BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi sistem merupakan tahap penerjemahan perancangan berdasarkan hasil analisis ke dalam suatu bahasa pemrograman yang digunakan serta penerapan perangkat lunak yang dibangun pada lingkungan yang sesungguhnya. Tujuan implementasi adalah untuk menerapkan perancangan yang telah dilakukan terhadap sistem, sehingga pengguna dapat memberi masukan demi berkembangnya sistem yang telah dibangun. Adapun pembahasan untuk implementasi sistem meliputi kebutuhan perangkat keras, perangkat lunak, pengujian alpha, dan pengujian beta.

4.1.1 Implementasi Perangkat Pembangun

Perangkat pembangun yang digunakan dalam pembangunan *game* Pertempuran Konvoy Sukabumi – Cianjur adalah sebagai berikut:

a. Perangkat Keras yang Digunakan

Perangkat keras yang digunakan untuk pembuatan *game* Pertempuran Konvoy Sukabumi – Cianjur adalah sebagai berikut:

Tabel 4.1 Perangkat Keras yang Digunakan

No	Perangkat Keras	Spesifikasi
1	*Processor*	Intel® Core™ i3-2330M CPU@2.20GHz
2	*RAM*	4GB
3	*Harddisk*	500 GB
4	*VGA*	NVIDIA GeForce GT 520M 1GB
5	Monitor	Monitor dengan resolusi 1366x768 *pixel*
6	*Mouse*	Ya
7	*Keyboard*	Ya
8	*Speaker*	Ya

b. Perangkat Lunak yang Digunakan

Spesifikasi perangkat lunak (*software*) yang dipasang pada sistem komputer yang digunakan untuk mengembangkan *game* Pertempuran Konvoy Sukabumi – Cianjur adalah sebagai berikut:

Tabel 4.2 Perangkat Lunak yang Digunakan

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Microsoft Windows 8
2	*Game Engine*	Game Maker 8.0 Pro
3	Bahasa Pemrograman	GML
4	*Image Editor*	Adobe Photoshop CS3
5	UML *Modeler*	Microsoft Visio 2010

4.1.2 Implementasi Antarmuka

Pada tahap ini dilakukan penerapan hasil perancangan antarmuka ke dalam *game* yang dibangun dengan menggunakan perangkat lunak yang telah dipaparkan sebelumnya. Adapun implementasi antarmuka dari *game* Pertempuran Konvoy Sukabumi – Cianjur adalah sebagai berikut:

1. Tampilan Menu Utama

   Merupakan antarmuka halaman menu utama *game* yang akan muncul ketika *game* dijalankan. Terdapat 5 menu yang dapat dipilih *user*, yaitu mulai, cara main, *highscore*, *credits*, info dan menu untuk keluar dari aplikasi.

2. Tampilan Cara Main

   Merupakan tampilan yang akan menampilkan petunjuk cara memainkan *game* Pertempuran Konvoy Sukabumi – Cianjur.

3. Tampilan *Highscore*

   Merupakan tampilan *highscore* pemain yang dibagi-bagi dalam setiap level.

4. Tampilan Info

   Menampilkan informasi singkat mengenai pembuat dan referensi yang digunakan dalam pembuatan *game* Pertempuran Konvoy Sukabumi – Cianjur.

5. Tampilan *Credits*

   Menampilkan *rolling credits* yang berisi mengenai nama-nama yang telah berkontribusi dalam pembuatan *game* Pertempuran Konvoy Sukabumi – Cianjur.

6. Tampilan Pilih *Stage*

   Merupakan tampilan menu pemilihan *stage* yang terdiri dari tutorial, prolog 1945, *stage* 1, *stage* 2, *stage* 3, epilog 1945, prolog 1946, *stage* 4, *stage* 5, *stage* 6, *stage* 7, epilog 1946 dan menu kembali ke menu utama.

7. Tampilan Tutorial

   Menampilkan tutorial singkat mengenai aturan dan cara memainkan *game* Pertempuran Konvoy Sukabumi – Cianjur. Pemain harus menyelesaikan tutorial terlebih dahulu jika ingin memainkan *stage* berikutnya.

8. Tampilan Prolog 1945

   Menampilkan prolog kisah sejarah yang terjadi pada tahun 1945, dimulai dari pembacaan proklamasi kemerdekaan RI sampai ke penyebab konflik antara TKR dan pasukan Sekutu yang melatarbelakangi terjadinya pertempuran konvoy Sukabumi - Cianjur.

9. Tampilan Cerita *Stage* 1

   Menampilkan kisah sejarah dan misi yang harus diselesaikan pada *stage* 1.

10. Tampilan *Stage* 1

    Tampilan *map* dengan misi yang harus diselesaikan pemain pada *stage* 1.

11. Tampilan Cerita *Stage* 2

    Menampilkan kisah sejarah dan misi yang harus diselesaikan pada *stage* 2.

12. Tampilan *Stage* 2

    Tampilan *map* dengan misi yang harus diselesaikan pemain pada *stage* 2.

13. Tampilan Cerita Stage 3

    Menampilkan kisah sejarah dan misi yang harus diselesaikan pada *stage* 3.

14. Tampilan *Stage* 3

    Tampilan *map* dengan misi yang harus diselesaikan pemain pada *stage* 3.

15. Tampilan Epilog 1945

    Menampilkan epilog kisah sejarah yang terjadi pada tahun 1945.

16. Tampilan Prolog 1946

    Menampilkan prolog kisah sejarah yang terjadi pada tahun 1946 yang melatarbelakangi terjadinya kembali konflik antara pasukan TKR yang sudah berubah nama menjadi TRI dengan Sekutu.

17. Tampilan Cerita *Stage* 4

    Menampilkan kisah sejarah dan misi yang harus diselesaikan pada *stage* 4.

18. Tampilan *Stage* 4

    Tampilan *map* dengan misi yang harus diselesaikan pemain pada *stage* 4.

19. Tampilan Cerita *Stage* 5

    Menampilkan kisah sejarah dan misi yang harus diselesaikan pada *stage* 5.

20. Tampilan *Stage* 5

    Tampilan *map* dengan misi yang harus diselesaikan pemain pada *stage* 5.

21. Tampilan Cerita *Stage* 6

    Menampilkan kisah sejarah dan misi yang harus diselesaikan pada *stage* 6.

22. Tampilan *Stage* 6

    Tampilan *map* dengan misi yang harus diselesaikan pemain pada *stage* 6.

23. Tampilan Cerita *Stage* 7

    Menampilkan kisah sejarah dan misi yang harus diselesaikan pada *stage* 7.

24. Tampilan *Stage* 7

    Tampilan *map* dengan misi yang harus diselesaikan pemain pada *stage* 7.

25. Tampilan Epilog 1946

    Menampilkan epilog kisah sejarah yang terjadi pada tahun 1946 yang mengakhiri peristiwa pertempuran konvoy Sukabumi – Cianjur.

26. Tampilan Kapitulasi

    Tampilan intisari dari kisah dan pelaku sejarah yang terlibat dalam setiap *stage* yang akan ditampilkan ketika pemain menyelesaikan setiap *stage* dalam *game* Pertempuran Konvoy Sukabumi - Cianjur.

27. Tampilan Pesan Keluar

    Tampilan pesan konfirmasi ketika pemain akan keluar dari *game*.

28. Tampilan Pesan Menang

    Tampilan pesan yang muncul ketika pemain menyelesaikan setiap *stage*.

29. Tampilan Pesan Kalah

    Tampilan pesan yang muncul ketika pemain kalah dalam *stage*.

30. Tampilan Pesan Dialog

    Tampilan dialog antar tokoh yang ditampilkan di awal permainan.

31. Tampilan Pesan Pertanyaan

    Tampilan pertanyaan untuk mengevaluasi pemahaman pemain mengenai cerita dan tokoh yang terlibat dalam *stage* yang baru saja diselesaikan.

32. Tampilan Pesan *Stage* Terkunci

    Tampilan yang akan muncul apabila pemain memilih *stage* yang masih terkunci karena pemain belum menyelesaikan *stage* sebelumnya.

4.2 Pengujian

Pengujian aplikasi dilakukan dengan tujuan untuk menemukan kesalahan atau kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

Pengujian yang dilakukan yaitu pengujian *alpha* dan pengujian *beta*. Pengujian *alpha* dilakukan dengan pengujian *black box* dan *white box*. Pengujian *black box* digunakan untuk mengetahui kesalahan proses secara fungsional sedangkan *white box* digunakan untuk menguji performa dari algoritma dari metode yang digunakan. Pengujian *beta* digunakan untuk mengetahui tanggapan pengguna terhadap aplikasi *game* dengan menggunakan kuesioner.

4.2.1 Pengujian *Alpha*

Pengujian *alpha* dilakukan pada sisi pengembangan yang merekan semua kesalahan dan masalah pemakaian. Pengujian *alpha* dilakukan dengan menggunakan metode *black box* dan *white box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak sedangkan *white box* berfokus pada algoritma dari metode yang digunakan. Skenario pengujian dari *game* pertempuran konvoy Sukabumi – Cianjur dapat dilihat pada table 4.3 dibawah ini.

Tabel 4.3 Skenario Pengujian *Alpha*

No	Komponen yang Diuji	Skenario Pengujian	Jenis Pengujian
1	Menu Utama	Memilih Menu Mulai	*Black box*
		Memilih Menu Cara Main	*Black box*
		Memilih Menu *Highscore*	*Black box*
		Memilih Menu Credits	*Black box*
		Memilih Menu Info	*Black box*
		Memilih Menu Keluar	*Black box*
2	Menu Cara Main	Menekan Tombol *Esc*	*Black box*
		Menekan Tombol Kiri *Mouse*	*Black box*
		Menekan Tombol Kanan *Mouse*	*Black box*
3	Menu *Highscore*	Menekan Tombol *Esc*	*Black box*
4	Menu *Credits*	Menekan Tombol *Esc*	*Black box*
5	Menu Info	Menekan tombol *Esc*	*Black box*
6	Menu Pilih Stage	Memilih Menu Tutorial	*Black box*
		Memilih Menu Prolog – 1945	*Black box*
		Memilih Menu *Stage* 1	*Black box*
		Memilih Menu *Stage* 2	*Black box*
		Memilih Menu *Stage* 3	*Black box*
		Memilih Menu Epilog – 1945	*Black box*
		Memilih Menu Prolog – 1946	*Black box*
		Memilih Menu *Stage* 4	*Black box*
		Memilih Menu *Stage* 5	*Black box*
		Memilih Menu *Stage* 6	*Black box*
		Memilih Menu *Stage* 7	*Black box*
		Memilih Menu Epilog – 1946	*Black box*
		Memilih Menu Menu Utama	*Black box*
7	Menu Tutorial	Menekan Tombol Kanan *Mouse*	*Black box*
		Menekan Tombol Kiri *Mouse*	*Black box*

No	Komponen yang Diuji	Skenario Pengujian	Jenis Pengujian
		Memilih Menu Serang	*Black box*
		Memilih Menu Tunggu	*Black box*
		Memilih Menu Kembali	*Black box*
		Memilih Menu Keluar	*Black box*
8	Menu Prolog – 1945	Menekan Tombol Kiri Mouse	*Black box*
		Menekan Tombol *Esc*	*Black box*
9	Menu *Stage* 1	Menekan Tombol Kiri *Mouse*	*Black box*
		Memilih Menu Serang	*Black box*
		Memilih Menu Tunggu	*Black box*
		Memilih Menu Kembali	*Black box*
		Memilih Menu Keluar	*Black box*
		Menekan Tombol F5	*Black box*
		Menekan Tombol F6	*Black box*
10	Menu *Stage* 2	Menekan Tombol Kiri *Mouse*	*Black box*
		Memilih Menu Serang	*Black box*
		Memilih Menu Tunggu	*Black box*
		Memilih Menu Kembali	*Black box*
		Memilih Menu Keluar	*Black box*
		Menekan Tombol F5	*Black box*
		Menekan Tombol F6	Black box
11	Menu *Stage* 3	Menekan Tombol Kiri *Mouse*	*Black box*
		Memilih Menu Serang	*Black box*
		Memilih Menu Tunggu	*Black box*
		Memilih Menu Kembali	*Black box*
		Memilih Menu Keluar	*Black box*
		Menekan Tombol F5	*Black box*
		Menekan Tombol F6	*Black box*
12	Menu Epilog – 1945	Menekan Tombol Kiri Mouse	*Black box*

No	Komponen yang Diuji	Skenario Pengujian	Jenis Pengujian
13	Menu Prolog – 1946	Menekan Tombol Kiri Mouse	*Black box*
		Menekan Tombol *Esc*	*Black box*
14	Menu *Stage* 4	Menekan Tombol Kiri *Mouse*	*Black box*
		Memilih Menu Serang	*Black box*
		Memilih Menu Tunggu	*Black box*
		Memilih Menu Kembali	*Black box*
		Memilih Menu Keluar	*Black box*
		Menekan Tombol F5	*Black box*
		Menekan Tombol F6	*Black box*
15	Menu *Stage* 5	Menekan Tombol Kiri *Mouse*	*Black box*
		Memilih Menu Serang	*Black box*
		Memilih Menu Tunggu	*Black box*
		Memilih Menu Kembali	*Black box*
		Memilih Menu Keluar	*Black box*
		Menekan Tombol F5	*Black box*
		Menekan Tombol F6	*Black box*
16	Menu *Stage* 6	Menekan Tombol Kiri *Mouse*	*Black box*
		Memilih Menu Serang	*Black box*
		Memilih Menu Tunggu	*Black box*
		Memilih Menu Kembali	*Black box*
		Memilih Menu Keluar	*Black box*
		Menekan Tombol F5	*Black box*
		Menekan Tombol F6	*Black box*
17	Menu *Stage* 7	Menekan Tombol Kiri *Mouse*	*Black box*
		Memilih Menu Serang	*Black box*
		Memilih Menu Tunggu	*Black box*
		Memilih Menu Kembali	*Black box*
		Memilih Menu Keluar	*Black box*

No	Komponen yang Diuji	Skenario Pengujian	Jenis Pengujian
		Menekan Tombol F5	*Black box*
		Menekan Tombol F6	*Black box*
18	Menu Epilog – 1946	Menekan Tombol Kiri *Mouse*	*Black box*
19	Pesan Keluar	Memilih Tombol Ya	*Black box*
		Memilih Tombol Tidak	*Black box*
20	Pencarian Rute Terpendek	Metode *A Star Pathfinding*	*White box*
21	FSM Menyerang Pemain	Algoritma *Finite State Machine*	*White box*

4.2.1.1 Kasus dan Hasil Pengujian *Black Box*

Pengujian perangkat lunak yang sudah dibangun dilakukan dengan menggunakan metode *black box*, berdasarkan pengamatan yang sudah dilakukan maka dihasilkan kesimpulan dari pengujian sebagai berikut:

Tabel 4.4 Pengujian *Black Box*

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
1	Menu Utama	Memilih Menu Mulai	Menampilkan halaman Pilih *Stage*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Cara Main	Menampilkan halaman Cara Main	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu *Highscore*	Menampilkan halaman daftar *Highscore*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Credits	Menampilkan halaman *Credits*	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
		Memilih Menu Info	Menampilkan halaman Info *game*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Keluar	Menutup / keluar dari aplikasi *game*	[√] Berhasil [ ] Tidak Berhasil
2	Menu Cara Main	Menekan Tombol *Esc*	Kembali ke Menu Utama	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol Kiri *Mouse*	Melanjutkan ke halaman Cara Main berikutnya	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol Kanan *Mouse*	Kembali ke halaman Cara Main sebelumnya	[√] Berhasil [ ] Tidak Berhasil
3	Menu *Highscore*	Menekan Tombol *Esc*	Kembali ke Menu Utama	[√] Berhasil [ ] Tidak Berhasil
4	Menu Credits	Menekan Tombol *Esc*	Kembali ke Menu Utama	[√] Berhasil [ ] Tidak Berhasil
5	Menu Info	Menekan tombol *Esc*	Kembali ke Menu Utama	[√] Berhasil [ ] Tidak Berhasil
6	Menu Pilih *Stage*	Memilih Menu Tutorial	Menampilkan *Stage* Tutorial	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Prolog – 1945	Menampilkan pesan "Stage Terkunci" apabila telah menyelesaikan Tutorial dan menampilkan Prolog – 1945 jika	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
			Tutorial telah diselesaikan	
		Memilih Menu *Stage* 1	Menampilkan pesan "Stage Terkunci" apabila belum menyelesaikan Tutorial dan menampilkan *Stage* 1 jika Tutorial telah diselesaikan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu *Stage* 2	Menampilkan pesan "Stage Terkunci" apabila belum menyelesaikan *Stage* 1 dan menampilkan *Stage* 2 jika *Stage* 1 telah diselesaikan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu *Stage* 3	Menampilkan pesan "Stage Terkunci" apabila belum menyelesaikan *Stage* 2 dan menampilkan *Stage* 3 jika *Stage* 2 telah diselesaikan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Epilog – 1945	Menampilkan pesan "Stage Terkunci" apabila belum menyelesaikan *Stage* 3 dan menampilkan	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
			Epilog – 1945 jika *Stage* 3 telah diselesaikan	
		Memilih Menu Prolog – 1946	Menampilkan pesan "Stage Terkunci" apabila belum menyelesaikan *Stage* 3 dan menampilkan Prolog – 1946 jika *Stage* 3 telah diselesaikan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu *Stage* 4	Menampilkan pesan "Stage Terkunci" apabila belum menyelesaikan *Stage* 3 dan menampilkan *Stage* 4 jika *Stage* 3 telah diselesaikan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu *Stage* 5	Menampilkan pesan "Stage Terkunci" apabila belum menyelesaikan *Stage* 4 dan menampilkan *Stage* 5 jika *Stage* 4 telah diselesaikan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu *Stage* 6	Menampilkan pesan "Stage Terkunci" apabila belum	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
			menyelesaikan *Stage* 5 dan menampilkan *Stage* 6 jika *Stage* 5 telah diselesaikan	
		Memilih Menu *Stage* 7	Menampilkan pesan "Stage Terkunci" apabila belum menyelesaikan *Stage* 6 dan menampilkan *Stage* 7 jika *Stage* 6 telah diselesaikan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Epilog – 1946	Menampilkan pesan "Stage Terkunci" apabila belum menyelesaikan *Stage* 7 dan menampilkan Epilog – 1946 jika *Stage* 7 telah diselesaikan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Menu Utama	Menampilkan halaman Menu Utama	[√] Berhasil [ ] Tidak Berhasil
7	Menu Tutorial	Menekan Tombol Kanan *Mouse*	Melanjutkan instruksi tutorial	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol Kiri	Memilih pasukan dan menampilkan menu	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
		Mouse		
		Memilih Menu Serang	Menyerang musuh	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Tunggu	Menunggu ditempat setelah pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Kembali	Kembali ke posisi awal sebelum pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Keluar	Menampilkan pesan keluar	[√] Berhasil [ ] Tidak Berhasil
8	Menu Prolog – 1945	Menekan Tombol Kiri Mouse	Melanjutkan cerita selanjutnya pada prolog – 1945	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol *Esc*	Menampilkan peasn keluar	[√] Berhasil [ ] Tidak Berhasil
9	Menu *Stage* 1	Menekan Tombol Kiri *Mouse*	Melanjutkan cerita selanjutnya pada saat cerita ditampilkan dan memilih pasukan lalu menampilkan menu ketika sudah dalam *map*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Serang	Menyerang musuh	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Tunggu	Menunggu ditempat setelah pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih	Kembali ke posisi awal	[√] Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
		Menu Kembali	sebelum pasukan digerakkan	[ ] Tidak Berhasil
		Memilih Menu Keluar	Menampilkan pesan keluar	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F5	Membuat cek poin (*save*)	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F6	Melanjutkan cek poin (*load*)	[√] Berhasil [ ] Tidak Berhasil
10	Menu *Stage* 2	Menekan Tombol Kiri *Mouse*	Melanjutkan cerita selanjutnya pada saat cerita ditampilkan dan memilih pasukan lalu menampilkan menu ketika sudah dalam *map*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Serang	Menyerang musuh	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Tunggu	Menunggu ditempat setelah pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Kembali	Kembali ke posisi awal sebelum pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Keluar	Menampilkan pesan keluar	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F5	Membuat cek poin (*save*)	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F6	Melanjutkan cek poin (*load*)	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
11	Menu *Stage* 3	Menekan Tombol Kiri *Mouse*	Melanjutkan cerita selanjutnya pada saat cerita ditampilkan dan memilih pasukan lalu menampilkan menu ketika sudah dalam *map*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Serang	Menyerang musuh	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Tunggu	Menunggu ditempat setelah pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Kembali	Kembali ke posisi awal sebelum pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Keluar	Menampilkan pesan keluar	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F5	Membuat cek poin (*save*)	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F6	Melanjutkan cek poin (*load*)	[√] Berhasil [ ] Tidak Berhasil
12	Menu Epilog – 1945	Menekan Tombol Kiri Mouse	Menampilkan cerita selanjutnya pada epilog – 1945	[√] Berhasil [ ] Tidak Berhasil
13	Menu Prolog – 1946	Menekan Tombol Kiri Mouse	Menampilkan cerita selanjutnya pada prolog – 1946	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol *Esc*	Menampilkan pesan keluar	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
14	Menu *Stage* 4	Menekan Tombol Kiri *Mouse*	Melanjutkan cerita selanjutnya pada saat cerita ditampilkan dan memilih pasukan lalu menampilkan menu ketika sudah dalam *map*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Serang	Menyerang musuh	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Tunggu	Menunggu ditempat setelah pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Kembali	Kembali ke posisi awal sebelum pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Keluar	Menampilkan pesan keluar	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F5	Membuat cek poin (*save*)	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F6	Melanjutkan cek poin (*load*)	[√] Berhasil [ ] Tidak Berhasil
15	Menu *Stage* 5	Menekan Tombol Kiri *Mouse*	Melanjutkan cerita selanjutnya pada saat cerita ditampilkan dan memilih pasukan lalu menampilkan menu ketika sudah dalam *map*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Serang	Menyerang musuh	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
		Memilih Menu Tunggu	Menunggu ditempat setelah pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Kembali	Kembali ke posisi awal sebelum pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Keluar	Menampilkan pesan keluar	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F5	Membuat cek poin (*save*)	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F6	Melanjutkan cek poin (*load*)	[√] Berhasil [ ] Tidak Berhasil
16	Menu *Stage* 6	Menekan Tombol Kiri *Mouse*	Melanjutkan cerita selanjutnya pada saat cerita ditampilkan dan memilih pasukan lalu menampilkan menu ketika sudah dalam *map*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Serang	Menyerang musuh	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Tunggu	Menunggu ditempat setelah pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Kembali	Kembali ke posisi awal sebelum pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Keluar	Menampilkan pesan keluar	[√] Berhasil [ ] Tidak Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
		Menekan Tombol F5	Membuat cek poin (*save*)	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F6	Melanjutkan cek poin (*load*)	[√] Berhasil [ ] Tidak Berhasil
17	Menu *Stage* 7	Menekan Tombol Kiri *Mouse*	Melanjutkan cerita selanjutnya pada saat cerita ditampilkan dan memilih pasukan lalu menampilkan menu ketika sudah dalam *map*	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Serang	Menyerang musuh	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Tunggu	Menunggu ditempat setelah pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Kembali	Kembali ke posisi awal sebelum pasukan digerakkan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Menu Keluar	Menampilkan pesan keluar	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F5	Membuat cek poin (*save*)	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol F6	Melanjutkan cek poin (*load*)	[√] Berhasil [ ] Tidak Berhasil
18	Menu Epilog – 1946	Menekan Tombol Kiri *Mouse*	Menampilkan cerita selanjutnya pada epilog – 1946	[√] Berhasil [ ] Tidak Berhasil
19	Pesan	Memilih	Keluar dari *stage* dan	[√] Berhasil

No	Kasus yang Akan Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang Diharapkan	Kesimpulan
	Keluar	Tombol Ya	kembali ke Menu Utama	[ ] Tidak Berhasil
		Memilih Tombol Tidak	Kembali ke permainan	[√] Berhasil [ ] Tidak Berhasil

4.2.1.2 Kasus dan Hasil Pengujian *White Box*

Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun dengan menggunakan metode *white box*.

A. Penerapan Algoritma A* (*A Star*)

```
1   //Inisiasi variable A*
    with (objNode)
    {
        aStarParent = noone;
        aStarF = 0;
        aStarG = 0;
        aStarH = 0;
    }

    var open, closed;
    var start, finish, current, neighbor;
    var tempF, tempG, i;
    var checkOnlyMovement, checkPath;

    start = argument0;
    finish = argument1;
    //Jika node awal dan node tujuan sama, maka tidak ada
    penghitungan jalur optimal
2   if (start == finish)
    {
        var path;
        path = ds_stack_create();
        ds_stack_push(path, start);
        return path;
    }
    //Buat list open node (node yang bisa dituju) dan closed
    node (node yang tidak bisa dituju)
3   open = ds_priority_create();
    closed = ds_list_create();
    //Tambahkan node ke daftar open node
4   ds_priority_add(open, start, start.aStarF);
    current = start;
```

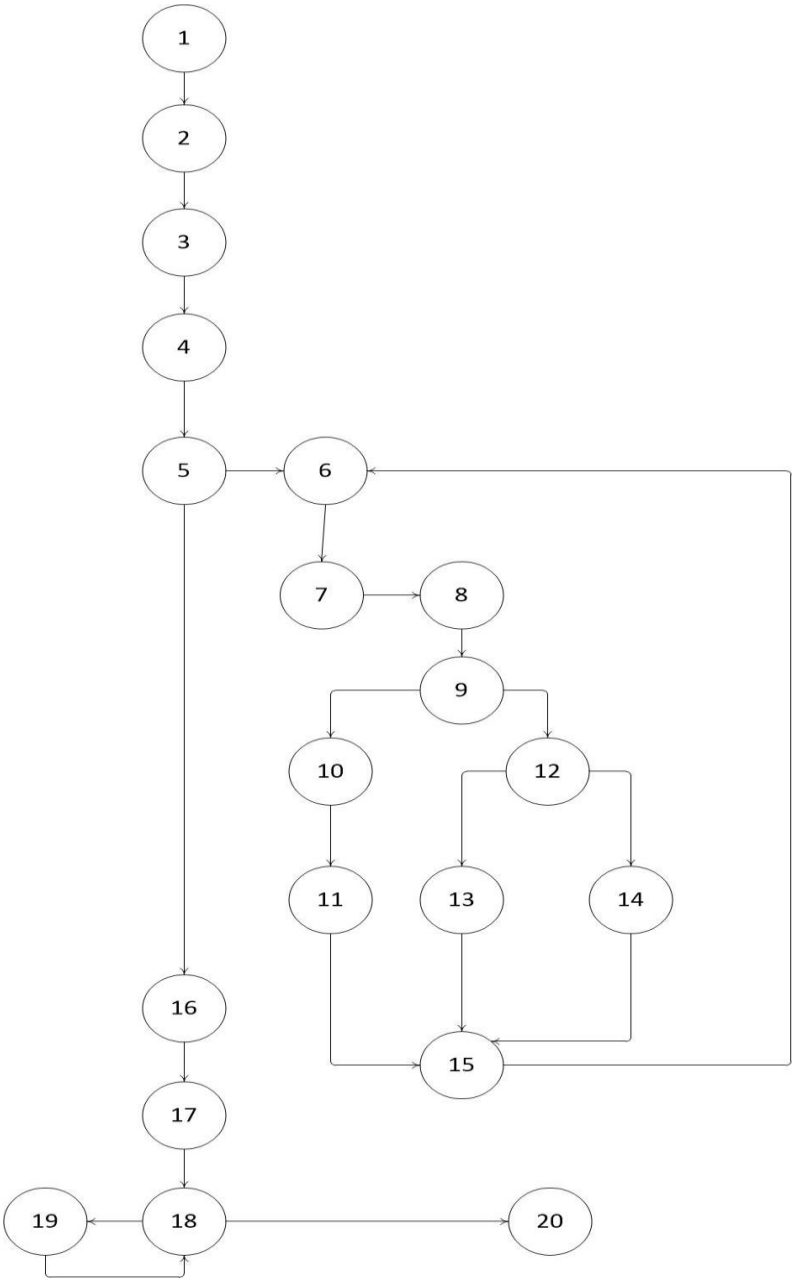
```
    //Ulangi sampai semua node dicek dan node tujuan sudah
    ditemukan
5   while (ds_priority_size(open) > 0 && current != finish)
    {
        //Cari node dengan nilai F terkecil, dan jadikan node
    ini closed node
        current = ds_priority_delete_min(open);
6       ds_list_add(closed, current);

    //Hitung nilai F dan tambahkan ke daftar open node
7       for (i = 0; i < ds_list_size(current.neighbors); i += 1)
        {
8           neighbor = ds_list_find_value(current.neighbors, i);
    //Cari node yang bukan closed node dan bisa dilewati
9           if (ds_list_find_index(closed, neighbor) < 0 &&
    neighbor.passable)
            {
                //Cek apakah node berada di daftar open node
10              if (ds_priority_find_priority(open, neighbor) ==
    0)
                {
                //jika tidak, hitung nilai F dan masukkan ke
    daftar open node
               neighbor.aStarParent = current;
               neighbor.aStarG = current.aStarG + neighbor.cost;
               neighbor.aStarH = heuristic(current, neighbor);
            neighbor.aStarF = neighbor.aStarG + neighbor.aStarH;
               ds_priority_add(open, neighbor, neighbor.aStarF);
                }
12  else
                {
    //Jika ya, cek apakah jika menuju neighbor dari node
    sekarang akan menghasilkan nilai F yang lebih baik, jika ya
    jadikan nilai tersebut sebagai nilai F yang baru.
                    tempG = current.aStarG + neighbor.cost;
                    tempF = neighbor.aStarG + neighbor.aStarH;
13                  if (tempF < neighbor.aStarF)
                    {
                        neighbor.aStarParent = current;
                        neighbor.aStarG = tempG;
                        neighbor.aStarF = tempF;
14                      ds_priority_change_priority(open,
    neighbor, neighbor.aStarF);
                    }
                } //endif
            } //endif
        } //endfor
15  } //endwhile

16  ds_priority_destroy(open);
    //Kini jalur sudah didapatkan, masukkan kedalam stack
17  var path;
    path = ds_stack_create();
    current = finish;
18  while (current != start && current != noone)
    {
```

```
        ds_stack_push(path, current);
        current = current.aStarParent;
19  }
    //tampilkan jalur yang didapat
20  return path;
```

a. *Flow Graph*



Gambar 4.1 *Flow Graph* Algoritma A* (*A Star*)

Keterangan :

◯ = Menggambarkan kondisi

→ = Menggambarkan aksi

b. *Cyclomatic Complexity* (V(G)

Dari diagram alir diatas, dapat dihitung *cyclomatic complexity*, yakni:

V(G) = E – N + 2

V(G) = 23 – 20 + 2

V(G) = 3 + 2

V(G) = 5

Dimana:

E = jumlah *edge* pada grafik alir

N = jumlah *node* pada grafik alir

c. *Independent Path*

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 15

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 12 – 13 – 15

Path 3= 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 12 – 14 – 15

Path 4 = 1 – 2 – 3 – 4 – 5 – 16 – 17 – 18 – 19

Path 5 = 1 – 2 – 3 – 4 – 5 – 16 – 17 – 18 – 20

d. *Graph Matrices*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	E
1		1																			0
2			1																		0
3				1																	0
4					1																0
5						1										1					1
6							1														0
7								1													0
8									1												0
9										1		1									1
10											1										0
11															1						0
12													1	1							1
13															1						0
14															1						0
15																1					0
16																	1				0
17																		1			0
18																			1	1	1
	SUM (E) + 1																				5

Keterangan:

1. Baris dan kolom merepresentasikan simpul
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul

Berdasarkan hassil pengujian diatas, maka dapat disimpulkan bahwa hasil pengujian *white box* algoritma a* adalah sebagai berikut:

Tabel 4.5 Hasil Pengujian Algoritma A* (*A Star*)

Path no	*Node (n)*	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 15	Mencari *node* tujuan dan menghitung nilai F	Mencari *node* tujuan dan menghitung nilai F	Sesuai yang diharapkan
2	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 12 – 13 – 15	Mencari *node* tujuan dan menghitung nilai F	Mencari *node* tujuan dan menghitung nilai F	Sesuai yang diharapkan
3	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 12 – 14 – 15	Mencari *node* tujuan dan menghitung nilai F	Mencari *node* tujuan dan menghitung nilai F	Sesuai yang diharapkan
4	1 – 2 – 3 – 4 – 5 – 16 – 17 – 18 – 19	Memasukkan daftar *path* kedalam *stack*	Memasukkan daftar *path* kedalam *stack*	Sesuai yang diharapkan
5	1 – 2 – 3 – 4 – 5 – 16 – 17 – 18 – 20	Mengembalikan jalur yang didapat	Mengembalikan jalur yang didapat	Sesuai yang diharapkan

B. FSM Menyerang Pemain

```
1   switch (state)
    {
2   case "finding enemies":
    //NPC akan mencari sasaran serang, jika ada, maka akan
    melakukan penyerangan, jika tidak ada, giliran NPC tersebut
    diakhiri. Pertama, NPC akan mencari semua node yang ada
    dalam wilayah serangnya
        attackNodes = findAttackRange(nodes[x div GRID_SIDE, y
    div GRID_SIDE], minRange, maxRange);
        var i;
        var enemyInRange;
        enemyInRange = false;

    //Kemudian, NPC akan mengecek apa di node-node tersebut
    terdapat pasukan musuhnya (pasukan pemain)
3       for (i = 0; i < ds_list_size(attackNodes); i += 1)
        {
            var unit;
            with (ds_list_find_value(attackNodes, i))
                unit = occupant;

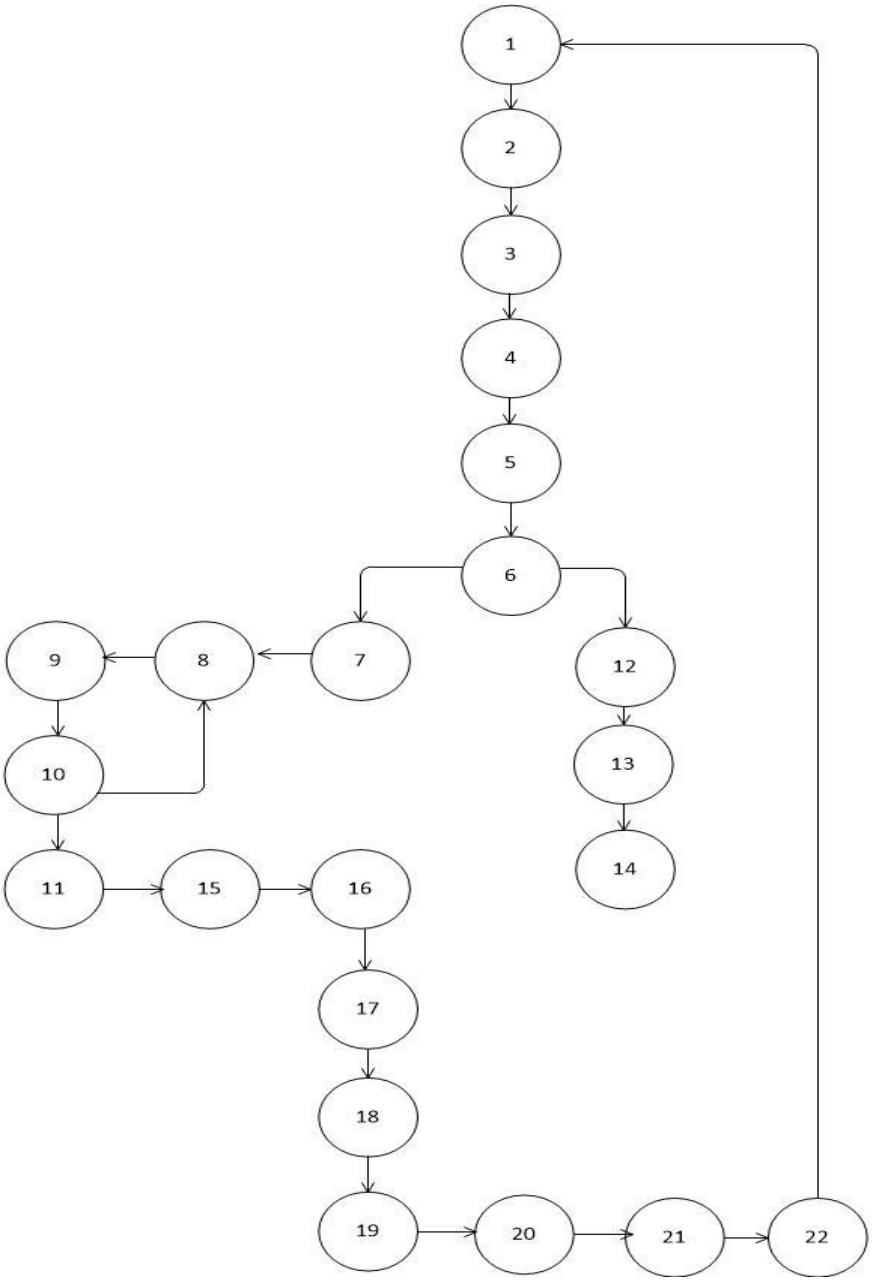
4           if (unit != noone)
            {
    //NPC mengecek apakah pasukan yang ada dalam wilayah
    serangnya merupakan pasukannya sendiri atau pasukan musuh.
5               if (unit.army != army)
                {
                    enemyInRange = true;
                    break;
                } //endif
            } //endif
6       } //endfor

    //Jika ada pasukan musuh dalam wilayah serang, maka lakukan
    penyerangan
7       if (enemyInRange)
        {
8           for (i = 0; i < ds_list_size(attackNodes); i += 1)
            {
                (ds_list_find_value(attackNodes, i)).color =
    c_red;
9           state = "attacking";
10          } //endfor
11      } //endif

    //Namun jika tidak ada, maka giliran NPC ini diakhiri
12      else
        {
13          state = "inactive";
            selected = false;
            cursor.selectedFinder = noone;
            ds_list_destroy(attackNodes);
14      }
    break;
```

```
15  case "attacking":
        var i;
        var attackingEnemy;
        attackingEnemy = false;
    //Ketika terjadi proses pemilihan node yang akan diserang,
    beberapa hal harus di cek. Pertama, NPC harus mengecek
    apakah node yang akan diserang apakah memang bisa diserang
    oleh NPC. Kemudian NPC memastikan bahwa node yang dia serang
    berisi pasukan musuhnya (pasukan pemain). Setelah menyerang,
    NPC mengakhiri giliran.
16      if (ds_list_find_index(attackNodes, cursor.node) != -1)
        {
17          if (cursor.node.occupant != noone)
18                  if (cursor.node.occupant.army != army)
                        attackingEnemy = true;
19          if (attackingEnemy)
                attack(id, cursor.node.occupant);
        }
20  //ubah state menjadi “inactive” setelah mengakhiri giliran
        selected = false;
        state = "inactive";
        cursor.selectedFinder = noone;
    //reset daftar node yang bisa diserang
21      for (i = 0; i < ds_list_size(attackNodes); i += 1)
            (ds_list_find_value(attackNodes, i)).color =
    c_black;
        ds_list_destroy(attackNodes);
        break;
22  }
```

a. *Flow Graph*



Gambar 4.2 *Flow Graph* FSM Menyerang Musuh

Keterangan :

◯ = Menggambarkan kondisi

→ = Menggambarkan aksi

b. *Cyclomatic Complexity* (V(G)

Dari diagram alir diatas, dapat dihitung *cyclomatic complexity*, yakni:

V(G) = E – N + 2

V(G) = 22 – 22 + 2

V(G) = 0 + 2

V(G) = 2

Dimana:

E = jumlah *edge* pada grafik alir

N = jumlah *node* pada grafik alir

c. *Independent Path*

Path 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22

Path 2 = 1 – 2 – 3 – 4 – 5 – 6 – 12 – 13 – 14

d. *Graph Matrices*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	E
1		1																					0
2			1																				0
3				1																			0
4					1																		0
5						1																	0
6							1					1											1
7								1															0
8									1														0
9										1													0
10											1												0
11															1								0
12													1										0
13														1									0
14																							0
15																1							0
16																	1						0
17																		1					0
18																			1				0
19																				1			0
20																					1		0
21																						1	0
	SUM (E) + 1																						2

Keterangan:

1. Baris dan kolom merepresentasikan simpul
2. Nilai 1 merepresentasikan adanya keterhubungan antar simpul

Berdasarkan hassil pengujian diatas, maka dapat disimpulkan bahwa hasil pengujian *white box* FSM menyerang musuh adalah sebagai berikut:

Tabel 4.6 Hasil Pengujian FSM Menyerang Musuh

Path no	*Node (n)*	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22	*State* NPC menjadi *"inactive"*	State NPC menjadi "inactive"	Sesuai yang diharapkan
2	1 – 2 – 3 – 4 – 5 – 6 – 12 – 13 – 14	Menyerang pasukan pemain	Menyerang pasukan pemain	Sesuai yang diharapkan

4.2.1.3 Kesimpulan Pengujian *Alpha*

Berdasarkan hasil pengujian *alpha* dengan menggunakan metode *white box* dan *black box* dengan kasus uji diatas maka dapat disimpulkan bahwa *game* Pertempuran Konvoy Sukabumi – Cianjur ini tidak memiliki kesalahan proses dan secara fungsional pun menghasilkan aplikasi yang sudah sesuai dengan perancangan dan kebutuhan *user*.

4.2.2 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif dengan melakukan pengujian secara langsung ke lapangan, dengan tujuan untuk mengetahui sejauh mana kualitas dari aplikasi *game* yang dibangun. Pengujian *beta* dilakukan dengan penelitian terhadap responden atau calon pengguna aplikasi untuk mengetahui penilaian *user* terhadap aplikasi. Teknik pengumpulan data dilakukan melalui pengisian kuesioner oleh *user* yang terdiri dari 30 orang responden yang berdomisili di Jawa Barat terutama di Sukabumi.

Untuk mengetahui tanggapan dan penilaian *user* terhadap aplikasi *game* pertempuran konvoy Sukabumi – Cianjur ini, telah disebar kuesioner kepada responden berdasarkan target *user*, kemudian dilakukan penghitungan dengan

menggunakan skala kontinum dimana data dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring setiap jawaban dari responden yang kemudian dijumlahkan.

4.2.2.1 Data Kuisioner

Adapun data kuesioner dari aplikasi *game* pertempuran konvoy Sukabumi – Cianjur adalah sebagai berikut:

1. Jumlah responden terdiri dari 30 orang.
2. *Point* yang menjadi bahan penilaian adalah:
    a. Memberikan *game* yang menarik
    b. Memberikan penyajian cerita sejarah yang menarik
    c. Membuat *gameplay* yang mudah dipahami
    d. Mengenalkan peristiwa pertempuran konvoy Sukabumi – Cianjur
    e. Menarik minat *user* untuk mengenal lebih jauh mengenai peristiwa konvoy Sukabumi – Cianjur
3. Pertanyaan yang diberikan kepada responden pada pengujian *beta* adalah sebagai berikut:
    1) Apakah *game* ini menarik?
    2) Apakah penyajian cerita sejarah dalam *game* ini menarik?
    3) Apakah *gameplay* dari *game* ini mudah untuk dipahami dan dimainkan?
    4) Apakah *game* ini membuat Anda mengenal peristiwa Pertempuran Konvoy Sukabumi – Cianjur?
    5) Apakah *game* ini membuat Anda tertarik untuk mengenal lebih jauh peristiwa Pertempuran Konvoy Sukabumi – Cianjur?
    6) Apakah Anda tertarik untuk memainkan *game* ini kembali?

4.2.2.2 Pengolahan Data Kuesioner

Adapun skala jawaban dari kuesioner adalah sebagai berikut:

Tabel 4.7 Skala Jawaban Kuesioner

Jawaban	Skor
SS = Sangat Setuju	5
S = Setuju	4
BS = Biasa Saja	3
KS = Kurang Setuju	2
TS = Tidak Setuju	1

Berikut adalah hasil pengumpulan data dari masing-masing jawaban untuk para responden. Jawaban dari masing-masing responden dapat dilihat pada table dibawah ini:

Tabel 4.8 Jawaban Kuesioner

No	Pertanyaan	Jawaban					Total
		SS	S	BS	KS	TS	
1	Apakah *game* ini menarik?	8	22				**30**
2	Apakah penyajian cerita sejarah dalam *game* ini menarik?	11	15	4			**30**
3	Apakah *gameplay* dari *game* ini mudah untuk dipahami dan dimainkan?	13	10	7			**30**
4	Apakah *game* ini membuat Anda mengenal peristiwa Pertempuran Konvoy Sukabumi – Cianjur?	15	14	1			**30**

No	Pertanyaan	Jawaban					Total
		SS	S	BS	KS	TS	
5	Apakah *game* ini membuat Anda tertarik untuk mengenal lebih jauh peristiwa Pertempuran Konvoy Sukabumi – Cianjur?	15	11	3	1		**30**
6	Apakah Anda tertarik untuk memainkan *game* ini kembali?	6	19	5			**30**

4.2.2.3 Hasil Pengolahan Kuesioner

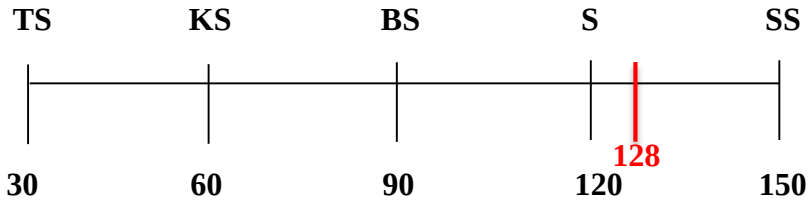
Data dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring setiap jawaban dari responden. Kemudian dijumlahkan lalu digambarkan ke dalam skala kontinum.

Tabel 4.9 Hasil Pengolahan Data Kuesioner Pertanyaan (1)

Jawaban	Skor	Jumlah Responden	Jumlah Skor (Skor x Jml Responden)
SS = Sangat Setuju	5	8	40
S = Setuju	4	22	88
BS = Biasa Saja	3		
KS = Kurang Setuju	2		
TS = Tidak Setuju	1		
		Total	**128**

Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30).

Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 128. Jadi, berdasarkan data tersebut, maka tingkat persetujuan terhadap pertanyaan nomor (1) adalah (128:150) x 100% = 85% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 128 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:

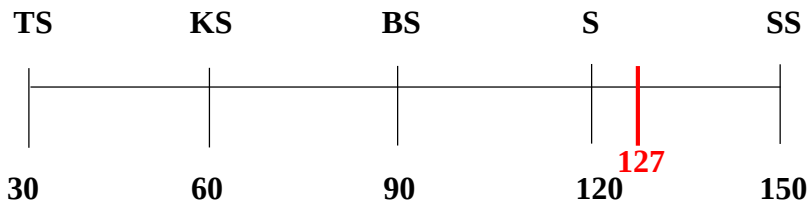


Gambar 4.3 Skala Kontinum Pertanyaan (1)

Tabel 4.10 Hasil Pengolahan Data Kuesioner Pertanyaan (2)

Jawaban	Skor	Jumlah Responden	Jumlah Skor (Skor x Jml Responden)
SS = Sangat Setuju	5	11	55
S = Setuju	4	15	60
BS = Biasa Saja	3	4	12
KS = Kurang Setuju	2		
TS = Tidak Setuju	1		
		Total	**127**

Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 127. Jadi, berdasarkan data tersebut, maka tingkat persetujuan terhadap pertanyaan nomor (2) adalah (127:150) x 100% = 85% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 127 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:

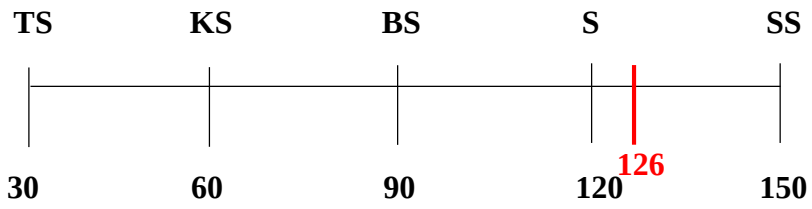


Gambar 4.4 Skala Kontinum Pertanyaan (2)

Tabel 4.11 Hasil Pengolahan Data Kuesioner Pertanyaan (3)

Jawaban	Skor	Jumlah Responden	Jumlah Skor (Skor x Jml Responden)
SS = Sangat Setuju	5	13	65
S = Setuju	4	10	40
BS = Biasa Saja	3	7	21
KS = Kurang Setuju	2		
TS = Tidak Setuju	1		
		Total	**126**

Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 126. Jadi, berdasarkan data tersebut, maka tingkat persetujuan terhadap pertanyaan nomor (3) adalah (126:150) x 100% = 84% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 126 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:

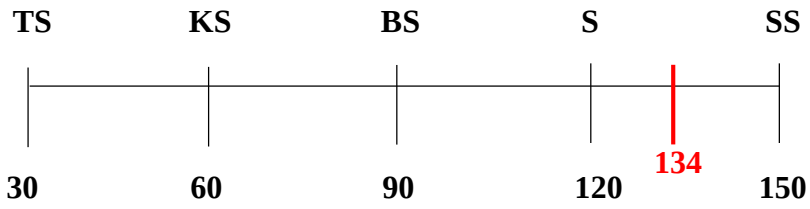


Gambar 4.5 Skala Kontinum Pertanyaan (3)

Tabel 4.12 Hasil Pengolahan Data Kuesioner Pertanyaan (4)

Jawaban	Skor	Jumlah Responden	Jumlah Skor (Skor x Jml Responden)
SS = Sangat Setuju	5	15	75
S = Setuju	4	14	56
BS = Biasa Saja	3	1	3
KS = Kurang Setuju	2		
TS = Tidak Setuju	1		
		Total	**134**

Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 134. Jadi, berdasarkan data tersebut, maka tingkat persetujuan terhadap pertanyaan nomor (4) adalah (134:150) x 100% = 89% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 134 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:

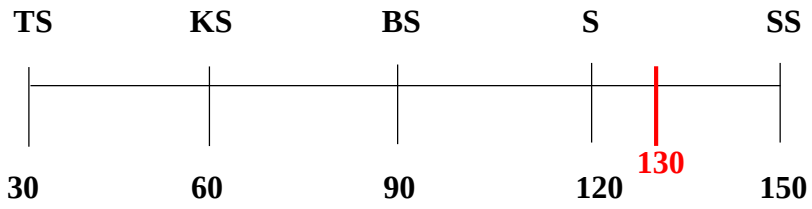


Gambar 4.6 Skala Kontinum Pertanyaan (4)

Tabel 4.13 Hasil Pengolahan Data Kuesioner Pertanyaan (5)

Jawaban	Skor	Jumlah Responden	Jumlah Skor (Skor x Jml Responden)
SS = Sangat Setuju	5	15	75
S = Setuju	4	11	44
BS = Biasa Saja	3	3	9
KS = Kurang Setuju	2	1	2
TS = Tidak Setuju	1		
		Total	**130**

Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 130. Jadi, berdasarkan data tersebut, maka tingkat persetujuan terhadap pertanyaan nomor (5) adalah (130:150) x 100% = 87% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 130 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:

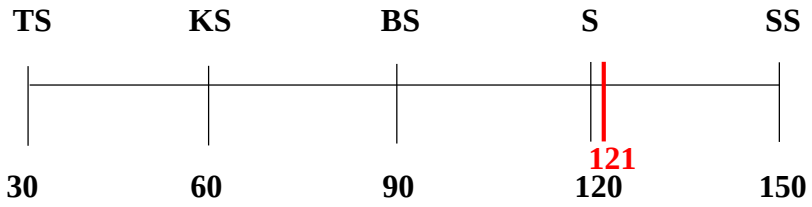


Gambar 4.7 Skala Kontinum Pertanyaan (5)

Tabel 4.14 Hasil Pengolahan Data Kuesioner Pertanyaan (6)

Jawaban	Skor	Jumlah Responden	Jumlah Skor (Skor x Jml Responden)
SS = Sangat Setuju	5	6	30
S = Setuju	4	19	76
BS = Biasa Saja	3	5	15
KS = Kurang Setuju	2		
TS = Tidak Setuju	1		
		Total	**121**

Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 121. Jadi, berdasarkan data tersebut, maka tingkat persetujuan terhadap pertanyaan nomor (6) adalah (121:150) x 100% = 81% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 121 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:



Gambar 4.8 Skala Kontinum Pertanyaan (6)

4.2.2.4 Kesimpulan Pengujian Beta

Berdasarkan jawaban hasil kuesioner yang diberikan kepada para responden, maka dapat disimpulkan bahwa:

1. *Game* Pertempuran Konvoy Sukabumi – Cianjur merupakan *game* yang menarik.

2. *Game* Pertempuran Konvoy Sukabumi – Cianjur menyajikan cerita sejarah dengan menarik.
3. *Gameplay* dari *game* ini mudah untuk dipahami dan dimainkan.
4. *Game* ini dapat mengenalkan kisah sejarah Pertempuran Konvoy Sukabumi – Cianjur pada pengguna.
5. *Game* ini membuat responden tertarik untuk mengenal lebih jauh mengenai kisah sejarah Pertempuran Konvoy Sukabumi – Cianjur.
6. Sebagian besar responden ingin memainkan *game* ini kembali.