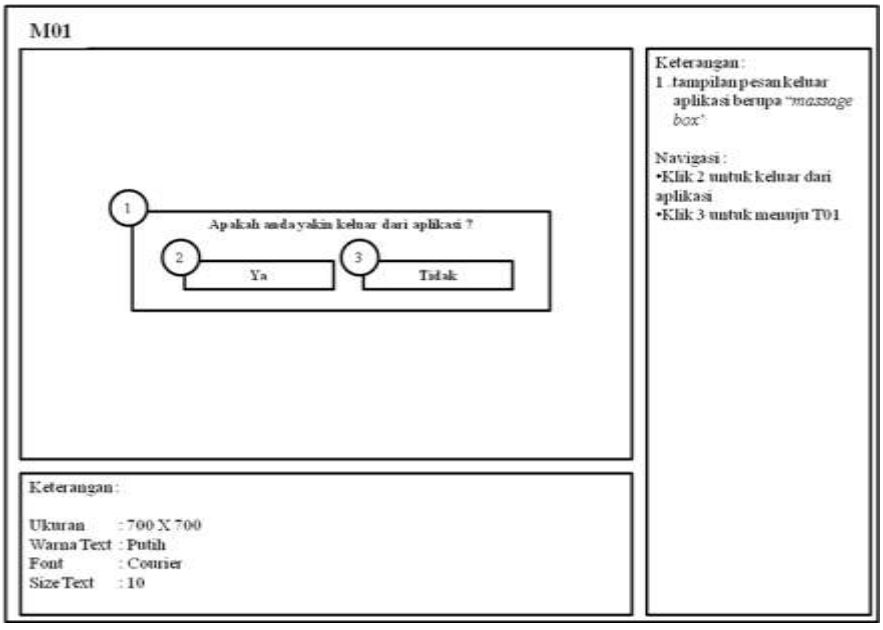
Berikut perancangan antarmuka menu option dimana pada bagian ini terdapat menu lain untuk menampilkan jalur pencarian musuh ke node player yang menggunakan A* (A Star) dan mengalahkan semua zombie pada tampat karakter berada.



Gambar 3.81 Perancangan Antar Muka Menu Option

6. Perancangan antarmuka pesan keluar aplikasi

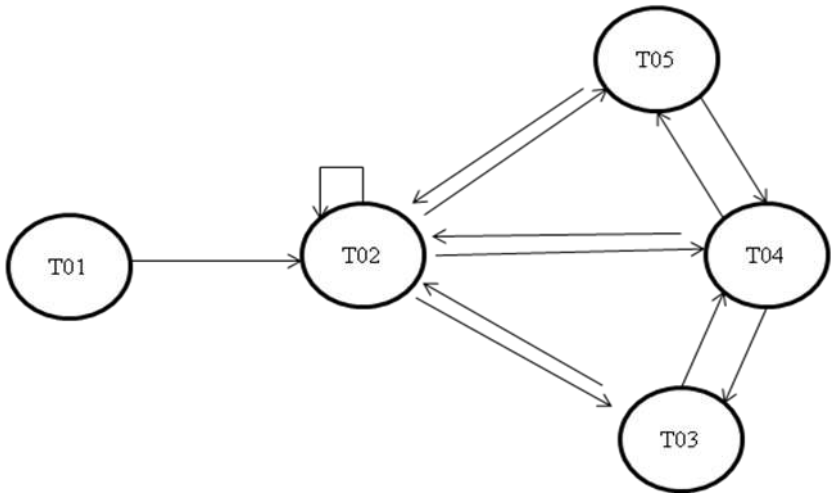
Berikut merupakan perancangan antarmuka pesan keluar aplikasi yang diberi identitas M01.



Gambar 3.82 Perancangan Antar Muka Pesan Keluar Aplikasi

3.1.3 Jaringan Semantik

Jaringan semantik adalah gambaran tentang relasi dari masing-masing halaman yang bisa diakses oleh pengguna pada *game* edukasi sejarah walisongo. Jaringan semantik ini, digunakan sebagai panduan penggunaan menu-menu dalam *game.* Berikut ini jaringan semantik *game* edukasi sejarah walisongo.



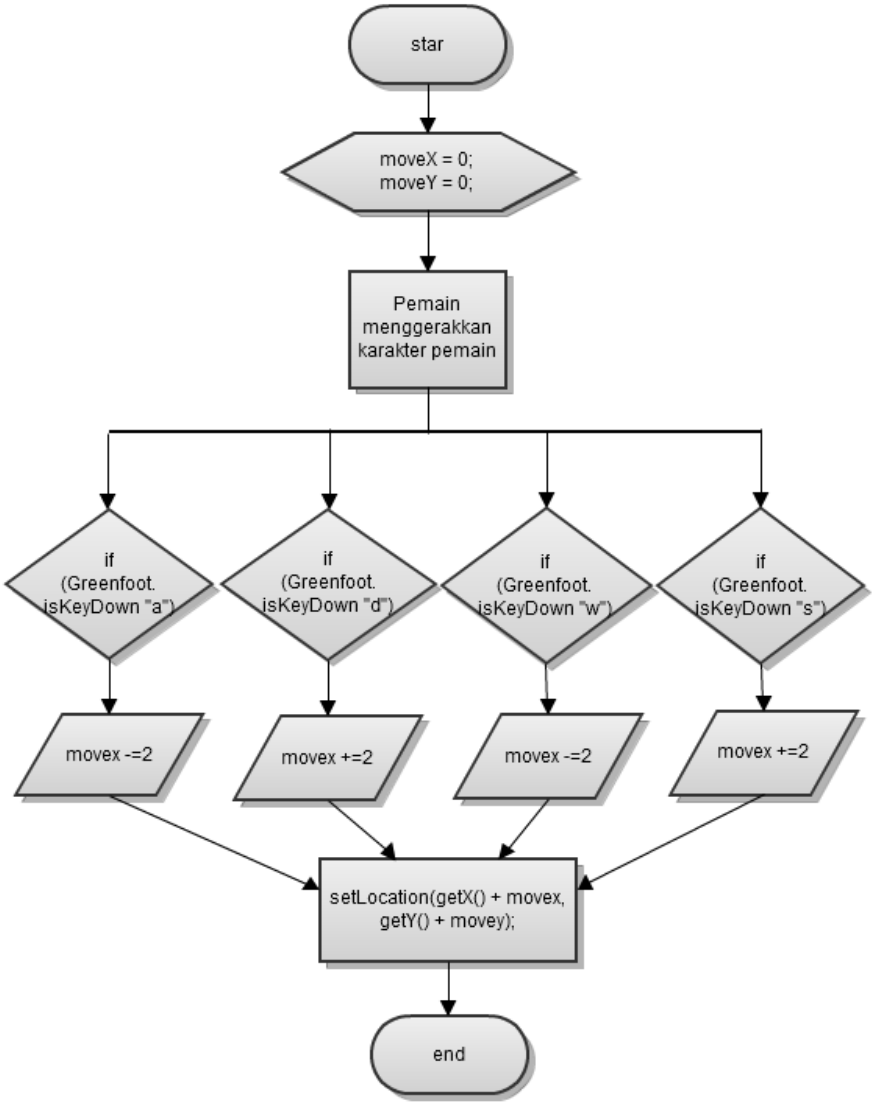
Gambar 3.83 Jaringan Semantik

3.1.4 Perancagan Prosedural

Perancangan prosedural merupakan perancangan yang berfungsi untuk mendeskripsikana *method-method* yang berada di dalam aplikasi. *Method* dapat dipanggil dengan menyertakan variable. Baik hanya satu variable, banyak variable atau bahkan tidak ada sama sekali. Adapun *method-method* yang terdapat dalam *game* edukasi sejarah walisongo adalah sebagai berikut:

1. Perancangan Prosedural Gerak Karakter Player

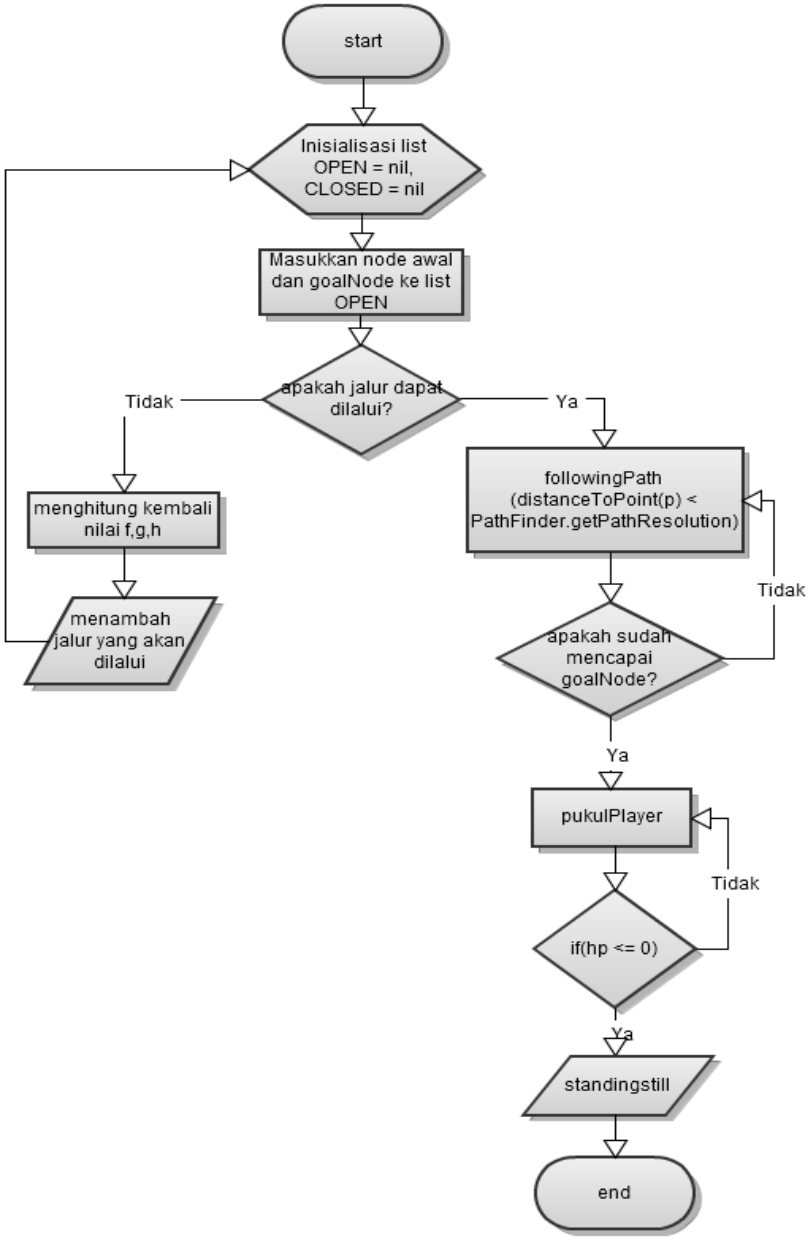
Method ini digunakan untuk menggerakkan karakter pemain. Untuk lebih jelasnya dapat dilihat dari *flowchart* pada gambar sebagai berikut :



Gambar 3.84 *Flowchart Method* Gerak Player

2. Perancangan Prosedural pathfinder

Method ini digunakan untuk pencarian jalur musuh ke tujuan atau lokasi karakter pemain. Untuk lebih jelasnya dapat dilihat dari *flowchart* pada gambar sebagai berikut :



Gambar 3.85 *Flowchart Method* Pathfinder