BAB 3
ANALISIS MASALAH DAN PERANCANGAN

3.1 Analisis

Analisis bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem serta menentukan kebutuhan dari sistem yang dibangun. Analisis merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, sehingga diperoleh solusi. Analisis merupakan tahapan yang paling penting, karena kesalahan dalam tahap ini akan menyebabkan kesalahan di tahap selanjutnya.

Pada bagian analisis sistem ini akan mencakup analisis masalah, analisis *game* sejenis, analisis kebutuhan non-fungsional dan analisis kebutuhan fungsional.

3.1.1 Analisis Masalah

Analisis masalah merupakan bentuk penjabaran mengenai masalah yang ada sebelum dibangunnya *game* Pertempuran Konvoy Sukabumi – Cianjur. Analisis masalah yang ada meliputi hal-hal sebagai berikut:

1. Kurang dikenalnya peristiwa Pertempuran Konvoy Sukabumi – Cianjur bahkan oleh masyarakat Jawa Barat sendiri, khususnya masyarakat Sukabumi dan Cianjur.
2. Referensi mengenai peristiwa Pertempuran Konvoy Sukabumi – Cianjur yang cenderung susah didapat, karena hanya dapat diperoleh dengan membeli langsung di Museum Palagan Bojongkokosan.
3. Referensi yang ada hanya berupa buku.

3.1.2 Analisis *Game* Sejenis

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

Game dengan tipe *gameplay turn-based strategy* merupakan turunan dari *game strategy*, dengan cara memainkannya adalah dengan setiap pemain diberi giliran masing-masing untuk menggerakkan semua pasukannya. Apabila satu pemain telah selesai menggerakkan semua pasukannya atau pemain tersebut telah merasa cukup dalam pergerakkan pasukannya, maka dia akan mengakhiri gilirannya untuk selanjutnya pemain lain menjalankan gilirnnya begitu pula seterusnya. Walaupun *game* pada *genre* ini tidak begitu banyak dibuat, namun *game* ini memiliki daya tarik tersendiri bagi pemain. Beberapa judul populer dari *game* ber-*genre turn-based strategy* ini antara lain adalah *Advance Wars, Ancient Empires, Fire Emblem, Final Fantasy Tactics, Civilization*, dsb.

Game yang akan dianalisis dan menjadi acuan disini ialah *game Advance Wars* dan *game Ancient Empires II*. Secara *gameplay*, *game Advance Wars* dan *Ancient Empires II* serupa dengan *gameplay game* Pertempuran Konvoy Sukabumi – Cianjur yang akan dibangun. Berikut ulasan mengenai *game Advance Wars* dan *game Ancient Empires II.*

3.1.3 Pengenalan *Game Advance Wars*

Advance Wars adalah *game* dengan *genre turn-based strategy* merupakan *game* yang dikembangkan oleh *Intelligent Systems* dan berjalan di *platform GameBoy Advance*. Dalam *game* ini pemain berperan sebagai CO (*Commanding Officer*) yang bertugas mengatur pasukannya untuk memenuhi tujuan yang diberikan di setiap *stage*, yang sebagian besarnya adalah untuk mengalahkan pasukan musuh.

3.1.3.1 *Gameplay Game Advance Wars*

Beberapa aturan main yang terdapat dalam *game Advance Wars* ini adalah sebagai berikut:

1. Pemain berperan sebagai CO (*Commanding Officer*) yang tidak selalu sama dalam setiap *stage*-nya.

2. Pemain diberi gilirannya masing-masing untuk menggerakkan pasukannya. Apabila gilirannya telah selesai, maka pemain lain (atau komputer) akan memulai gilirannya. Begitu pula sebaliknya.
3. Misi utama adalah untuk mengalahkan semua pasukan musuh atau mengambil alih ibukota musuh.
4. Pemain dapat mengambil alih bangunan yang akan memberikan pendapatan yang nantinya bisa untuk membeli pasukan baru di *barrack.*
5. Dalam *map game* terdapat beberapa jenis *terrain* yang dapat memberikan bonus *defense* pada pasukan yang menempati terrain itu.
6. Dalam *game* ini terdapat beberapa CO yang semuanya memiliki bonus tertentu pada pasukan yang dipimpinnya.

3.1.3.2 Tampilan *Game Advance Wars*

Berikut ini merupakan tampilan pada *game Advance Wars*. Tampilan menu utama *game Advance Wars* dapat dilihat pada gambar 3.1 dibawah ini.



Gambar 3.1 Tampilan Menu Utama *Game Advance Wars*

Tampilan permainan *game Advance Wars* dapat dilihat pada gambar 3.2 dibawah ini.



Gambar 3.2 Tampilan Permainan *Game Advance Wars*

3.1.3.3 Hasil Analisis *Game Advance Wars*

Adapun hasil analisis dari *Game Advance Wars* adalah sebagai berikut:

1. *Game* ini merupakan *game* ber-*genre turn-based strategy* yang dapat dimainkan di *console Gameboy Advance* yang relatif kurang populer di Indonesia, sehingga *game* ini kurang dikenal di Indonesia.
2. Menggunakan grafis 2 dimensi.
3. Begitu banyak aturan dan tipe unit dalam *game* ini, sehingga cukup sulit untuk dipahami.
4. Tidak terdapat unsur edukasi dalam cerita yang diangkat.

3.1.4 Pengenalan *Game Ancient Empires II*

Ancient Empires II merupakan sebuah *game mobile* ber-*genre turn-based strategy* yang dikembangkan oleh *Macrospace* (sekarang *Glu*). *Game* ini dapat dijalankan di *platform* J2ME pada ponsel yang mendukungnya. Dalam *game* ini, pemain berperan sebagai King Galamar atau adiknya Valadron yang memiliki tujuan akhir merebut kembali *Crystal of Courage* dari Saeth yang bertujuan untuk menggunakan *Crystal of Courage* untuk mengaktifkan senjata mematikan.

3.1.4.1 *Gameplay Game Ancient Empires II*

Beberapa aturan main yang terdapat dalam *game Ancient Empires II* ini tidak terlalu berbeda dengan *game Advance Wars*. Namun tetap terdapa beberapa perbedaan. Aturan main dalam game ini adalah sebagai berikut:

1. Pemain diberi gilirannya masing-masing untuk menggerakkan pasukannya. Apabila gilirannya telah selesai, maka pemain lain (atau komputer) akan memulai gilirannya. Begitu pula sebaliknya.
2. Misi utama dalam setiap level adalah untuk megalahkan semua musuh dan mangambil alih kastil musuh.
3. Pemain dapat mengambil alih bangunan (rumah) yang akan memberikan pendapatan yang nantinya bisa untuk membeli pasukan baru di kastil.
4. Dalam *map game* terdapat beberapa jenis *terrain* yang dapat memberikan bonus *defense* pada pasukan yang menempati *terrain* itu.

3.1.4.2 Tampilan *Game Ancient Empires II*

Berikut ini merupakan tampilan pada *game Ancient Empires II*. Tampilan menu utama *game Ancient Empires II* dapat dilihat pada gambar 3.3 dibawah ini.



Gambar 3.3 Tampilan Menu Utama Game Ancient Empires II

Tampilan permainan *game Ancient Empires II* dapat dilihat pada gambar 3.4 dibawah ini.



Gambar 3.4 Tampilan Permainan Game Ancient Empires II

3.1.4.3 Hasil Analisis *Game Ancient Empires II*

Hasil analisis dari *game Ancient Empires II* adalah sebagai berikut:

1. *Game* ini hanya dapat berjalan di ponsel yang mendukung platform J2ME.
2. Menggunakan grafis 2 dimensi.
3. Cerita yang disampaikan cukup menarik, namun menggunakan bahasa Inggris, sehingga pemain sulit memahaminya.
4. Cerita yang disampaikan tidak mengandung unsur edukasi.

3.1.5 Pengenalan *Game* Pertempuran Konvoy Sukabumi – Cianjur

Game yang akan dibangun berjudul Pertempuran Konvoy Sukabumi – Cianjur dan ber-*genre turn-based strategy*. Game ini dibangun untuk menyampaikan kisah pertempuran antara pasukan TKR/TRI melawan konvoy sekutu yang terjadi di Sukabumi dan Cianjur pada tahun 1945 dan 1946.

Berikut ini merupakan fitur-fitur yang ada pada game ini:

1. *Game* dapat dimainkan satu pemain / *single player*.
2. Mengangkat kisah pertempuran konvoy Sukabumi – Cianjur.
3. Menggunakan grafis 2D (2 Dimensi)
4. Jenis/*genre game* adalah *turn-based strategy.*
5. Memiliki fitur *Highscore*.

3.1.5.1 *Storyline*

Game pertempuran konvoy Sukabumi – Cianjur mengisahkan mengenai pertempuran yang terjadi di daerah Sukabumi dan Cianjur pada tanggal 9-12 Desember 1945 dan pada tanggal 10-14 Maret 1946 antara pasukan TKR/TRI dengan konvoy yang ditunggangi NICA, pasukan sekutu yang akan berangkat ke Bandung yang dicurigai ingin mengumpulkan kekuatan di Bandung untuk merebut kembali kekuasaan Belanda dari Indonesia.

Berikut ini adalah *storyline* untuk setiap *stage* dari *game* pertempuran konvoy Sukabumi – Cianjur:

1. *Stage* 1: Tanggal 9 Desember 1945, konvoy sekutu yang bertujuan ke bandung memasuki wilayah Cigombong. Pasukan TKR yang sudah bersiaga dari pos cigombong sampai bojongkokosan, melakukan taktik *hit and run* dalam mengahadapi pasukan sekutu.
2. *Stage* 2: Masih pada tanggal 9 Desember 1945, pasukan sekutu yang panik karena serangan TKR, melanjutkan perjalanan kembali diserang pasukan TKR di daerah Parungkuda sampai ke wilyah Kota Sukabumi.
3. *Stage* 3: 11 Desember 1945, Batalyon III TKR yang ditempatkan di daerah Gekbrong sampai Ciranjang, menghadang sekutu dari arah Bandung yang menuju Kota Sukabumi dengan tujuan membantu tentara sekutu yang sebelumnya diserang di Bojongkokosan sampai Kota Sukabumi.
4. *Stage* 4: 10 – 11 Maret 1946 konvoy pasukan sekutu yang terdiri dari Batalyon Patiala dari arah Bogor diserang di daerah Cikukulu sampai ke Kota Sukabumi oleh pasukan TKR yang sekarang telah diganti namanya menjari TRI (Tentara Republik Indonesia).
5. *Stage* 5: Pada tanggal 12 maret 1946, datang balabantuan berupa Tank Sherman dari Squadron 13 Lancer dengan dikawal pasukan Grenadier dari arah Bogor, namun kembali diserang di daerah Cikukulu.
6. *Stage* 6: 12 Maret 1946, bersamaan dengan diberangkatkannya Tank Sherman dari Squadron 13 Lancer dan pasukan pengawal Grenadier dari Bogor, dari Bandung dikirimkan juga balabantuan Batalyon 5/6

Rajputana Rifles (Rajrif) ke Sukabumi. Namun perjalanan mereka diserang mulai dari Cisokan Ciranjang sampai dengan Gekbrong. Lalu pada tanggal 13 Maret 1946, sekutu kembali mengirimkan balabantuan dengan dipimpin oleh Brigadier (Jenderal) N.D. Wingrove yang kembali diserang di wilayah Cisokan Ciranjang.

7. *Stage* 7: 13 Maret 1946, pasukan sekutu yang terdiri dari Batalyon Patiala, Batalyon Rajputana Rifles (Rajrif), Squadron 13 Lancer dan Grenadier diserang di pusat kota Sukabumi oleh TRI pada malam hari sekitar pukul 20.00 dengan menggunakan taktik Kirikumi. Sebelum penyerangan, penerangan di seluruh Kota Sukabumi dipadamkan oleh GBO (PLN), agar taktik Kirikumi berjalan dengan baik.

3.1.5.2 *Gameplay*

Misi utama pemain dalam *game* ini berbeda di setiap *stage* nya disesuaikan dengan sejarah pertempuran konvoy Sukabumi – Cianjur. Tipe atau *genre* permainan ini adalah *turn-based strategy* dimana pemain dan komputer menggerakkan pasukannya dalam gilirannya masing-masing. Berikut ini adalah *gameplay* dari *game* Pertempuran Konvoy Sukabumi – Cianjur:

1. Pemain diberi gilirannya masing-masing untuk menggerakkan pasukannya. Apabila gilirannya telah selesai, maka komputer akan memulai gilirannya. Begitu pula sebaliknya.
2. Misi utama dalam setiap level berbeda-beda dan telah ditentukan sebelumnya.
3. Dalam *map game* terdapat beberapa jenis *terrain* yang dapat memberikan bonus *defense* (pertahanan) ataupun bonus *attack* (serangan) pada pasukan yang menempati *terrain* itu.
4. *Terrain* akan memberikan penalti pergerakan pada pasukan yang melewatinya. Beberapa *terrain* (gunung dan hutan) tidak dapat dilalui oleh kendaraan perang sekutu.
5. Permainan akan berakhir jika misi dari *stage* yang dimainkan dapat diselesaikan oleh pemain.

6. Di awal permainan, pemain hanya dapat memilih tutorial, tutorial ini harus dimainkan untuk memberikan pemahaman mengenai aturan main dalam *game*.
7. Setelah memenangkan *stage*, *game* akan meng-*unlock stage* selanjutnya dan mengecek *highscore*. Jika *score* pemain lebih besar dari *highscore* sebelumnya, maka pemain diberikan tampilan untuk meng-*input* namanya kedalam daftar *highscore*.
8. Pemain kalah apabila seluruh pasukannya dikalahkan.

3.1.5.3 *Scoring*

Scoring atau pemberian nilai dilakukan untuk menumbuhkan keinginan pemain untuk memainkan kembali *game* untuk mendapatkan *score* yang lebih baik untuk ditampilkan pada daftar *highscore*.

Score dihitung dengan rumus sebagai berikut:

Score pemain = (jumlah pasukan*1000) + ((100 - giliran) * 100) + pertanyaan bonus

Keterangan:

- Jumlah pasukan = jumlah pasukan pemain yang tersisa di akhir permainan
- Giliran = jumlah giliran (*turn*) yang dibutuhkan untuk memenangkan permainan
- Pertanyaan bonus = nilai dari pertanyaan bonus yang diberikan pada pemain di akhir permainan dikali jumlah pertanyaan bonus yang diberikan. Jumlah pertanyaan bonus tidak selalu sama untuk setiap *stage*.

Sebagai contoh, misalnya pemain menyelesaikan *stage* 1 dalam 8 giliran (*turn*) dengan sisa pasukannya berjumlah 15 pasukan. Kemudian pemain menjawab 2 pertanyaan bonus dengan benar yang masing-masing nilainya 5000 poin, maka:

Score pemain = (jumlah pasukan*1000) + ((100 - giliran) * 100) + pertanyaan bonus

Score pemain = (15*1000) + ((100 - 8) * 100) + (2 * 5000)

Score pemain = (15000) + ((92) * 100) + (10000)

Score pemain = (15000) + (9200) + (10000)

Score pemain = 34200

Sehingga, *score* akhir yang didapat oleh pemain untuk *stage* 1 adalah 34200.

3.2 Analisis Algoritma

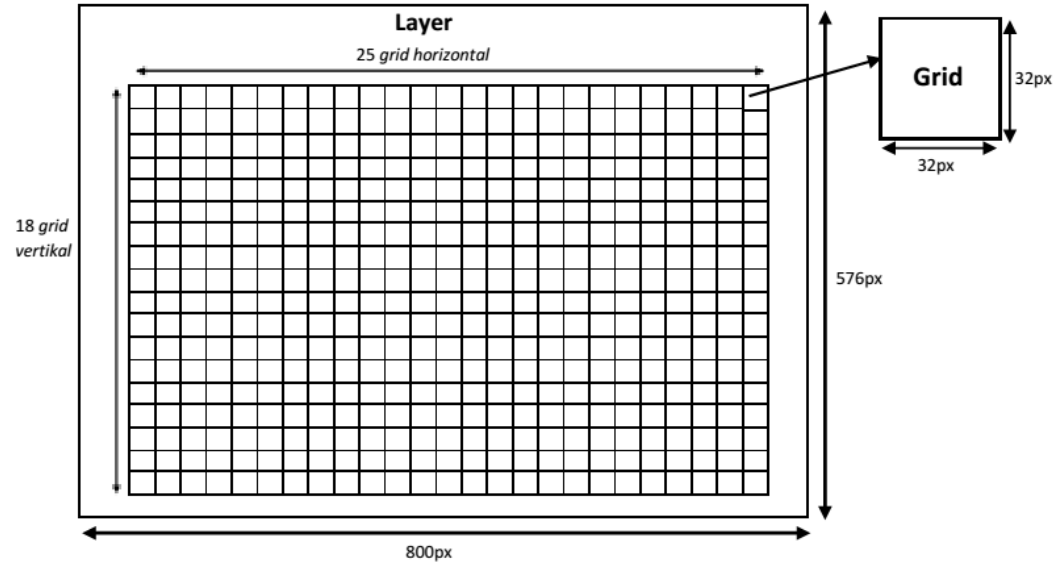
Algoritma yang akan diterapkan pada pencarian jalur terpendek pada *game* Pertempuran Konvoy Sukabumi – Cianjur adalah algoritma A* (*A Star*). Algoritma ini akan digunakan untuk pencarian jalur terdekat ke lokasi musuh.

Sedangkan untuk pengambilan keputusan akan diterapkan algoritma *Finite State Machine* (FSM). Algoritma ini digunakan untuk menentukan aksi apa saja yang bisa dilakukan NPC.

3.2.1 Analisis Algoritma A* *(A Star)*

Analisis Algoritma A* *(A Star)* akan diterapkan pada analisis kasus pencarian jalur terpendek menuju lokasi musuh. Sebelum menghitung menggunakan algoritma A* *(A Star)* alangkah baiknya bila kita ketahui terlebih dahulu nilai *cost* yang terdapat pada jalur *game* itu sendiri.

Layer game Pertempuran Konvoy Sukabumi – Cianjur dibuat dengan skala *pixel* yang berukuran lebar 800 pixel dan tinggi 576 *pixel*. Dari satu *layer* tersebut dibagi-bagi menjadi beberapa *grid* (suatu kotak-kotak kecil) didalamnya dengan dengan ukuran lebar dan tinggi 32x32 *pixel*, setelah dibagi, diperoleh 25 *grid* horizontal dan 18 *grid* vertical. Sehingga total keseluruhan jumlah *grid* dalam satu *layer* adalah 450. Lihat Gambar 3.5 untuk lebih jelasnya.



Gambar 3.5 Keterangan ukuran *layer* dan *grid* (*node*) pada *game*

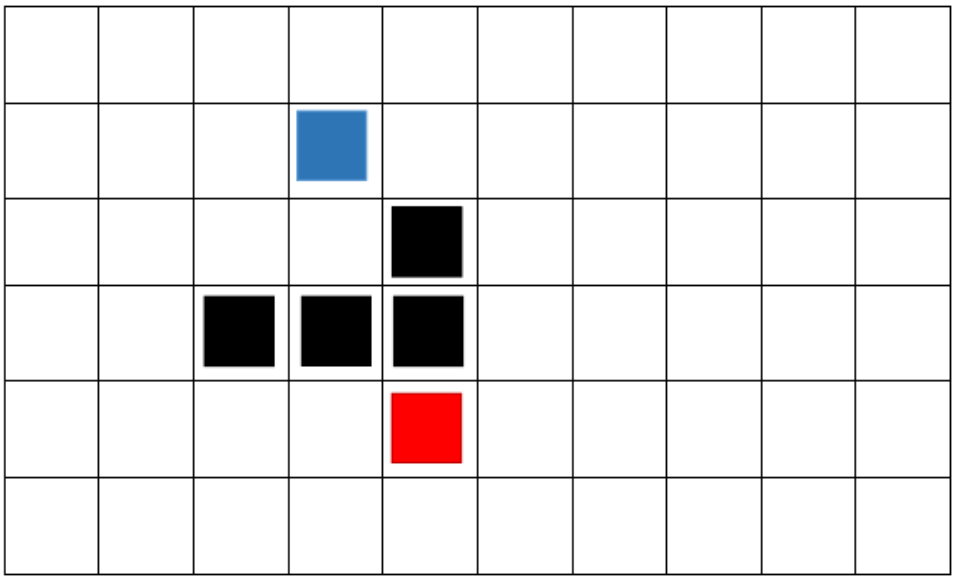
Untuk memudahkan perhitungan, maka ada beberapa nilai yang akan diubah / disederhanakan. Pertama, untuk memudahkan penghitungan, hanya akan diambil *grid* sebanyak 10 *grid horizontal* dan 6 *grid vertical*, sehingga total *grid* adalah 60 *grid*.

Kedua, nilai *node (grid)* yang tadinya 32x32 *pixel* diubah menjadi 1x1, yang mengibaratkan jarak *node* 1 ke *node* yang lain adalah 1, jadi setiap *node* memiliki *cost* yang sama yaitu 1. Lihat gambar 3.6.

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

Gambar 3.6 Nilai *g(n)* pada sebuah *node horizontal* dan *vertical*

Untuk lebih memperjelas cara kerja algoritma A*, maka akan dibuat simulasi cara penghitungan jalur terpendek menggunakan algoritma A* pada *game* Pertempuran Konvoy Sukabumi – Cianjur. Lihat Gambar 3.7

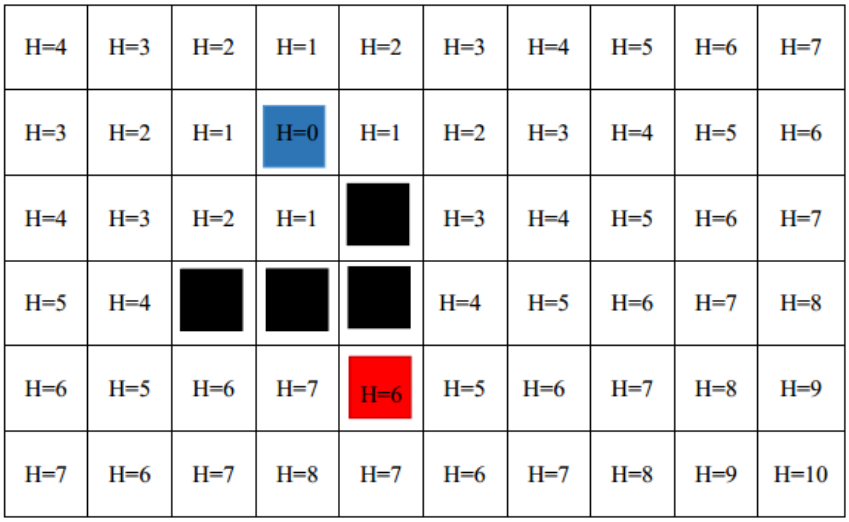


Gambar 3.7 Contoh kasus algoritma A* *pathfinding*

Keterangan:

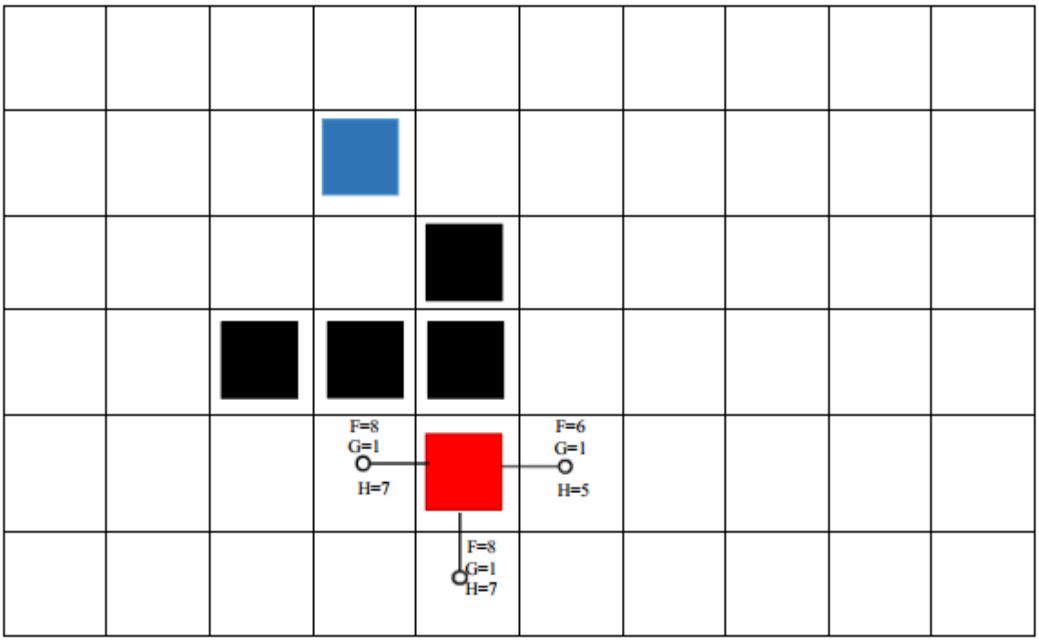
- Kotak merah adalah *node* awal (AI Musuh)
- Kotak hitam adalah *node* halangan yang tidak bisa dilewati
- Kotak biru adalah *node* tujuan (pasukan pemain yang akan diserang)

Nilai *h(n)* didapat dari jarak perkiraan dai *node* merah (AI musuh) ke *node* biru (pasukan pemain), setiap jarak perkiraanya benilai kelipatan 1. Dalam *game* ini, pasukan hanya dapat bergerak secara *horizontal* dan *vertical*, pergerakan *diagonal* tidak bisa dilakukan, sehigga tidak ada perhitungan *heuristic diagonal*. Berikut nilai *heuristic* yang didapat.



Gambar 3.8 Nilai *heuristic* pada kasus pencarian jalur terpendek

Di bawah ini merupakan langkah-langkah pencarian jalur terpendek menggunakan algoritma A* *(A Star)*. Langkah pertama dapat dilihat di Gambar 3.9.



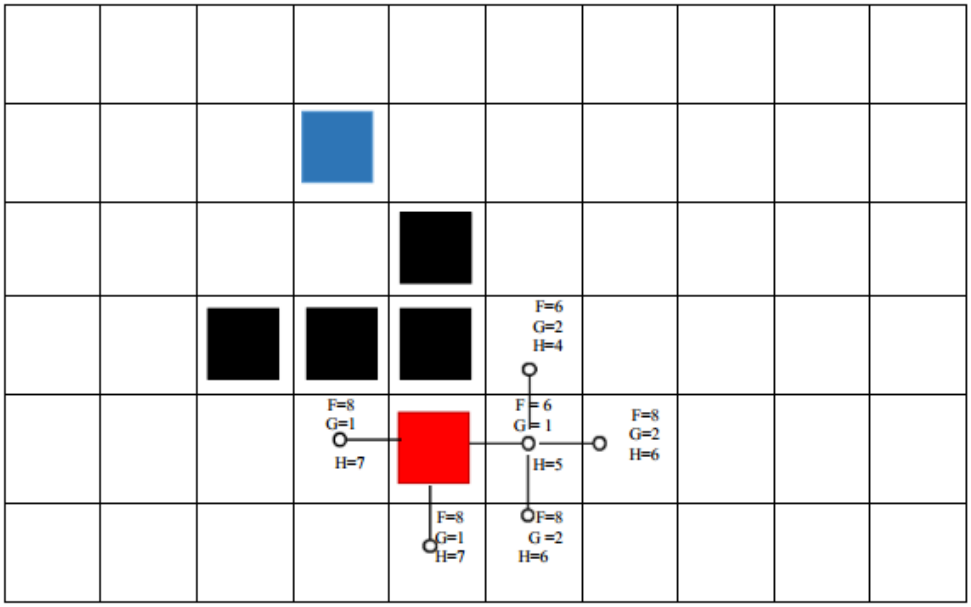
Gambar 3.9 Pencarian jalur terpendek dengan algoritma A* (1)

Dari *node* merah (pasukan musuh) akan melakukan pencarian secara *horizontal* dan *vertical* menuju *node* biru (pasukan pemain). Mula-mula, pencarian dilakukan ke arah:

- *Vertical* bawah : $g = 1$, $h = 7$, dan $f = g + h = 8$
- *Horizontal* kiri : $g = 1$, $h = 7$, dan $f = g + h = 8$
  kanan : $g = 1$, $h = 5$, dan $f = g + h = 6$

Dari perhitungan, didapat nilai *cost* terkecil adalah ke *node horizontal* ke kanan, maka kita akan mengambill jalur *horizontal* ke kanan.

Kemudian, kita masuk ke langkah kedua, yang dapat dilihat di Gambar 3.10.



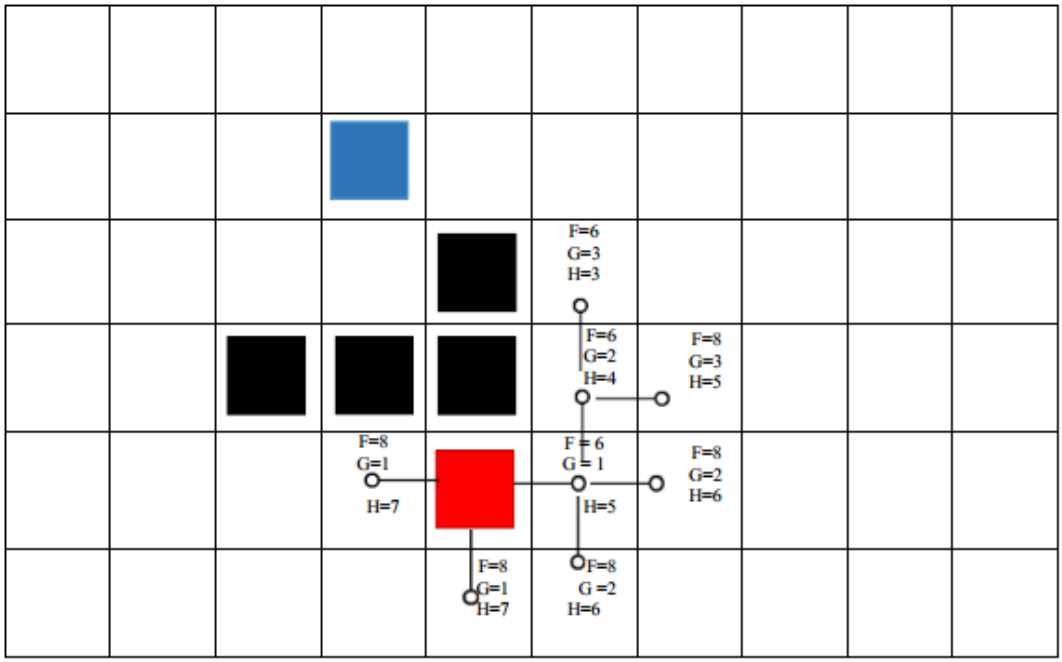
Gambar 3.10 pencarian jalur terpendek dengan algoritma A* (2)

Karena sebelumnya telah dipilih *node horizontal* ke kanan, maka sekarang *node* ini dijadikan sebagai *starting point*. perhitungan lalu dimulai dari *node* ini, ke arah:

- *Horizontal* kanan : $g = 2$, $h = 6$, dan $f = g + h = 8$
- *Vertical* atas : $g = 2$, $h = 4$, dan $f = g + h = 6$

  bawah : $g = 2$, $h = 6$, dan $f = g + h = 8$

Nilai *g* = 2, didapat dari nilai *g* sebelumnya yaitu 1 ditambah *cost* ke *node* selanjutnya yaitu 1. Kemudian kita melanjutkan ke langkah ke tiga, yang dapat dilihat pada Gambar 3.11.

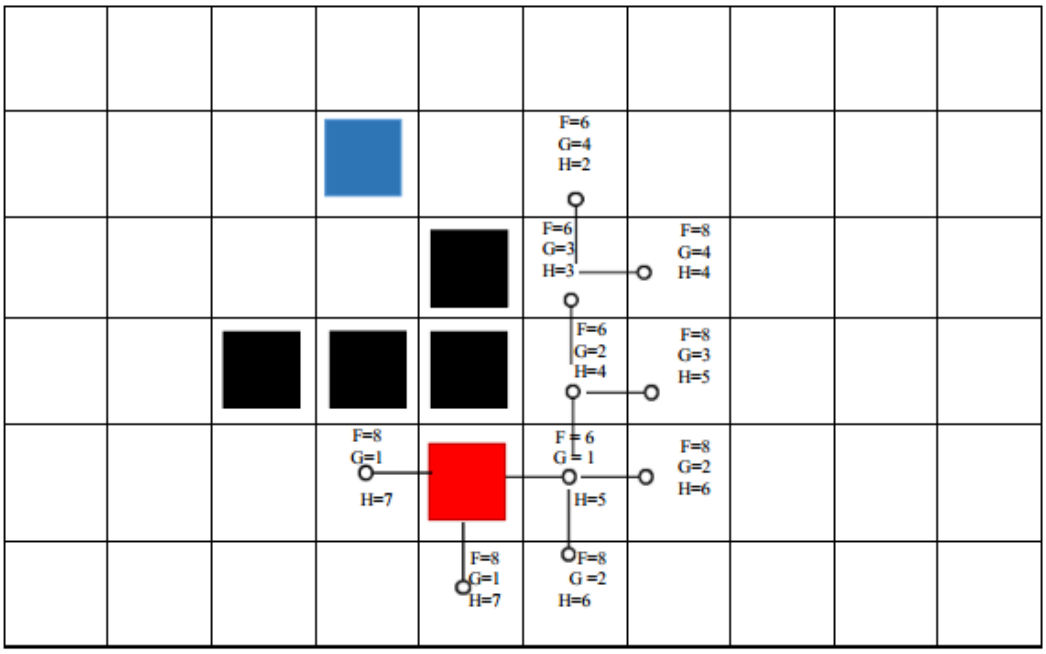


Gambar 3.11 Pencarian jalur terpendek dengan algoritma A* (3)

Karena *node* dengan nilai *f* = 6 (*vertical* atas) adalah *cost* yang paling murah, maka kini pencarian dimulai dari *node* ini. Ke arah:

- *Horizontal* kanan : *g* = 3, *h* = 5, dan *f* = *g* + *h* = 8
- *Vertical* atas : *g* = 3, *h* = 3, dan *f* = *g* + *h* = 6

Nilai *g* = 3, didapat dari nilai *g* sebelumnya yaitu 2 ditambah *cost* ke *node* selanjutnya yaitu 1. Kemudian kita melanjutkan ke langkah ke empat, yang dapat dilihat pada gambar 3.12.

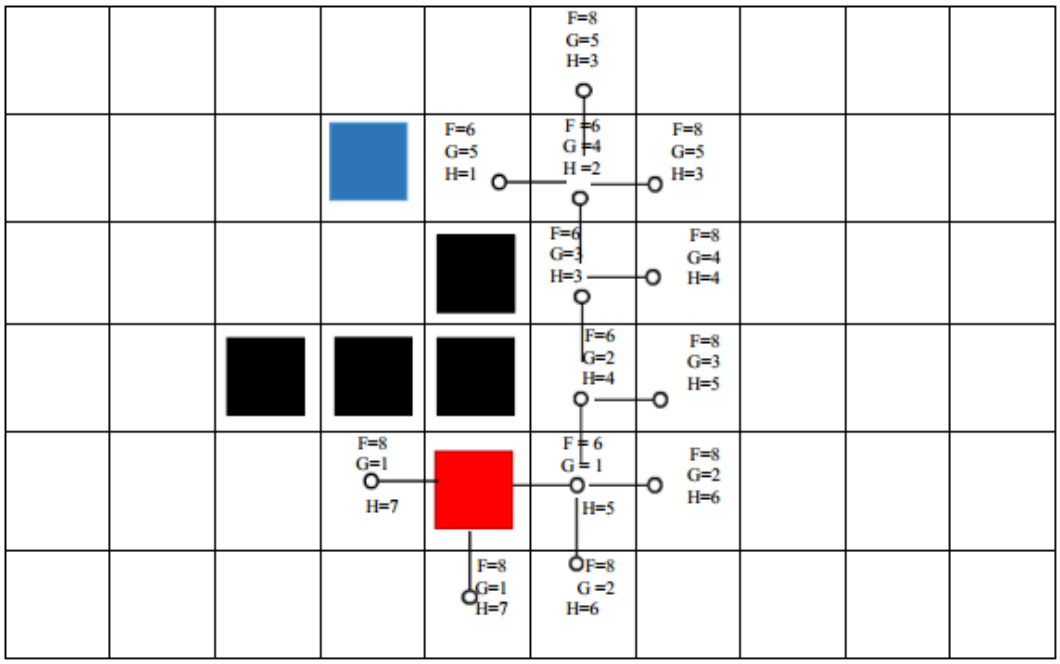


Gambar 3.12 Pencarian jalur terpendek dengan algoritma A* (4)

Karena *node* dengan nilai f = 6 (*vertical* atas) adalah *cost* yang paling murah, maka kini pencarian dimulai dari *node* ini. Ke arah:

- *Horizontal* kanan : $g = 4$, $h = 4$, dan $f = g + h = 8$
- *Vertical* atas : $g = 4$, $h = 2$, dan $f = g + h = 6$

Nilai g = 4, didapat dari nilai g sebelumnya yaitu 3 ditambah *cost* ke *node* selanjutnya yaitu 1. Kemudian kita melanjutkan ke langkah ke lima, yang dapat dilihat pada gambar 3.13.

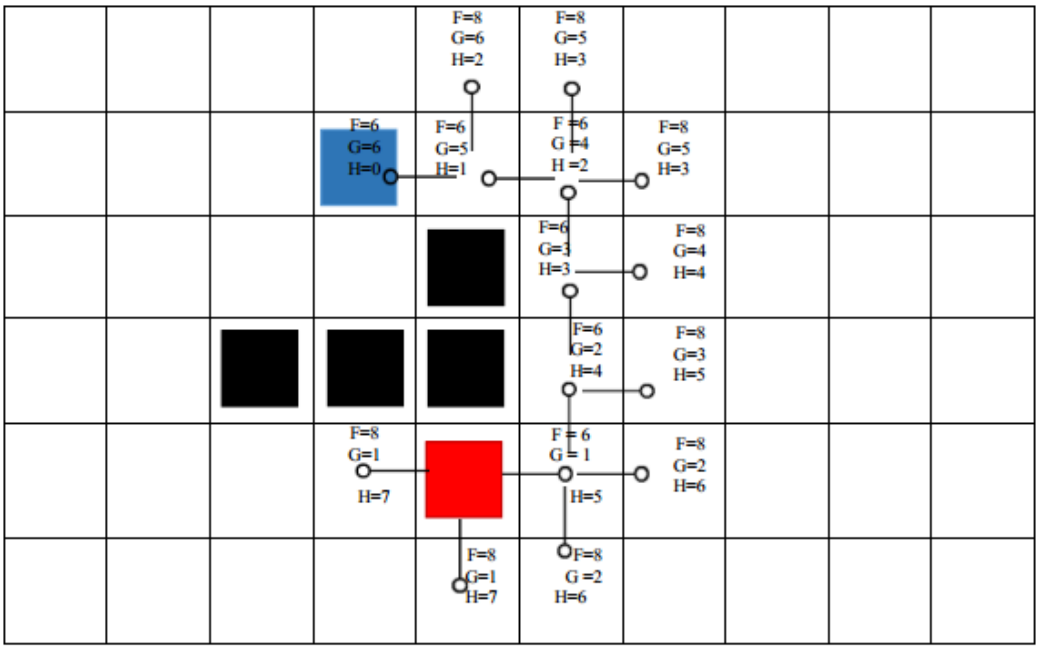


Gambar 3.13 Pencarian jalur terpendek dengan algoritma A* (5)

Karena *node* dengan nilai $f = 6$ (*horizontal* kiri) adalah *cost* yang paling murah, maka kini pencarian dimulai dari *node* ini. Ke arah:

- *Horizontal* kiri : $g = 5$, $h = 1$, dan $f = g + h = 6$
  kanan : $g = 5$, $h = 3$, dan $f = g + h = 8$
- *Vertical* atas : $g = 5$, $h = 3$, dan $f = g + h = 8$

Nilai $g = 5$, didapat dari nilai g sebelumnya yaitu 4 ditambah *cost* ke *node* selanjutnya yaitu 1. Kemudian kita melanjutkan ke langkah ke enam, yang dapat dilihat pada gambar 3.14.



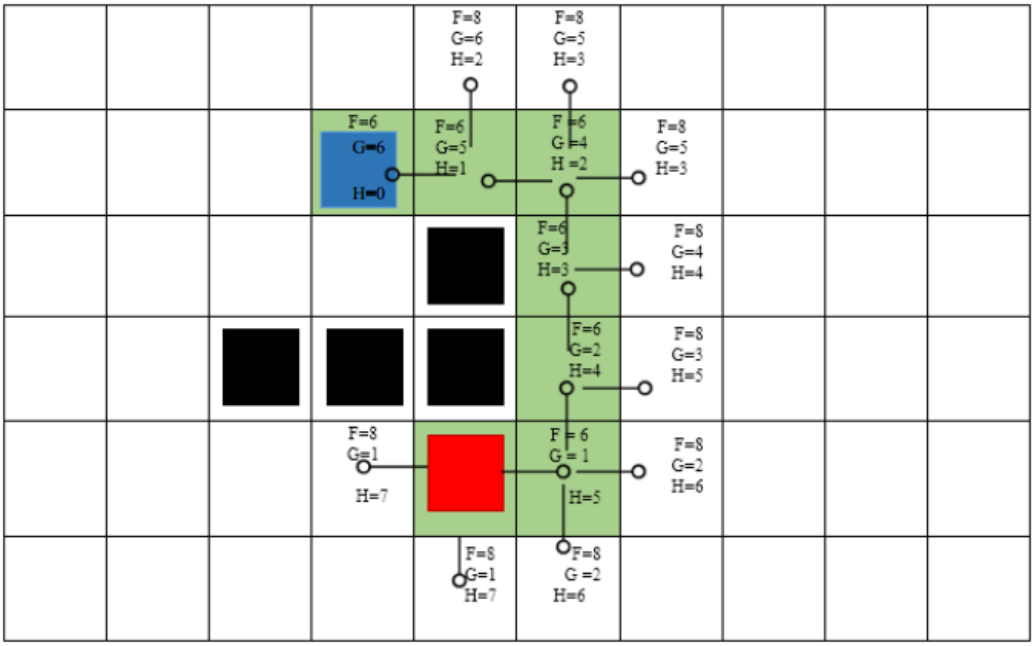
Gambar 3.14 Pencarian jalur terpendek dengan algoritma A* (6)

Karena *node* dengan nilai $f = 6$ (*horizontal* kiri) adalah *cost* yang paling murah, maka kini pencarian dimulai dari *node* ini. Ke arah:

- *Horizontal* kiri : $g = 6$, $h = 0$, dan $f = g + h = 6$
- *Vertical* atas : $g = 6$, $h = 2$, dan $f = g + h = 8$

Nilai $g = 6$, didapat dari nilai g sebelumnya yaitu 5 ditambah *cost* ke *node* selanjutnya yaitu 1.

Dari perhitungan diatas, maka hasil jalur terpendek dengan algoritma A* *(A Star)* terdapat pada *node horizontal* kiri dengan nilai $f = 6$ yang dapat dilihat pada gambar 3.15.



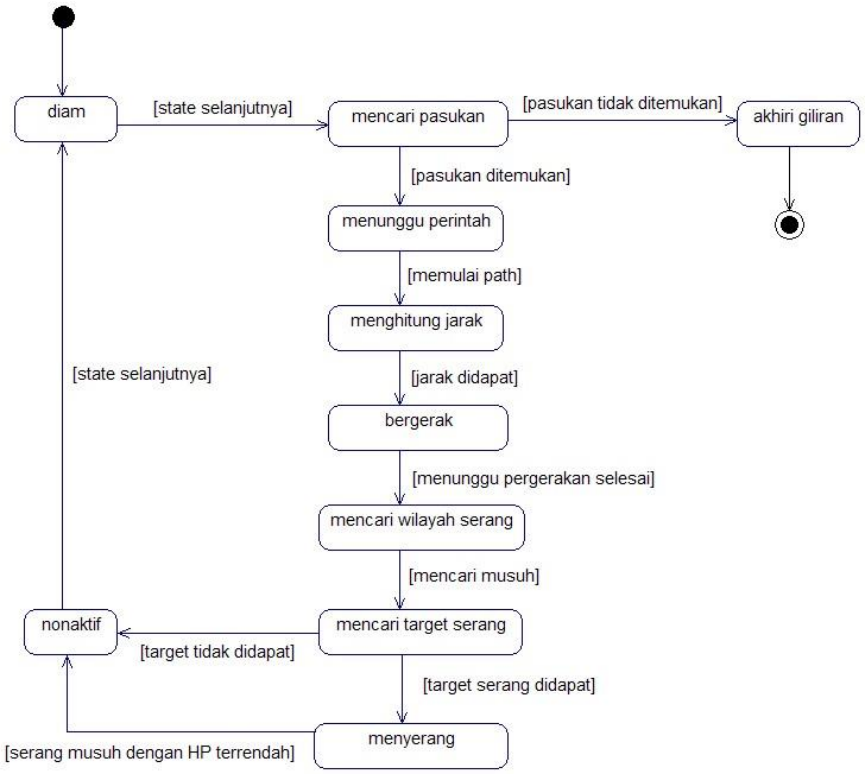
Gambar 3.15 Hasil pencarian jalur terpendek dengan algoritma A* *(A Star)*

Hasil pencarian jalur terpendek menggunakan algoritma A* *(A Star)* dari *node* berwarna merah (pasukan musuh) menuju *node* berwarna biru (pasukan pemain) yang didapatkan adalah sebagai berikut:

1. Dari *node* berwarna merah (pasukan musuh) mengambil jalur ke arah *node horizontal* kanan dengan nilai *f* = 6.
2. Dari *node horizontal* kanan bernilai *f* = 6 mengambil jalur ke arah *node vertical* atas dengan nilai *f* = 6.
3. Dari *node vertical* atas bernilai *f* = 6 mengambil jalur ke arah *node vertical* atas dengan nilai *f* = 6.
4. Dari *node vertical atas* bernilai *f* = 6 mengambil jalur ke arah *node vertical* atas dengan nilai *f* = 6.
5. Dari *node vertical* atas bernilai *f* = 6 mengambil jalur ke arah *node horizontal* kiri dengan nilai *f* = 6.
6. Dari *node horizontal* kiri bernilai *f* = 6 mengambil jalur ke arah *node horizontal* kiri dan sampailah ke *node* berwarna biru (pasukan pemain) dengan nilai *f* = 6.

3.2.2 Analisis Algoritma *Finite State Machine* (FSM)

Algoritma *Finite State Machine* (FSM) dalam *game* Pertempuran Konvoy Sukabumi – Cianjur, digunakan dalam proses pengambilan keputusan dan aksi yang dapat dilakukan NPC. Secara *default*, setiap awal memulai permainan, NPC akan berada dalam *state* "diam" atau *"idle"*, kemudian NPC akan menjalankan *state* selanjutnya yang telah sebelumnya didefinisikan. Untuk mempermudah dalam memahami *finite state machine* yang diimplementasikan dalam *game* Pertempuran Konvoy Sukabumi – Cianjur, lihat Gambar 3.16.



Gambar 3.16 *State Machine* pada *game* Pertempuran Konvoy Sukabumi – Cianjur

3.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan aplikasi *game*. Spesifikasi ini juga meliputi elemen atau komponen-komponen yang dibutuhkan untuk *game* yang akan dibangun sampai dengan *game* tersebut diimplementasikan. Analisis

kebutuhan ini juga menentukan spesifikasi masukan yang dibutuhkan *game*, keluaran yang akan dihasilkan *game* dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Analisis kebutuhan *game* yang diperlukan antara lain:

1. Analisis Kebutuhan Perangkat Lunak
2. Analisis Kebutuhan Perangkat Keras
3. Analisis Pengguna

3.3.1 Analisis Kebutuhan Perangkat Lunak

Dalam membangun sebuah *game* diperlukan perangkat lunak – perangkat lunak yang dapat mendukung proses pembuatannya. Adapun perangkat lunak yang dibutuhkan antara lain:

a. Sistem Operasi Windows
b. Game Maker 8
c. Adobe Photoshop CS3

3.3.2 Analisis kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam membangun aplikasi *game* Pertempuran Konvoy Sukabumi – Cianjur adalah sebagai berikut:

a. Prosesor Intel i3 2.20 GHz
b. RAM 4GB
c. VGA nVidia GT520M 1GB
d. Hard Disk 500GB
e. Mouse, Keyboard, Speaker dan Monitor

Sedangkan perangkat keras yang dibutuhkan (minimum) untuk menjalankan *game* Pertempuran Konvoy Sukabumi – Cianjur ini adalah sebagai berikut:

a. Prosesor 1GHz
b. RAM 1GB
c. VGA 256MB 32bit

d. Hard Disk 100MB
e. Mouse, Keyboard, Speaker, Monitor

3.3.3 Analisis Pengguna

Berdasarkan hasil penelitian, spesifikasi minimal bagi pengguna yang dapat menggunakan aplikasi ini adalah sebagai berikut:

a. Peran : *User*
b. Usia : 13 tahun atau lebih
c. Tingkat Keterampilan : Dapat mengoperasikan komputer
d. Pengalaman : Pernah memainkan *game* PC

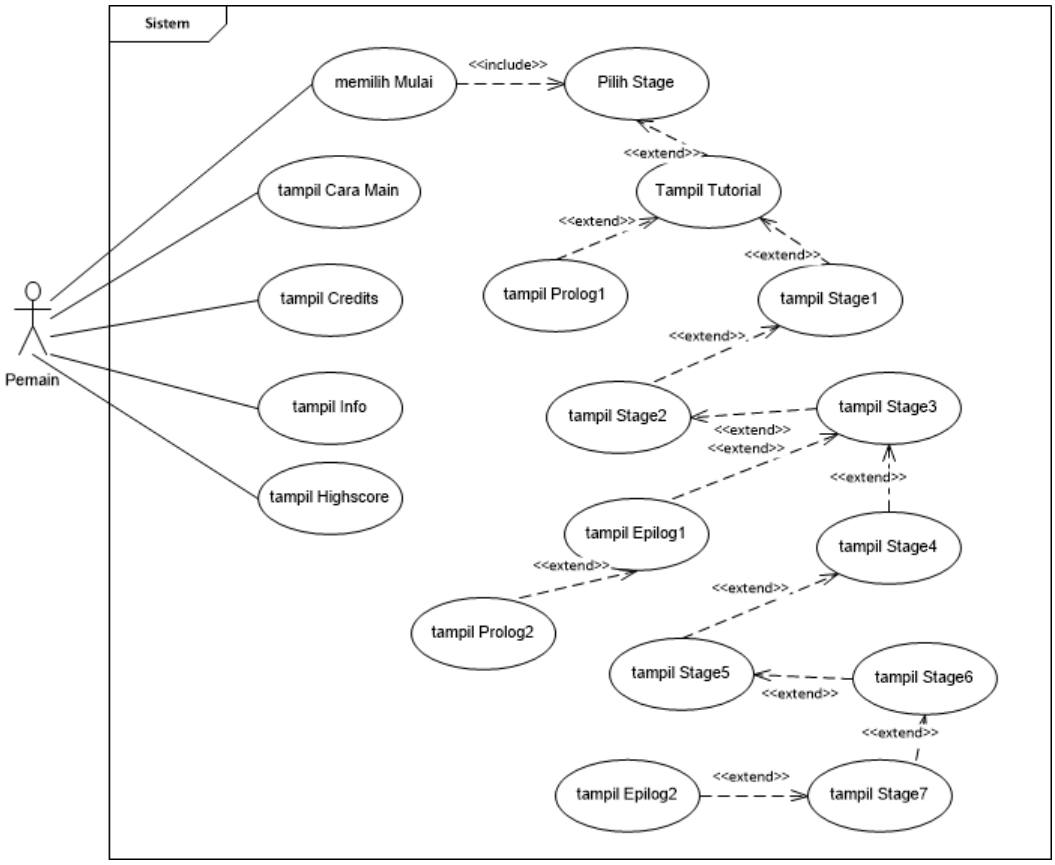
3.4 Analisis Kebutuhan Fungsional

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

Kebutuhan Fungsional dari aplikasi *game* Pertempuran Konvoy Sukabumi – Cianjur yang akan dibuat terdiri dari penggambaran soesifikasi kebutuhan menggunakan *tool* pemodelan berorientasi objek yaitu *Unified Modeling Language* (UML).

3.4.1 Diagram *Use Case*

Use case diagram merupakan gambaran fungsional dari interaksi antara aktor dengan sistem. Sebuah diagram *use case* menggambarkan keterhubungan antara aktor dengan aksi yang dapat dilakukannya terhadap sistem.



Gambar 3.17 *Use Case Diagram game* Pertempuran Konvoy Sukabumi – Cianjur

3.4.2 Definisi *Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan proses yang terdapat pada setiap *Use Case*. *Use case* yang ada pada game yang akan dibangun akan dijelaskan sebagi berikut:

Tabel 3.1 Definisi *Use Case*

No	*Use Case*	Deskripsi
1	Memilih Mulai	Proses memilih memulai *game*
2	Tampil Cara Main	Proses menampilkan cara memainkan *game*
3	Tampil *Credits*	Proses menampilkan *credits*
4	Tampil Info	Proses menampilkan info mengenai *game* dan

No	*Use Case*	Deskripsi
		referensi yang dipakai dalam pembuatan *game*
5	Tampil *Highscore*	Proses menampilkan daftar *highscore*
6	Pilih *Stage*	Proses memilih *stage game* yang ingin dimainkan
7	Tampil Tutorial	Proses menampilkan Tutorial *game*
8	Tampil Prolog1	Proses menampilkan Prolog 1 *game*
9	Tampil Stage1	Proses menampilkan Stage1 *game*
10	Tampil Stage2	Proses menampilkan Stage2 *game*
11	Tampil Stage3	Proses menampilkan Stage3 *game*
12	Tampil Epilog1	Proses menampilkan Epilog1 *game*
13	Tampil Prolog2	Proses menampilkan Prolog2 *game*
14	Tampil Stage4	Proses menampilkan Stage4 *game*
15	Tampil Stage5	Proses menampilkan Stage5 *game*
16	Tampil Stage6	Proses menampilkan Stage6 *game*
17	Tampil Stage7	Proses menampilkan Stage7 *game*
18	Tampil Epilog2	Proses menampilkan Epilog2 *game*

3.4.3 Skenario *Use Case*

Skenario *Use Case* menggambarkan alur penggunaan sistem dimana setiap scenario digambarkan dari sudut pandang actor, seseorang atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara.

Tabel 3.2 Skenario *Use Case* Memilih Mulai

Identifikasi	
Nomor	1
Nama	Memilih Mulai
Tujuan	Mulai permainan
Deskripsi	Proses untuk memulai permainan

Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di menu utama	
Aksi Aktor		**Reaksi Sistem**
1. Memilih menu Mulai		
		2. Menampilkan menu pilih *stage* yang ingin dimainkan
3. Menekan tombol Menu Utama		
		4. Kembali ke menu utama
Kondisi akhir	Aplikasi *game* menampilkan menu pilihan *stage* yang ingin dimainkan	

Tabel 3.3 Skenario *Use Case* Tampil Cara Main

Identifikasi		
Nomor	2	
Nama	Menampilkan Cara Main	
Tujuan	Menampilkan cara memainkan *game*	
Deskripsi	Proses menampilkan cara memainkan *game*	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di menu utama	
Aksi Aktor		**Reaksi Sistem**
1. Memilih menu Cara Main		
		2. Menampilkan halaman cara memainkan *game*
3. Memilih Kembali		
		4. Menampilkan halaman menu utama
Kondisi akhir	Aplikasi *game* cara memainkan *game*	

Tabel 3.4 Skenario *Use Case* Tampil Credits

Identifikasi	
Nomor	3
Nama	Menampilkan Credits
Tujuan	Menampilkan credits
Deskripsi	Proses untuk menampilkan credits
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu Credits	
	2. Menampilkan halaman credits
3. Memilih Kembali	
	4. Menampilkan halaman menu utama
Kondisi akhir	Aplikasi *game* menampilkan credits

Tabel 3.5 Skenario *Use Case* Tampil Info

Identifikasi	
Nomor	4
Nama	Menampilkan Info
Tujuan	Menampilkan info mengenai *game*
Deskripsi	Proses menampilkan info mengenai *game* dan referensi yang dipakai dalam pembuatan *game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu Info	

	2. Menampilkan informasi mengenai *game*
3. Menekan tombol Esc	
	4. Kembali ke menu utama
Kondisi akhir	Aplikasi *game* menampilkan info

Tabel 3.6 Skenario *Use Case* Tampil *Highscore*

Identifikasi	
Nomor	5
Nama	Menampilkan *Highscore*
Tujuan	Menampilkan daftar *highscore game*
Deskripsi	Proses menampilkan daftar *highscore* dari pemain *game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu utama
Aksi Aktor	**Reaksi Sistem**
1. Memilih menu *Highscore*	
	2. Menampilkan daftar *highscore game*
3. Menekan tombol Esc	
	4. Kembali ke menu utama
Kondisi akhir	Aplikasi *game* menampilkan daftar *highscore*

Tabel 3.7 Skenario *Use Case* Pilih *Stage*

Identifikasi	
Nomor	6
Nama	Pilih *Stage*
Tujuan	Menampilkan pilihan *stage*
Deskripsi	Proses memilih *stage game* yang ingin dimainkan
Aktor	Pemain

Skenario Utama	
Kondisi awal	Pemain telah memilih menu Mulai
Aksi Aktor	**Reaksi Sistem**
1. Memilih *stage* yang ingin dimainkan	
	2. Menampilkan cerita *stage*
3. Memilih Lanjutkan ke Permainan	
	4. Menampilkan *stage* yang telah dipilih
Kondisi akhir	Aplikasi *game* menampilkan *stage* yang telah dipilih

Tabel 3.8 Skenario *Use Case* Tampil Tutorial

Identifikasi	
Nomor	7
Nama	Tampil Tutorial
Tujuan	Menampilkan tutorial *game*
Deskripsi	Proses menampilkan dan memainkan tutorial *game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*
Aksi Aktor	**Reaksi Sistem**
1. Memilih Tutorial	
	2. Menampilkan instruksi
	3. Menampilkan map

	4. Menampilkan informasi misi pemain
	5. Menempatkan pasukan pemain dan musuh
	6. Mengecek apakah seluruh pesukan musuh telah dikalahkan, apabila terpenuhi tampilkan halaman menang, apabila kalah menampilkan halaman game over
7. Klik kanan tombol *mouse*	
	8. Menampilkan instruksi selanjutnya
9. Menggerakkan cursor	
	10. Menampilkan *range* gerak pasukan apabila *cursor* berada diatas pasukan musuh
11. Memilih pasukan yang ingin digerakkan dengan menggunakan tombol spasi	
	12. Menampilkan *range* gerak pasukan yang dipilih
13. Memilih tujuan gerak pasukan	
	14. Memindahkan pasukan ke tujuan
15. Memilih serang ketika menemui pasukan musuh	
	16. Mengurangi HP pasukan yang diserang
17. Memilih kembali	
	18. Mengembalikan pasukan ke posisi awal

19. Memilih tunggu	
	20. Menempatkan pasukan di tujuan yang telah dipilih
21. Memilih akhiri	
	22. Mengakhiri giliran pemain dan memulai giliran komputer
23. Memilih Keluar	
	24. Menampilkan validasi apakah yakin ingin keluar dari aplikasi *game*
	25. Jika memilih "tidak" maka kembali ke permainan
	26. Jika memilih "ya" maka kembali ke menu utama
Kondisi akhir	Kondisi kemenangan terpenuhi dan pemain menang atau semua pasukan pemain habis dan pemain kalah. Kembali ke menu utama

Tabel 3.9 Skenario Use Case Tampil Prolog1

Identifikasi	
Nomor	8
Nama	Tampil Prolog1
Tujuan	Menampilkan Prolog1 cerita *game*
Deskripsi	Proses menampilkan prolog tahun 1945 dari Pertempuran Konvoy Sukabumi - Cianjur

Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di menu Pilih *Stage*	
Aksi Aktor		**Reaksi Sistem**
1. Memilih Prolog - 1945		
		2. Menampilkan cerita prolog - 1945
		3. Menampilkan *background* gambar yang sesuai dengan cerita
4. Klik pada *mouse* atau menekan spasi		
		5. Menampilkan cerita selanjutnya
		6. Menuju *Stage* selanjutnya ketika cerita selesai
7. Tekan tombol Esc		
		8. Kembali ke menu utama
Kondisi akhir	Pemain melanjutkan ke stage selanjutnya jika cerita telah selesau, atau kembali ke menu utama ketika menekan tombol Esc	

Tabel 3.10 Skenario Use Case Tampil *Stage1*

Identifikasi	
Nomor	9
Nama	Tampil *Stage1*
Tujuan	Menampilkan *stage1 game*
Deskripsi	Proses menampilkan dan memainkan *stage1 game*

Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*

Aksi Aktor	**Reaksi Sistem**
1. Memilih *Stage1*	
	2. Menampilkan map
	3. Menampilkan informasi misi pemain
	4. Menempatkan pasukan pemain dan musuh
	5. Mengecek apakah 50% (9) pasukan musuh telah dikalahkan , apabila terpenuhi tampilkan halaman menang, apabila kalah menampilkan halaman *game over*
	6. Menghitung *score* dan mengecek apakah *score* tersebut adalah *highscore* baru, jika ya, maka pemain dapat memasukkan namanya ke daftar *highscore*
7. Memasukkan nama dan tekan OK	
8. Menggerakkan cursor	
	9. Menampilkan *range* gerak pasukan apabila *cursor* berada diatas pasukan musuh
10. Memilih pasukan yang ingin digerakkan dengan menggunakan tombol spasi	

	11. Menampilkan *range* gerak pasukan yang dipilih
12. Memilih tujuan gerak pasukan	
	13. Memindahkan pasukan ke tujuan
14. Memilih serang ketika menemui pasukan musuh	
	15. Mengurangi HP pasukan yang diserang
16. Memilih kembali	
	17. Mengembalikan pasukan ke posisi awal
18. Memilih tunggu	
	19. Menempatkan pasukan di tujuan yang telah dipilih
20. Memilih akhiri	
	21. Mengakhiri giliran pemain dan memulai giliran komputer
22. Memilih Keluar	
	23. Menampilkan validasi apakah yakin ingin keluar dari aplikasi *game*
	24. Jika memilih "tidak" maka kembali ke permainan
	25. Jika memilih "ya" maka kembali ke menu utama
26. Menekan tombol F5	
	27. Membuat *checkpoint* baru (*save*)

Kondisi akhir	Misi terpenuhi dan pemain menang atau semua pasukan pemain habis dan pemain kalah. Kembali ke menu utama

Tabel 3.11 Skenario Use Case Tampil *Stage2*

Identifikasi	
Nomor	10
Nama	Tampil *Stage2*
Tujuan	Menampilkan *stage2 game*
Deskripsi	Proses menampilkan dan memainkan *stage2 game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*

Aksi Aktor	**Reaksi Sistem**
1. Memilih *Stage1*	
	2. Menampilkan map
	3. Menampilkan informasi misi pemain
	4. Menempatkan pasukan pemain dan musuh
	5. Mengecek apakah pasukan musuh telah tersisa 5 pasukan, apabila terpenuhi tampilkan halaman menang, apabila kalah menampilkan halaman *game over*
	6. Menghitung *score* dan mengecek apakah *score* tersebut adalah

	highscore baru, jika ya, maka pemain dapat memasukkan namanya ke daftar *highscore*
7. Memasukkan nama dan tekan OK	
8. Menggerakkan cursor	
	9. Menampilkan *range* gerak pasukan apabila *cursor* berada diatas pasukan musuh
10. Memilih pasukan yang ingin digerakkan dengan menggunakan tombol spasi	
	11. Menampilkan *range* gerak pasukan yang dipilih
12. Memilih tujuan gerak pasukan	
	13. Memindahkan pasukan ke tujuan
14. Memilih serang ketika menemui pasukan musuh	
	15. Mengurangi HP pasukan yang diserang
16. Memilih kembali	
	17. Mengembalikan pasukan ke posisi awal
18. Memilih tunggu	
	19. Menempatkan pasukan di tujuan yang telah dipilih
20. Memilih akhiri	
	21. Mengakhiri giliran pemain dan memulai giliran komputer

22. Memilih Keluar	
	23. Menampilkan validasi apakah yakin ingin keluar dari aplikasi *game*
	24. Jika memilih "tidak" maka kembali ke permainan
	25. Jika memilih "ya" maka kembali ke menu utama
26. Menekan tombol F5	
	27. Membuat *checkpoint* baru (*save*)
Kondisi akhir	Misi terpenuhi dan pemain menang atau semua pasukan pemain habis dan pemain kalah. Kembali ke menu utama

Tabel 3.12 Skenario Use Case Tampil *Stage3*

Identifikasi	
Nomor	11
Nama	Tampil *Stage3*
Tujuan	Menampilkan *stage3 game*
Deskripsi	Proses menampilkan dan memainkan *stage3 game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*
Aksi Aktor	**Reaksi Sistem**
1. Memilih *Stage3*	
	2. Menampilkan map

	3. Menampilkan informasi misi pemain
	4. Menempatkan pasukan pemain dan musuh
	5. Mengecek apakah 80% (19) pasukan musuh telah dikalahkan , apabila terpenuhi tampilkan halaman menang, apabila kalah menampilkan halaman *game over*
	6. Menghitung *score* dan mengecek apakah *score* tersebut adalah *highscore* baru, jika ya, maka pemain dapat memasukkan namanya ke daftar *highscore*
7. Memasukkan nama dan tekan OK	
8. Menggerakkan cursor	
	9. Menampilkan *range* gerak pasukan apabila *cursor* berada diatas pasukan musuh
10. Memilih pasukan yang ingin digerakkan dengan menggunakan tombol spasi	
	11. Menampilkan *range* gerak pasukan yang dipilih
12. Memilih tujuan gerak pasukan	
	13. Memindahkan pasukan ke tujuan
14. Memilih serang ketika menemui pasukan musuh	
	15. Mengurangi HP pasukan yang

	diserang
16. Memilih kembali	
	17. Mengembalikan pasukan ke posisi awal
18. Memilih tunggu	
	19. Menempatkan pasukan di tujuan yang telah dipilih
20. Memilih akhiri	
	21. Mengakhiri giliran pemain dan memulai giliran komputer
22. Memilih Keluar	
	23. Menampilkan validasi apakah yakin ingin keluar dari aplikasi *game*
	24. Jika memilih "tidak" maka kembali ke permainan
	25. Jika memilih "ya" maka kembali ke menu utama
26. Menekan tombol F5	
	27. Membuat *checkpoint* baru (*save*)
Kondisi akhir	Misi terpenuhi dan pemain menang atau semua pasukan pemain habis dan pemain kalah. Kembali ke menu utama

Tabel 3.13 Skenario Use Case Tampil Epilog1

Identifikasi	
Nomor	12
Nama	Tampil Epilog1

Tujuan	Menampilkan Epilog1 cerita *game*	
Deskripsi	Proses menampilkan Epilog tahun 1945 Pertempuran Konvoy Sukabumi - Cianjur	
Aktor	Pemain	
Skenario Utama		
Kondisi awal	Pemain berada di menu Pilih *Stage*	
Aksi Aktor		**Reaksi Sistem**
1. Memilih Epilog - 1945		
		2. Menampilkan cerita epilog - 1945
		3. Menampilkan *background* gambar yang sesuai dengan cerita
4. Klik pada *mouse* atau menekan spasi		
		5. Menampilkan cerita selanjutnya
		6. Menuju menu Pilih *Stage* ketika cerita selesai
7. Tekan tombol Esc		
		8. Kembali ke menu utama
Kondisi akhir	Pemain kembali ke menu Pilih Stage jika cerita telah selesai, atau kembali ke menu utama ketika menekan tombol Esc	

Tabel 3.14 Skenario Use Case Tampil Prolog2

Identifikasi	
Nomor	13

Nama	Tampil Prolog2
Tujuan	Menampilkan Prolog2 cerita *game*
Deskripsi	Proses menampilkan prolog tahun 1946 dari Pertempuran Konvoy Sukabumi - Cianjur
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*

Aksi Aktor	**Reaksi Sistem**
1. Memilih Prolog - 1946	
	2. Menampilkan cerita prolog - 1946
	3. Menampilkan *background* gambar yang sesuai dengan cerita
4. Klik pada *mouse* atau menekan spasi	
	5. Menampilkan cerita selanjutnya
	6. Menuju *Stage* selanjutnya ketika cerita selesai
7. Tekan tombol Esc	
	8. Kembali ke menu utama

Kondisi akhir	Pemain melanjutkan ke stage selanjutnya jika cerita telah selesau, atau kembali ke menu utama ketika menekan tombol Esc

Tabel 3.15 Skenario Use Case Tampil *Stage4*

Identifikasi	
Nomor	14
Nama	Tampil *Stage4*
Tujuan	Menampilkan *stage4 game*
Deskripsi	Proses menampilkan dan memainkan *stage4 game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*

Aksi Aktor	**Reaksi Sistem**
1. Memilih *Stage4*	
	2. Menampilkan map
	3. Menampilkan informasi misi pemain
	4. Menempatkan pasukan pemain dan musuh
	5. Mengecek apakah 80% (28) pasukan musuh telah dikalahkan , apabila terpenuhi tampilkan halaman menang, apabila kalah menampilkan halaman *game over*
	6. Menghitung *score* dan mengecek apakah *score* tersebut adalah *highscore* baru, jika ya, maka pemain dapat memasukkan namanya ke daftar *highscore*

7. Memasukkan nama dan tekan OK	
8. Menggerakkan cursor	
	9. Menampilkan *range* gerak pasukan apabila *cursor* berada diatas pasukan musuh
10. Memilih pasukan yang ingin digerakkan dengan menggunakan tombol spasi	
	11. Menampilkan *range* gerak pasukan yang dipilih
12. Memilih tujuan gerak pasukan	
	13. Memindahkan pasukan ke tujuan
14. Memilih serang ketika menemui pasukan musuh	
	15. Mengurangi HP pasukan yang diserang
16. Memilih kembali	
	17. Mengembalikan pasukan ke posisi awal
18. Memilih tunggu	
	19. Menempatkan pasukan di tujuan yang telah dipilih
20. Memilih akhiri	
	21. Mengakhiri giliran pemain dan memulai giliran komputer
22. Memilih Keluar	
	23. Menampilkan validasi apakah

	yakin ingin keluar dari aplikasi *game*
	24. Jika memilih "tidak" maka kembali ke permainan
	25. Jika memilih "ya" maka kembali ke menu utama
26. Menekan tombol F5	
	27. Membuat *checkpoint* baru (*save*)
Kondisi akhir	Misi terpenuhi dan pemain menang atau semua pasukan pemain habis dan pemain kalah. Kembali ke menu utama

Tabel 3.16 Skenario Use Case Tampil *Stage5*

Identifikasi	
Nomor	15
Nama	Tampil *Stage5*
Tujuan	Menampilkan *stage5 game*
Deskripsi	Proses menampilkan dan memainkan *stage5 game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*
Aksi Aktor	**Reaksi Sistem**
1. Memilih *Stage5*	
	2. Menampilkan map
	3. Menampilkan informasi misi pemain

	4. Menempatkan pasukan pemain dan musuh
	5. Mengecek apakah 80% (28) pasukan musuh telah dikalahkan , apabila terpenuhi tampilkan halaman menang, apabila kalah menampilkan halaman *game over*
	6. Menghitung *score* dan mengecek apakah *score* tersebut adalah *highscore* baru, jika ya, maka pemain dapat memasukkan namanya ke daftar *highscore*
7. Memasukkan nama dan tekan OK	
8. Menggerakkan cursor	
	9. Menampilkan *range* gerak pasukan apabila *cursor* berada diatas pasukan musuh
10. Memilih pasukan yang ingin digerakkan dengan menggunakan tombol spasi	
	11. Menampilkan *range* gerak pasukan yang dipilih
12. Memilih tujuan gerak pasukan	
	13. Memindahkan pasukan ke tujuan
14. Memilih serang ketika menemui pasukan musuh	
	15. Mengurangi HP pasukan yang diserang
16. Memilih kembali	

	17. Mengembalikan pasukan ke posisi awal
18. Memilih tunggu	
	19. Menempatkan pasukan di tujuan yang telah dipilih
20. Memilih akhiri	
	21. Mengakhiri giliran pemain dan memulai giliran komputer
22. Memilih Keluar	
	23. Menampilkan validasi apakah yakin ingin keluar dari aplikasi *game*
	24. Jika memilih "tidak" maka kembali ke permainan
	25. Jika memilih "ya" maka kembali ke menu utama
26. Menekan tombol F5	
	27. Membuat *checkpoint* baru (*save*)
Kondisi akhir	Misi terpenuhi dan pemain menang atau semua pasukan pemain habis dan pemain kalah. Kembali ke menu utama

Tabel 3.17 Skenario Use Case Tampil *Stage6*

Identifikasi	
Nomor	16
Nama	Tampil *Stage6*
Tujuan	Menampilkan *stage6 game*
Deskripsi	Proses menampilkan dan memainkan *stage6 game*

Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*

Aksi Aktor	**Reaksi Sistem**
1. Memilih *Stage6*	
	2. Menampilkan map
	3. Menampilkan informasi misi pemain
	4. Menempatkan pasukan pemain dan musuh
	5. Mengecek apakah 80% (26) pasukan musuh telah dikalahkan , apabila terpenuhi tampilkan halaman menang, apabila kalah menampilkan halaman *game over*
	6. Menghitung *score* dan mengecek apakah *score* tersebut adalah *highscore* baru, jika ya, maka pemain dapat memasukkan namanya ke daftar *highscore*
7. Memasukkan nama dan tekan OK	
8. Menggerakkan cursor	
	9. Menampilkan *range* gerak pasukan apabila *cursor* berada diatas pasukan musuh
10. Memilih pasukan yang ingin digerakkan dengan menggunakan tombol spasi	

	11. Menampilkan *range* gerak pasukan yang dipilih
12. Memilih tujuan gerak pasukan	
	13. Memindahkan pasukan ke tujuan
14. Memilih serang ketika menemui pasukan musuh	
	15. Mengurangi HP pasukan yang diserang
16. Memilih kembali	
	17. Mengembalikan pasukan ke posisi awal
18. Memilih tunggu	
	19. Menempatkan pasukan di tujuan yang telah dipilih
20. Memilih akhiri	
	21. Mengakhiri giliran pemain dan memulai giliran komputer
22. Memilih Keluar	
	23. Menampilkan validasi apakah yakin ingin keluar dari aplikasi *game*
	24. Jika memilih "tidak" maka kembali ke permainan
	25. Jika memilih "ya" maka kembali ke menu utama
26. Menekan tombol F5	
	27. Membuat *checkpoint* baru (*save*)

Kondisi akhir	Misi terpenuhi dan pemain menang atau semua pasukan pemain habis dan pemain kalah. Kembali ke menu utama

Tabel 3.18 Skenario Use Case Tampil *Stage7*

Identifikasi	
Nomor	17
Nama	Tampil *Stage7*
Tujuan	Menampilkan *stage7 game*
Deskripsi	Proses menampilkan dan memainkan *stage7 game*
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*

Aksi Aktor	**Reaksi Sistem**
28. Memilih *Stage7*	
	29. Menampilkan map
	30. Menampilkan informasi misi pemain
	31. Menempatkan pasukan pemain dan musuh
	32. Mengecek apakah pasukan musuh telah tersisa 8 pasukan, apabila terpenuhi tampilkan halaman menang, apabila kalah menampilkan halaman *game over*
	33. Menghitung *score* dan mengecek apakah *score* tersebut adalah

	highscore baru, jika ya, maka pemain dapat memasukkan namanya ke daftar *highscore*
34. Memasukkan nama dan tekan OK	
35. Menggerakkan cursor	
	36. Menampilkan *range* gerak pasukan apabila *cursor* berada diatas pasukan musuh
37. Memilih pasukan yang ingin digerakkan dengan menggunakan tombol spasi	
	38. Menampilkan *range* gerak pasukan yang dipilih
39. Memilih tujuan gerak pasukan	
	40. Memindahkan pasukan ke tujuan
41. Memilih serang ketika menemui pasukan musuh	
	42. Mengurangi HP pasukan yang diserang
43. Memilih kembali	
	44. Mengembalikan pasukan ke posisi awal
45. Memilih tunggu	
	46. Menempatkan pasukan di tujuan yang telah dipilih
47. Memilih akhiri	
	48. Mengakhiri giliran pemain dan memulai giliran komputer

49. Memilih Keluar	
	50. Menampilkan validasi apakah yakin ingin keluar dari aplikasi *game*
	51. Jika memilih "tidak" maka kembali ke permainan
	52. Jika memilih "ya" maka kembali ke menu utama
53. Menekan tombol F5	
	54. Membuat *checkpoint* baru (*save*)
Kondisi akhir	Misi terpenuhi dan pemain menang atau semua pasukan pemain habis dan pemain kalah. Kembali ke menu utama

Tabel 3.19 Skenario Use Case Tampil Epilog2

Identifikasi	
Nomor	18
Nama	Tampil Epilog2
Tujuan	Menampilkan Epilog2 cerita *game*
Deskripsi	Proses menampilkan epilog tahun 1946 dari Pertempuran Konvoy Sukabumi - Cianjur
Aktor	Pemain
Skenario Utama	
Kondisi awal	Pemain berada di menu Pilih *Stage*
Aksi Aktor	**Reaksi Sistem**
1. Memilih Epilog - 1946	

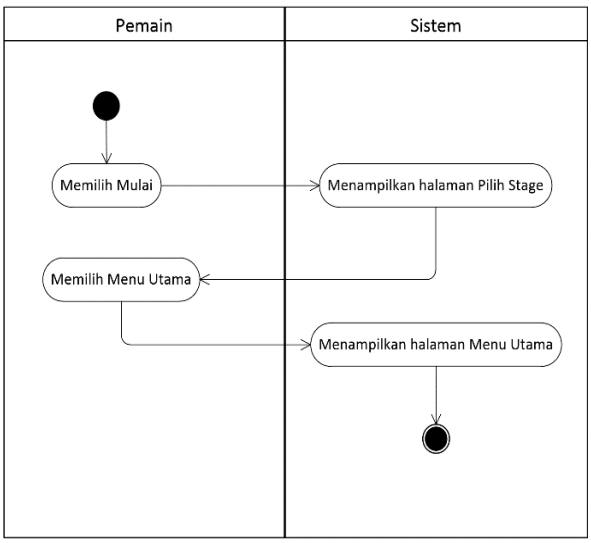
	2. Menampilkan cerita epilog - 1946
	3. Menampilkan *background* gambar yang sesuai dengan cerita
4. Klik pada *mouse* atau menekan spasi	
	5. Menampilkan cerita selanjutnya
	6. Kembali ke menu Pilih *Stage* ketika cerita selesai
7. Tekan tombol Esc	
	8. Kembali ke menu utama
Kondisi akhir	Pemain melanjutkan ke stage selanjutnya jika cerita telah selesau, atau kembali ke menu utama ketika menekan tombol Esc

3.4.4 *Activity Diagram*

Diagram Aktifitas atau *Activity Diagram* menggambarkan alur aktifitas didalam sebuah proses (*use case*). *Activity diagram* menggambarkan proses-proses dan jalur-jalur aktifitas dari level atas secara umum. Sebuah aktifitas dapat direalisasikan oleh satu *use case* atau lebih.

3.4.4.1 *Activity Diagram* Memilih Mulai

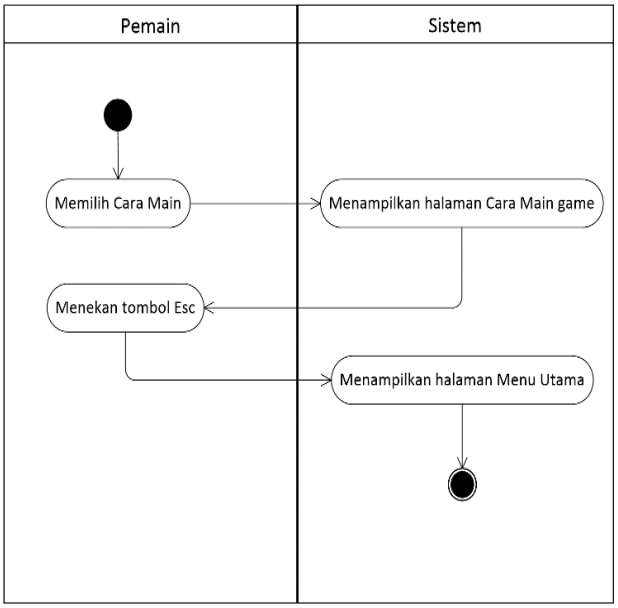
Activity diagram memilih Mulai merupakan alur aktifitas dari memilih menu mulai pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.18 *Activity Diagram* Memilih Mulai

3.4.4.2 *Activity Diagram* Tampil Cara Main

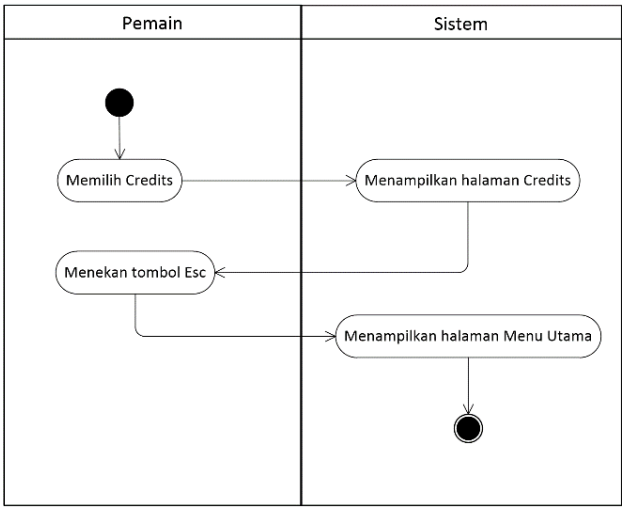
Activity diagram tampil cara main merupakan alur aktifitas dari proses menampilkan cara main pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.19 *Activity Diagram* Menampilkan Cara Main

3.4.4.3 *Activity Digram* Tampil *Credits*

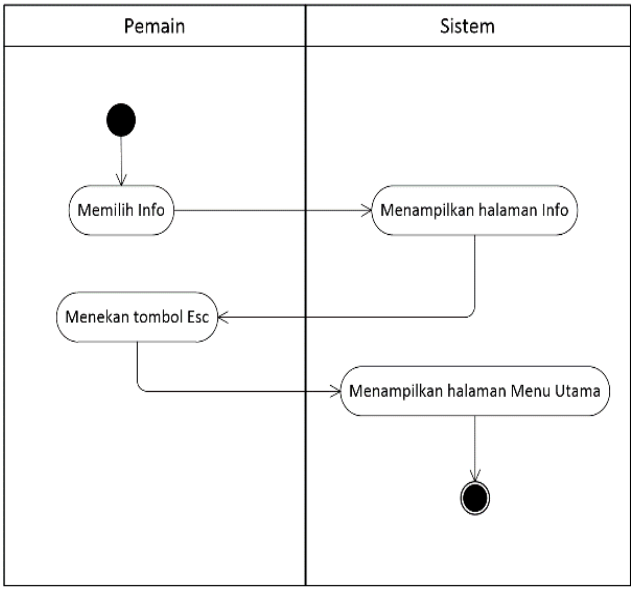
Activity diagram tampil *credits* merupakan alur aktifitas dari proses menampilkan *credits* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.20 *Activity Diagram* Menampilkan *Credits*

3.4.4.4 *Activity Diagram* Tampil Info

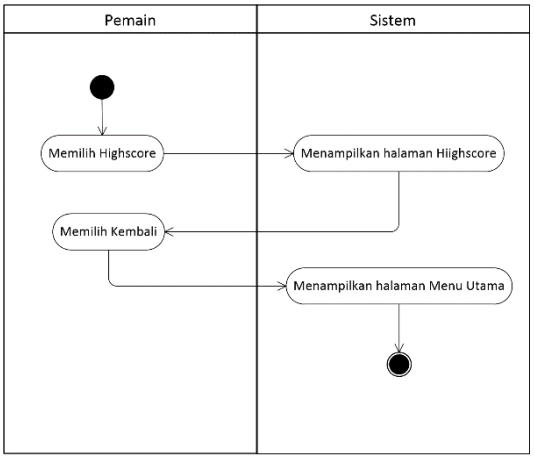
Activity diagram tampil info merupakan alur aktifitas dari proses menampilkan info pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.21 *Activity Diagram* Menampilkan Info

3.4.4.5 *Activity Diagram* Tampil *Highscore*

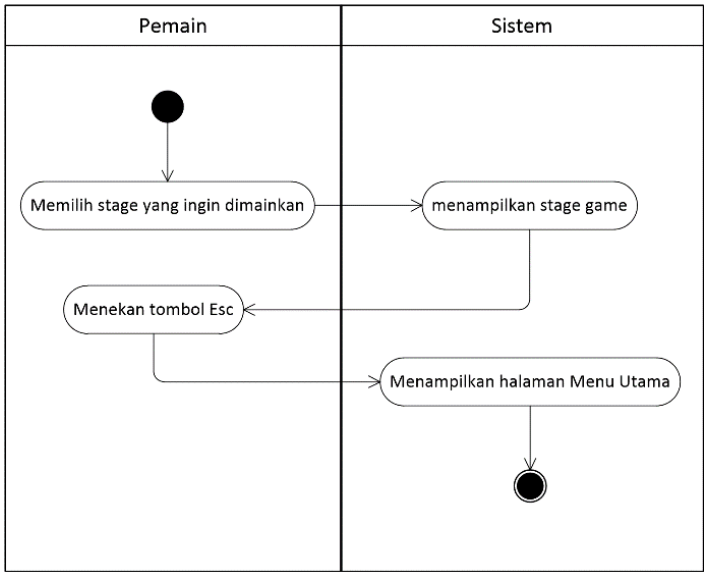
Activity diagram tampil *highscore* merupakan alur aktifitas dari proses menampilkan *highscore* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.22 ***Activity Diagram*** **Menampilkan** ***Highscore***

3.4.4.6 *Activity Diagram* Pilih *Stage*

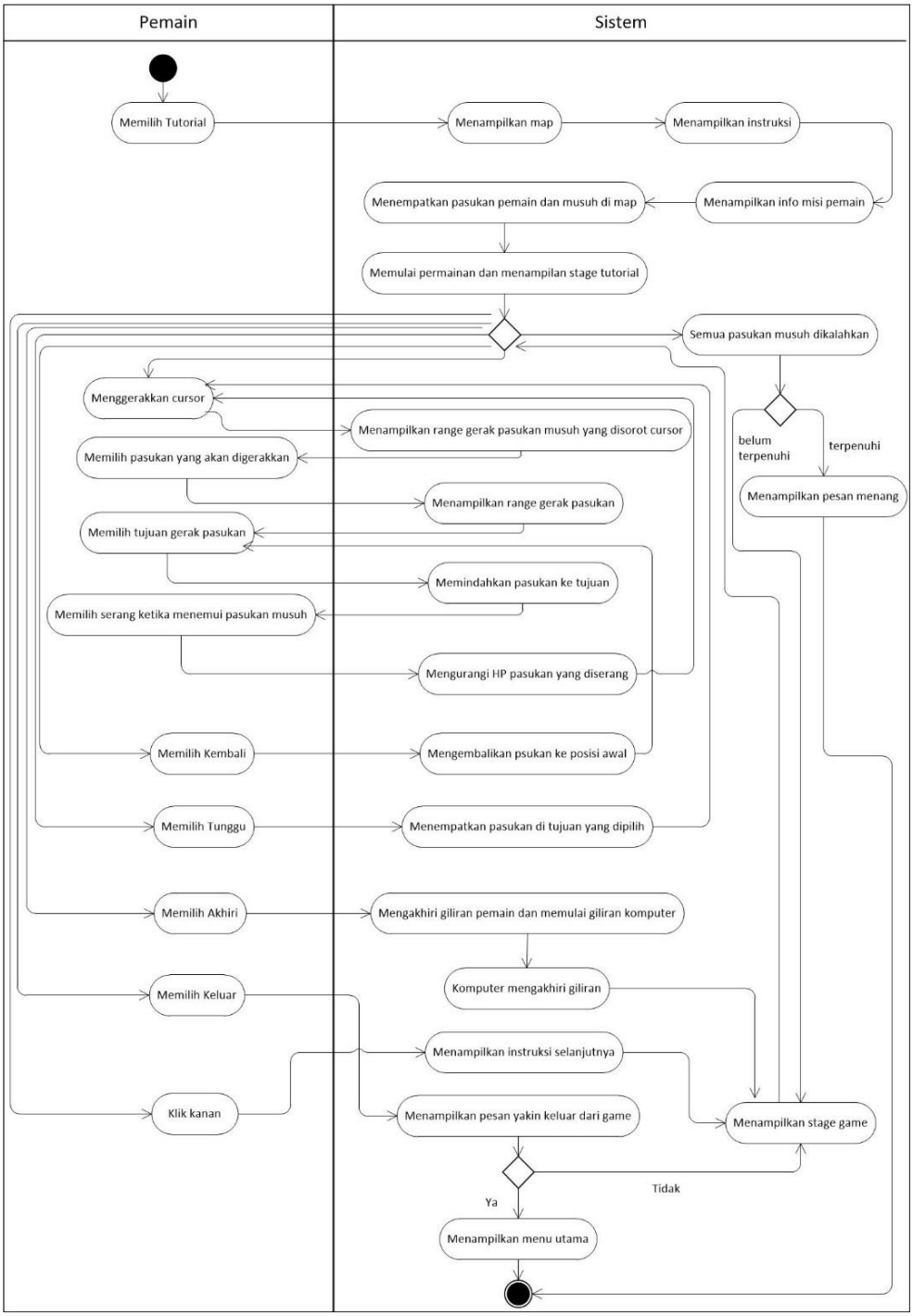
Activity diagram pilih *stage* merupakan alur aktifitas dari proses pilih *stage* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.23 ***Activity Diagram*** **Pilih** ***Stage***

3.4.4.7 *Activity Diagram* Tampil Tutorial

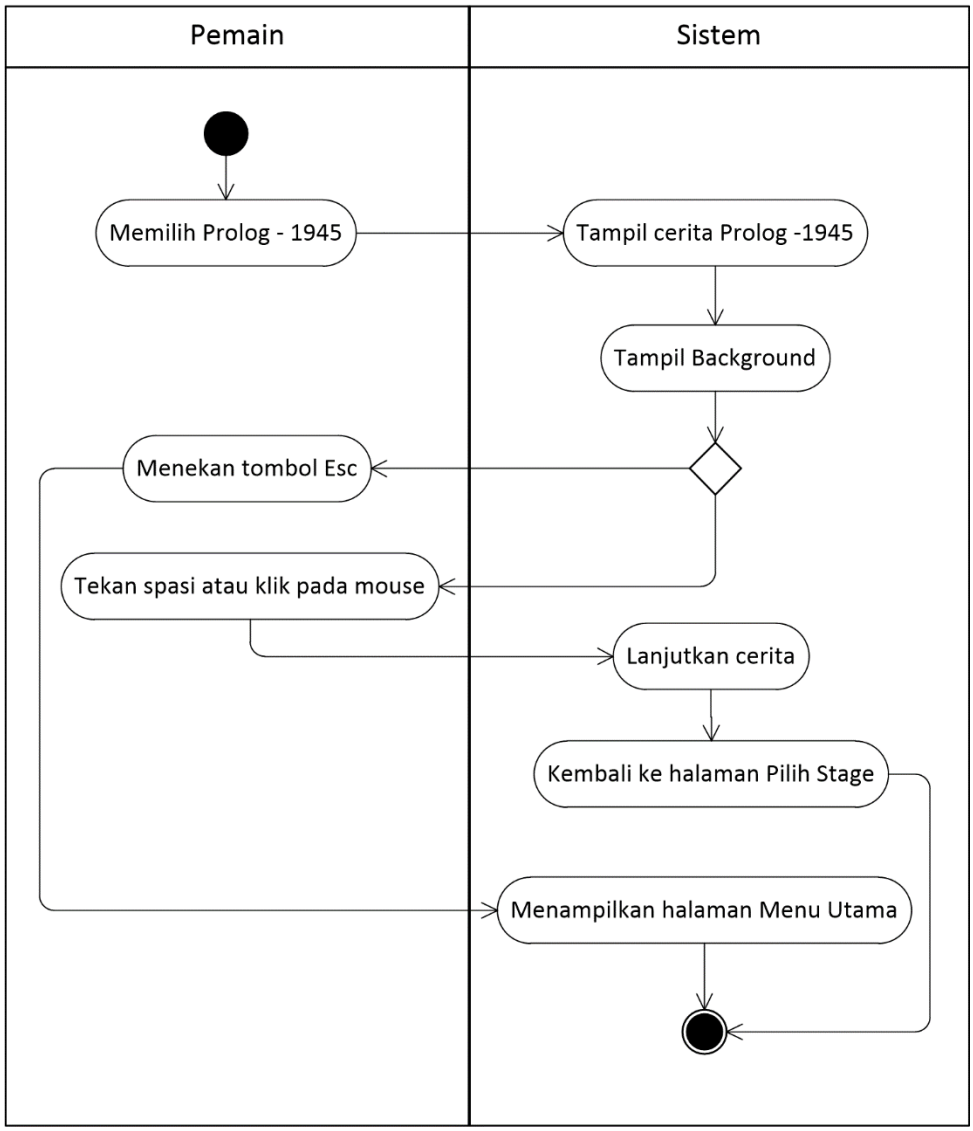
Activity diagram pilih tutorial merupakan alur aktifitas dari proses tampil tutorial pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.24 *Activity Diagram* Tampil Tutorial

3.4.4.8 *Activity Diagram* Tampil Prolog1

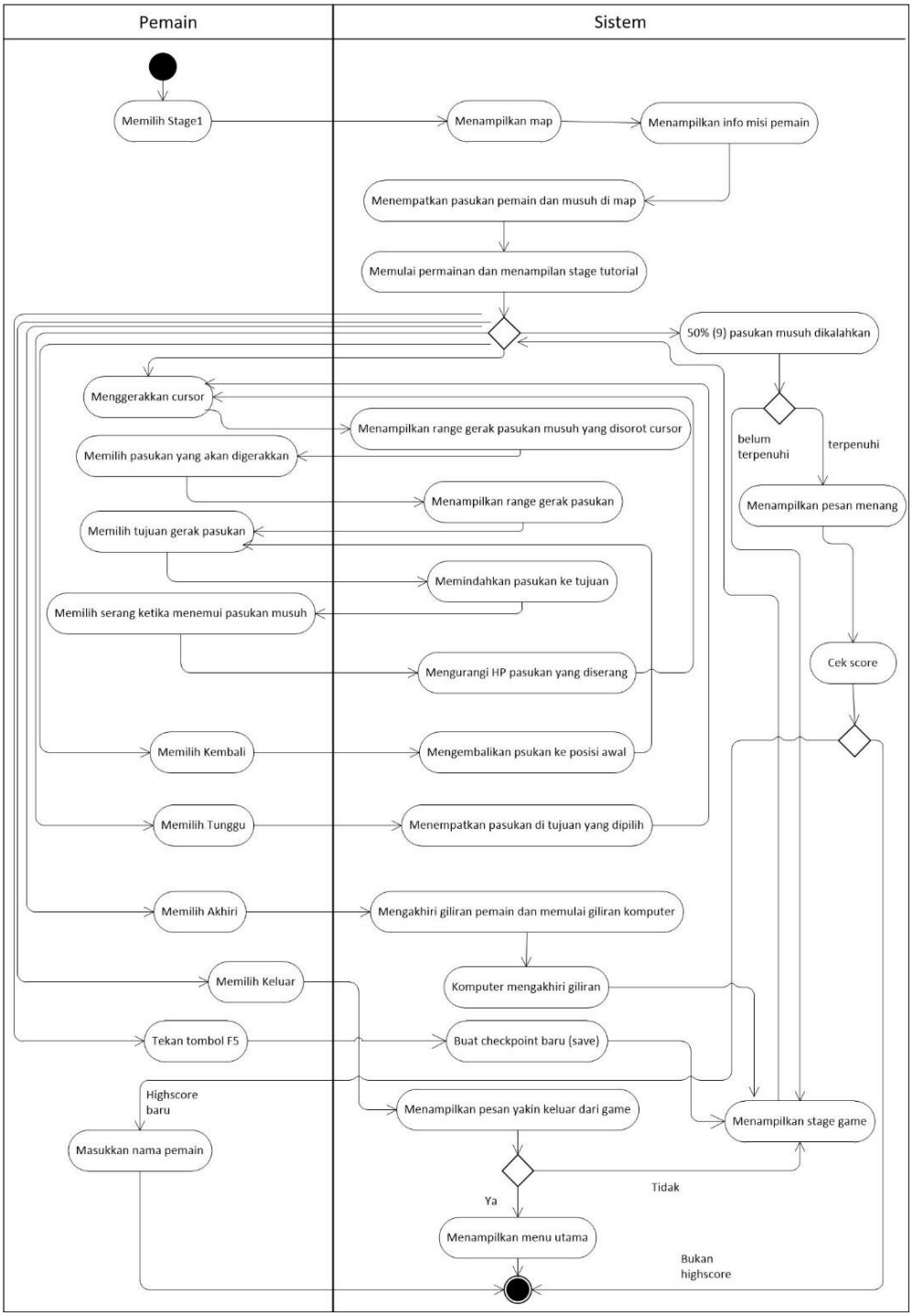
Activity diagram tampil prolog1 merupakan alur aktifitas dari proses tampil prolog - 1945 pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.25 Activity Diagram Tampil Prolog

3.4.4.9 *Activity Diagram* Tampil *Stage1*

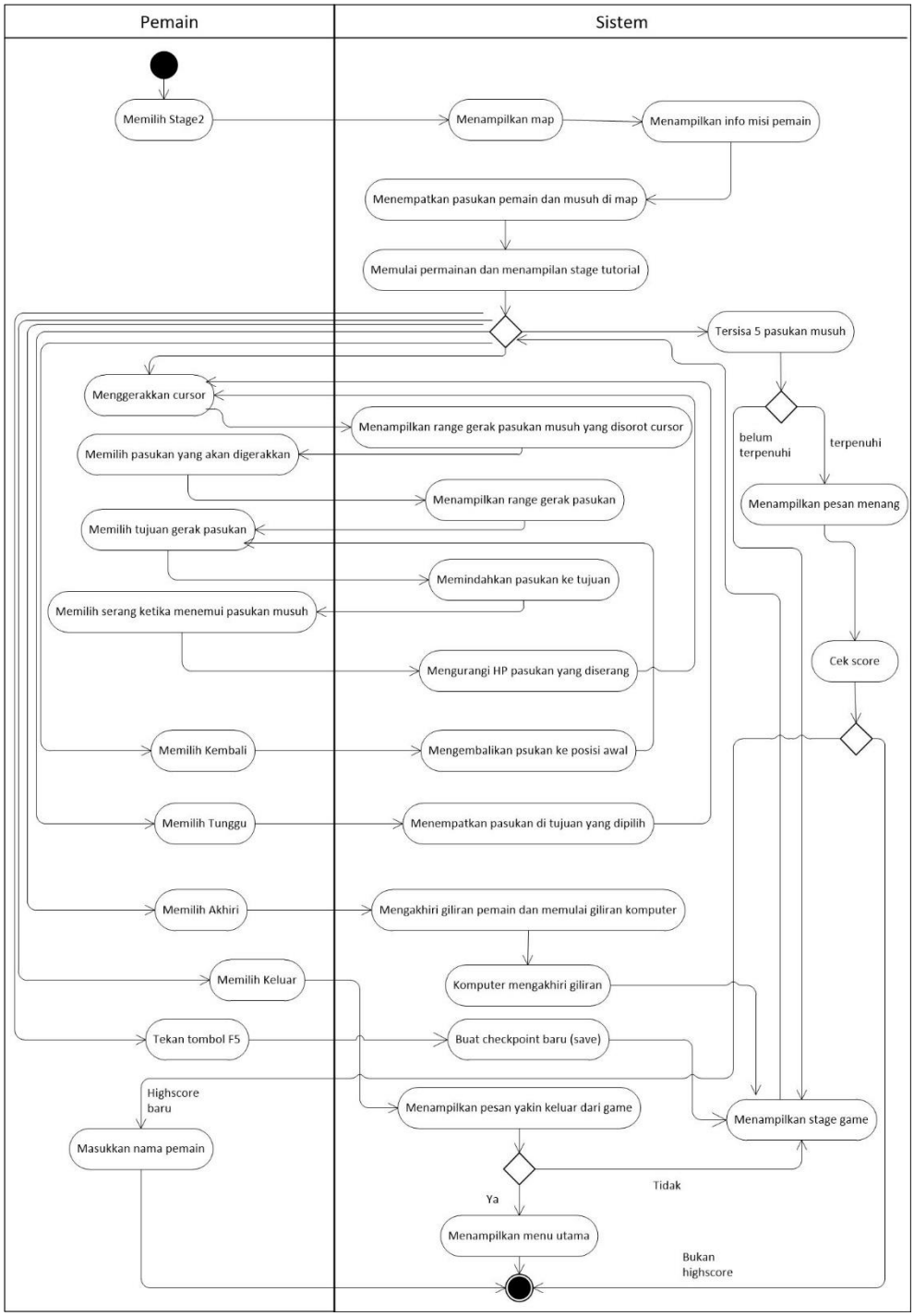
Activity diagram tampil *stage1* merupakan alur aktifitas dari proses tampil *stage1* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.26 Activity Diagram Tampil Stage1

3.4.4.10 *Activity Diagram* Tampil *Stage2*

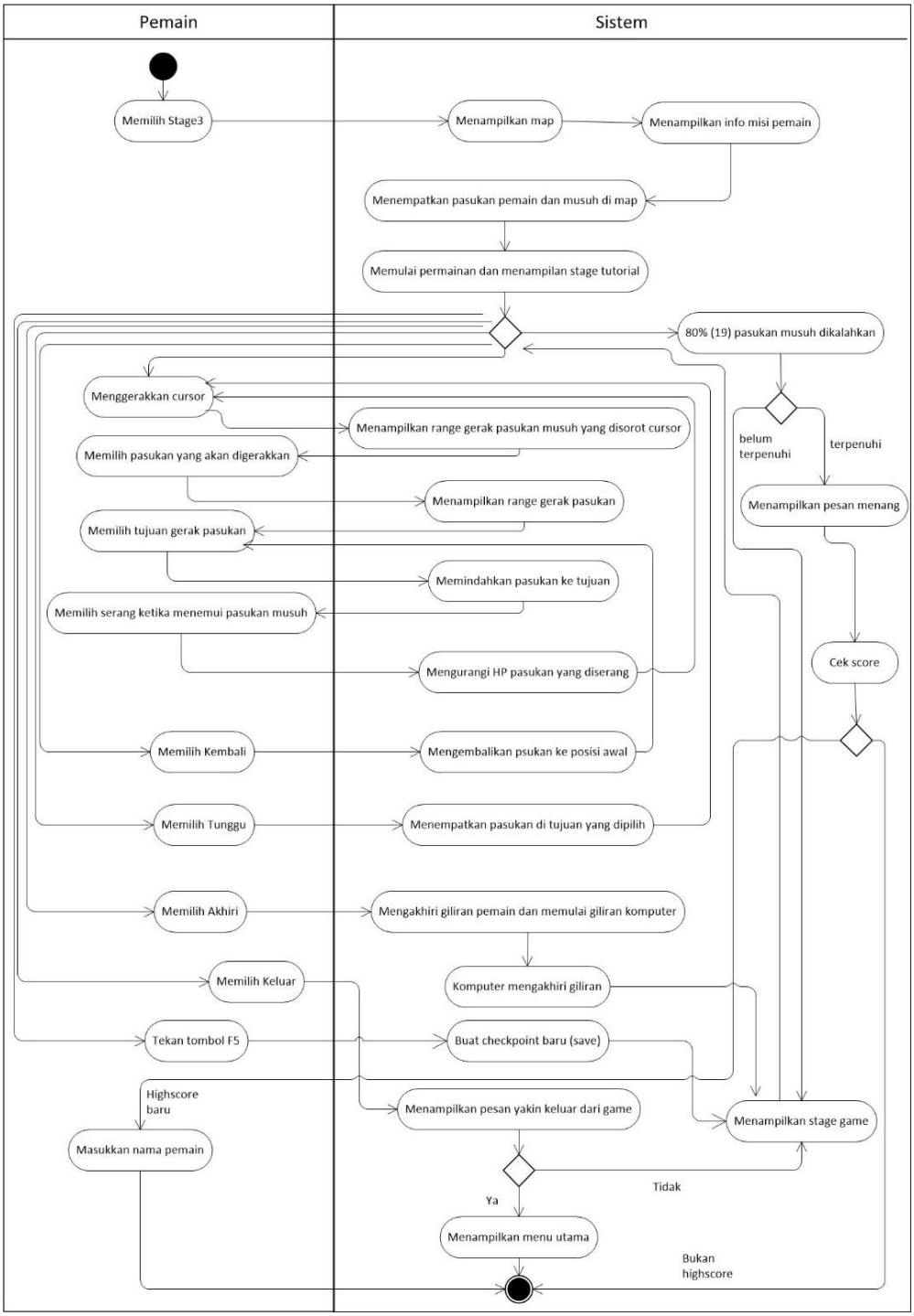
Activity diagram tampil *stage2* merupakan alur aktifitas dari proses tampil *stage2* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.27 Activity Diagram Tampil Stage2

3.4.4.11 *Activity Diagram* Tampil *Stage3*

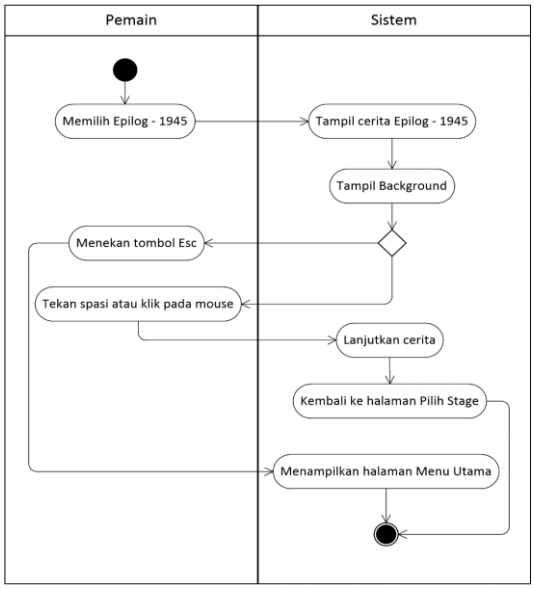
Activity diagram tampil *stage3* merupakan alur aktifitas dari proses tampil *stage3* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.28 Activity Diagram Tampil Stage3

3.4.4.12 *Activity Diagram* Tampil Epilog1

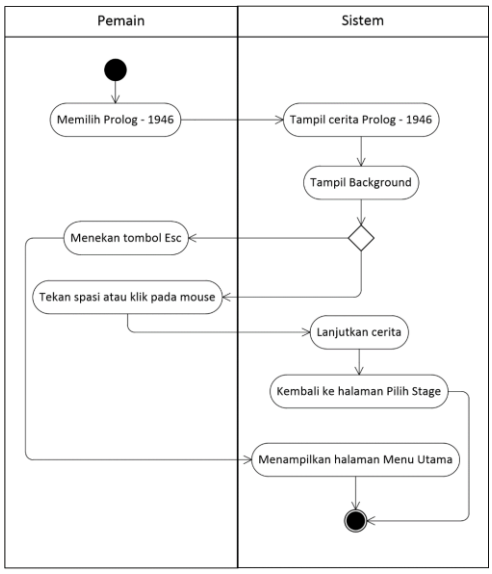
Activity diagram tampil epilog1 merupakan alur aktifitas dari proses tampil epilog - 1945 pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.29 Activity Diagram Tampil Epilog1

3.4.4.13 Activity Diagram Tampil Prolog2

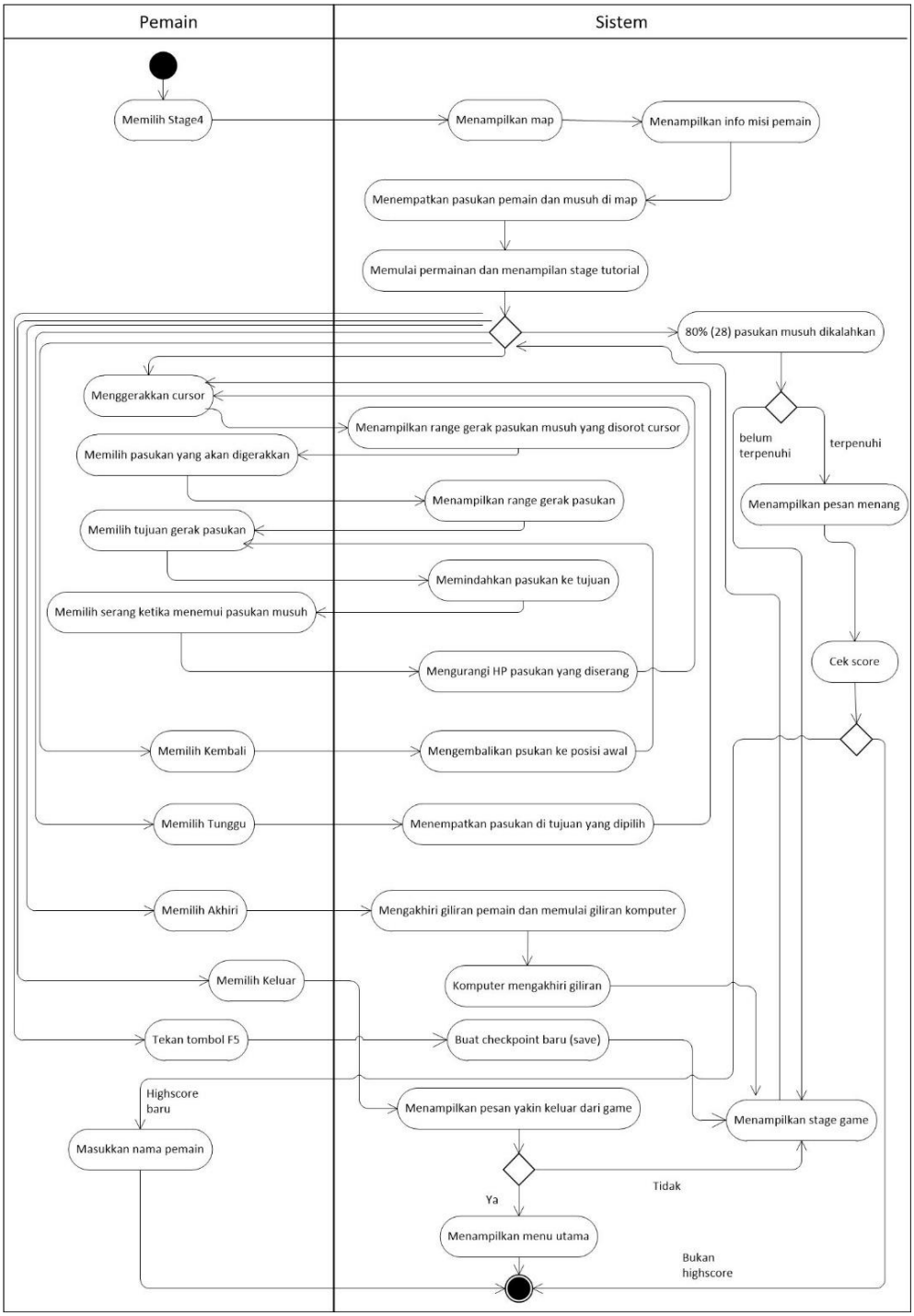
Activity diagram tampil epilog1 merupakan alur aktifitas dari proses tampil prolog - 1946 pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.30 Activity Diagram Tampil Prolog2

3.4.4.14 *Activity Diagram* Tampil *Stage4*

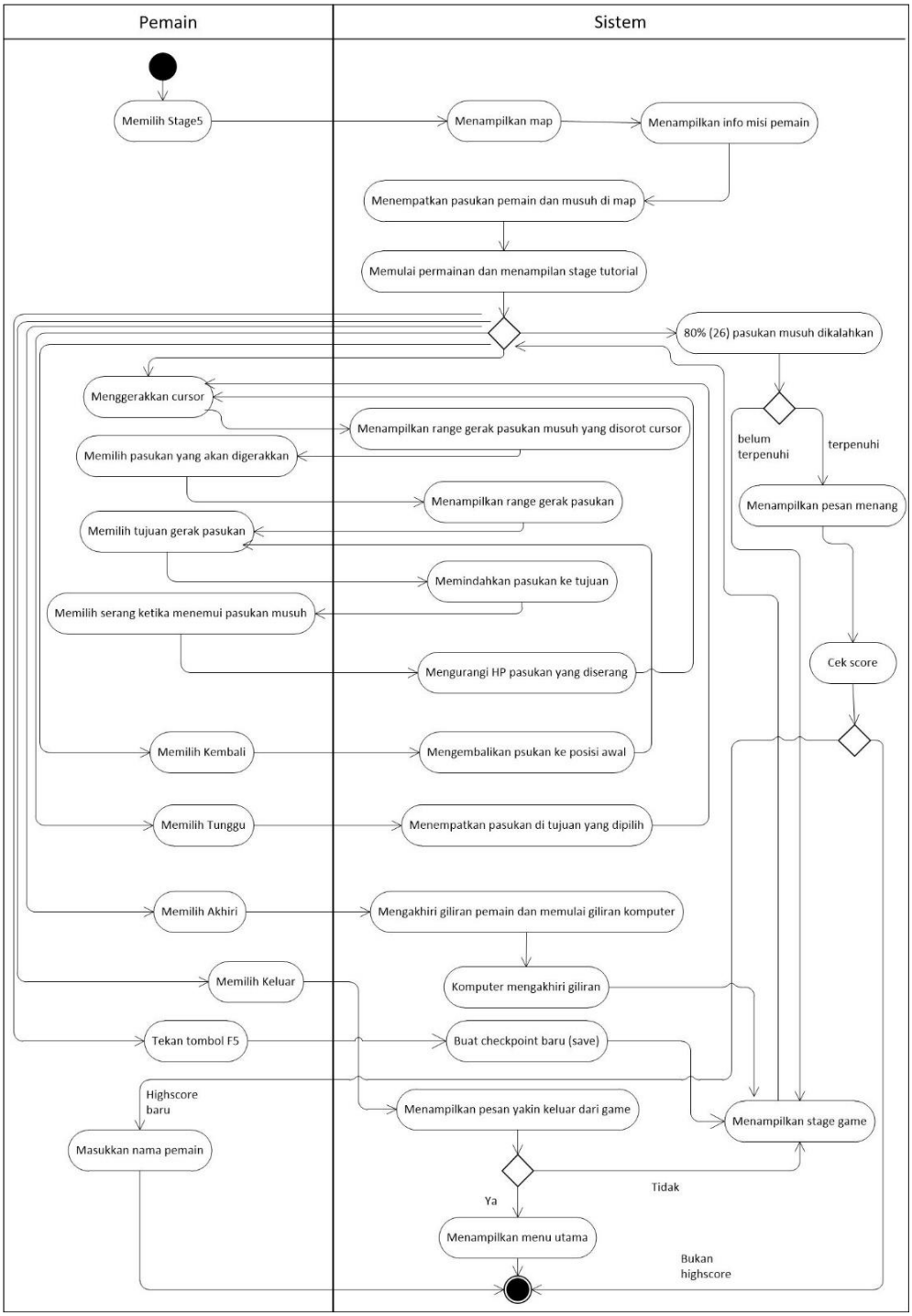
Activity diagram tampil *stage4* merupakan alur aktifitas dari proses tampil *stage4* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.31 Activity Diagram Tampil Stage4

3.4.4.15 *Activity Diagram* Tampil *Stage5*

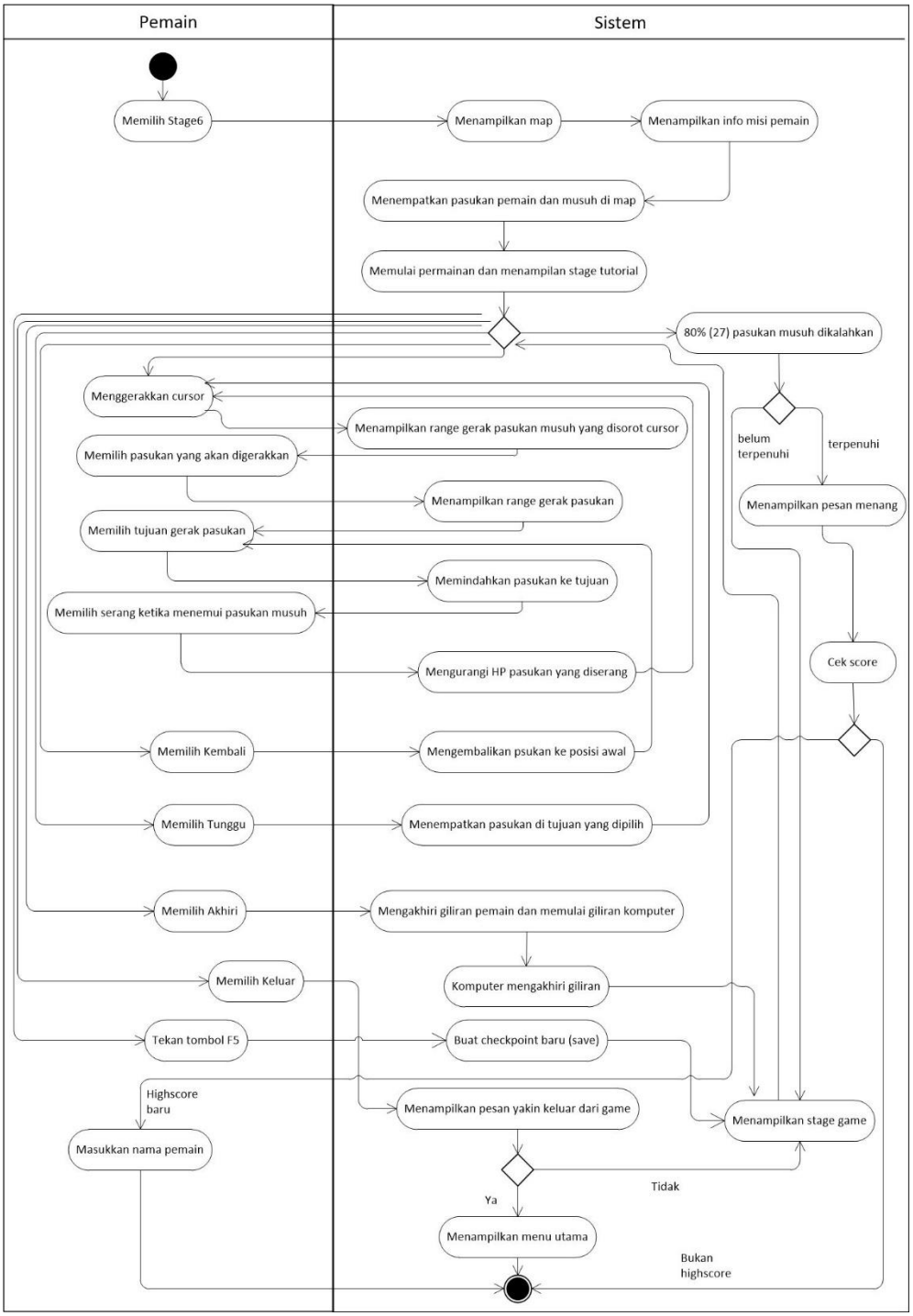
Activity diagram tampil *stage5* merupakan alur aktifitas dari proses tampil *stage5* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.32 Activity Diagram Tampil Stage5

3.4.4.16 *Activity Diagram* Tampil *Stage6*

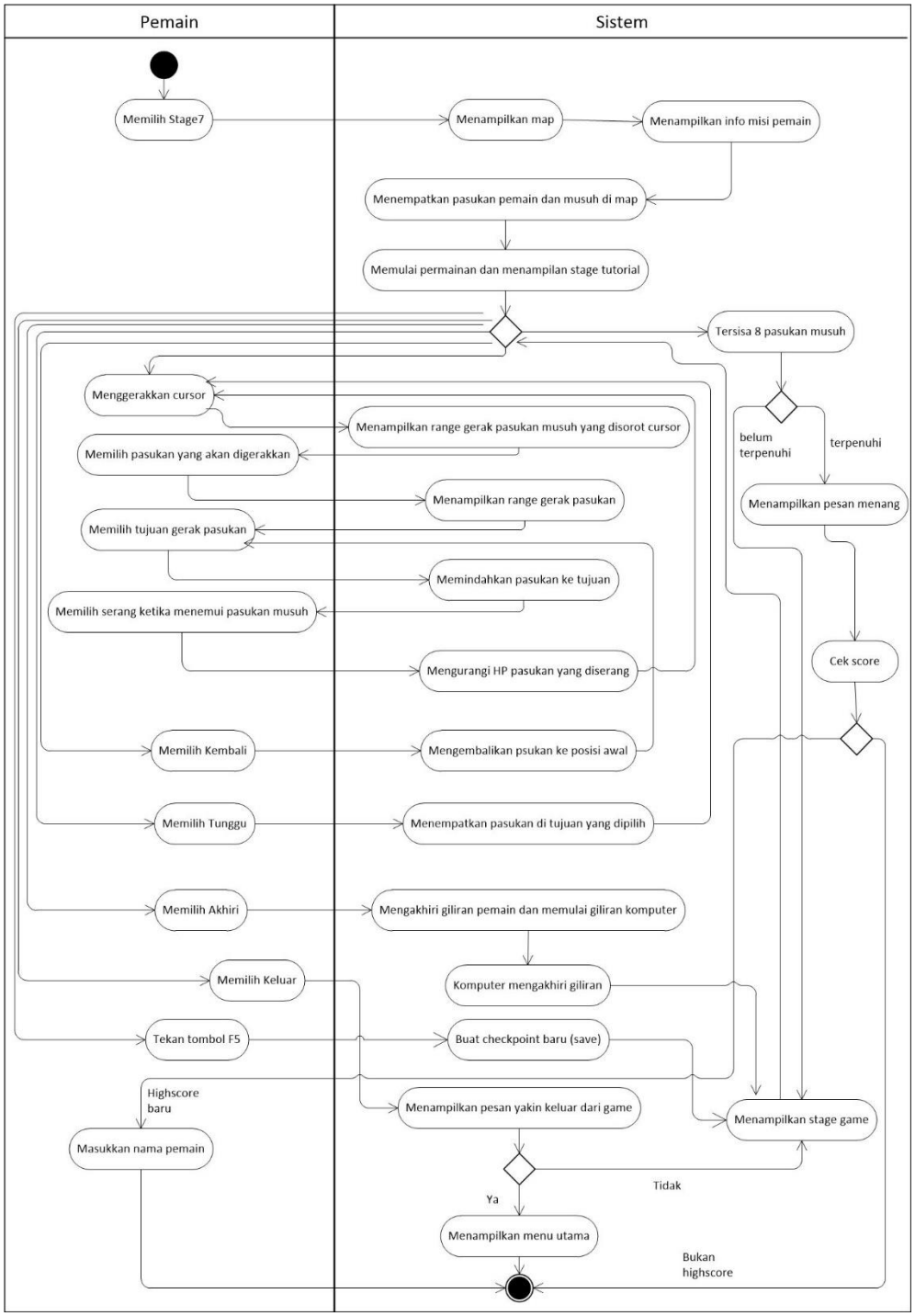
Activity diagram tampil *stage6* merupakan alur aktifitas dari proses tampil *stage6* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.33 Activity Diagram Tampil Stage6

3.4.4.17 *Activity Diagram* Tampil *Stage7*

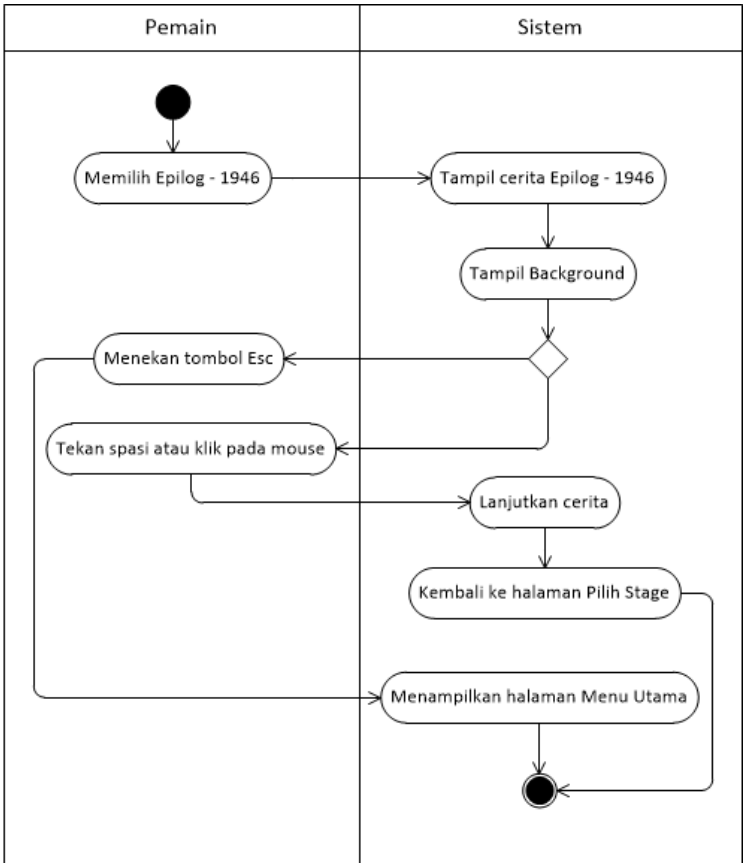
Activity diagram tampil *stage7* merupakan alur aktifitas dari proses tampil *stage7* pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.34 Activity Diagram Tampil Stage7

3.4.4.18 *Activity Diagram* Tampil Epilog2

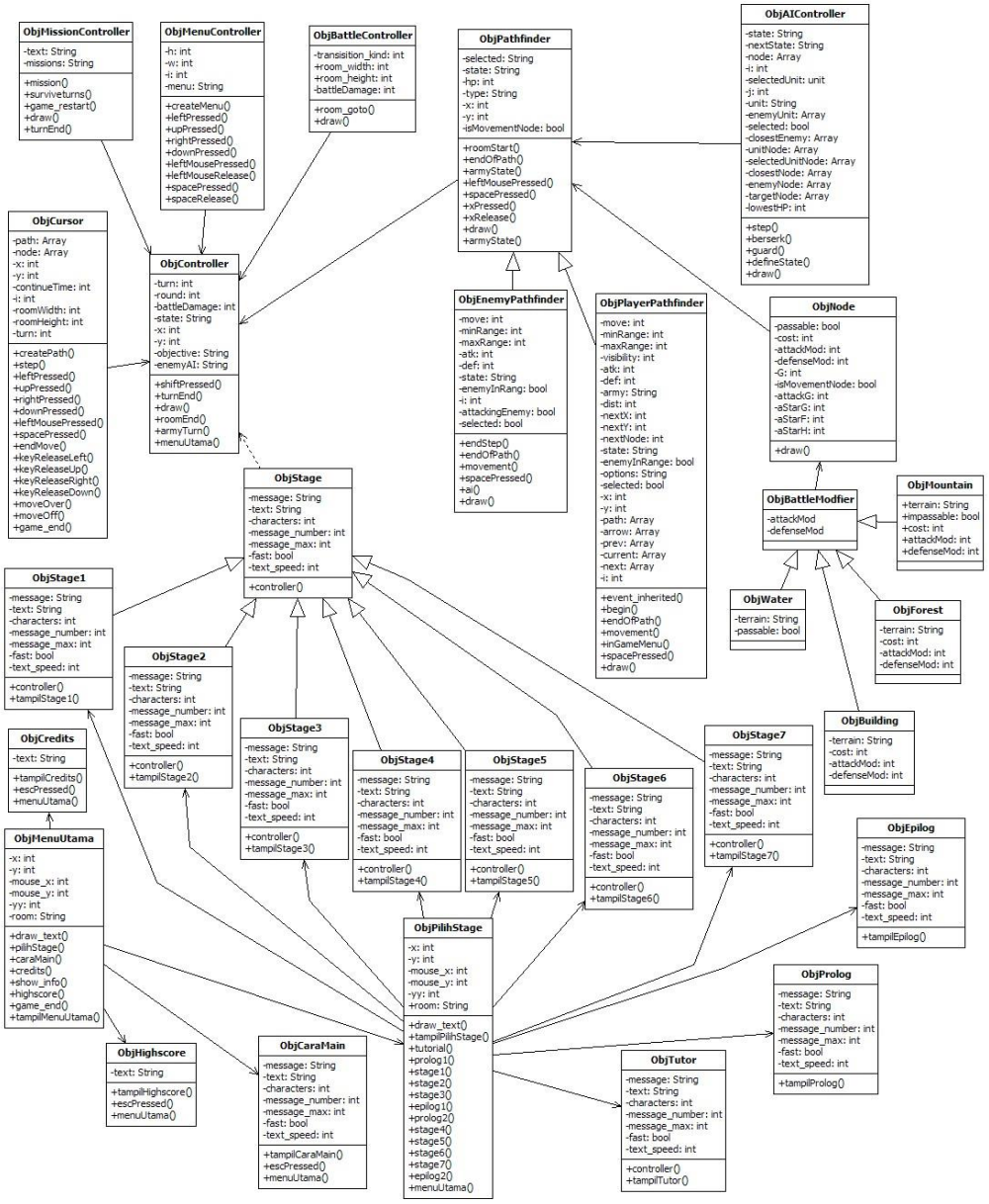
Activity diagram tampil epilog2 merupakan alur aktifitas dari proses tampil epilog - 1946 pada *game* Pertempuran Konvoy Sukabumi – Cianjur.



Gambar 3.35 Activity Diagram Tampil Epilog2

3.4.5 *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Berikut ini adalah perancangan struktur sistem yang terdapat pada aplikasi *game* Pertempuran Konvoy Sukabumi - Cianjur yang digambarkan dengan *Class Diagram* yang dapat dilihat pada gambar 3.36.

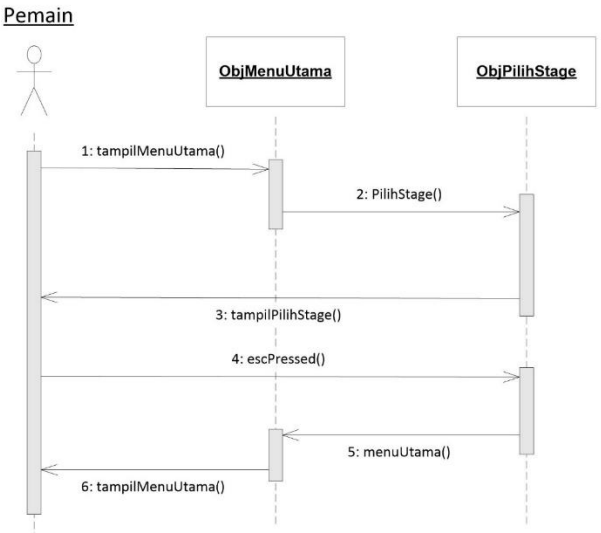


Gambar 3.36 *Class Diagram* Aplikasi *Game* Pertempuran Konvoy Sukabumi - Cianjur

3.4.6 *Sequence Diagram*

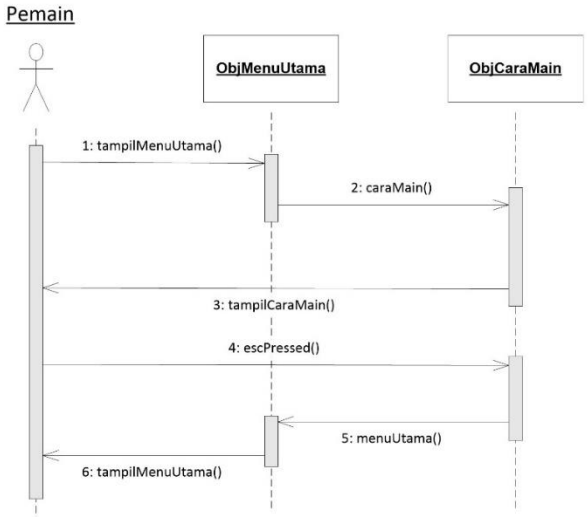
Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan masing-masing operasi.

3.4.6.1 *Sequence Diagram* Memilih Mulai



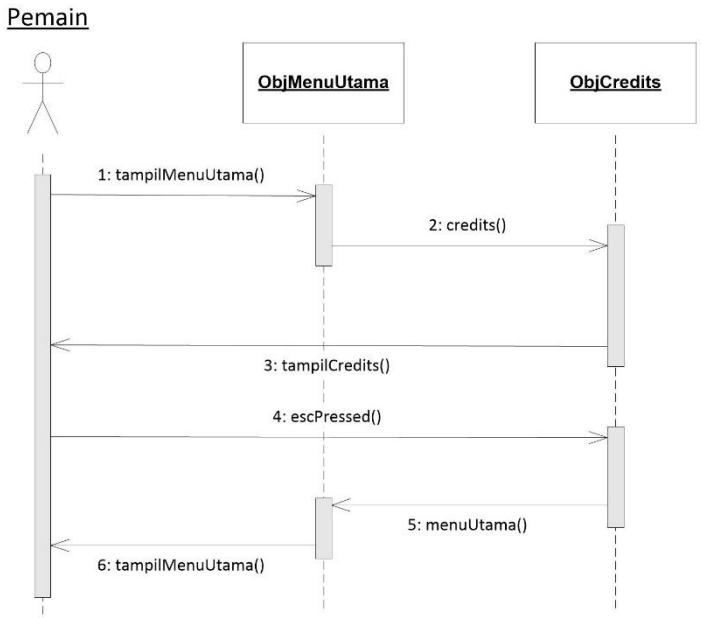
Gambar 3.37 *Sequence Diagram* Memilih Mulai

3.4.6.2 *Sequence Diagram* Tampil Cara Main



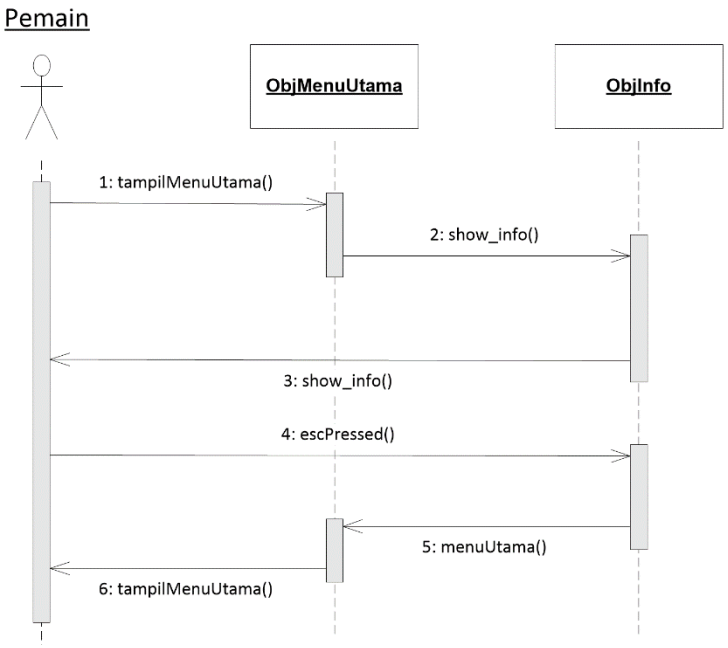
Gambar 3.38 *Sequence Diagram* Cara Main

3.4.6.3 *Sequence Diagram* Tampil Credits



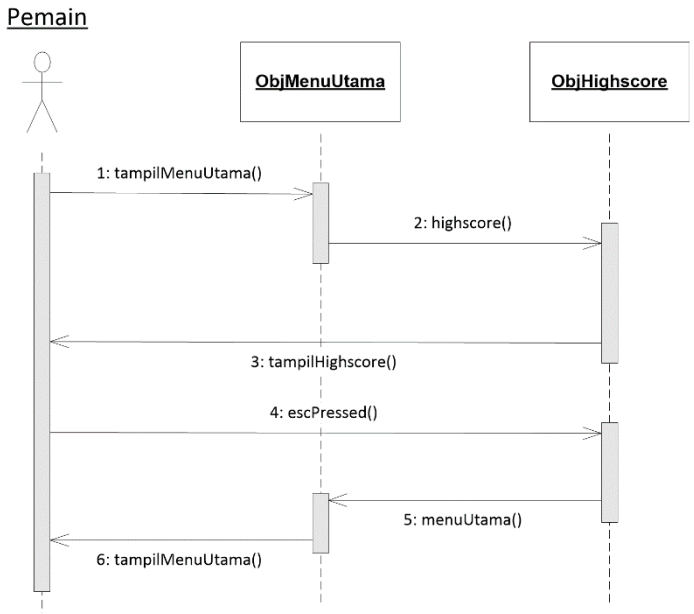
Gambar 3.39 *Sequence Diagram* Menampilkan Credits

3.4.6.4 *Sequence Diagram* Tampil Info



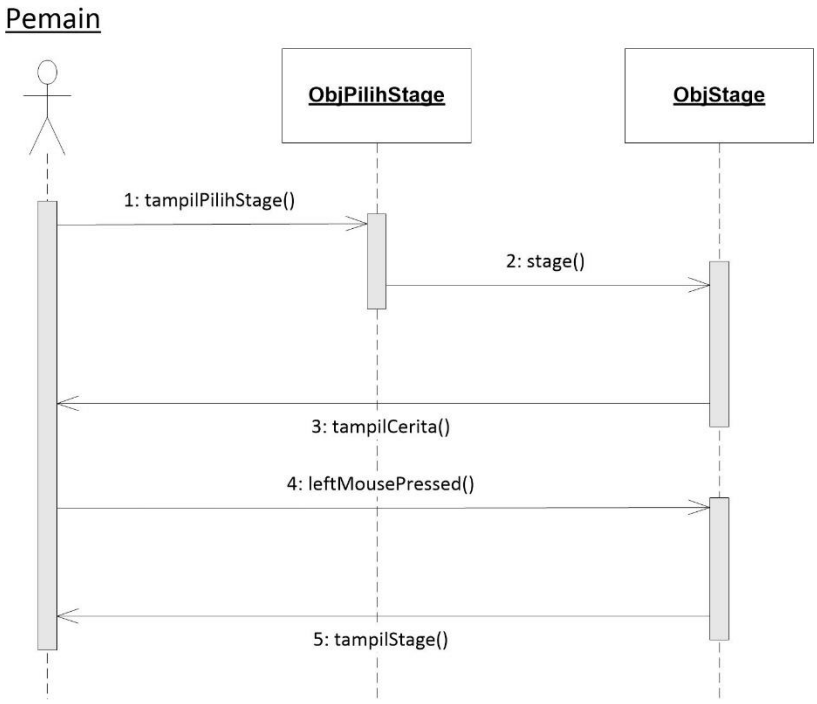
Gambar 3.40 *Sequence Diagram* Menampilkan Info

3.4.6.5 *Sequence Diagram* Tampil *Highscore*



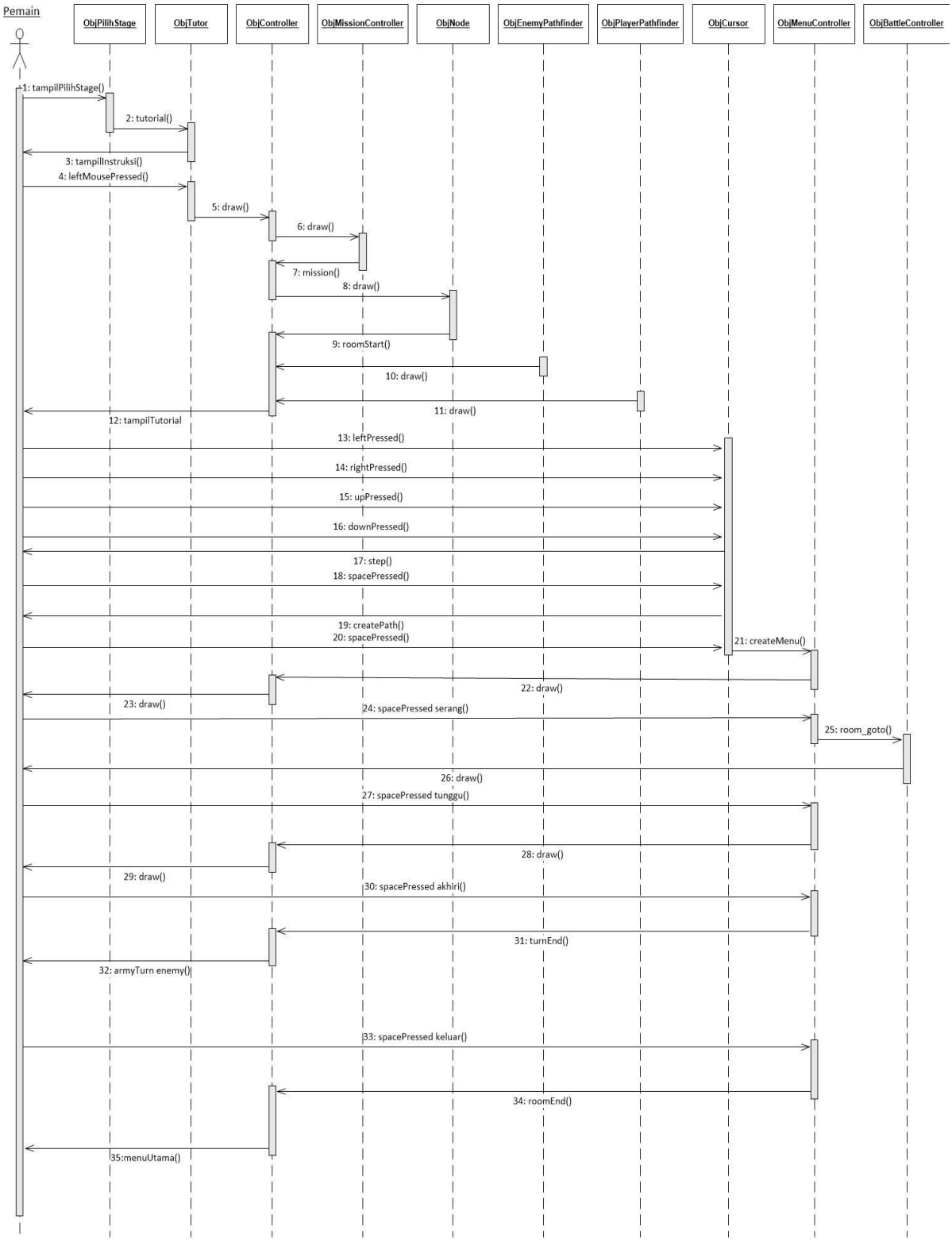
Gambar 3.41 *Sequence Diagram* Menampilkan *Highscore*

3.4.6.6 *Sequence Diagram* Pilih *Stage*



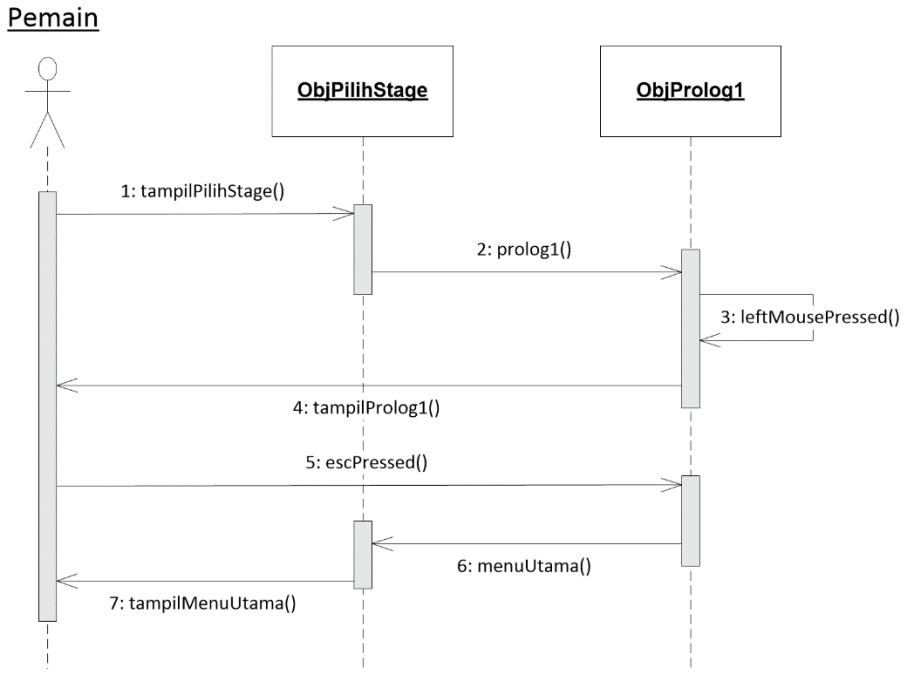
Gambar 3.42 *Sequence Diagram* Pilih *Stage*

3.4.6.7 *Sequence Diagram* Tampil Tutorial



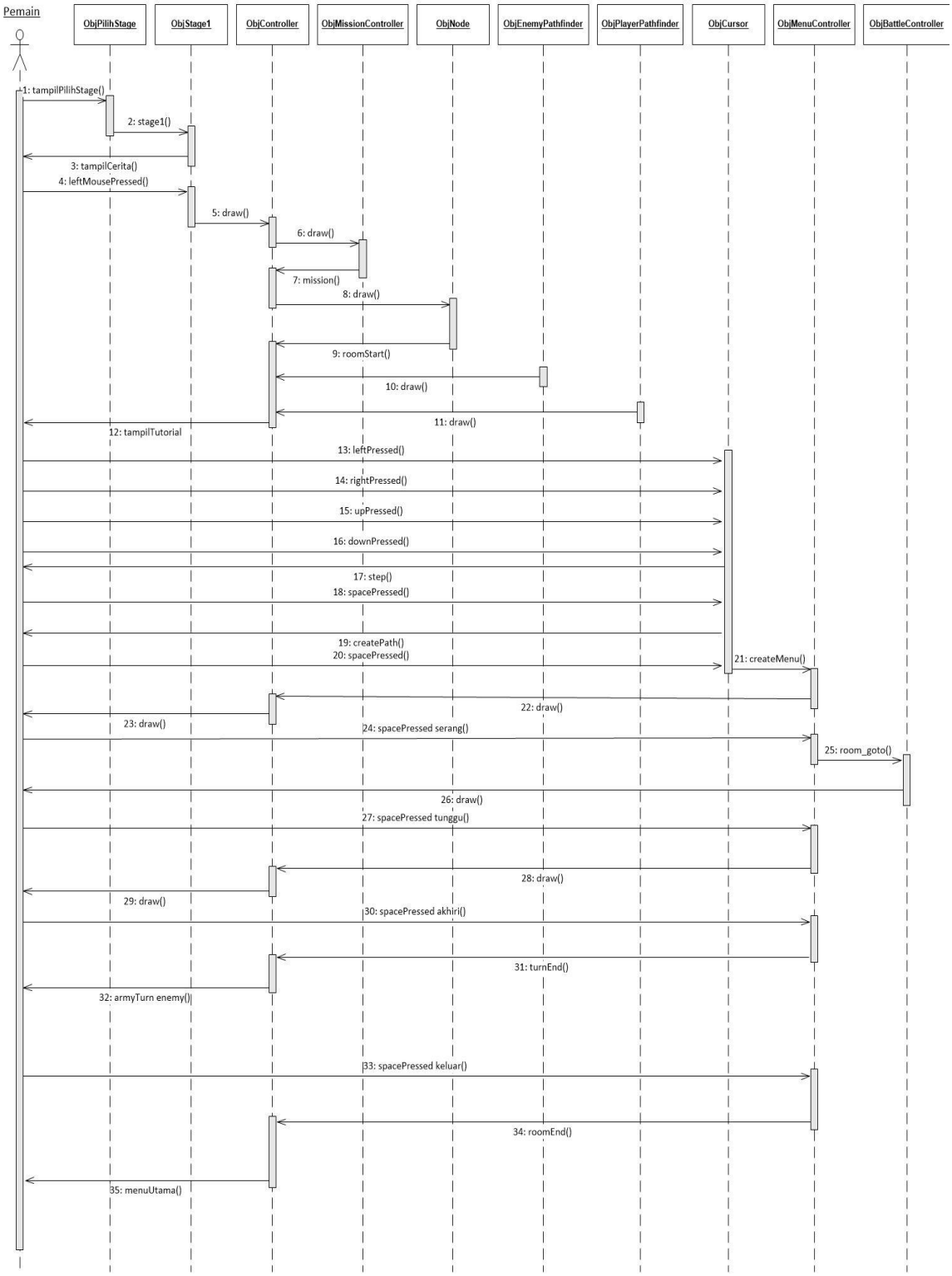
Gambar 3.43 *Sequence Diagram* Tampil Tutorial

3.4.6.8 *Sequence Diagram* Tampil Prolog1



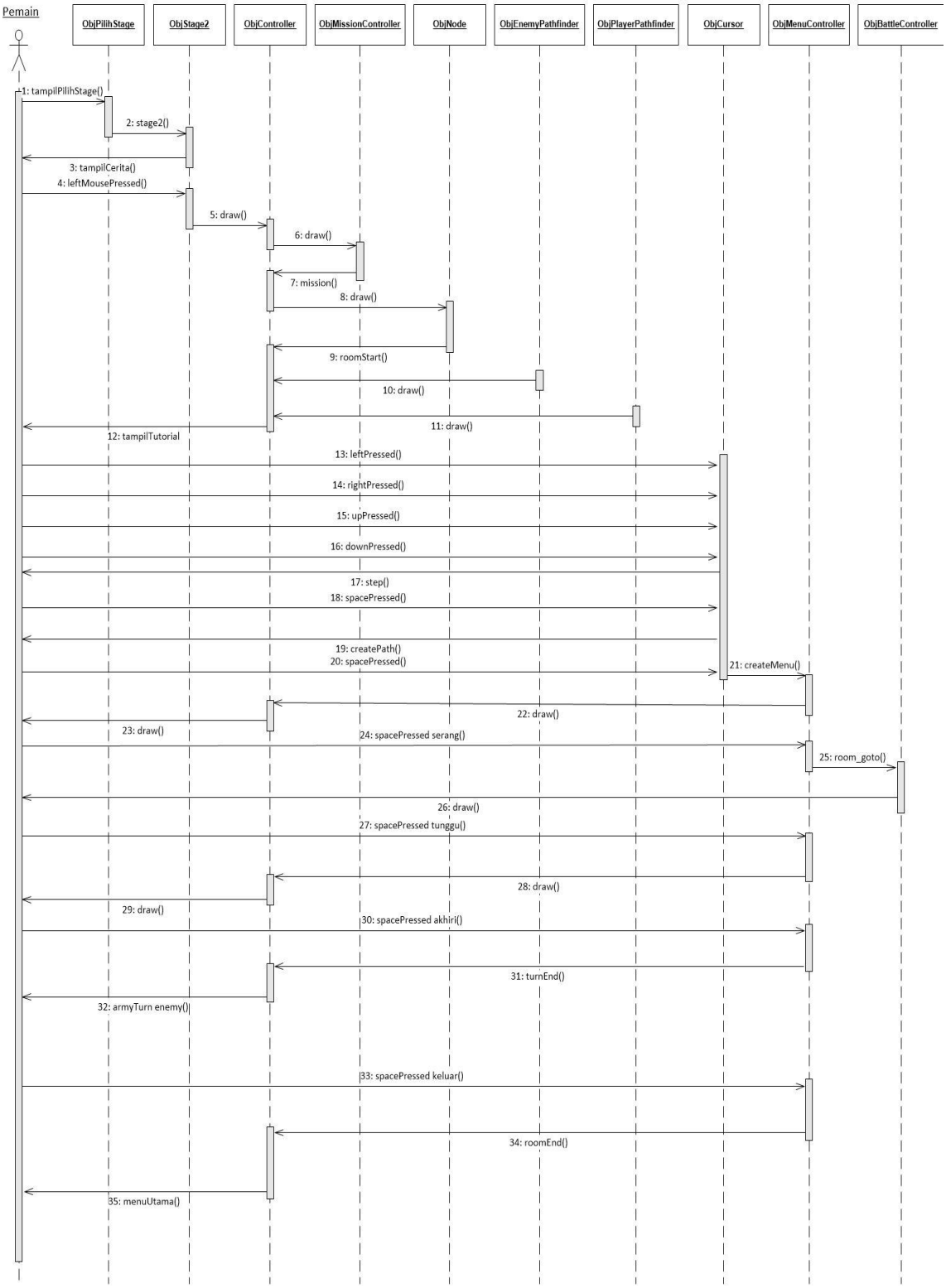
Gambar 3.44 Sequence Diagram Tampil Prolog1

3.4.6.9 *Sequence Diagram* Tampil *Stage1*



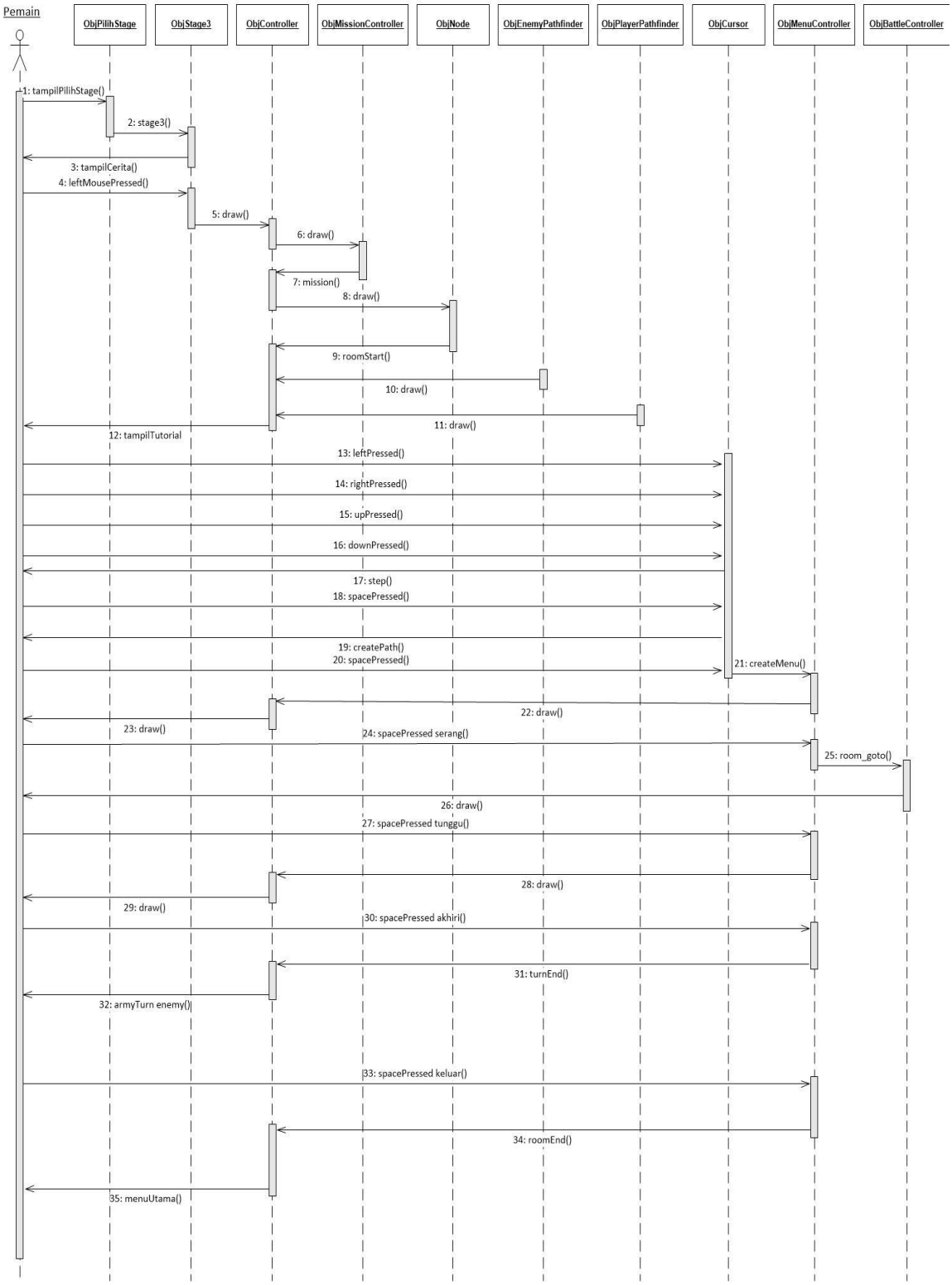
Gambar 3.45 *Sequence Diagram* Tampil *Stage1*

3.4.6.10 *Sequence Diagram* Tampil *Stage2*



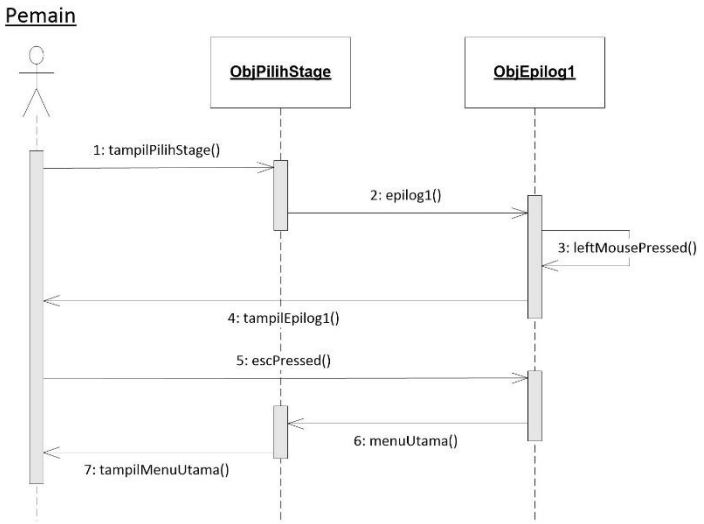
Gambar 3.46 *Sequence Diagram* Tampil *Stage2*

3.4.6.11 *Sequence Diagram* Tampil *Stage3*



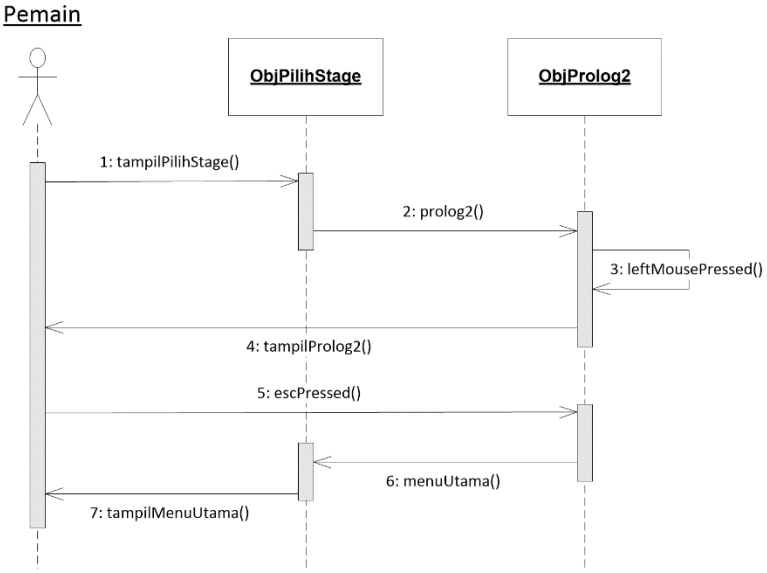
Gambar 3.47 *Sequence Diagram* Tampil *Stage3*

3.4.6.12 *Sequence Diagram* Tampil Epilog1



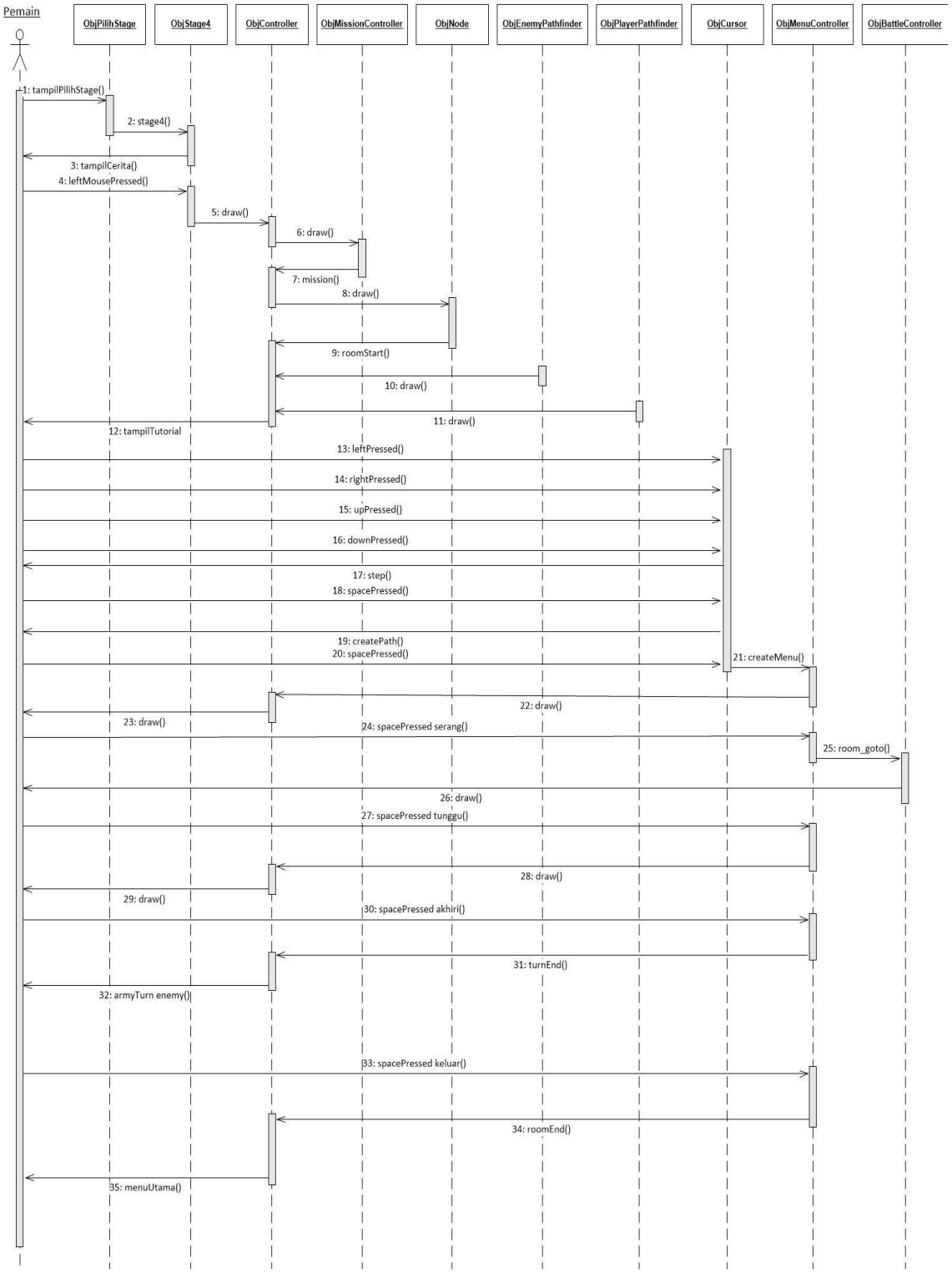
Gambar 3.48 Sequence Diagram Tampil Epilog1

3.4.6.13 *Sequence Diagram* Tampil Prolog2



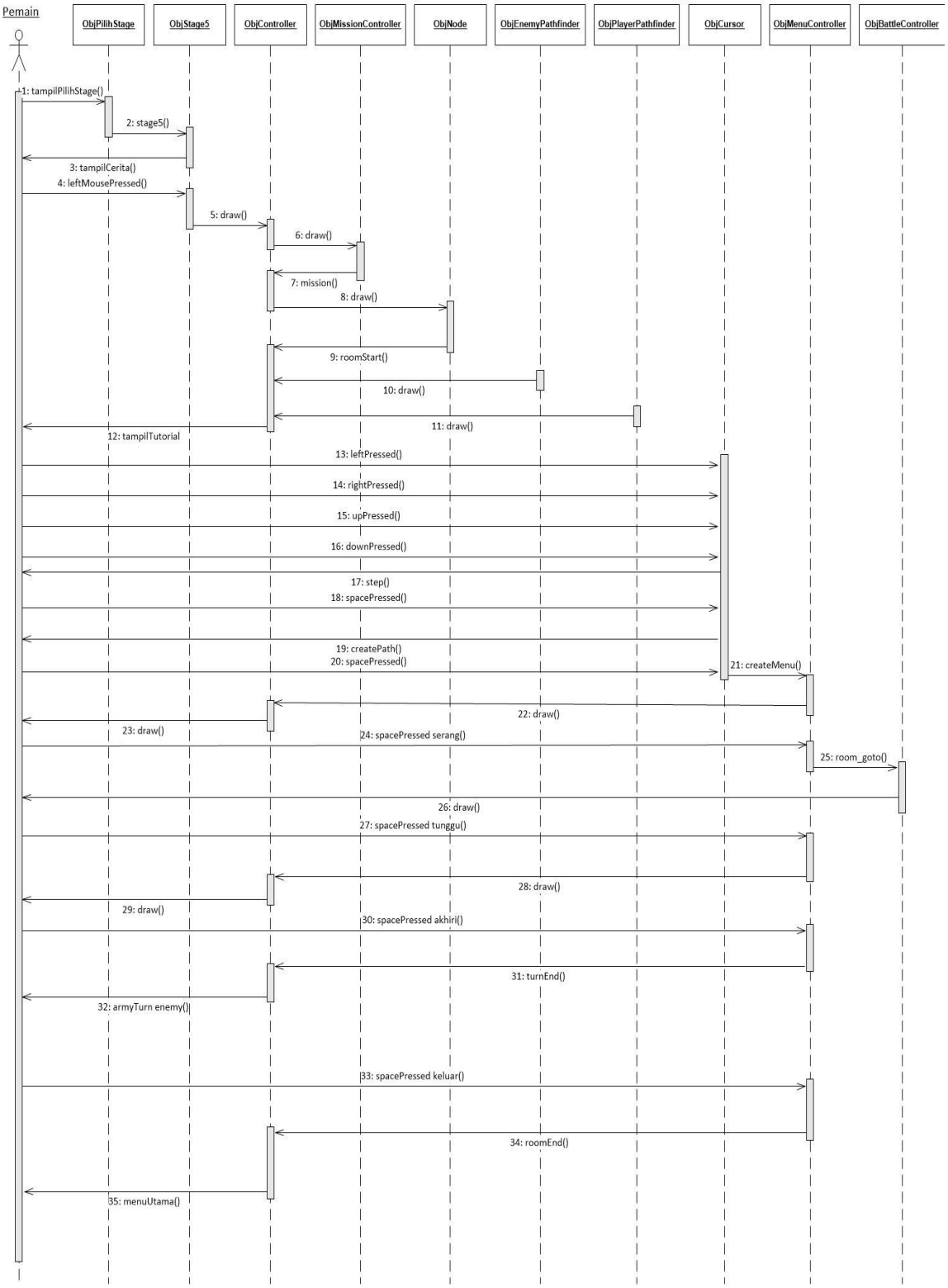
Gambar 3.49 Sequence Diagram Tampil Prolog2

3.4.6.14 *Sequence Diagram* Tanpil *Stage4*



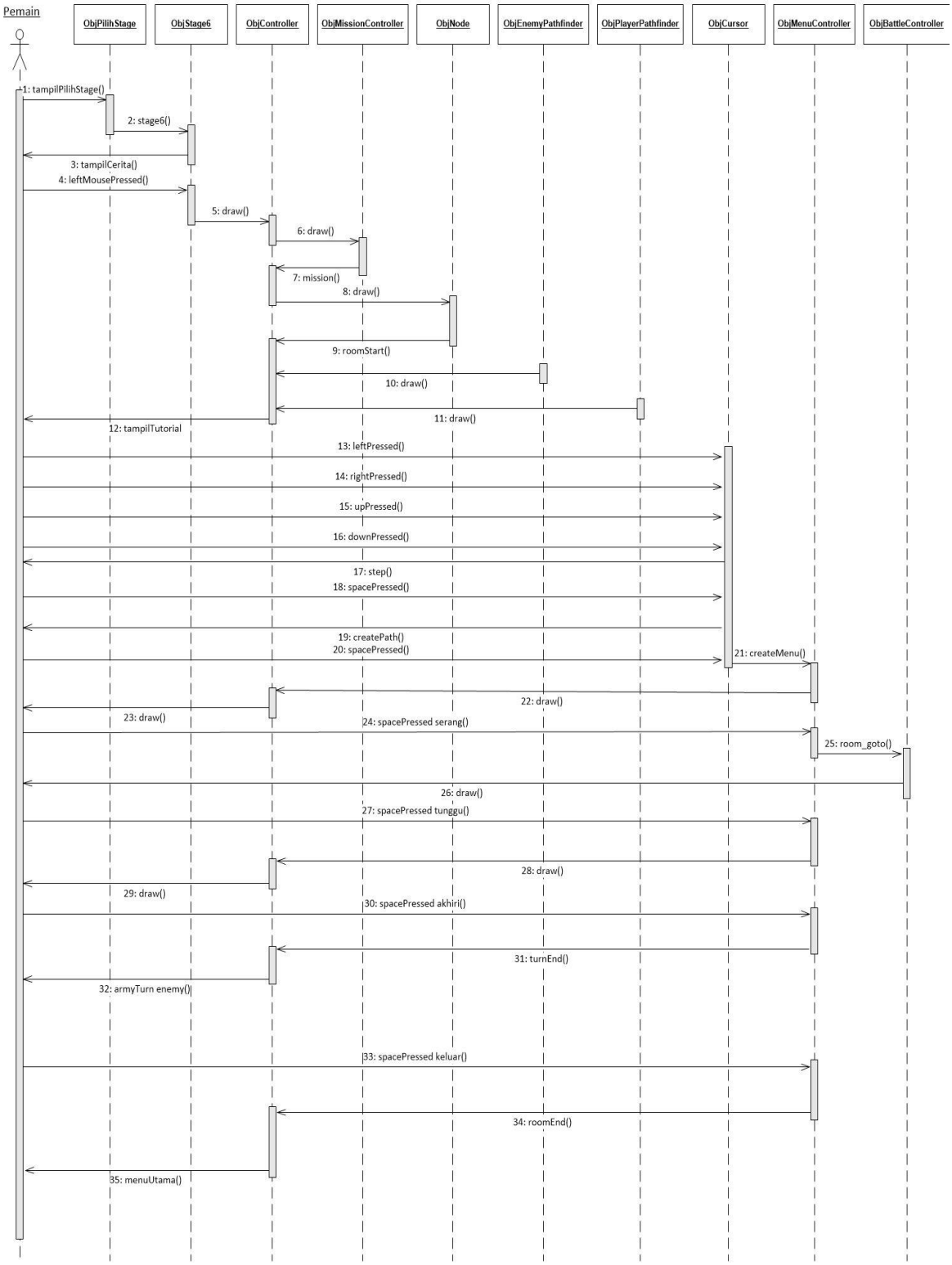
Gambar 3.50 *Sequence Diagram* Tampil *Stage4*

3.4.6.15 *Sequence Diagram* Tanpil *Stage5*



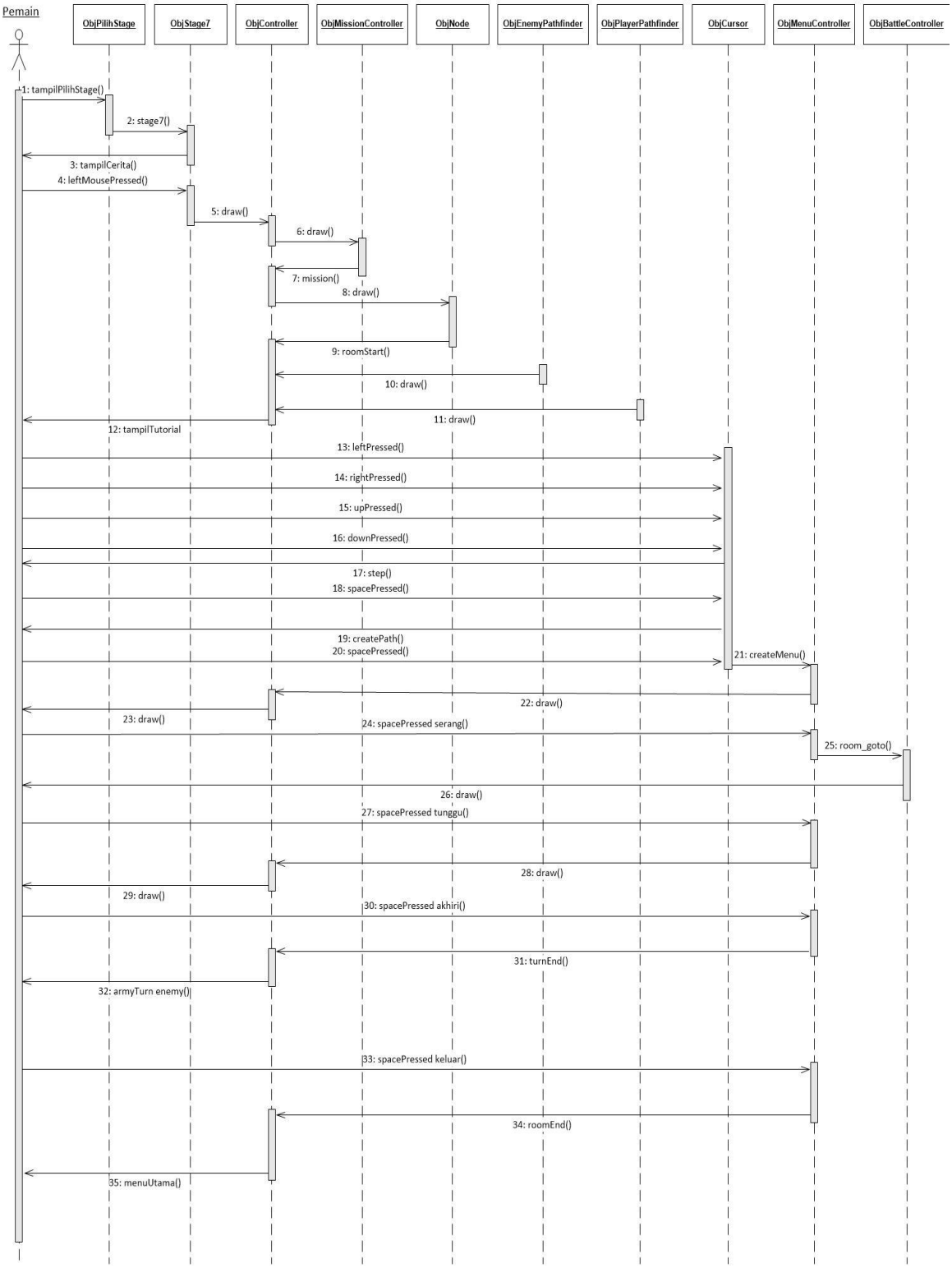
Gambar 3.51 *Sequence Diagram* Tampil *Stage5*

3.4.6.16 *Sequence Diagram* Tanpil *Stage6*



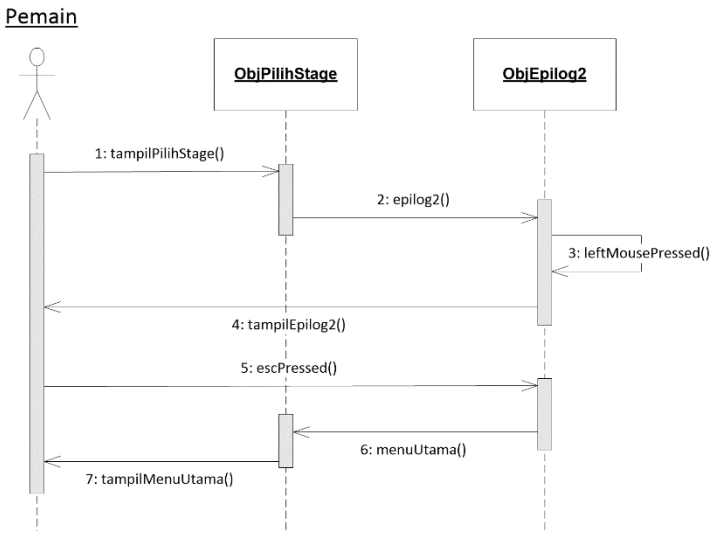
Gambar 3.52 *Sequence Diagram* Tampil *Stage6*

3.4.6.17 *Sequence Diagram* Tanpil *Stage7*



Gambar 3.53 *Sequence Diagram* Tampil *Stage7*

3.4.6.13 *Sequence Diagram* Tampil Epilog2



Gambar 3.54 Sequence Diagram Tampil Epilog2

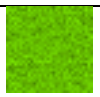
3.5 Perancangan

3.5.1 Perancangan Objek Pasukan dan *Terrain*

Pada *game* Pertempuran Konvoy Sukabumi – Cianjur terdapat beberapa objek yang berperan dalam permainannya. Berikut ini adalah daftar objek yang terdapat dalam *game* Pertempuran Konvoy Sukabumi – Cianjur:

Tabel 3.20 Objek-objek pada *game* "Pertempuran Konvoy Sukabumi – Cianjur"

No	Nama	Gambar	Keterangan
1	Infantri TKR		Pergerakan: 4 Maks. *range* serangan: 2 Serangan: 25 Pertahanan: 10
2	Pasukan Bom TKR		Pergerakan: 3 Maks. range serangan: 2 Serangan: 35 Pertahanan: 10

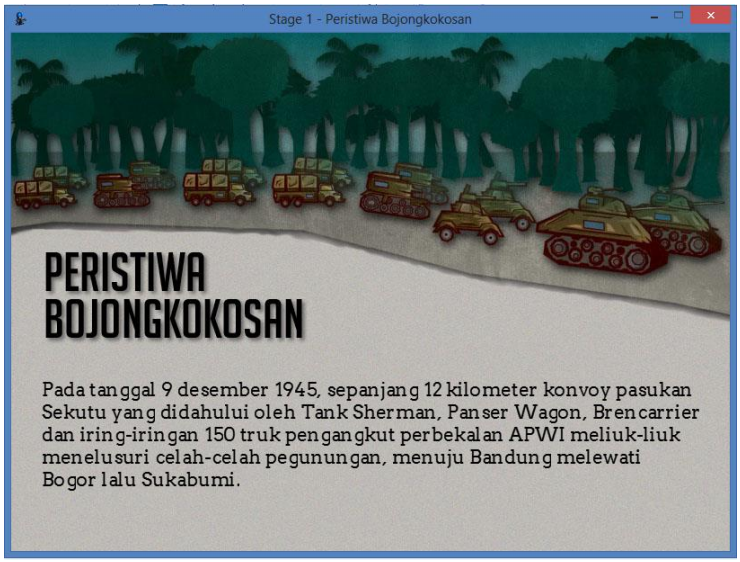
No	Nama	Gambar	Keterangan
3	Sniper TKR		Pergerakan: 4 Maks. range serangan: 3 Serangan: 30 Pertahanan: 10
4	Infantri Sekutu		Pergerakan: 4 Maks. range serangan: 2 Serangan: 25 Pertahanan: 10
5	Truk Sekutu		Pergerakan: 5 Maks. range serangan: 1 Serangan: 10 Pertahanan: 15
6	Brencarrier Sekutu		Pergerakan: 4 Maks. range serangan: 2 Serangan: 30 Pertahanan: 15
7	Tank Sherman Sekutu		Pergerakan: 3 Maks. range serangan: 2 Serangan: 45 Pertahanan: 25
8	Panser Wagon Sekutu		Pergerakan: 5 Maks. range serangan: 2 Serangan: 30 Pertahanan: 15
9	Rumput		Bisa dilewati: Ya Bonus serangan: 0 Bonus pertahanan: 0 Penalti: 0
10	Hutan/pohon		Bisa dilewati: Ya, hanya

No	Nama	Gambar	Keterangan
			oleh infantri Bonus serangan: 0 Bonus pertahanan: 5 Penalti: 1
11	Jalan Raya		Bisa dilewati: Ya Bonus serangan: 0 Bonus pertahanan: 0 Penalti: 0
12	Gunung		Bisa dilewati: Ya, hanya oleh infantri Bonus serangan: 0 Bonus pertahanan: 10 Penalti: 2
13	Rumah		Bisa dilewati: Ya Bonus serangan: 2 Bonus pertahanan: 3 Penalti: 1
14	Sungai		Bisa dilewati: Tidak Bonus serangan: 0 Bonus pertahanan: 0 Penalti: 0

3.5.2 *Storyboard*

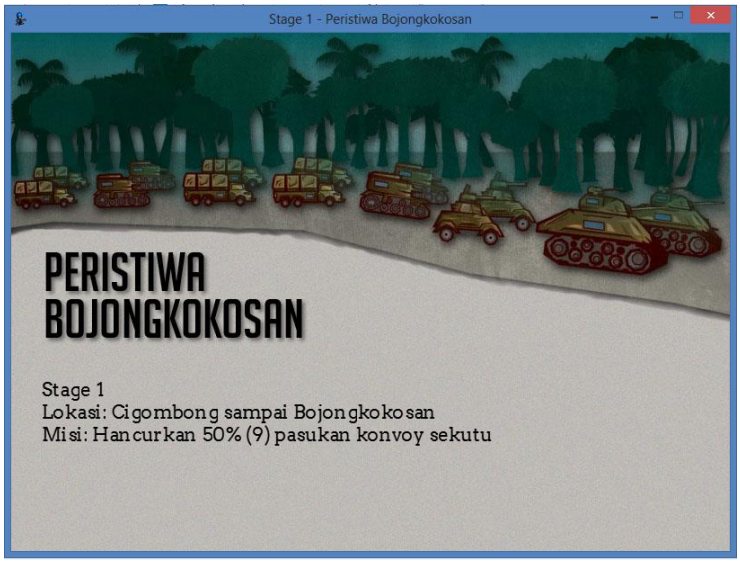
Storyboard merupakan sketsa gambar yang disusun berurutan sesuai dengan naskah yang ada, *storyboard* digunakan untuk menyampaikan ide cerita dari *game* yang akan dibangun kepada orang lin demham mudah. Berikut ini merupakan *template storyboard* dari *game* Pertempuran Konvoy Sukabumi – Cianjur:

1. Menampilkan kisah di setiap awal *stage*



Gambar 3.55 Kisah awal *stage*

2. Menampilkan informasi misi *stage* yang akan dimainkan



Gambar 3.56 Informasi misi *stage*

3. Menampilkan dialog tokoh



Gambar 3.57 Dialog tokoh

4. Menampilkan *map* dari *stage* yang akan dimainkan



Gambar 3.58 *Map stage*

5. Menampilkan pertanyaan ketika *stage* diselesaikan



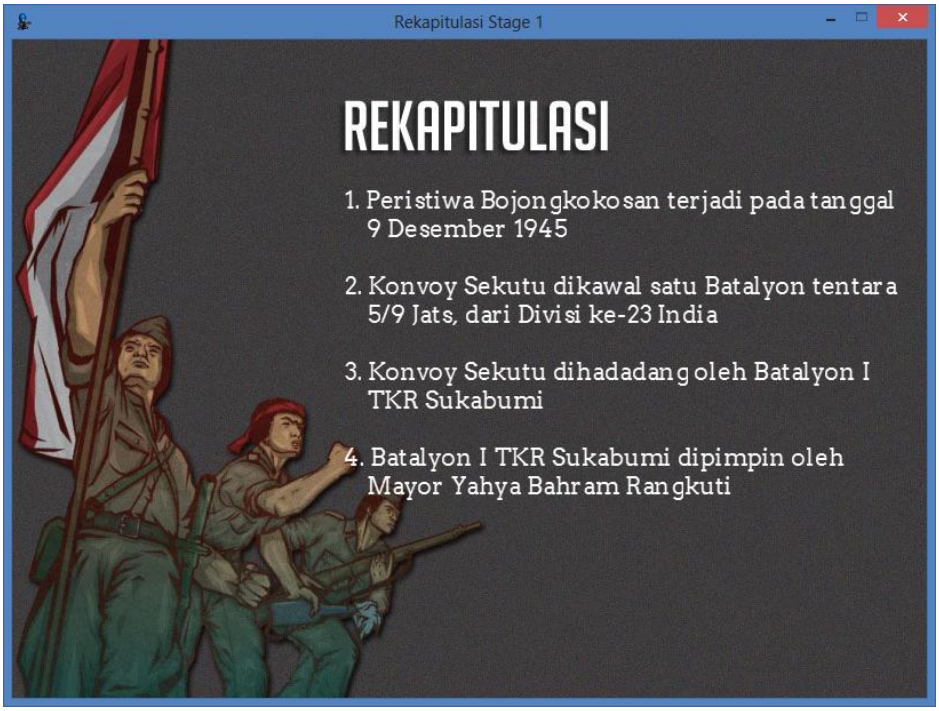
Gambar 3.59 Pertanyaan Bonus

6. Menampilkan isian nama jika *score* pemain merupakan *highscore* baru



Gambar 3.60 Isi Nama

7. Menampilkan rekapitulasi kisah *stage*



Gambar 3.61 Rekapitulasi *Stage*

3.5.3 Perancangan *Map* dan Misi Pemain

Map merupakan tempat *game* dimainkan. Setiap *stage* memiliki *map* dan misi yang berbeda-beda yang disesuaikan dengan jalan cerita yang disampaikan pada awal permainan. Berikut ini merupakan perancangan *map* dan misi dari setiap *stage* dalam *game* Pertempuran Konvoy Sukabumi – Cianjur.

1. *Map Stage* 1



Gambar 3.62 *Map Stage* 1

2. *Map Stage* 2



Gambar 3.63 *Map Stage* 2

3. *Map Stage* 3



Gambar 3.64 Map Stage 3

4. *Map Stage* 4



Gambar 3.65 *Map Stage* 4

5. *Map Stage* 5



Gambar 3.66 Map Stage 5

6. *Map Stage* 6



Gambar 3.67 Map Stage 6

7. *Map Stage* 7



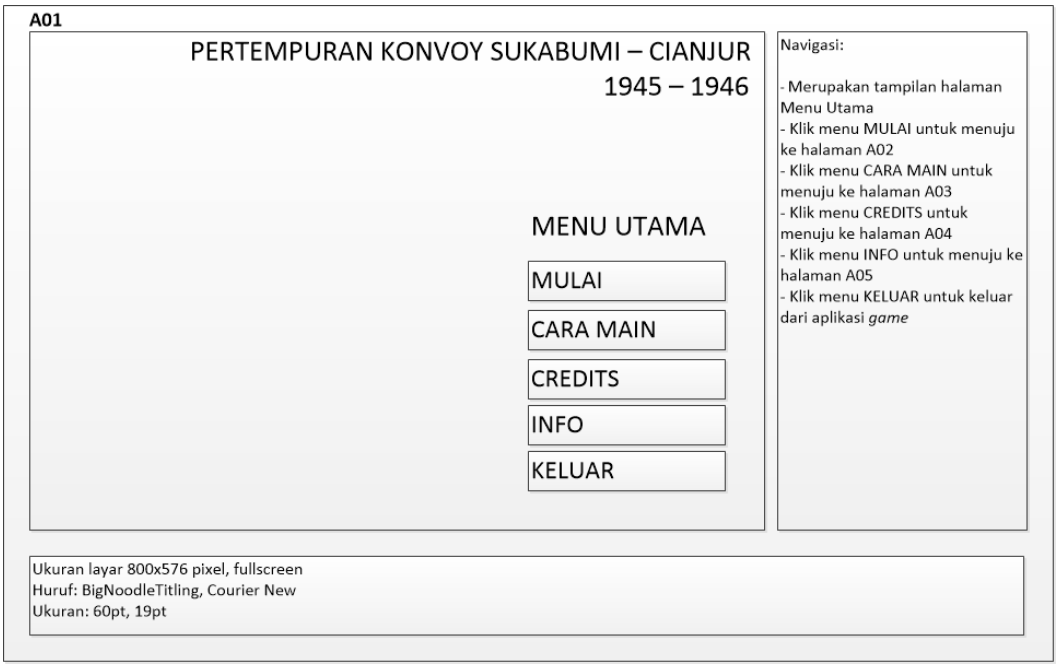
Gambar 3.68 Map Stage 7

Misi setiap *stage* disesuaikan dengan kisah pertempuran konvoy Sukabumi – Cianjur. Adapun misi dari setiap *stage* dalam game Pertempuran Konvoy Sukabumi – Cianjur adalah sebagai berikut:

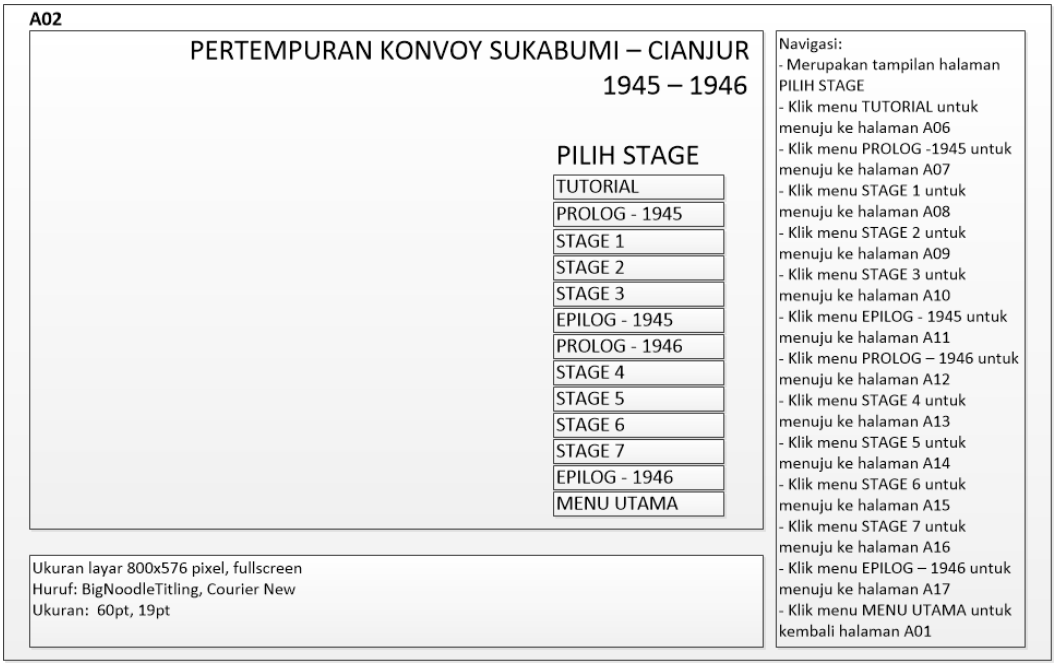
1. Misi *Stage* 1: Kalahkan 50% (9) pasukan musuh
2. Misi *Stage* 2: Kalahkan pasukan musuh sampai hanya tersisa 5 pasukan
3. Misi *Stage* 3: Kalahkan 80% (19) pasukan musuh
4. Misi *Stage* 4: Kalahkan 80% (28) pasukan musuh
5. Misi *Stage* 5: Kalahkan 80% (26) pasukan musuh
6. Misi *Stage* 6: Kalahkan 80% (27) pasukan musuh
7. Misi *Stage* 7: Kalahkan pasukan musuh sampai hanya tersisa 8 pasukan

3.5.4 Perancangan Antarmuka

Perancangan antarmuka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi *game* yang akan dibangun. Berikut ini disajikan perancangan antarmuka dari aplikasi *game* yang akan dibangun:



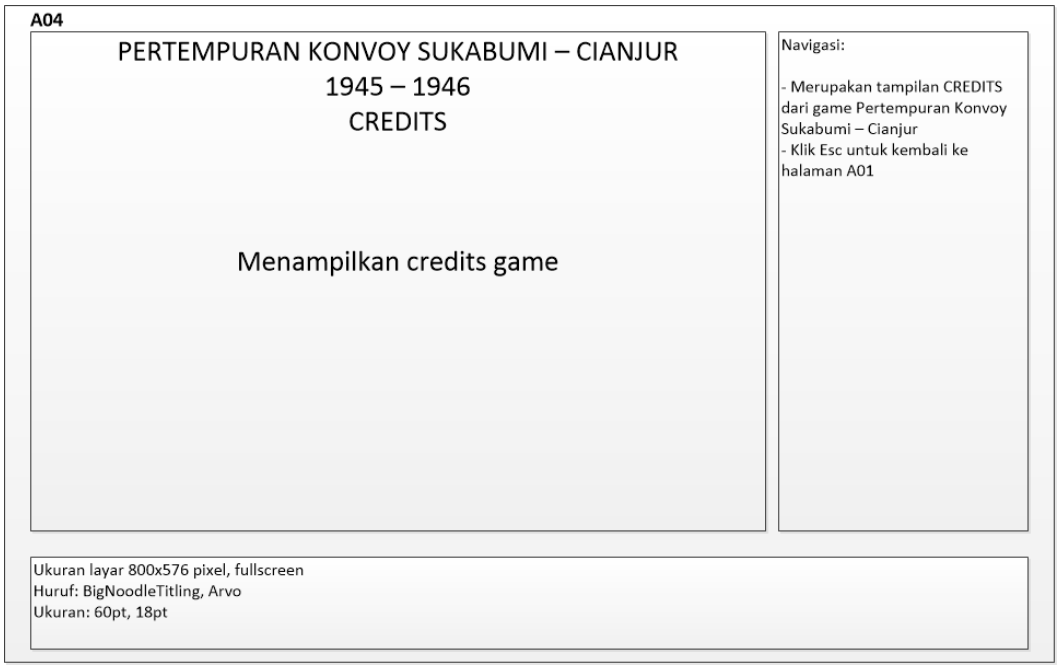
Gambar 3.69 Antarmuka halaman Menu Utama (A01)



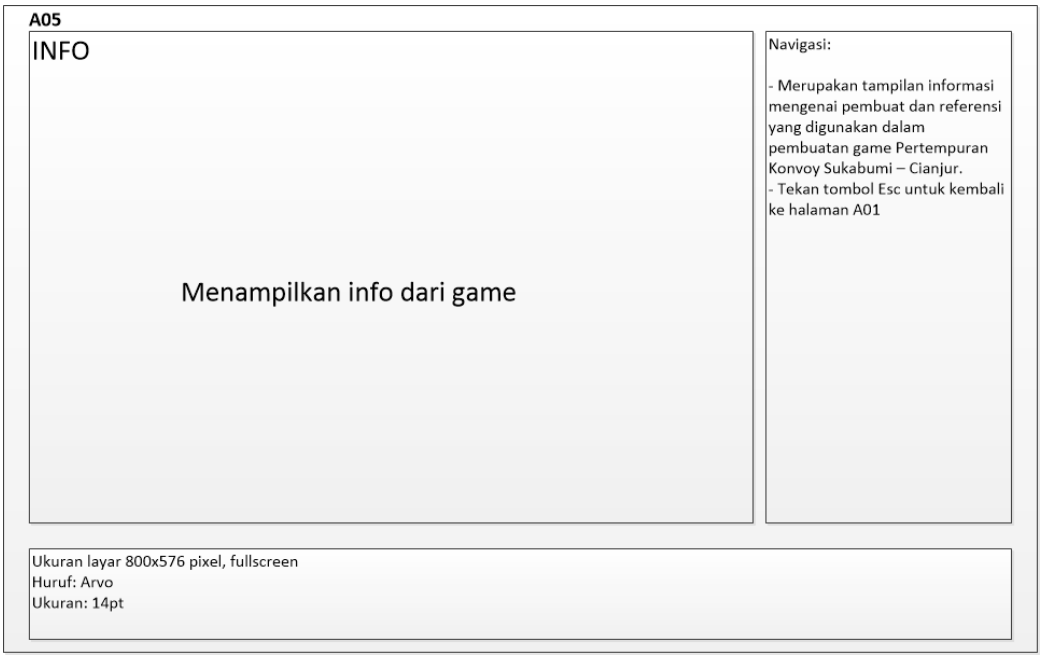
Gambar 3.70 Antarmuka halaman Pilih Stage (A02)



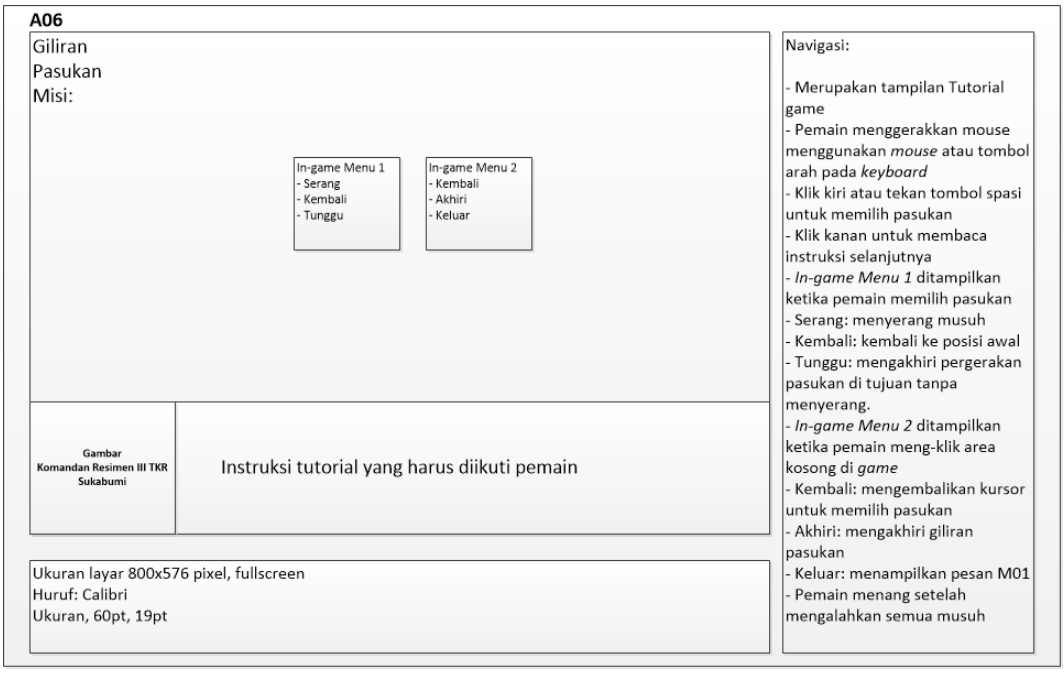
Gambar 3.71 Antarmuka halaman Cara Main (A03)



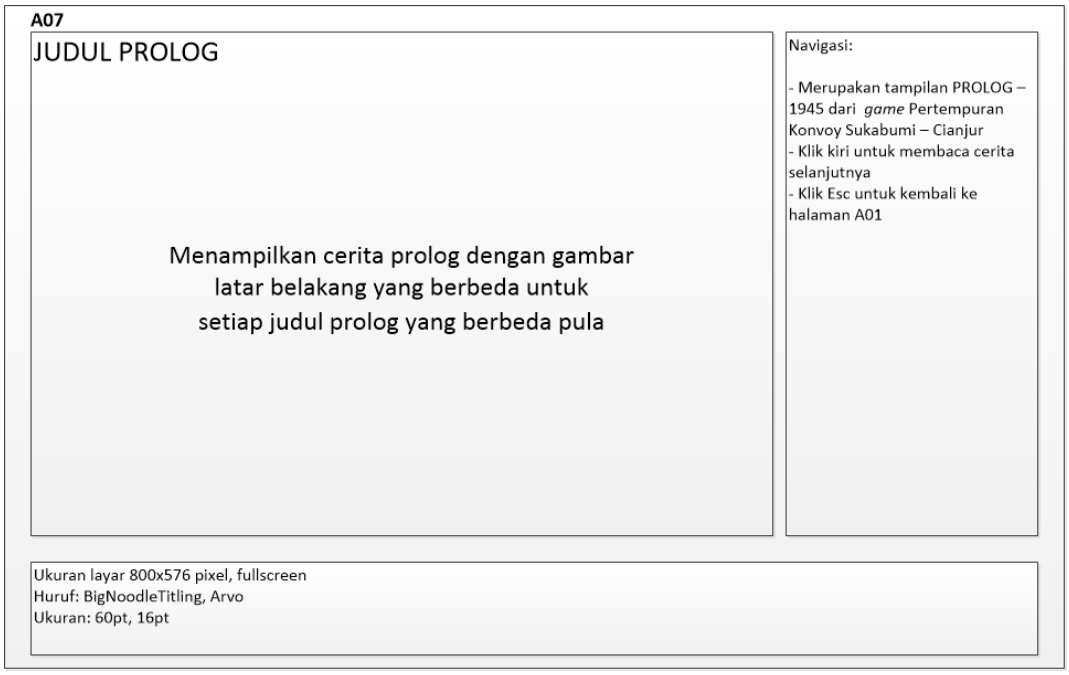
Gambar 3.72 Antarmuka halaman Credits (A04)



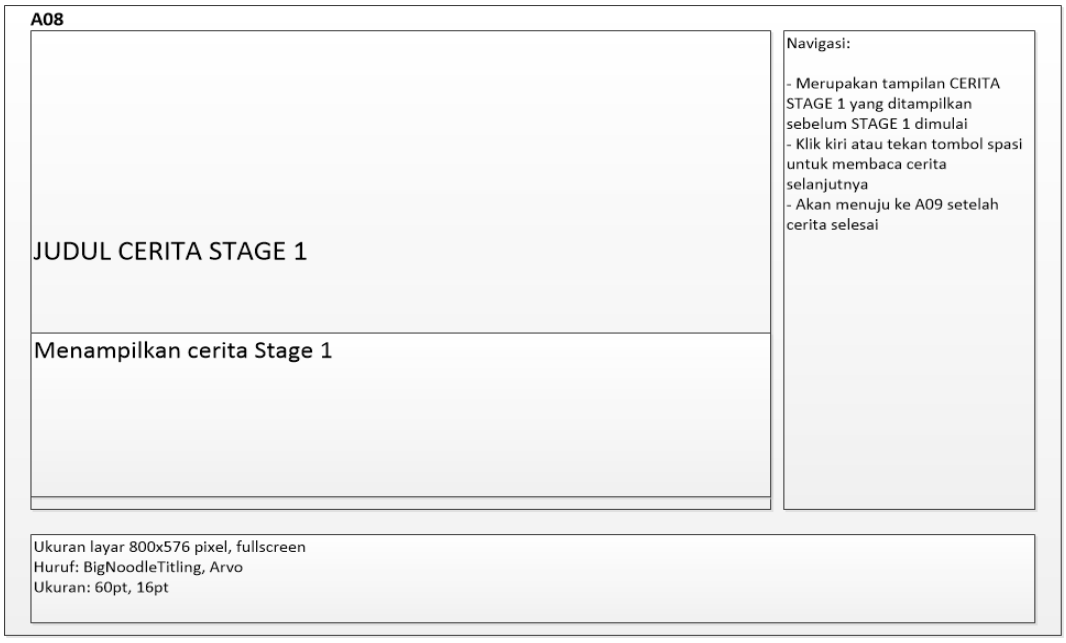
Gambar 3.73 Antarmuka halaman Info (A05)



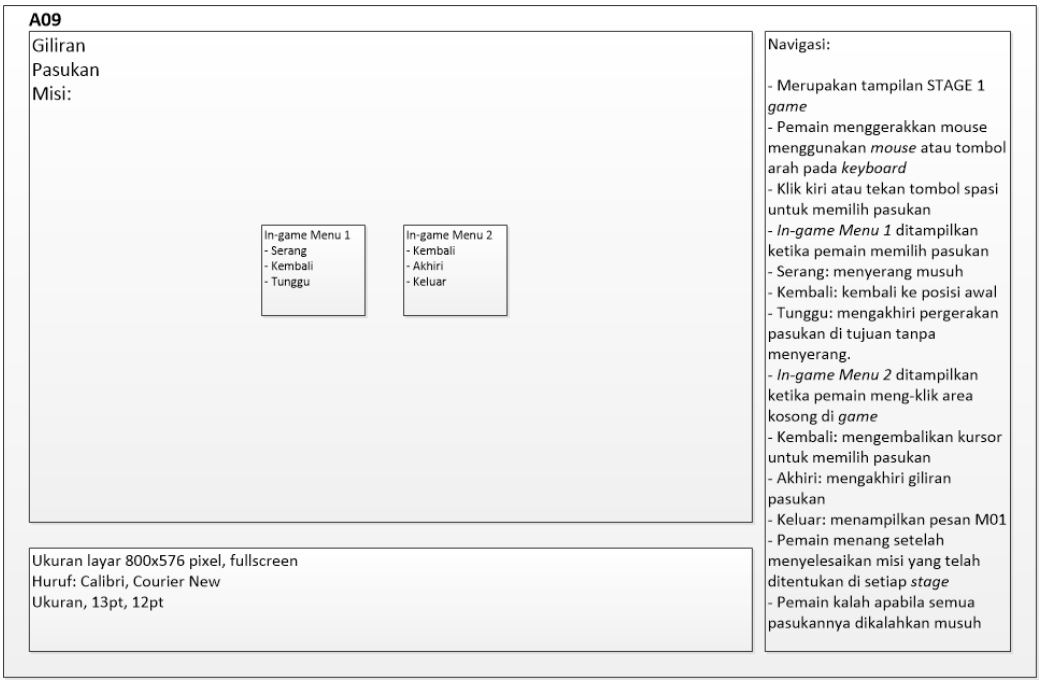
Gambar 3.74 Antarmuka halaman Tutorial (A06)



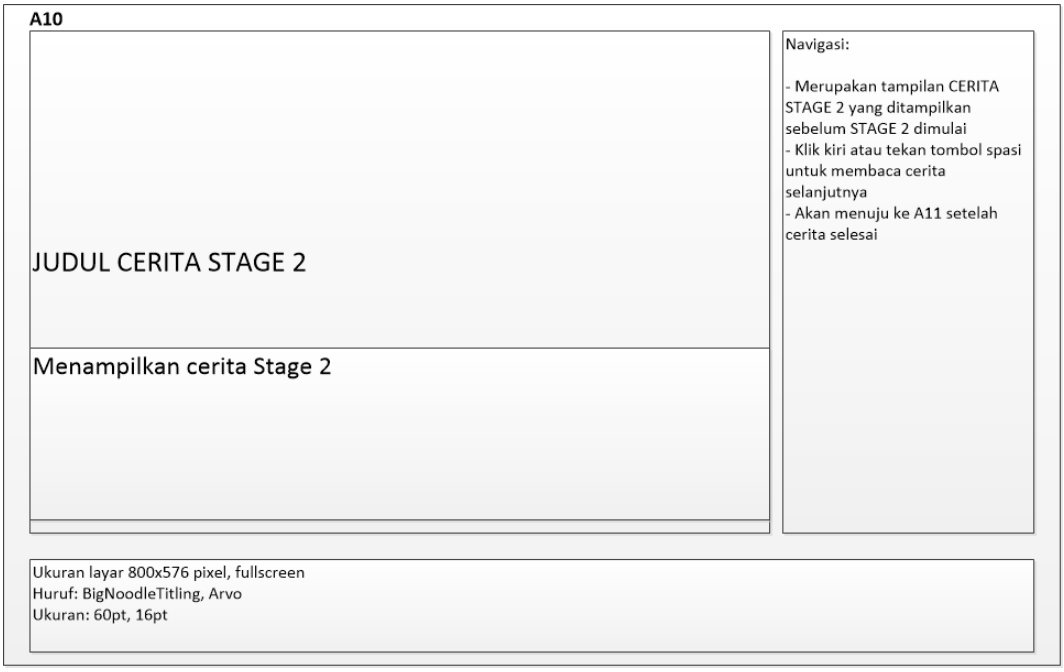
Gambar 3.75 Antarmuka halaman Prolog - 1945 (A07)



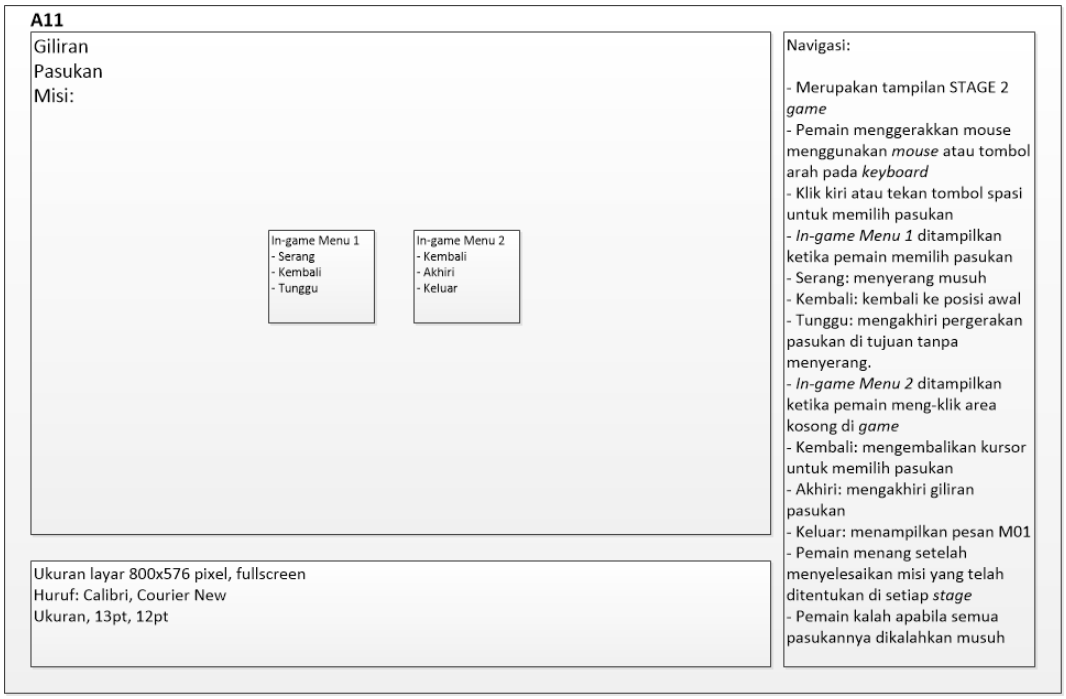
Gambar 3.76 Antarmuka halaman Cerita *Stage* 1 (A08)



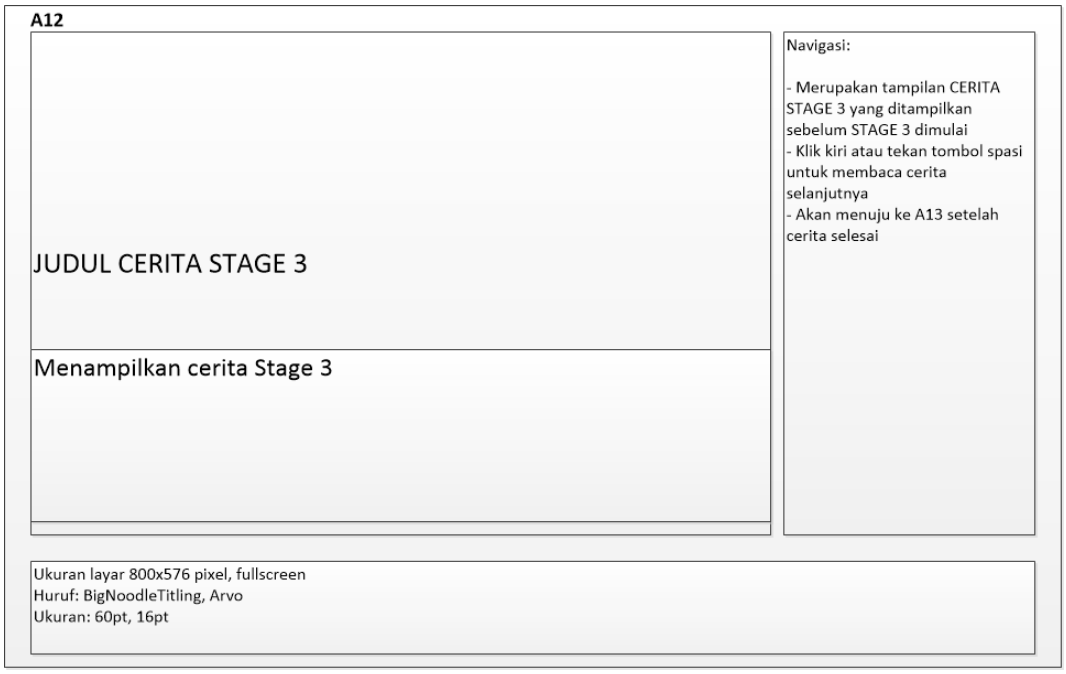
Gambar 3.77 Antarmuka halaman *Stage* 1 (A09)



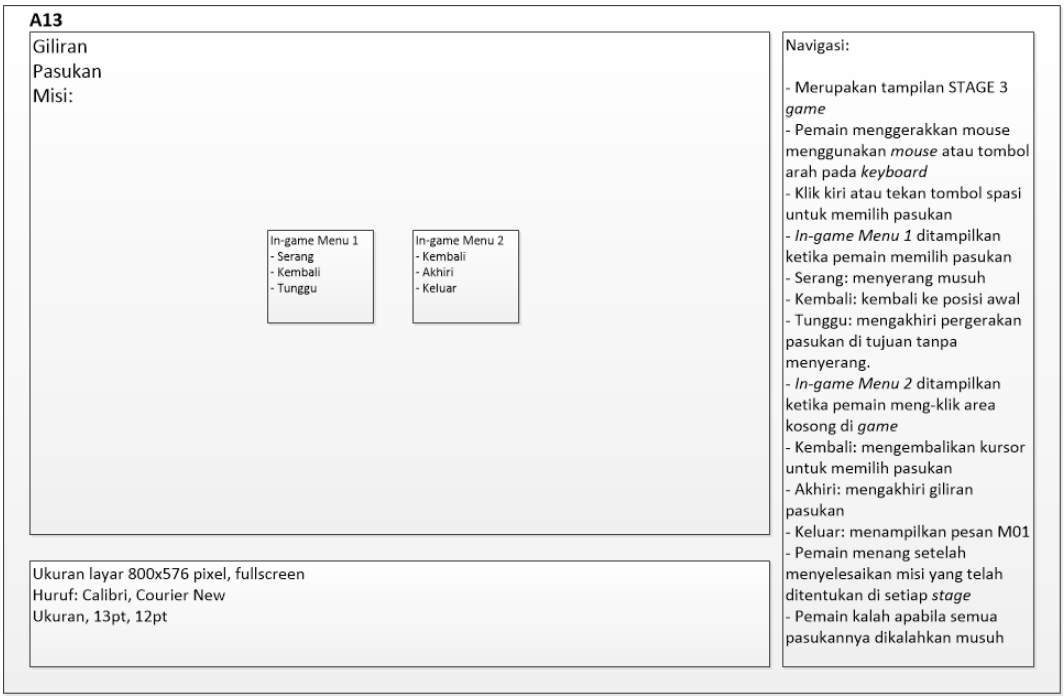
Gambar 3.78 Antarmuka halaman Cerita *Stage* 2 (A10)



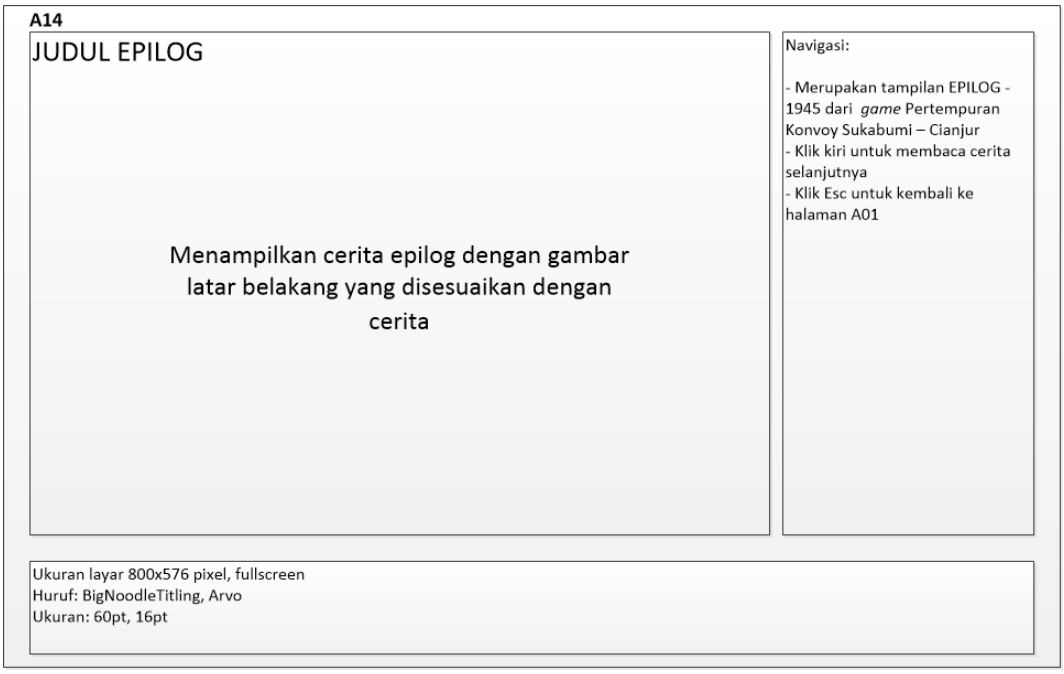
Gambar 3.79 Antarmuka halaman *Stage* 2 (A11)



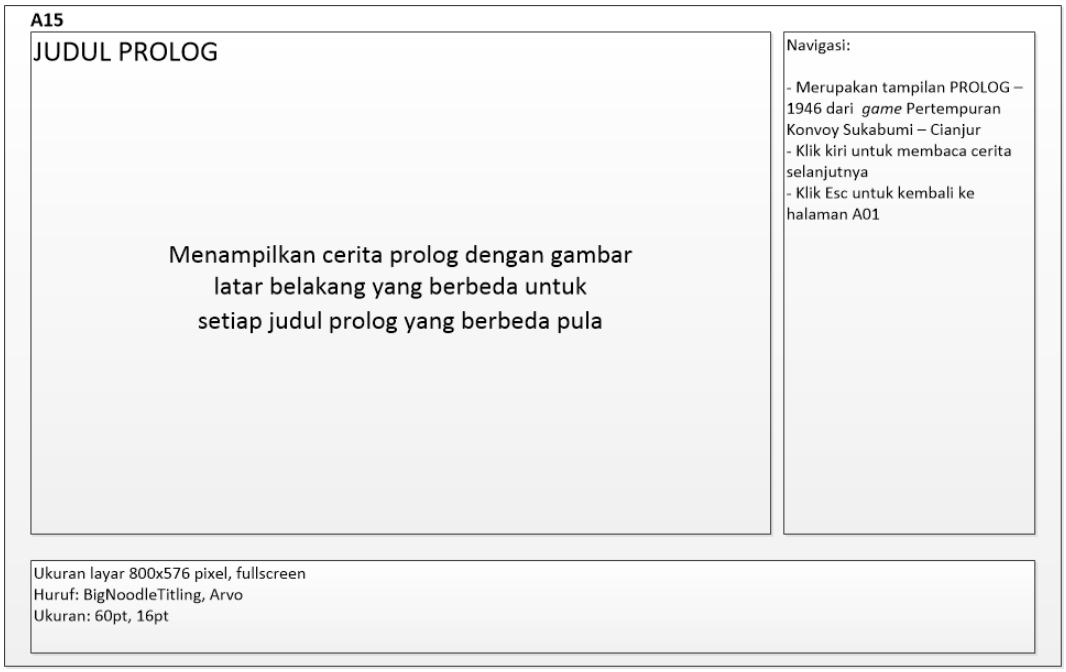
Gambar 3.80 Antarmuka halaman Cerita *Stage* 3 (A12)



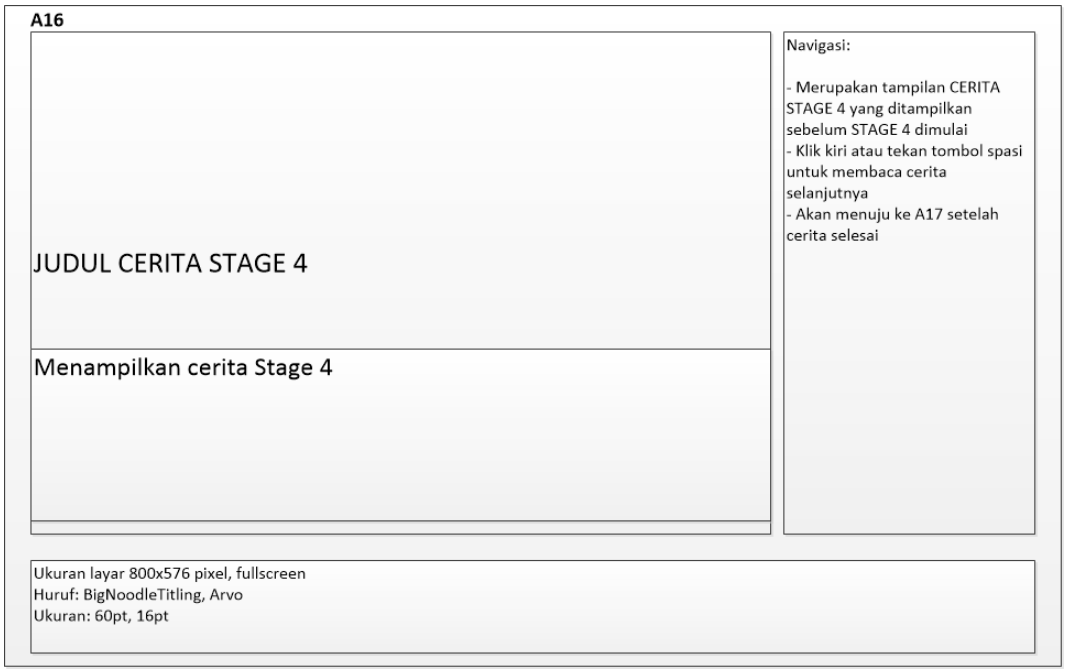
Gambar 3.81 Antarmuka halaman *Stage* 3 (A13)



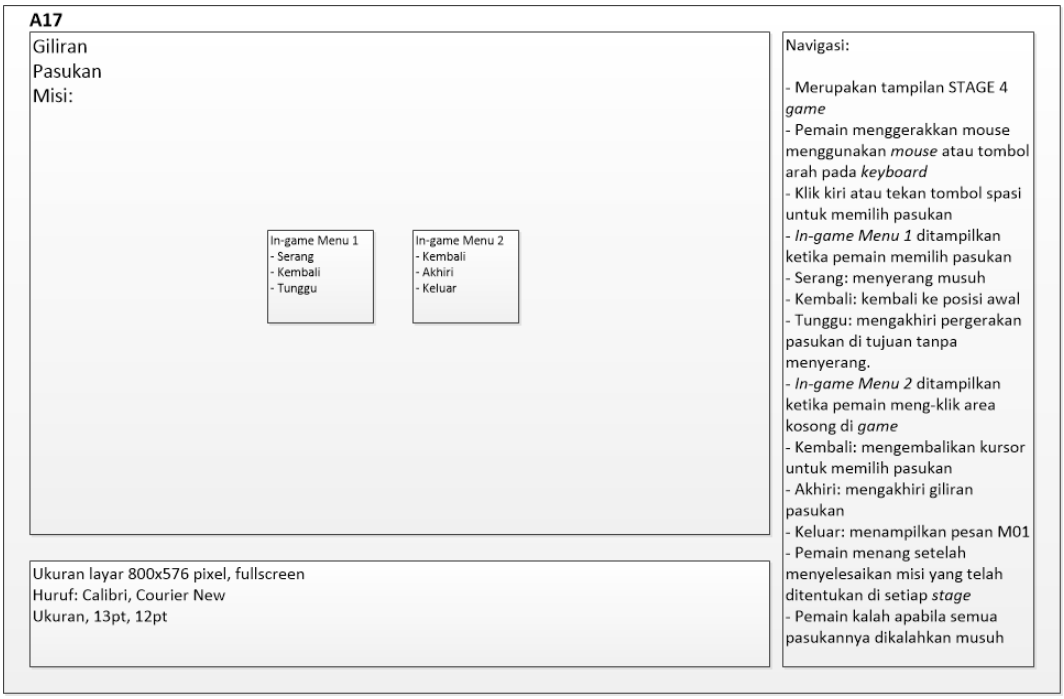
Gambar 3.82 Antarmuka halaman Epilog - 1945 (A14)



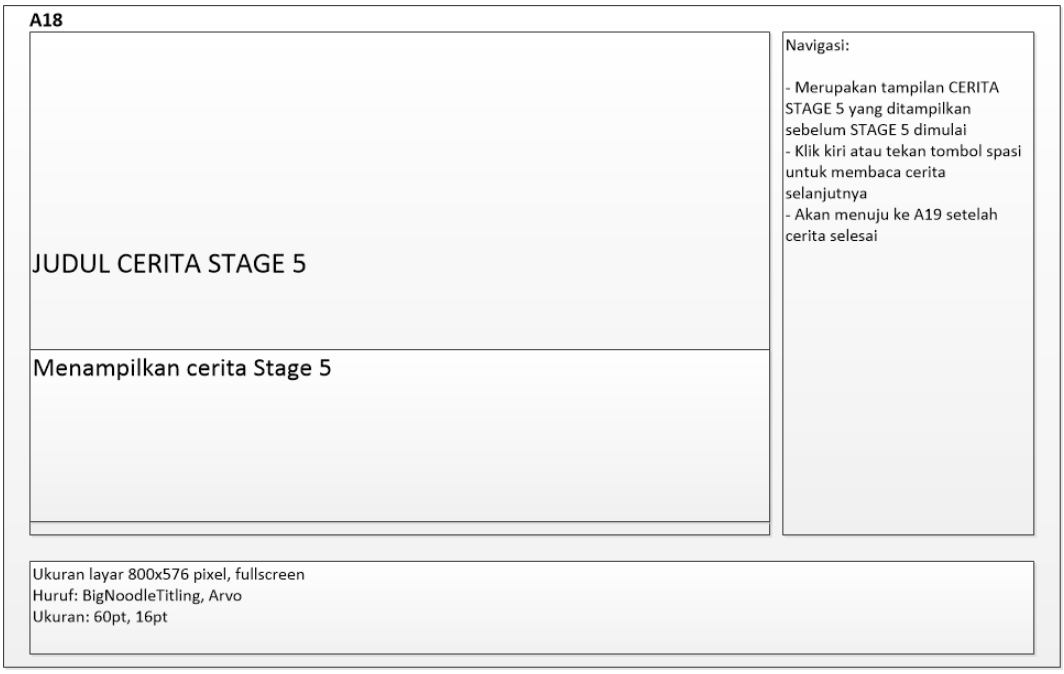
Gambar 3.83 Antarmuka halaman Prolog - 1946 (A15)



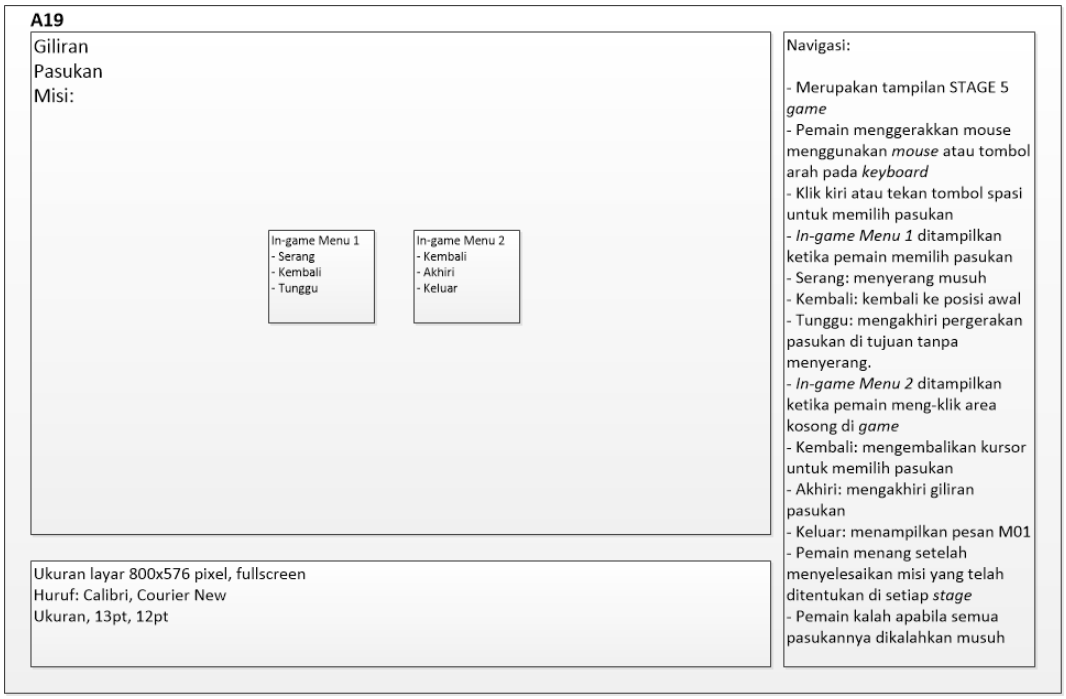
Gambar 3.84 Antarmuka halaman Cerita *Stage* 4 (A16)



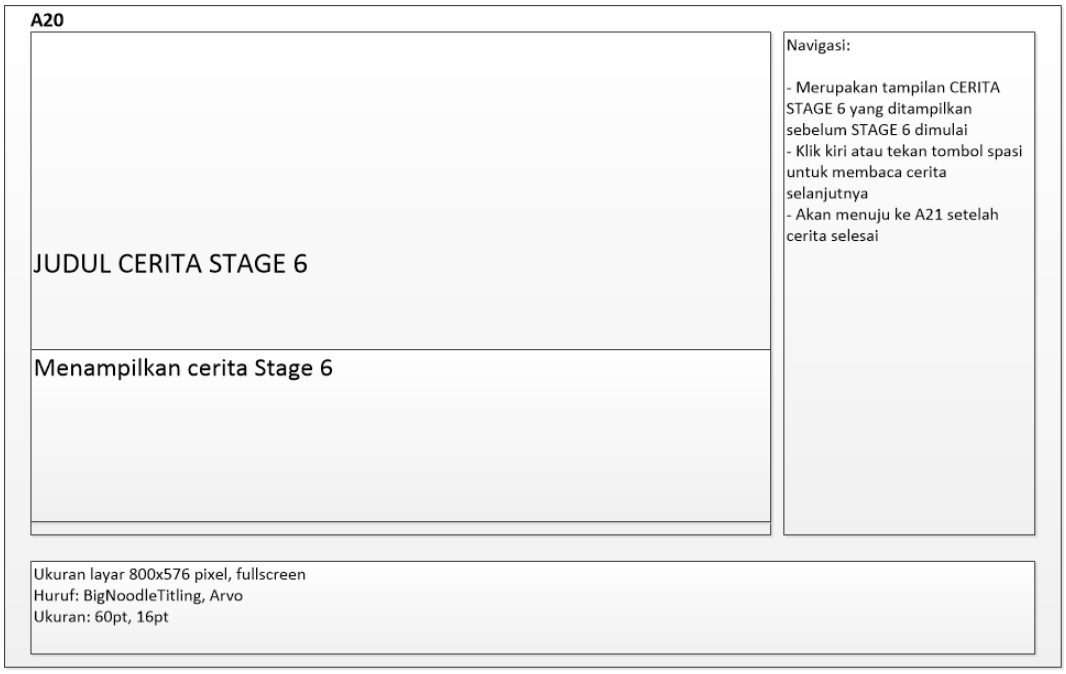
Gambar 3.85 Antarmuka halaman *Stage* 4 (A17)



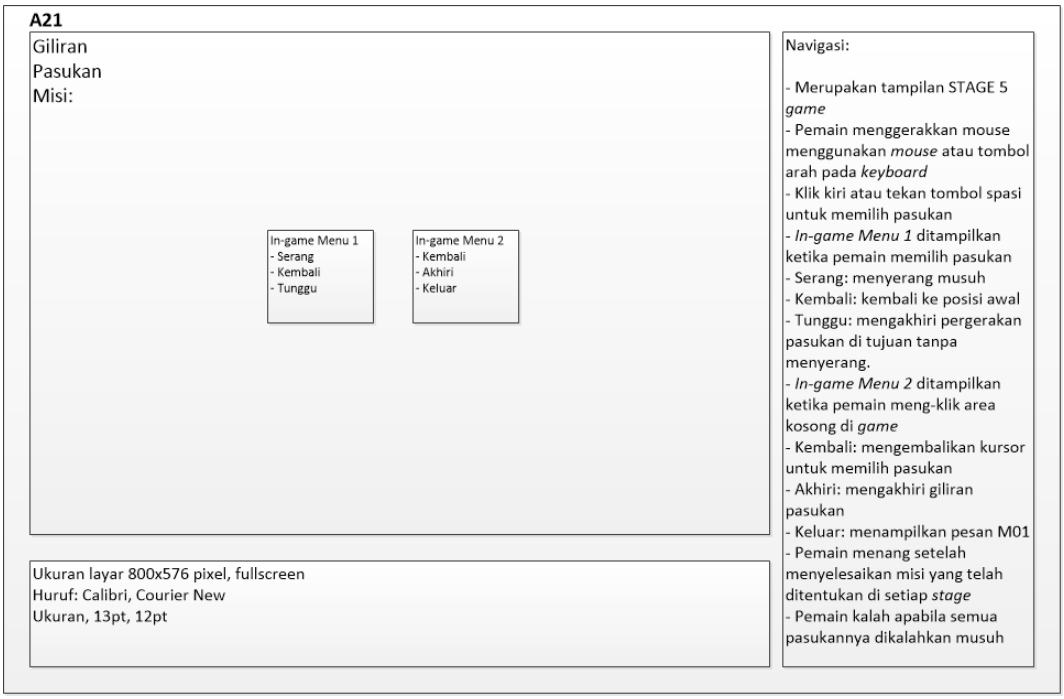
Gambar 3.86 Antarmuka halaman Cerita *Stage* 5 (A18)



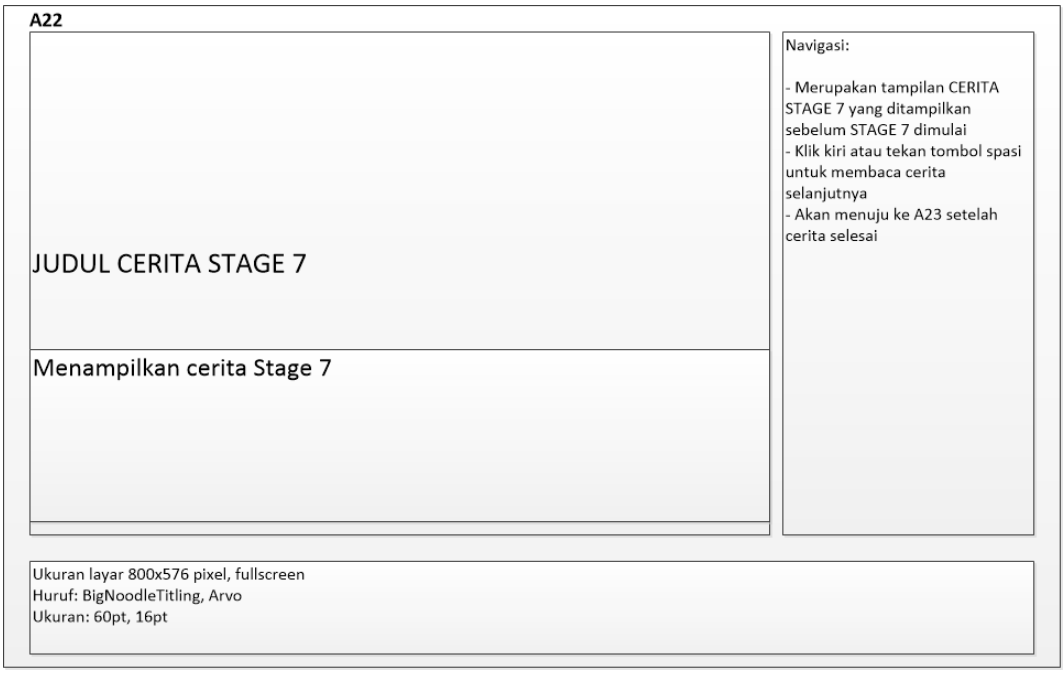
Gambar 3.87 Antarmuka halaman *Stage* 5 (A19)



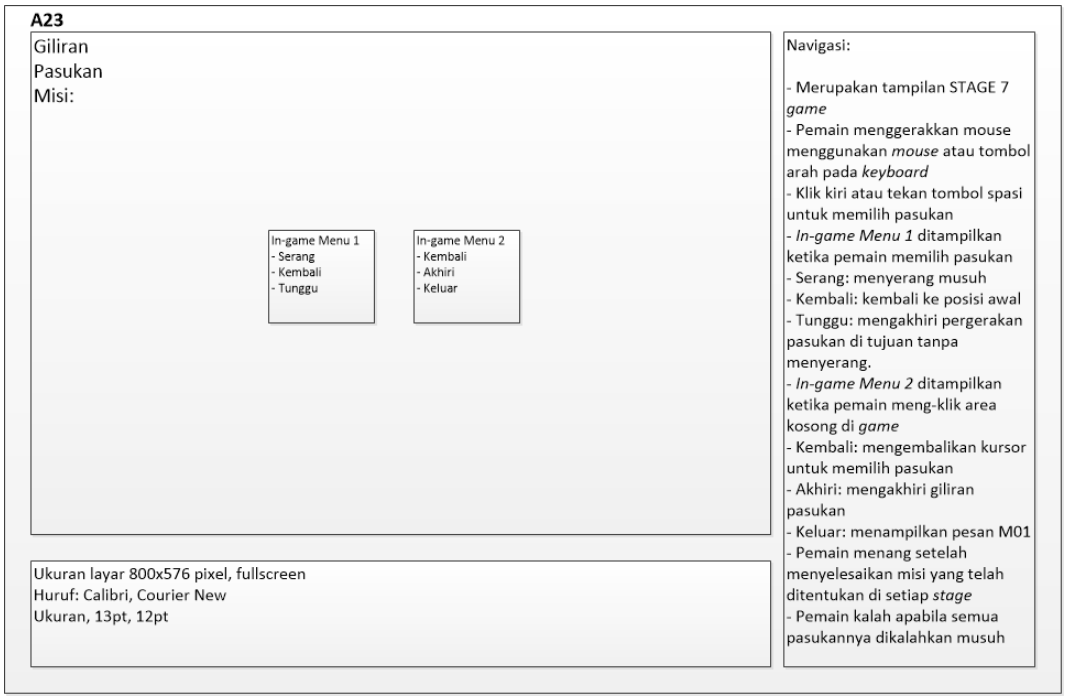
Gambar 3.88 Antarmuka halaman Cerita *Stage* 6 (A20)



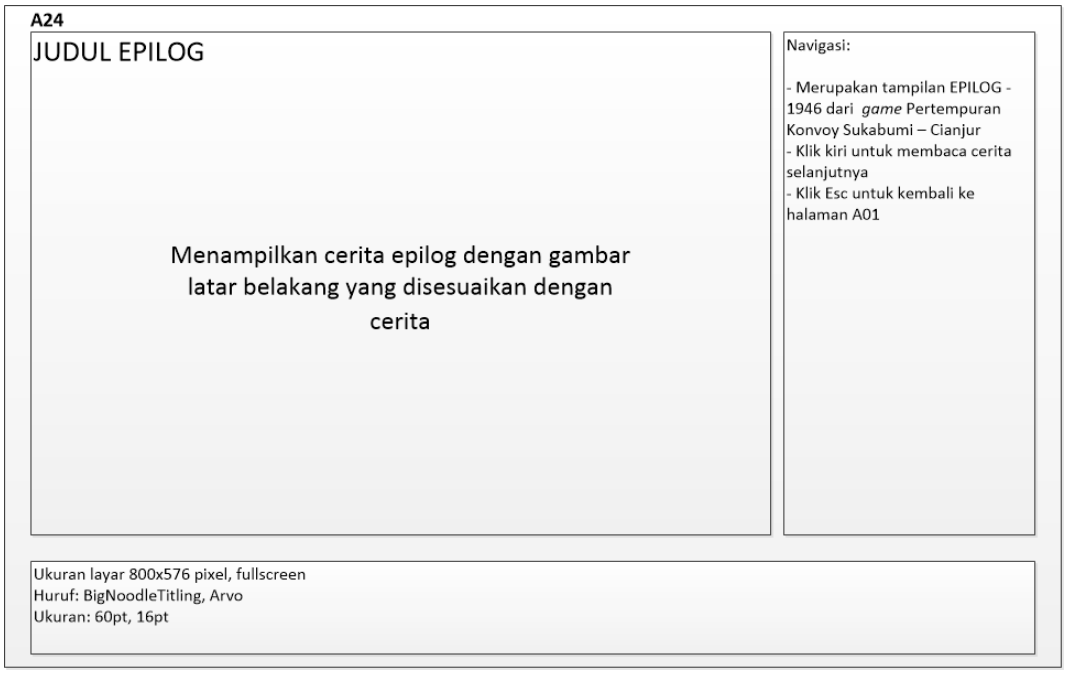
Gambar 3.89 Antarmuka halaman *Stage* 6 (A21)



Gambar 3.90 Antarmuka halaman Cerita *Stage* 7 (A22)



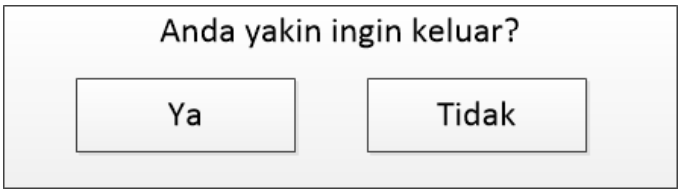
Gambar 3.91 Antarmuka halaman *Stage* 7 (A23)



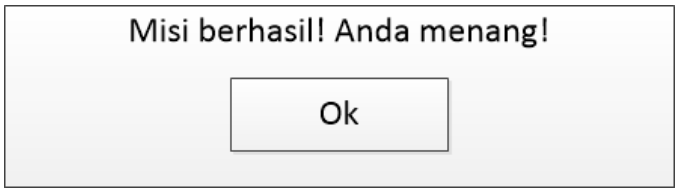
Gambar 3.92 Antarmuka halaman Epilog - 1946 (A24)

3.5.5 Perancangan Pesan

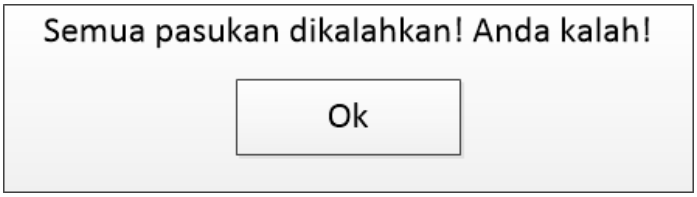
Tahap perancangan pesan dilakukan untuk merancang pesan agar dapat digunakan oleh *user* dalam menggunakan aplikasi ini. Adapun perancangan pesan dari *game* pertempuran konvoy sukabumi - cianjur adalah sebagai berikut:



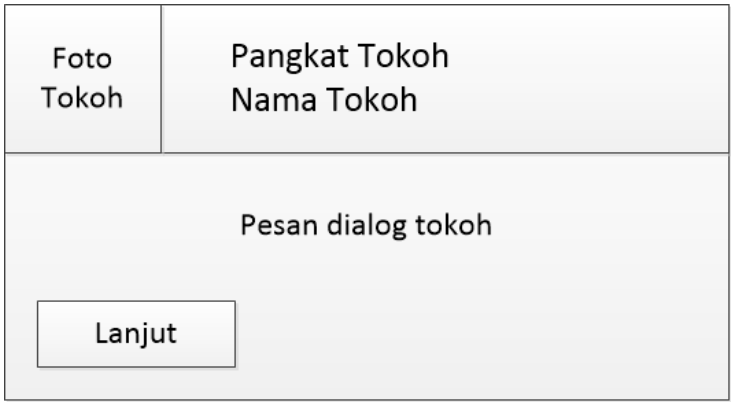
Gambar 3.93 Pesan validasi akan keluar dari stage (M01)



Gambar 3.94 Pesan memenangkan stage (M02)



Gambar 3.95 Pesan kalah stage (M03)



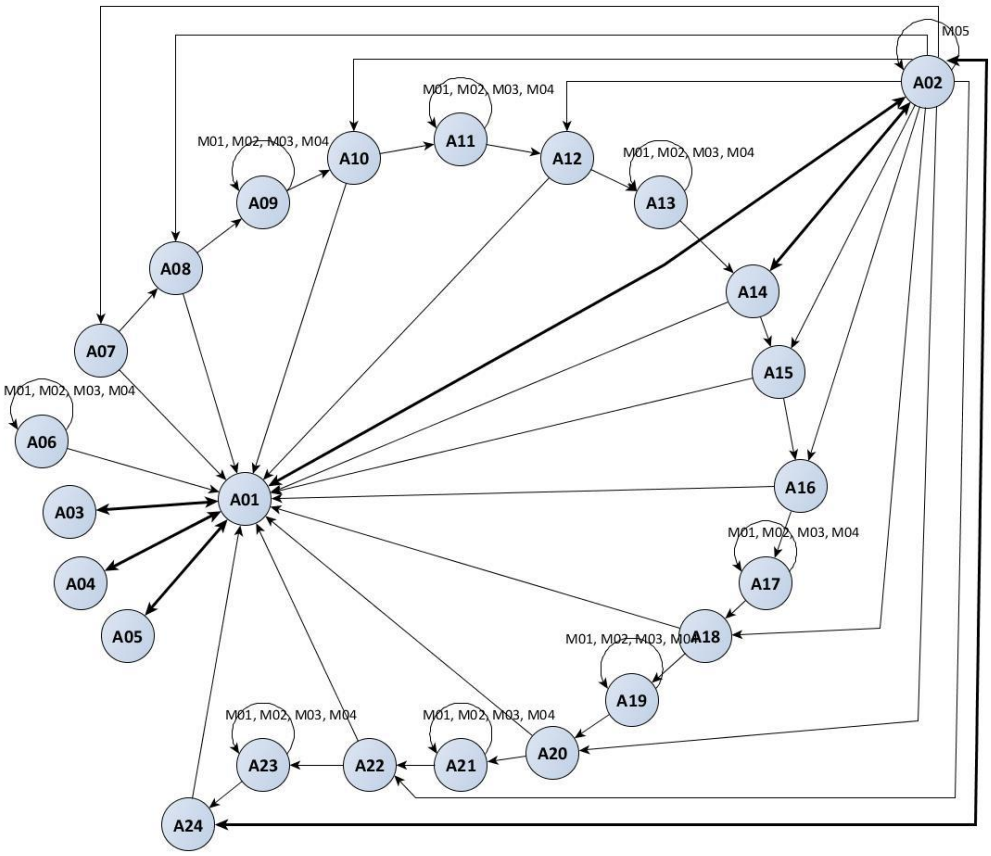
Gambar 3.96 Pesan Dialog Tokoh (M04)



Gambar 3.97 Pesan *Stage* Terkunci (M05)

3.5.6 Jaringan Semantik

Setelah melakukan perancangan antarmuka maka selanjutnya yaitu perancangan terhadap aliran dari menu-menu yang ada pada *game* Pertempuran Konvoy Sukabumi - Cianjur yang akan digambarkan dalam sebuah jaringan semantik seperti dibawah ini:



Gambar 3.98 Jaringan Semantik