

## BAB 3 METODOLOGI PENELITIAN

### 3.1 Metodologi Penelitian

Tahapan-tahapan penelitian untuk merancang dan membangun Maze Game yaitu studi literatur, perancangan game, pembangunan game, pengujian game, evaluasi dan penulisan laporan.

#### A Studi Literatur

Pada tahap studi literatur ini dilakukan pencarian referensi, teori dan pendalaman materi mengenai materi-materi yang berhubungan dengan game labirin yang dibuat, yaitu Procedural Content Generation, Cellular Automata, dan Marching Square.

#### B Perancangan Game

Pada tahap ini perancangan terhadap Game dilakukan dengan membuat *flowchart* dan *mockup*.

#### C Pembangunan Game

Pada tahap ini game mulai dibangun sesuai rancangan di unity dengan C-Sharp (C#) serta mengimplementasi PCG Cellular Automata.

#### D Pengujian Game

Pada tahap ini dilakukan pengujian terhadap game yang telah dibangun untuk mencari *error* dan *bug* serta apakah game telah mencapai target yang diinginkan. Beberapa pengujian dilakukan untuk melihat hasil *map* yang telah di generasi serta penempatan lawan atau musuh, lalu seluruh hasil ujian dipastikan tidak sama atau acak, pengujian dilakukan menggunakan *Blackbox testing*.

## **E Evaluasi**

Pada tahap ini akan dipastikan apakah *map* yang telah dibuat menggunakan algoritma CA dan MS sudah bagus dan menyimpulkan hasil evaluasi dari pendapat pemain menggunakan *Game User Experience Satisfaction Scale*.

## **F Penulisan laporan**

Tahap ini mencakup laporan mengenai cara implementasi Procedural Content Generation Cellular Automata dan Marching Square pada Maze Game dan penjelasan mengenai cara kerja algoritma serta *assets* yang digunakan dalam pembuatan game. Melalui *Game User Experience Satisfaction Scale* dikumpulkan tingkat kepuasan pemain setelah game di uji coba oleh pemain baik itu sampai selesai atau tidak.

### **3.2 Perancangan Aplikasi**

#### **3.2.1 Struktur Permainan**

Judul : *Perdita Perdita* yang berarti *lost* dalam bahasa Inggris dan tersasar dalam bahasa Indonesia, kata tersebut berasal dari bahasa Latin.

#### **A Player Type**

Game dimainkan oleh satu pemain saja atau *Single-player*.

#### **B Objectives**

Objektif pemain dalam game ini adalah untuk mengumpulkan sepuluh berlian dan menghindari musuh-musuh yang ada. Lalu setelah pemain berhasil mengumpulkan sepuluh berlian jalan keluar akan muncul, pemain harus mencapai jalan keluar tanpa menyentuh musuh atau pemain harus mengulang lagi.

#### **C Procedures**

1. Tampilan awal setelah aplikasi dijalankan adalah *Main Menu* dimana didalamnya terdapat judul dari game dan tiga opsi yang dapat dipilih oleh pemain, yaitu *Start*, *Instruction* dan *Exit*

2. Opsi *Exit* berfungsi untuk menutup aplikasi game apabila dipilih.
3. Opsi *Instruction* berfungsi untuk menampilkan instruksi mengenai game kepada pemain apabila dipilih.
4. Opsi *Start* berfungsi untuk memulai game dimana pemain akan diletakan di posisi *random* didalam *map*.
5. Pemain harus mencari sepuluh *item* atau berlian sambil menghindari musuh yaitu *zombie*.
6. Pemain yang telah mengumpulkan sepuluh berlian akan mendapatkan pesan di atas layar bahwa jalan keluar muncul di suatu tempat dalam *map* dan pemain harus mencari jalan keluar tersebut.
7. Pemain setelah berhasil menemukan jalan keluar yang muncul tanpa terkena sentuh musuh akan dianggap telah memenangkan game. Lalu pemain akan dibawa ke tampilan *main menu*.
8. Jika pemain tersentuh oleh musuh maka pemain dianggap kalah dan akan dibawa ke tampilan *lose* dimana pemain diberi dua opsi, yaitu *Menu* dan *Retry*.
9. Opsi *Menu* pada tampilan *lose* berfungsi untuk membawa pemain ke tampilan *main menu*.
10. Opsi *Retry* pada tampilan *lose* berfungsi mengulang game dari awal dengan *map* baru serta posisi *item* dan *zombie* yang telah diacak.

#### **D Rules**

1. Pemain hanya dapat menghindari musuh dan tidak dapat menyerang.
2. Pemain kalah apabila tersentuh oleh musuh.
3. Pemain dipindah ke *scene* kalah apabila tersentuh oleh musuh dan pemain mempunyai opsi untuk mengulang atau kembali ke *menu*.
4. Jalan keluar hanya akan muncul apabila pemain berhasil mengumpulkan 10 berlian.

5. Pemain dipindah ke *scene* apabila berhasil mengumpulkan sepuluh berlian dan mencapai jalan keluar.
6. Game tidak ada tingkat kesusahan atau *level*.
7. Pemain dapat bergerak menggunakan w, a, s, d, mouse
8. Pemain dapat melakukan *sprint* atau lari kencang dengan *shift*.

#### **E Resources**

*Resource* yang digunakan pada game ini hanya ada satu yaitu berlian yang harus dikumpulkan oleh pemain untuk memunculkan jalan keluar.

#### **F Conflict**

Konflik yang terdapat pada game ini adalah musuh berupa zombie yang di *spawn* secara random didalam map bersamaan dengan berlian. Dimana musuh tersebut akan mencoba untuk mengejar apabila posisi pemain terlalu dekat.

#### **G Boundaries**

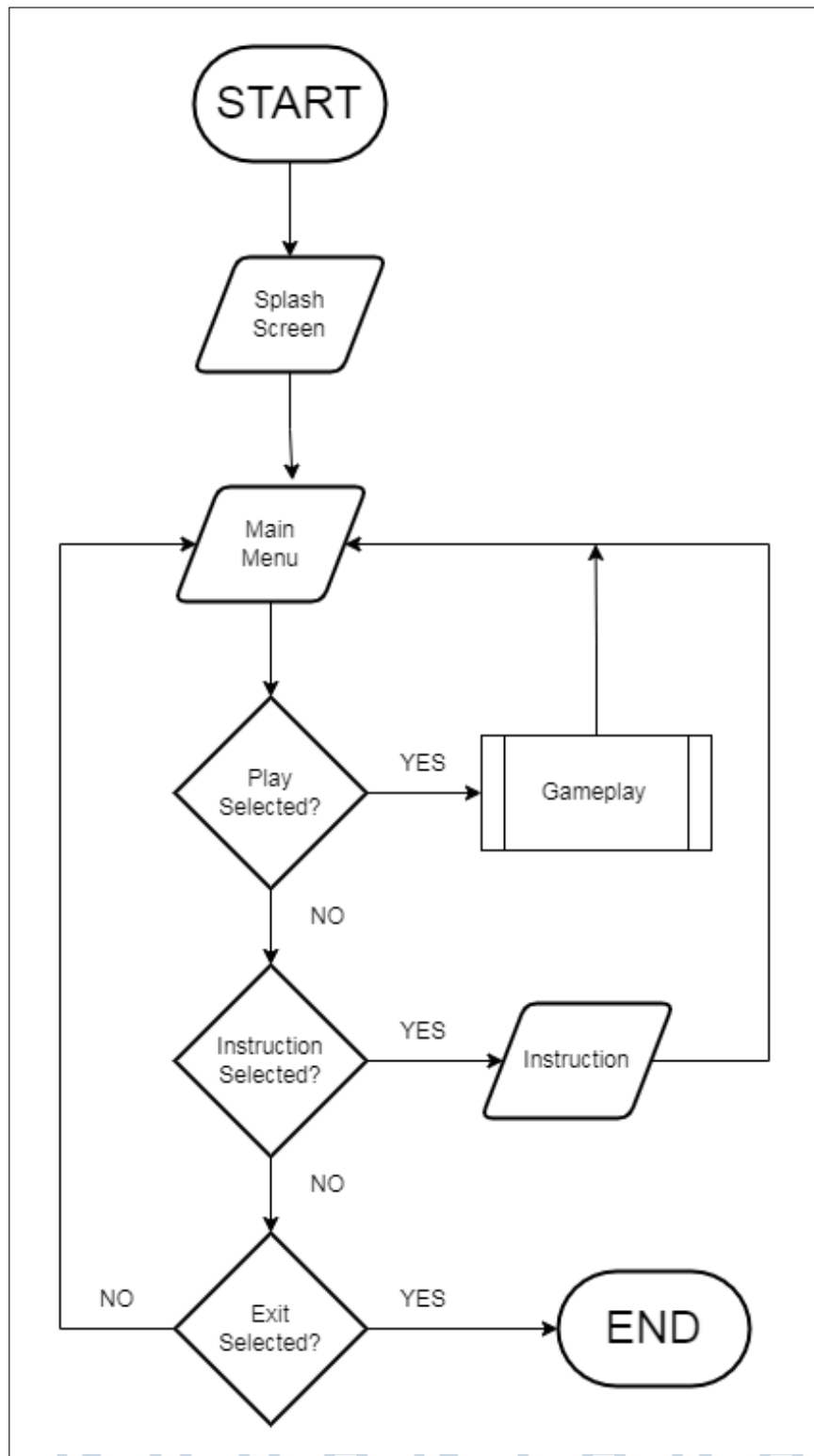
Batasan yang diberikan pada game ini adalah *map* dimana pemain tidak dapat keluar melewati batasan *map* yang ditentukan.

#### **H Outcome**

Tujuan akhir pemain di dalam game ini adalah dimana pemain berhasil mengumpulkan sepuluh berlian dan mencapai jalan keluar sehingga pemain dinyatakan menang.

### **3.2.2 Flowchart**

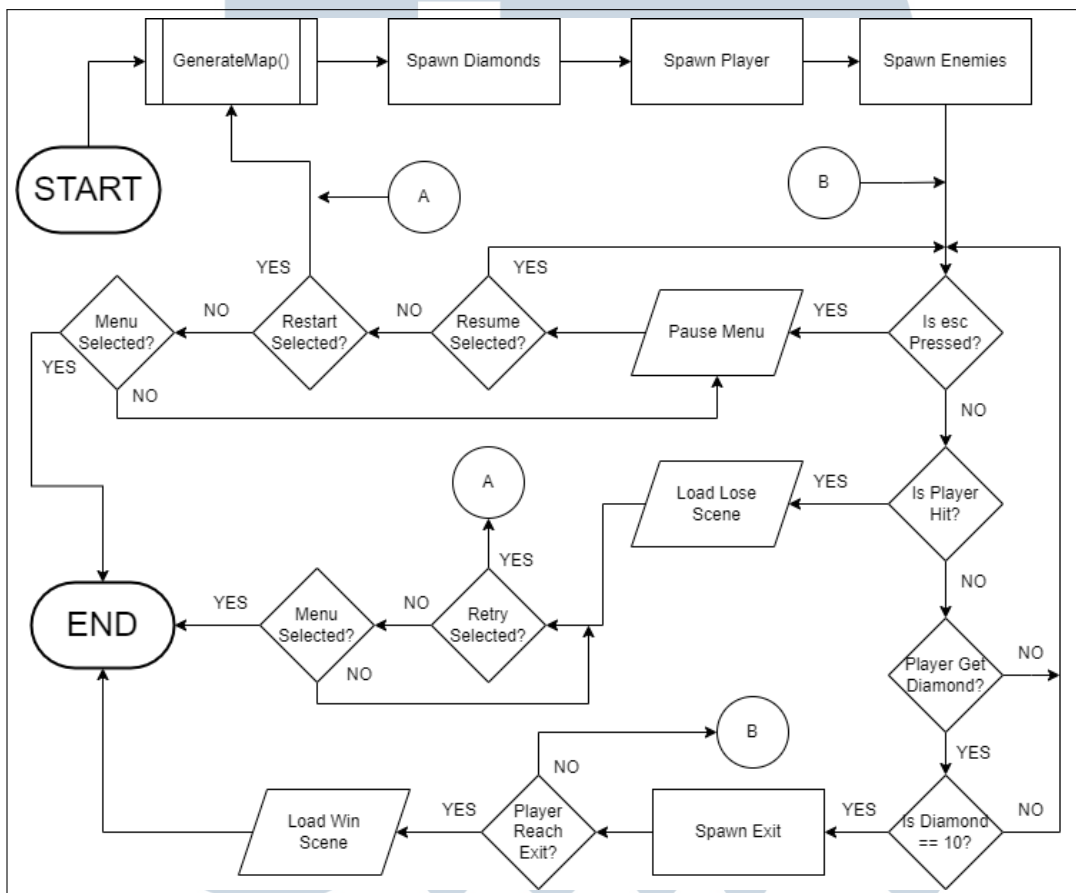
*Flowchart* dirancang sebagai landasan pengerjaan program. Berikut adalah perancangan permainan secara umum



Gambar 3.1. Flowchart Aplikasi

Gambar 3.1 menunjukkan alur dari game, *splash screen* Unity merupakan

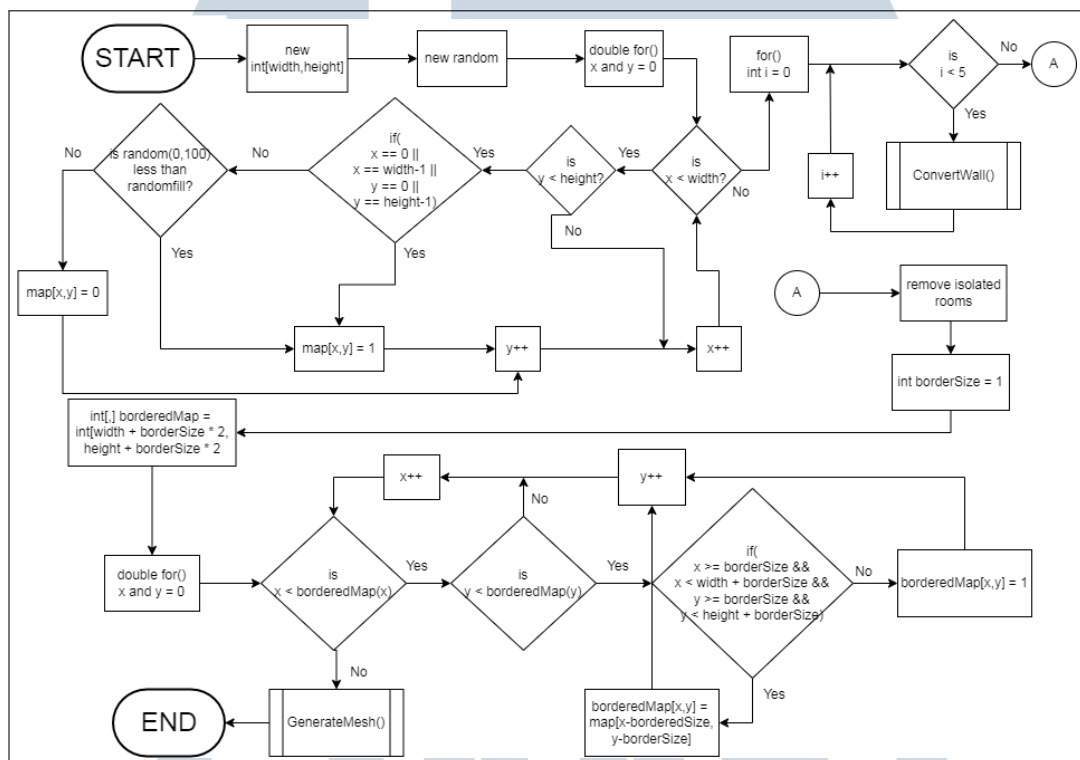
hal pertama yang dilihat pemain saat game dijalankan, setelah itu pemain akan dibawa ke *main menu*. Didalam *main menu* pemain diberikan tiga pilihan yaitu *play*, *instruction* dan *exit*. *Play* akan membawa pemain kedalam *gameplay* yang lalu akan kembali ke *main menu* baik itu karena pemain keluar atau menyelesaikan permainan. *Instruction* akan menampilkan instruksi untuk pemain dalam memainkan permainan, setelah itu pemain dapat kembali ke *main menu*. *Exit* mempunyai fungsi untuk menutup aplikasi.



Gambar 3.2. Flowchart *Gameplay*

Gambar 3.2 menunjukkan *flowchart* dari *gameplay*. Ketika pemain memilih *play* dari *main menu* *map* akan digenerate lalu *diamond*, *player* dan musuh akan di munculkan pada posisi acak. Pemain dapat men-*pause* game, pada *pause menu* pemain dapat mengulang permainan dari awal generasi konten, melanjutkan permainan atau kembali ke *main menu*. Objektif dari game ini adalah untuk mencari sepuluh berlian lalu mencapai jalan keluar tanpa menyentuh musuh sekalipun. Apabila pemain terkena musuh, maka pemain akan dibawa ke *scene*

kalah dimana pemain dapat mengulang permainan dengan generasi konten yang baru atau kembali ke *main menu*. Saat pemain berhasil mengumpulkan sepuluh berlian maka sebuah pesan akan muncul untuk memberitahukan pemain bahwa jalan telah muncul di posisi acak dan pemain harus mencari jalan keluar tersebut. Pemain akan dibawa ke *scene* menang apabila pemain berhasil mencapai jalan keluar, lalu pemain dapat kembali ke dalam *main menu*.

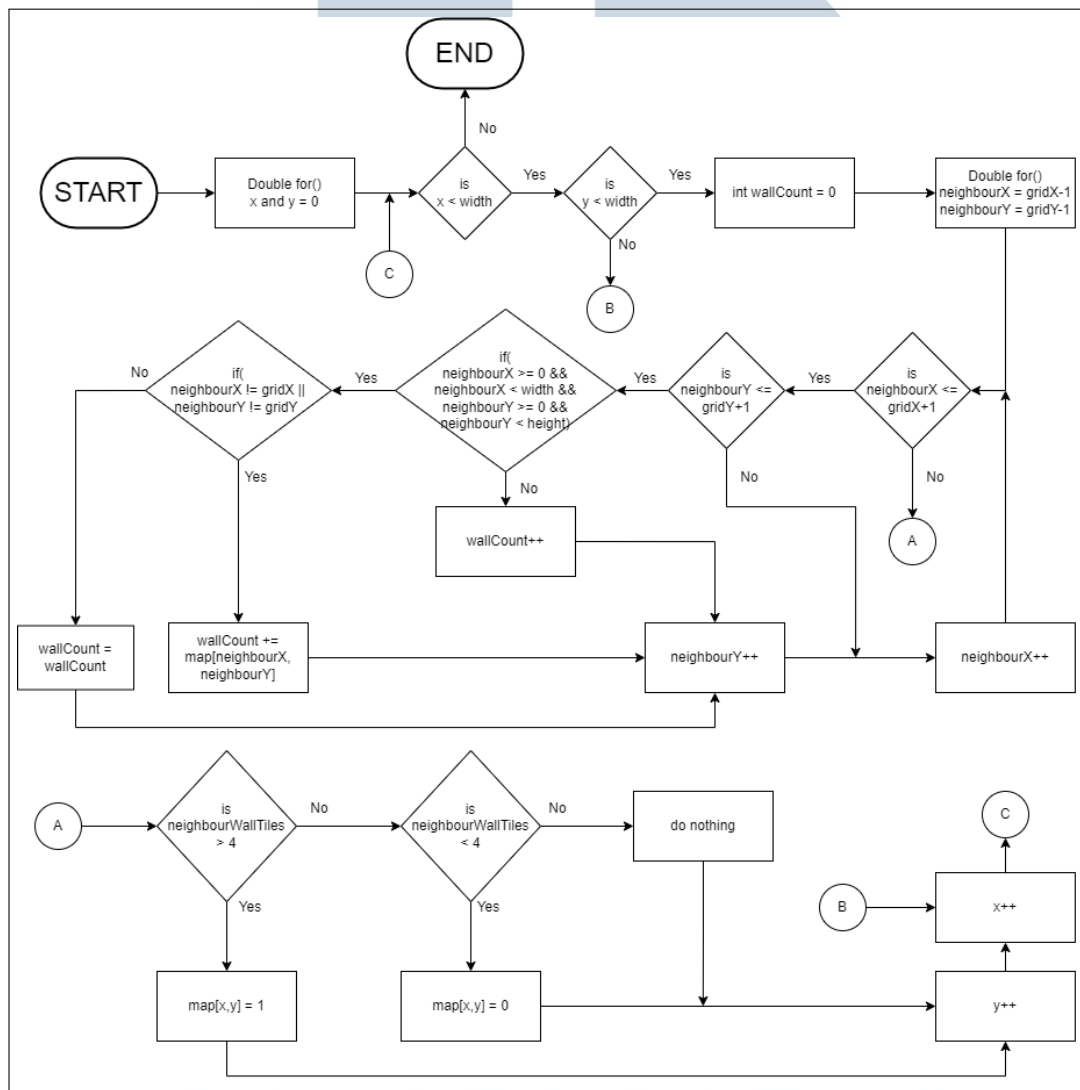


Gambar 3.3. Flowchart GenerateMap

Gambar 3.3 menunjukkan proses pembentukan *map* menggunakan PCG Cellular Automata dan Marching Square. Pertama dilakukan inialisasi *random* menggunakan *system* dan ukuran *map* berupa *width* dan *height* yang telah ditentukan yaitu 164x92. Setelah itu akan dijalankan dua *for()* *loop* untuk mengecek seluruh *width* dan *height*, apabila sel yang telah dicek berada pada tepi *map* maka sel tersebut dianggap *border* dan akan dijadikan *wall* atau tembok. Apabila sel tidak berada di tepi *map* maka akan dilakukan *random* dengan skala 1-100 dan jika hasil *random* lebih rendah dari persentase yang ditentukan *developer* maka sel itu akan dijadikan tembok dan sebaliknya jika hasil *random* lebih besar maka sel itu adalah lantai atau *floor*.

Setelah seluruh sel telah melalui pengecekan *random* akan dilakukan *loop*

sebanyak lima kali untuk membaguskan bentuk dari *map* menggunakan Cellular Automata. Lalu ruangan yang terisolasi akan dijadikan tembok dan akan digunakan *loop* untuk memberikan batasan pinggir pada *map* sesuai dengan ukuran yang ditentukan lalu langkah terakhir adalah pembuatan *mesh*.

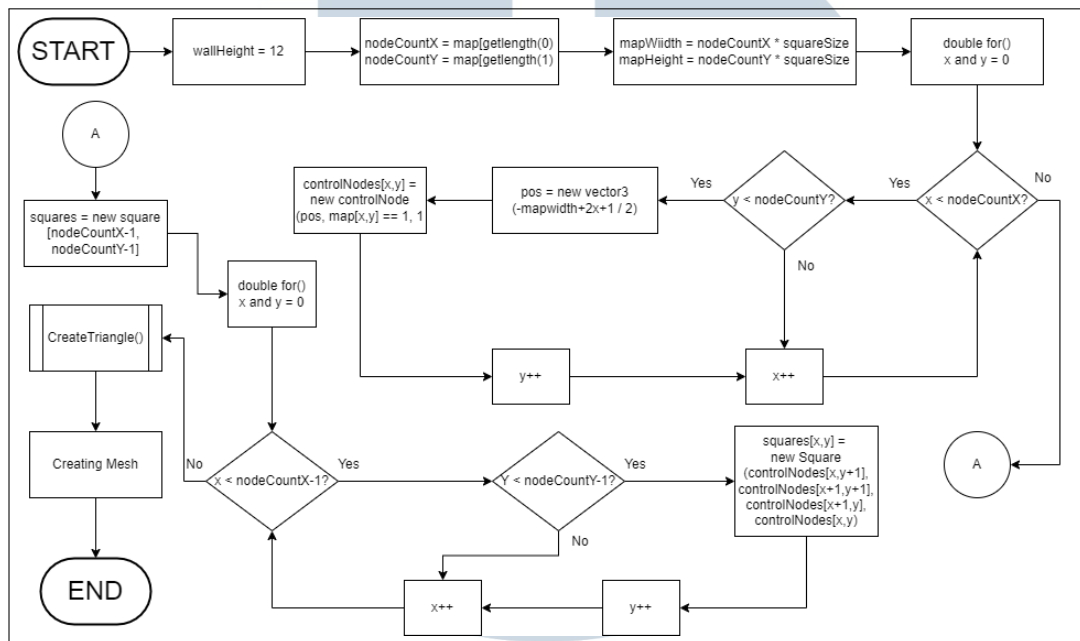


Gambar 3.4. Flowchart ConvertWall (Cellular Automata)

Gambar 3.4 menunjukkan ConvertWall yaitu algoritma Cellular Automata yang berfungsi untuk mengubah sel dari lantai menjadi tembok dan sebaliknya jika memenuhi sebuah ketentuan atau *rule*. Pertama akan dijalankan *loop* untuk semua sel dan di lakukan pengecekan dalam bentuk 3x3 terhadap sel-sel yang mengelilingi sel yang akan dicek lalu wallCount akan ditambah dengan  $map[neighbourX,neighbourY]$ , apabila  $map[neighbourX,neighbourY]$  bernilai 1



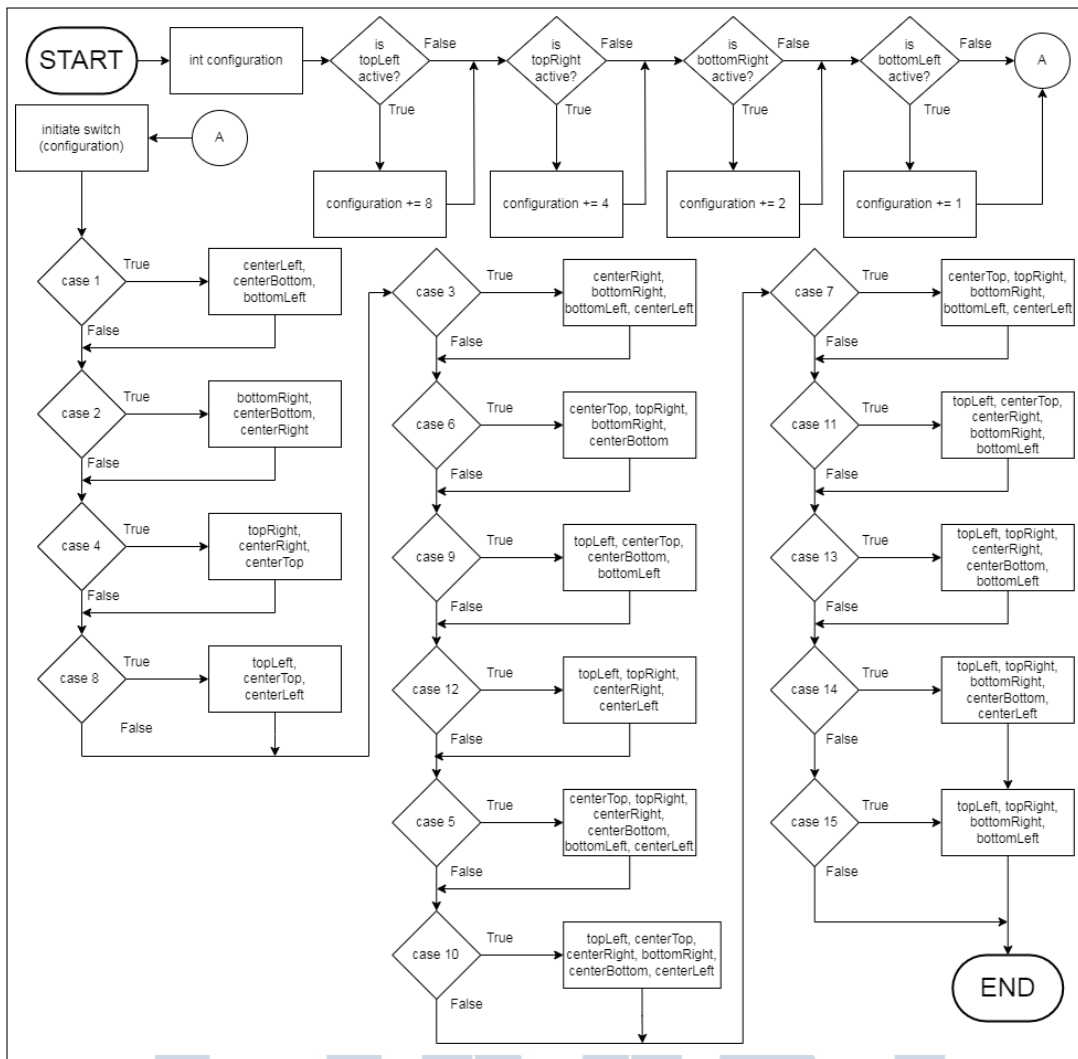
maka sel tersebut adalah tembok. Setelah pengecekan sel tetangga akan ditentukan apabila wallCount kurang dari 4 maka sel tetap dianggap sebagai lantai dan sebaliknya jika wallCount lebih dari 4 maka sel tersebut akan dijadikan tembok. Proses ini dilakukan untuk seluruh sel *map*.



Gambar 3.5. Flowchart GenerateMesh

Gambar 3.5 menunjukkan GenerateMesh yang berfungsi untuk membuat *mesh* pada *map* yang telah dibuat menggunakan Cellular Automata. Pertama di inialisasi untuk tinggi tembok, nodeCountX, nodeCountY, mapWidth dan mapHeight lalu digunakan *loop* untuk memberikan posisi untuk setiap controlNodes serta kondisi controlNodes apabila aktif atau tidak. Setelah itu akan di inialisasi *square* untuk memberikan kotak berdasarkan controlNodes yang dibuat menggunakan *loop*. Setelah itu akan digunakan Marching Square untuk menentukan nilai dari kotak untuk dibentuk segitiga lalu setelah itu adalah pembentukan *mesh*.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA








Gambar 3.6. Flowchart CreateTriangle (Marching Square)

Gambar 3.6 menunjukkan *flowchart* Marching Square. Pertama dilakukan inialisasi konfigurasi lalu kotak yang telah dibentuk dari controlNodes sebelumnya dicek kondisi *active* dengan nilai maksimum yaitu 15. Setelah nilai konfigurasi sudah didapatkan akan di gunakan switch dengan 15 kemungkinan untuk menghubungkan nodes. Contohnya pada kemungkinan konfigurasi bernilai 1 atau *case 1* didapat dari nilai *binary* 0 0 0 1 dan pada kotak maka urutan keempat adalah controlNode bawah kiri. Peraturan Marching Square untuk konfigurasi bawah kiri adalah menghubungkan node dari bagian tengah kiri, bawah kiri dan tengah bawah.

### 3.2.3 Penggunaan Asset

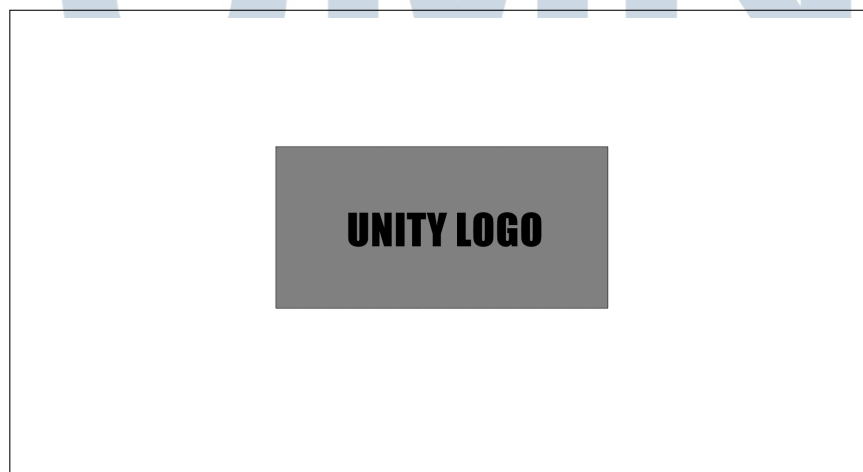
Tabel 3.1. Tabel Penggunaan *Assets*

Nama <i>Assets</i>	Sumber <i>Assets</i>	Gambar <i>Assets</i>
Mini First Person Controller	<a href="https://assetstore.unity.com/">https://assetstore.unity.com/</a> by Simon Pasi	
Zombie Model	<a href="https://assetstore.unity.com/">https://assetstore.unity.com/</a> by PxlTiger	
Diamond Model	<a href="https://assetstore.unity.com/">https://assetstore.unity.com/</a> by AurynSky	
Textures	<a href="https://pixabay.com/">https://pixabay.com/</a>	
Sounds	<a href="https://www.freepik.com/">https://www.freepik.com/</a>	

Tabel 3.1 menunjukkan *assets* yang digunakan dalam pembangunan game.

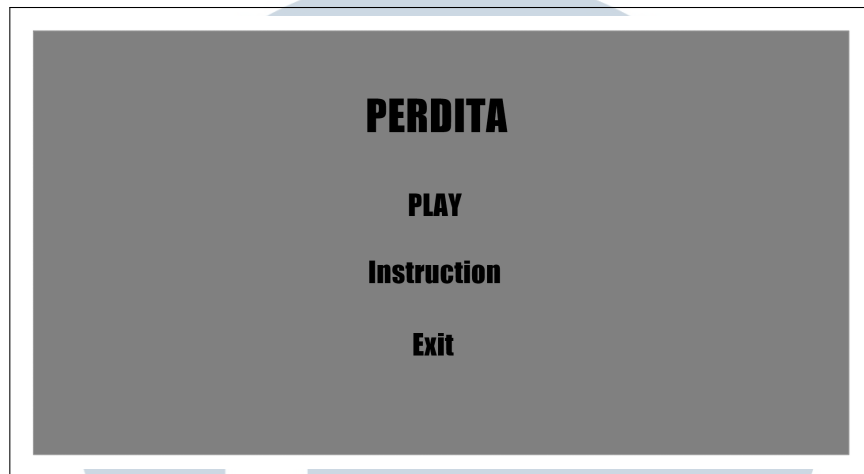
### 3.2.4 Desain Mockup

Berikut adalah *Mockup Design* dari aplikasi.



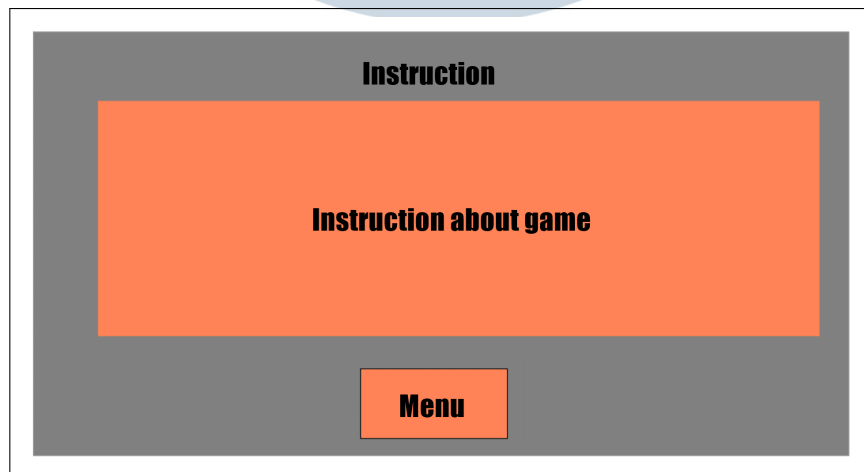
Gambar 3.7. Tampilan *Splash Screen*

Gambar 3.7 menunjukkan tampilan awal yang dilihat oleh pemain saat aplikasi dijalankan yaitu *splash screen* yang menampilkan logo Unity.



Gambar 3.8. Tampilan *Main Menu*

Gambar 3.8 menunjukkan tampilan setelah *splash screen* yaitu *main menu* dimana pemain dapat memilih *play*, *instruction* dan *exit*.



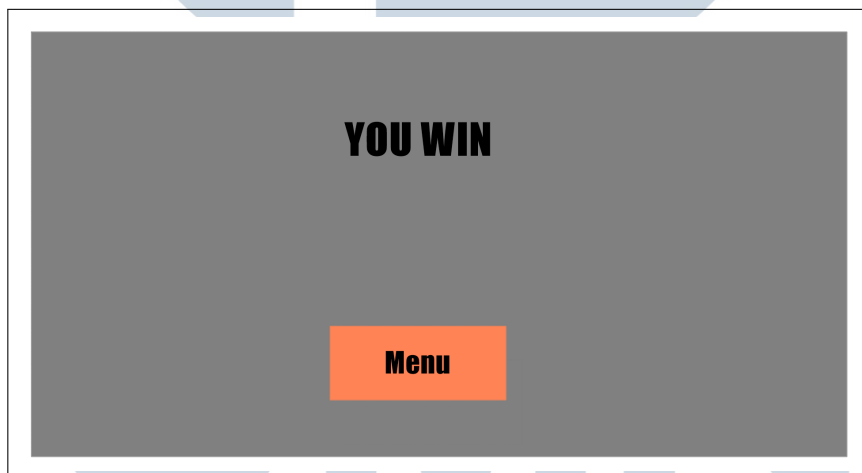
Gambar 3.9. Tampilan *Instruction*

Gambar 3.9 menunjukkan tampilan pada *Instruction* yang berisikan instruksi cara bermain game.



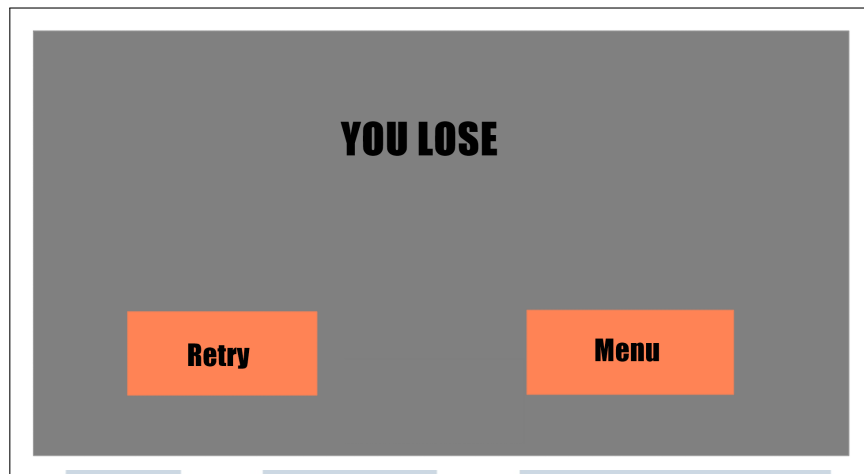
Gambar 3.10. Tampilan permainan

Gambar 3.10 menunjukkan tampilan utama permainan dimana pemain mempunyai *minimap* untuk membantu penelusuran *map* dan harus mengumpulkan item.



Gambar 3.11. Tampilan Menang

Gambar 3.11 menunjukkan tampilan apabila pemain berhasil mengumpulkan seluruh item tanpa tersentuh oleh musuh, setelah itu pemain dapat kembali ke *main menu*.



Gambar 3.12. Tampilan Kalah

Gambar 3.12 menunjukkan tampilan apabila pemain tersentuh oleh musuh. Pemain memiliki pilihan untuk mengulang permainan atau kembali ke *main menu*.

UMMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA