

BAB III

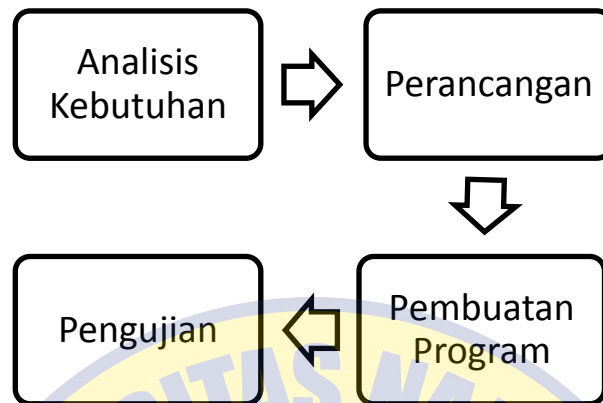
METODOLOGI PENELITIAN

3.1 Desain Penelitian

Metode yang digunakan pada penelitian ini ialah metode penelitian dan Pengembangan (*Research and Development*). Metode Penelitian dan Pengembangan ialah sebuah proses yang digunakan untuk mengembangkan dan memverifikasi terhadap suatu produk pendidikan. Langkah – langkah pada proses ini biasanya disebut dengan siklus *research* dan *development*. Menurut Gay (1990), metode Penelitian Pengembangan ialah sebuah usaha untuk mengembangkan suatu produk yang sangat efektif untuk digunakan sekolah, dan untuk menguji perangkat lunak.

Pada penelitian ini, metode yang digunakan untuk pengembangan perangkat lunak ialah metode *waterfall*. Metode ini pertama kali diperkenalkan oleh Winston Royce pada tahun 1970. Metode *waterfall* menggunakan pendekatan yang sistematis dan berurutan secara linear pada pengembangan perangkat lunak. Pada pengembangan perangkat lunak dengan metode *Waterfall*, tahapan yang harus dilakukan ialah Analisis Kebutuhan, Desain, Pengkodean, Pengujian, dan Pemeliharaan. Kelebihan menggunakan metode *waterfall* ialah sedikitnya kesalahan yang akan terjadi karena memiliki tahapan yang urut, mulai dari analisis sampai pemeliharaan.

Tahapan penelitian yang digunakan dalam penelitian ini ialah:



Gambar 3.1. Diagram Alur Penelitian

3.2 Metode Pengembangan

3.2.1 Analisis (Analisis)

Adalah proses dalam pengumpulan data, identifikasi masalah, dan analisis kebutuhan sistem untuk mendapatkan pemahaman yang lebih jauh dari substansi yang kompleks yang difokuskan dalam pembuatan perangkat lunak. Berikut ialah hasil dari analisis kebutuhan berupa kebutuhan fungsional, kebutuhan perangkat lunak, dan kebutuhan perangkat keras dan penjelasannya seperti di bawah ini:

1. Analisis Kebutuhan Fungsional

Berdasarkan hasil dari studi literatur dan beberapa observasi, dapat disimpulkan bahwa kebutuhan fungsional yang dibutuhkan pada permainan Edantara ialah:

- a) Setiap pulau memiliki beberapa *stage* yang harus dilewati.
- b) Setiap *stage* memiliki 5 soal yang membahas tentang kebudayaan dari suatu provinsi.

- c) Pada akhir *stage*, skor muncul sesuai dengan jumlah soal yang berhasil dijawab dengan benar. Skor maksimal yang bisa didapatkan ialah 100.
- d) Soal akan muncul apabila karakter mengambil sebuah objek yang nanti dijelaskan pada “cara bermain” dalam permainan.
- e) Setiap soal bernilai 20 angka yang untuk jawaban yang benar dan bernilai 0 untuk jawaban yang salah.
- f) Dalam setiap rintangan terdapat beberapa rintangan yang harus dilewati oleh karakter.
- g) Ada batas nyawa sebanyak 4 nyawa. Jika nyawa sampai angka 0, maka karakter akan kehabisan nyawa dan mulai permainan dari awal *stage*.
- h) Tersedia tombol bantuan untuk mengetahui tata cara bermain
- i) Terdapat tombol *mute* dan *unmute* untuk mengaktifkan atau menonaktifkan suara.
- j) Terdapat tombol mulai yang akan membawa pemain ke halaman kesulitan untuk memilih kesulitan, dan setelah tingkat kesulitan sudah dipilih maka pemain akan menuju peta pemilihan *stage*.

2. Analisis Kebutuhan *Software*

Hasil analisis kebutuhan perangkat lunak yang dibutuhkan untuk pembuatan *game* “Edantara” ialah sebagai berikut:

- a) Construct 2 sebagai *game engine* untuk membuat permainan.
- b) C2Buildozer untuk mengkonversi HTML 5 menjadi *file .apk*.
- c) New Mockup 3 untuk membuat storyboard.

- d) YEd – Graph Editor untuk membuat flowchart.
- e) Chrome Browser untuk menjalankan *game* pada PC.
- f) Perangkat Android untuk menjalankan *file* .apk.

3. Analisis Kebutuhan *Hardware*

Perangkat yang digunakan dalam pengujian *game* Edantara ialah *smartphone* yang berbasis sistem operasi Android versi 4.4 (Kitkat) keatas, karena pada saat ini *smartphone* dengan Android dibawah versi 4.4 sudah sulit ditemukan di pasaran.

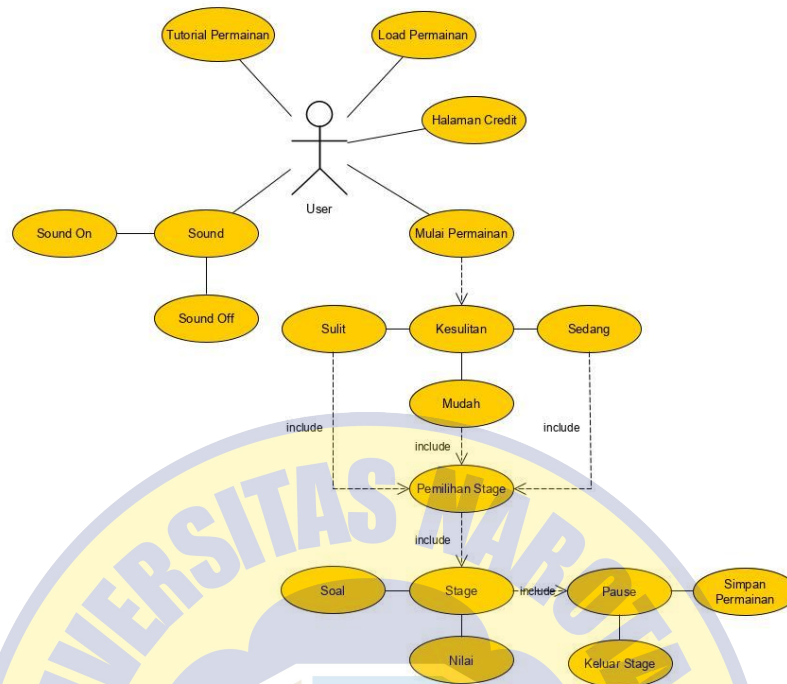
3.2.2 Desain (Perancangan)

1. Desain *Game*

Pada suatu pengembangan *game*, tahap ini merupakan tahap untuk pengembang dalam menentukan jenis atau *genre* permainan, tata cara permainan, alur cerita, perancangan karakter, dan tingkat kesulitan permainan.

2. Desain Sistem dengan UML

Proses pengembangan selanjutnya ialah perancangan *use case* diagram *game* edukasi kebudayaan nusantara oleh *user*. diagram ini nantinya akan digunakan untuk acuan dalam penulisan kode. Dengan adanya *use case* diagram, pengembang akan lebih mudah dalam pembuatan permainan Edantara. Untuk diagramnya bisa dilihat di bawah ini:



Gambar 3.2. Use Case Diagram permainan Edantara

Definisi use case dapat dilihat pada table di bawah.

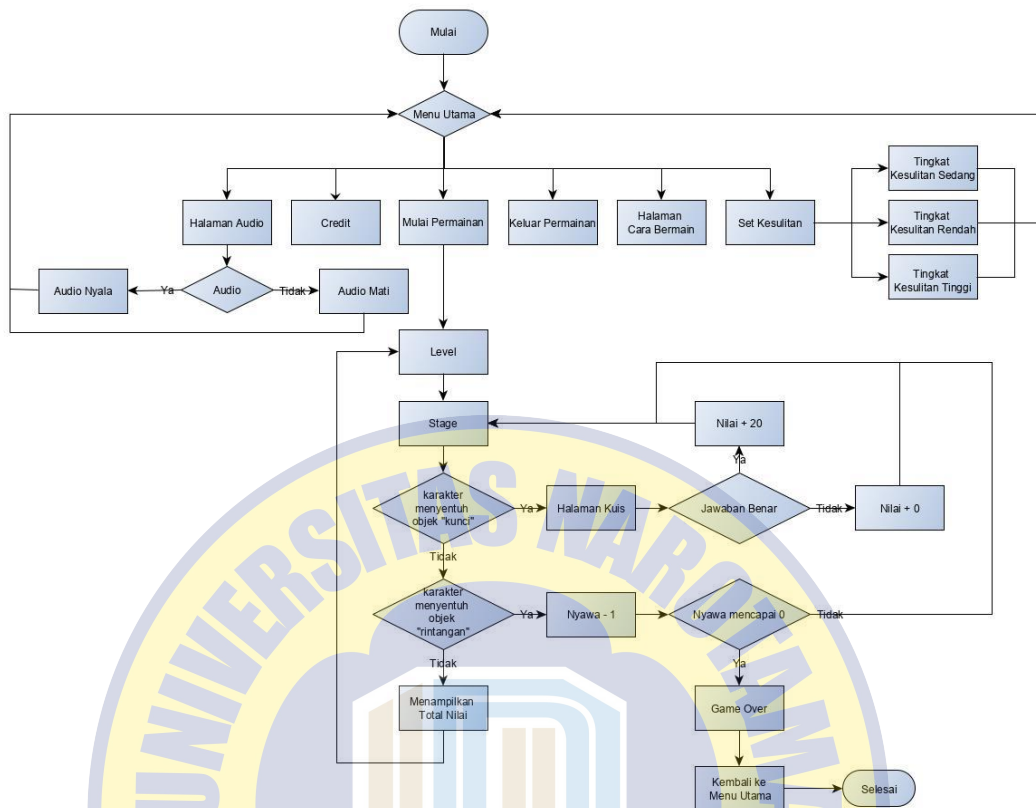
Tabel 3.1 Definisi Use Case PRO PATRIA

No	Use Case	Deskripsi
1	Mulai Permainan	Merupakan fitur untuk masuk dalam permainan Edantara
2	Load Permainan	Merupakan fitur untuk me-load data yang tersimpan pada <i>local storage</i> dari permainan Edantara
3	Sound	Untuk melakukan pengaturan pada audio
4	Tutorial Permainan	Untuk memberikan petunjuk permainan Edantara
5	Halaman Credit	Halaman untuk menempatkan credit <i>source</i> yang digunakan dalam permainan
6	Sound On	Fitur untuk mematikan suara pada permainan
7	Sound Off	Fitur untuk menyalakan suara pada permainan
8	Kesulitan	Halaman untuk memilih kesulitan pada permainan
9	Mudah	Tingkat kesulitan paling rendah pada permainan

10	Sedang	Tingkat kesulitan menengah pada permainan
11	Sulit	Tingkat kesulitan paling tinggi dalam permainan
7	Pemilihan <i>Stage</i>	Merupakan fitur untuk memilih level pada permainan
8	<i>Stage</i>	Merupakan <i>stage</i> dimana <i>user</i> bisa mengendalikan sebuah karakter untuk bisa menghindari rintangan dan menuju pada kuis, yang nantinya nilai kuis akan diakumulasikan pada akhir <i>stage</i>
9	Nilai	Merupakan proses yang terjadi untuk menghitung total nilai kuis yang telah didapat pada akhir <i>stage</i>
10	Soal	Merupakan fitur yang menampilkan soal seputar budaya pada permainan
11	<i>Pause</i>	Merupakan fitur untuk menghentikan permainan sementara
12	Simpan Permainan	Merupakan fitur untuk menyimpan data permainan ke dalam <i>local storage</i>
13	Keluar <i>Stage</i>	Merupakan fitur untuk keluar <i>stage</i> dan kembali ke halaman pemilihan <i>stage</i>

3. Flowchart

Dalam pengembangan permainan edukasi budaya nusantara ini, dibutuhkan flowchart untuk menggambarkan alur sistem yang ada dalam permainan. Adapun flowchart dari permainan “Edantara” ini bisa dilihat di bawah ini:


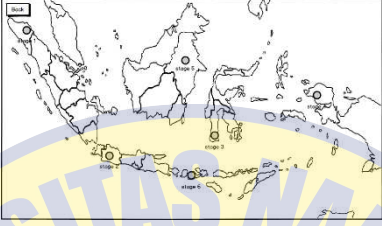
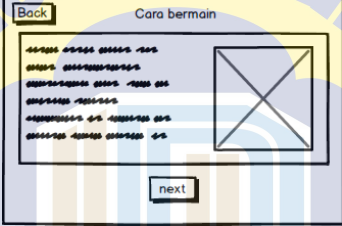
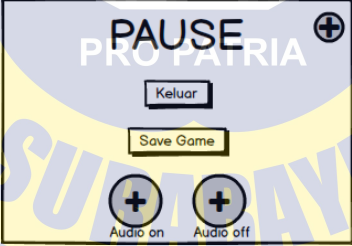
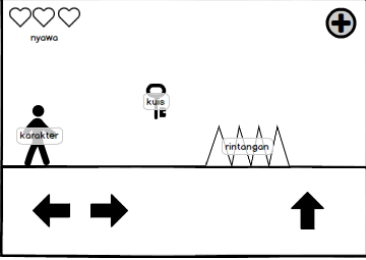


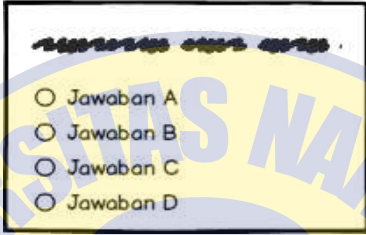
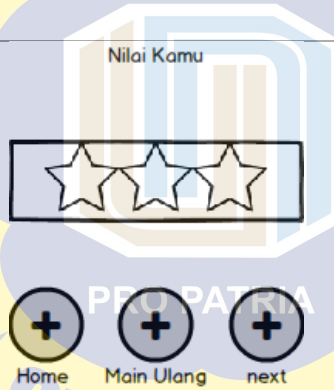
Gambar 3.3. Flowchart Game Edantara

4. Story Board

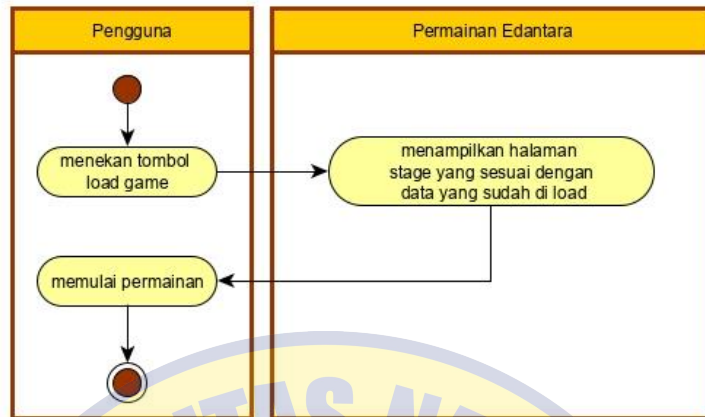
Tabel 3.2 Story Board *Game* "Edantara"

Scene	Nama	Desain	Isi
1	Menu Utama		<ul style="list-style-type: none"> • Logo "Edantara" • Tombol pengaturan audio • Tombol tutorial <i>game</i> • Tombol mulai yang akan menuju pada pemilihan level permainan • Tombol <i>Load Game</i> yang akan melanjutkan <i>Game</i> yang sudah di simpan sebelumnya

2	Tingkat Kesulitan		<ul style="list-style-type: none"> • Tombol untuk mengatur ke tingkat ke kesulitan permainan menjadi mudah, agak sulit, dan sulit.
3	Pemilihan Stage		<ul style="list-style-type: none"> • Gambar backround peta Indonesia sebagai latar untuk penempatan stage. • tombol pada setiap pulau yang akan terhubung pada stage yang berbeda.
4	Tutorial Permainan		<ul style="list-style-type: none"> • Tulisan dan gambar yang berisi cara untuk memainkan permainan, seperti penjelasan tombol dalam game. • Tombol untuk kembali ke menu utama. • Tombol untuk pindah ke tutorial selanjutnya
5	Pause		<ul style="list-style-type: none"> • Tombol untuk keluar permainan. • Tombol untuk menyimpan permainan • Tombol untuk menyalakan dan mematikan audio • Tombol untuk kembali pada permainan
6	Stage		<ul style="list-style-type: none"> • Tombol arah kanan untuk karakter berjalan ke depan. • Tombol arah kiri untuk karakter berjalan ke belakang. • Tombol panah atas untuk karakter meloncat. • Terdapat objek “kunci” yang jika tersentuh oleh karakter, maka akan

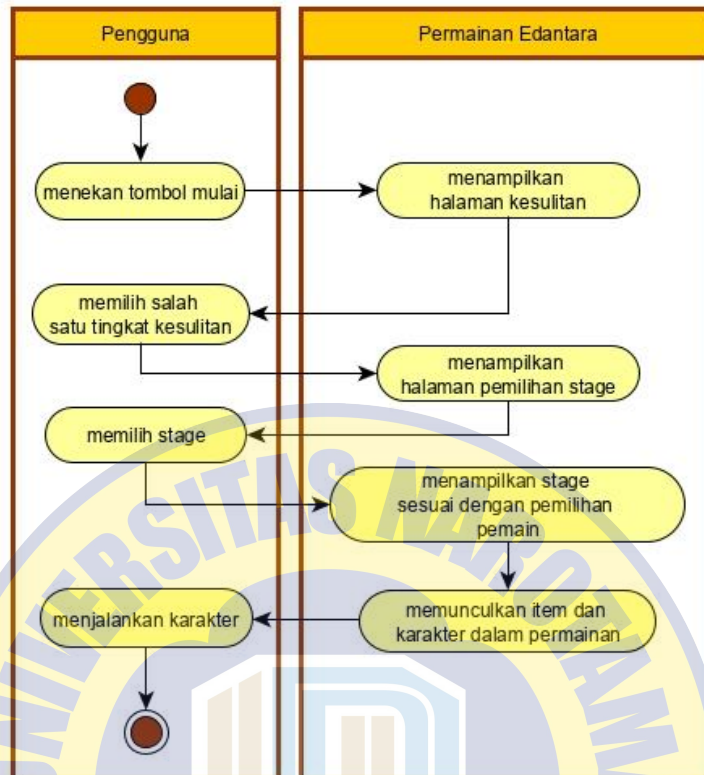
			<p>langsung menuju ke halaman kuis.</p> <ul style="list-style-type: none"> • Terdapat objek “rintangan” yang jika tersentuh karakter maka nyawa akan berkurang. • Tombol <i>Pause</i> yang akan menghentikan permainan
7	Soal	<p>Soal</p> 	<ul style="list-style-type: none"> • Tuliskan soal untuk dasar menjawab pertanyaan. • Tombol jawaban A • Tombol jawaban B • Tombol jawaban C • Tombol jawaban D
8	Halaman Nilai / <i>Finish</i>	<p>Nilai Kamu</p> 	<ul style="list-style-type: none"> • Nilai akumulasi dari semua soal dalam stage akan ditampilkan dalam bentuk bintang. • Tombol main ulang permainan • Tombol menuju ke level selanjutnya • Tombol kembali ke halaman utama

5. Activity Diagram



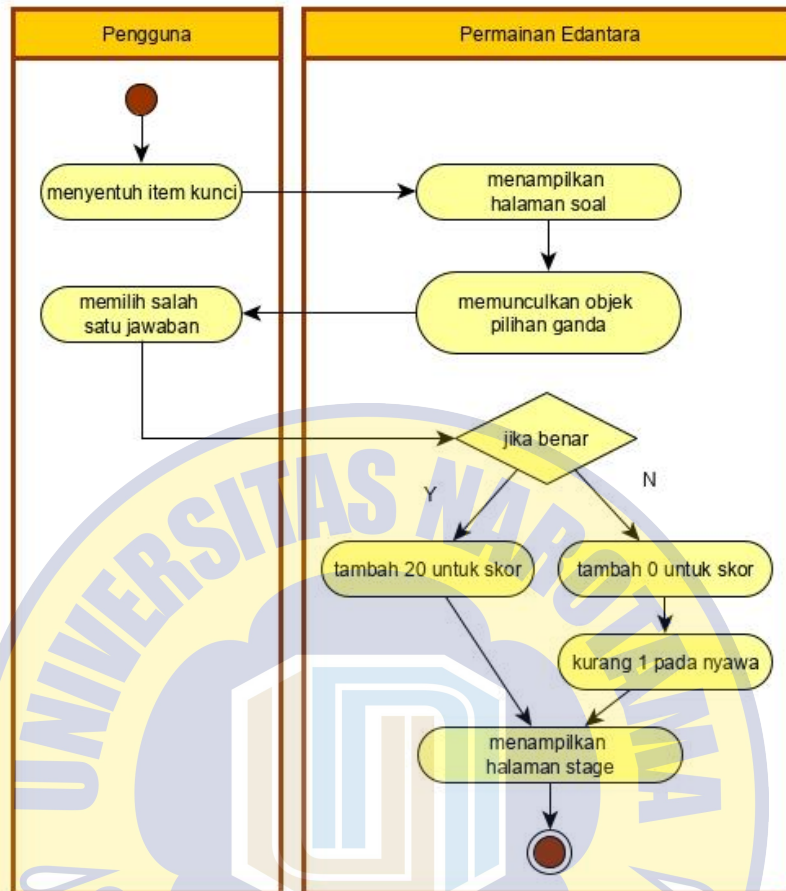
Gambar 3.4. Activity Diagram *Load Permainan*

Saat pemain menekan tombol *load* permainan, sistem pada permainan “Edantara” akan memeriksa data yang terdapat pada *local storage* dan menyesuaikan permainan dengan data yang sudah diambil, sehingga pemain bisa melanjutkan *stage* tanpa harus memulai permainan dari awal.



Gambar 3.5. Activity Diagram memulai permainan

Untuk memulai permainan, pemain harus menekan tombol “mulai” pada menu utama yang akan menampilkan halaman kesulitan. Pada halaman ini pemain bisa memilih tingkat kesulitan sesuai hati sesuai dengan kemampuan. Setelah memilih tingkat kesulitan, halaman pemilihan *stage* yang berlatar belakang peta Indonesia akan muncul dan menyajikan tombol pada beberapa provinsi. Pada saat pemain menekan tombol pada provinsi tertentu, *stage* yang keluar sesuai dengan provinsi letak tombol yang di tekan berada. Di dalam *stage*, pemain akan menjalankan karakter dan mempunyai misi menjawab pertanyaan tentang kebudayaan Indonesia dengan benar.

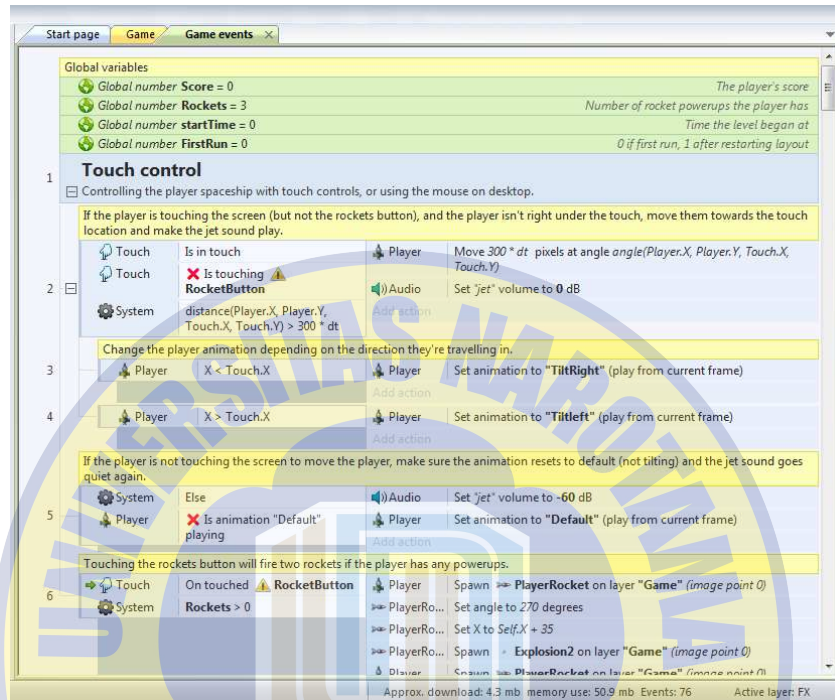


Gambar 3.6. Activity Diagram pada Halaman Soal

3.2.3 Code Generation (Pengkodean)

Adalah tahap mengubah hasil dari tahap perancangan menjadi bentuk yang bisa dibaca dan dieksekusi oleh mesin. Dalam permainan ini, pengkodean dilakukan di dalam *game engine* Construct 2. Construct 2 menggunakan bahasa pemrograman HTML 5 untuk menjalankan seluruh sistem permainan. Di dalam Construct 2 pengkodean dilakukan pada *Event Sheet*, *Event Sheet* ialah pusat untuk mengendalikan seluruh logika dalam permainan. Termasuk proses *looping* dan *conditions* yang akan digunakan dalam permainan.

1) Event Sheet



Gambar 3.7. Halaman event sheet pada Construct 2

Event sheet ialah kumpulan dari beberapa *event* yang bisa diubah melalui *event sheet view*. *Event sheet* juga bisa diganti nama atau dihapus, jadi memudahkan para developer untuk mengatur *event* yang diinginkan. Semua *event* yang ada dalam projek Construct 2 akan ditampilkan pada *project bar*. Ada 3 komponen utama yang terdapat dalam *event sheet* yaitu *variable*, *conditions*, dan *action*.

Global number	startTime = 0		Time the level began at
Global number	FirstRun = 0		0 if first run, 1 after restarting layout
Touch control			
<input type="checkbox"/> Controlling the player spaceship with touch controls, or using the mouse on desktop.			
If the player is touching the screen (but not the rockets button), and the player isn't right under the touch location and make the jet sound play.			
<input type="checkbox"/> Touch	Is in touch	<input type="checkbox"/> Player	Move 300 * dt pixels at angle angle(Player.X, Player.Y, Touch.X, Touch.Y)
<input type="checkbox"/> Touch	<input checked="" type="checkbox"/> Is touching	<input type="checkbox"/> Audio	Set 'jet*' volume to 0 dB
	RocketButton		

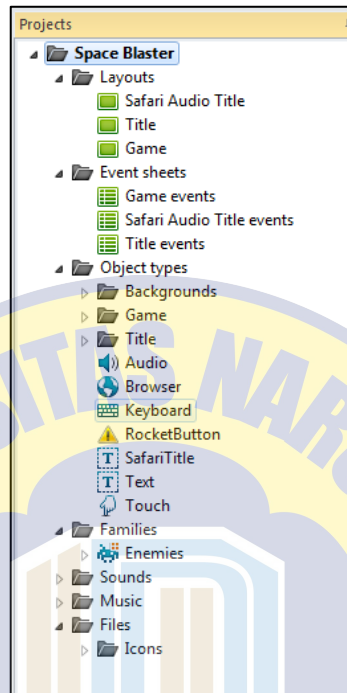
Gambar 3.8. Variabel, conditions, dan actions pada event sheet

- a) *Variable*, digunakan untuk menyimpan nilai data sesuai dengan tipe variabelnya pada semua bahasa pemrograman. Variabel sangat dibutuhkan dalam proses pembuatan sebuah *game*. Contohnya misalkan, sebuah karakter mempunyai variabel bernama nyawa dengan *value* 3, jika ingin halaman *game over* muncul maka *value* nyawa harus bernilai 0. Ini ialah peran variabel untuk memunculkan *page* lain ketika *game* berjalan.
- b) *Conditions*, sebuah kondisi ialah sebuah percabangan yang memiliki beberapa kemungkinan yang bisa terjadi. Kondisi yang muncul nantinya tergantung berdasarkan objek. Jadi setiap objek mempunyai kondisi yang berbeda.
- c) *Actions*, merupakan sebuah hasil yang didapatkan ketika sebuah kondisi terpenuhi. *Actions* bisa mempengaruhi objek atau variabel yang dimaksud.

2) *Layout*

Layout ialah tampilan dari kumpulan objek yang telah diatur sebelumnya. *Layout* dapat menampilkan sebuah *game level*, menu atau judul layer. Dalam fungsi

yang lain *layout* bisa juga disebut *scene* atau *stage*. *Layout* bisa ditambah, diubah, dan dihapus melalui *project bar*.



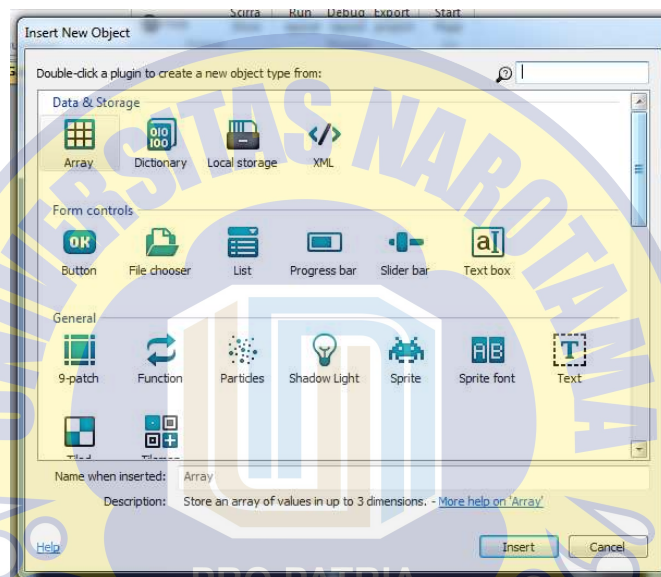
Gambar 3.9. Project Bar pada Construct 2



Gambar 3.10. Layout beserta objek pada Construct 2

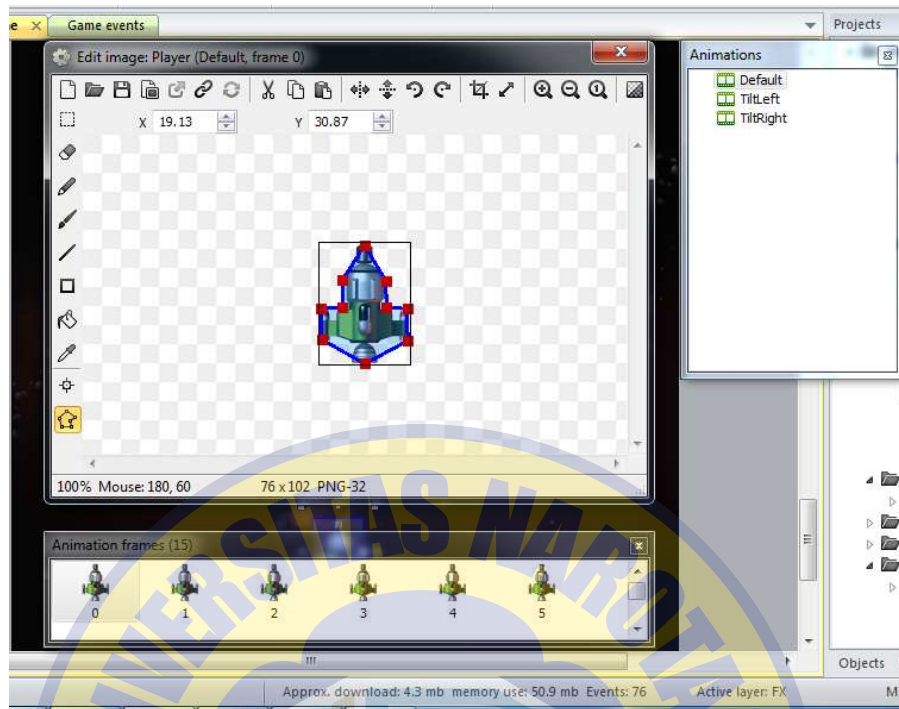
3) Objek

Pada Construct 2, objek paling banyak digunakan dalam sebuah project. Hal yang paling banyak dilihat pada Construct 2 ialah ditampilkan oleh objek, dan ada juga sebagian objek yang tersembunyi untuk kepentingan tertentu. Saat membuat objek baru, hal pertama yang dilakukan ialah memilih tipe objek, contohnya: *sprite*.



Gambar 3.11. Berbagai jenis objek pada Construct 2

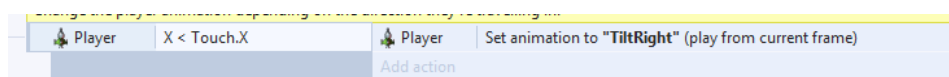
Objek yang sudah dimasukkan ke dalam projek, bisa juga diubah atau dijadikan gambar bergerak dengan memasukkan beberapa gambar dan mengatur *speed* perpindahan antara gambar satu dengan yang lain agar gambar bergerak dengan lebih lembut. Fitur ini disebut *animation frame*.



Gambar 3.12. Frame animasi pada objek

4) *Event*

Dalam Construct 2, *event* berada di dalam *event sheet* yang merupakan sebuah baris berisi *conditons* dan *actions*. Satu *event* bisa mempunyai lebih dari satu *actions*. *Event* ialah fitur utama yang dimiliki oleh Construct 2, sebagai pengganti dari proses *scripting* atau penulisan program yang rawan akan terjadinya eror dan *syntax* yang rumit. *Event* dapat mendefinisikan bagaimana *game* bekerja melalui logika sistem blok.



Gambar 3.10 Contoh *event* dalam *event sheet*

Pada *event* yang ditunjukkan pada gambar 13, merupakan *event* dengan satu *action*. Event di atas menunjukkan bahwa jika posisi x pesawat kurang dari titik sentuh x, maka animasi objek pesawat akan beralih menjadi animasi “TiltRight” yang dimulai dari frame yang tereksekusi saat itu.

3.2.4 Test (Pengujian)

Adalah tahap uji coba permainan untuk memastikan apakah semua perintah bisa berfungsi dengan baik dan eksternal input bisa memberikan keluaran sesuai dengan yang diinginkan. Ada beberapa hal yang menjadi fokus dalam tahap pengujian:

1. Berbagai fungsi dalam permainan yang tidak sesuai seperti yang diinginkan.
2. Interface yang tidak pada tempatnya.
3. Kesalahan kinerja.
4. *Compatibility* pada terhadap berbagai macam versi Android.

1. Pengujian Fungsi

Pada pengujian fungsi pada permainan “Edantara” digunakan metode *white box*. Pengujian dengan metode *white box* bertujuan untuk mengamati berbagai macam fungsi yang digunakan dalam perangkat lunak seperti input, interaksi antar objek, dan perubahan *value* variabel. Kelebihan dalam metode ini ialah:

- a) Efisiensi dalam menemukan eror dan penyelesaian sangat tinggi
- b) Memungkinkan menemukan kesalahan yang lebih banyak.

- c) Memerlukan pengetahuan tentang program yang dipakai dalam perangkat lunak

Dalam pengujian melalui metode *white box*, tester dituntut untuk setidaknya mengerti tentang internal perangkat lunak yang sedang diuji. Kekurangan dalam metode *white box* ialah diperlukannya akses kode. Jadi saat menguji sebuah perangkat lunak, diperlukan file *code program*.

2. Pengujian Interface

Dalam pengujian *interface*, hal yang harus diperhatikan ialah *assets*, *background*, dan karakter. *Assets* merupakan semua yang ada di dalam permainan, seperti efek suara, *sprites*, 3D Models, modul dan bahkan suatu proyek dapat digunakan oleh *game engine* sebagai *assets*. Contoh berbagai macam *assets*:

- a) Karakter
- b) Objek
- c) Lingkungan
- d) Alat – alat
- e) Music *background*
- f) Efek suara
- g) *Icons*

3. Pengujian Kinerja

Dalam sebuah perangkat lunak, kinerja perangkat lunak pada saat dijalankan harus dalam keadaan yang optimal. Optimal yang dimaksud ialah

seluruh bagian dari sistem dan *interface* berjalan sesuai dengan apa yang diinginkan oleh pengembang. Hal – hal yang perlu diperhatikan dalam pengujian kinerja ialah:

- a) Ketepatan perangkat lunak dalam merespon sebuah input dari pengguna.
- b) Perubahan *value* variabel berdasarkan interaksi antar objek.
- c) Perpindahan *page* yang dipicu oleh input dari pengguna atau fungsi dalam permainan.

4. Pengujian *Compatibility*

Compatibility ialah kemampuan perangkat lunak untuk berjalan pada berbagai macam sistem operasi *device*. Berbagai perangkat Android tentunya memiliki konfigurasi yang berbeda – beda, seperti *pixel per inch* (ppi) pada layar dan resolusi yang dimiliki oleh masing – masing *smartphone*. Game “Edantara” dibuat untuk versi Android 4.4 keatas, jadi *smartphone* yang menjadi fokus dalam tahap pengujian *compatibility* ialah yang mempunyai sistem operasi diatas versi Android Kitkat. Semakin banyak versi Android yang bisa menjalankan *game* “Edantara”, maka akan semakin bagus *compatibility* yang dimiliki. Pada pengujian ini, terdapat beberapa macam pengujian.

- a) Pengujian terhadap *smartphone* android yang mempunyai lebar layar yang berbeda.
- b) Pengujian terhadap *smartphone* yang memiliki versi Android yang berbeda.

- c) Pengujian terhadap *smartphone* yang mempunyai resolusi layar yang berbeda.

Pengujian *compatibility* berguna dalam penentuan set lingkungan yang akan dipakai untuk menjalankan *game* “Edantara”.

