

## BAB 3

### METODOLOGI PENELITIAN

#### 3.1. Metodologi Penelitian

Penelitian terhadap “Rancang Bangun *Game Dungeon Crawler* dengan Procedural Content Generation Menggunakan Algoritma Drunkard’s Walk” akan didasari beberapa tahap meliputi.

a. Identifikasi Masalah

Masalah yang diteliti adalah masalah *content generation*, dan efek yang ditimbulkan kepada para pemain. Penelitian ini dilakukan dengan merancang dan membangun *game Dungeon Crawler* dengan menggunakan metode PCG *Drunkard’s Walk*, dan mengukur efek terhadap pemain dengan menggunakan GUESS-18.

b. Telaah Literatur

Pengumpulan informasi yang terkait dengan penelitian yang dilakukan. Topik informasi tersebut meliputi *game* secara sederhana, konsep design *video game*, elemen-elemen esensial yang terdasar dalam suatu *video game*, metode PCG, algoritma *Drunkard’s Walk* sebagai penggerak metode *content generation*, *Dungeon Crawler*, dan GUESS-18 sebagai tolak ukur kepuasan pemain.

c. Perancangan dasar dan Analisa *game Dungeon Crawler*

Berfokus terhadap perancangan dan pengertian dasar konsep rancang bangun *game* dengan genre *Dungeon Crawler*. Referensi yang akan digunakan berupa dokumentasi yang disediakan oleh Unity.

d. Penerapan algoritma

Setelah proses perancangan dasar dan melakukan Analisa terhadap pembuatan *game* dengan genre *Dungeon Crawler*, akan dilakukannya penerapan algoritma *Drunkard's Walk*. Perancangan ini akan dilakukan pada Unity 2D dengan Bahasa C#

e. Testing dan Perbandingan

Testing akan dilakukan dengan jumlah minimal 30 pemain atau user sesuai dengan ketentuan metode penelitian kombinasi oleh Sugiono (2015). *Testing* akan diukur dengan menggunakan bentuk GUESS terbaru yakni GUESS-18, dimana kepuasan pemain akan diukur terhadap *game* PCG tersebut.

f. Kesimpulan dan Evaluasi

Setelah perancangan dan *testing* sudah selesai, akan dilakukan penarikan kesimpulan sebagai hasil akhir penentuan tingkat kepuasan pemain yang telah diukur terhadap *game* PCG *Drunkard's Walk*. Berdasarkan hasil apapun yang diberikan akan dilakukan evaluasi, serta segala rincian kendala yang terjadi selama proses studi.

g. Dokumentasi

Dokumentasi akan dilakukan selama proses penelitian berjalan, dimana segala bentuk pengerjaan akan didokumentasikan dengan cara yang sesuai dan terkait. Bentuk dokumentasi dapat meliputi catatan penting, *screenshot*, ataupun foto.

### 3.2. Perancangan Game

Perancangan game dibagi menjadi 4 bagian atau fase berurut yang bermulai dari struktur *game*, *flowchart*, *assets*, dan yang terakhir dan tidak kalah penting *mockup*. Struktur *game* merupakan konsep awal segala proses yang akan dilakukan oleh para pemain. Pada bagian *flowchart* akan diperjelas dengan detil bagaimana game tersebut akan bekerja, mulai dari *flowchart* utama hingga beberapa modul *predefined process*. *Listing assets* dalam bentuk tabel yang digunakan dalam game, dan *attribution* yang diberikan kepada para pencipta *assets* terkait. *Mockup* merupakan gambaran secara visual bagaimana bentuk fitur-fitur yang akan tersedia dalam game.

#### 3.2.1. Struktur Game

Judul: Evil Frontier

Evil Frontier merupakan game *Dungeon Crawler* dimana para pemain akan menjelajahi *dungeon* dan melawan *monster* untuk meningkatkan level.

*Formal elements* yang tersedia dalam game adalah sebagai berikut.

##### 1. Player

Evil frontier merupakan game single player, atau game yang hanya bisa dimainkan oleh 1 pemain pada sesi tertentu.

##### 2. Objective

Pemain harus menyelesaikan *quest* yang dihasilkan oleh *game* agar dapat menaikkan *stats* mereka, yang akan berguna untuk melawan 3 *bounties* yang tersedia dalam *game*.

##### 3. Procedures

Beberapa prosedur tersedia agar *flow* permainan jelas dan mudah untuk dimainkan oleh para pemain. Prosedur tersebut meliputi:

- a. Pemain menjalankan *game* dan akan bertemu dengan *splash screen* sebelum mencapai *main menu*.
- b. Pemain yang sedang berhadapan dengan *main menu* bisa memilih antara *start game* secara baru maupun melakukan *load game* yang menyimpan semua *progress* yang telah dilakukan oleh pemain, jika tersedia. Pemain juga bisa melihat intruksi permainan, *credits*, maupun keluar permainan.
- c. Setelah memilih antara *start game* maupun *load game* pemain akan diarahkan menuju *quest menu*. Pada *quest menu* pemain bisa memilih *quest* untuk dimainkan. *Quest* yang diberikan akan selalu berbeda setiap iterasi, dan jika pemain berhasil mengalahkan suatu *quest* akan menerima *gold* yang sepadan. Jika pemain gagal melakukan *quest* terkait maka akan muncul tulisan *game over* dan akan dikembalikan pada *quest menu*.
- d. Pemain bisa melakukan *upgrade* pada stats saat ini dengan menggunakan *gold* yang tersedia.
- e. Pemain bisa melakukan *save* terhadap *progress* yang telah dilakukan, agar data tersebut dapat disimpan dan digunakan kembali saat ingin melanjutkan permainan.
- f. Untuk mengalahkan *game* pemain harus mengalahkan 3 *bounties* yang tersedia, jika berhasil mengalahkan ketiga bos tersebut permainan akan tamat dan pemain akan dikembalikan pada *main*

*menu*. Jika pemain gagal maka pemain akan dikembalikan pada *quest menu*.

#### 4. Rules

- a. Control yang disediakan untuk pemain hanya berupa input *keyboard*, dan *mouse*.
- b. Pemain harus mengalahkan 3 *bounties* untuk menyelesaikan permainan.

#### 5. Resource

- a. Gold: merupakan *in-game* yang digunakan dalam *game*. *Gold* hanya bisa didapatkan dari menyelesaikan *quest* ataupun mengalahkan 1 *bounties*. *Gold* bisa digunakan untuk melakukan *upgrade* pada *stats* guna mempermudah proses permainan.
- b. Health: merupakan darah yang dimiliki oleh pemain. Darah akan bertambah jika pemain naik level maupun melakukan *upgrade stats*. Darah dapat berkurang jika terkena serangan oleh musuh.
- c. Mana: adalah energi yang dimiliki pemain untuk melakukan *casting spell*. Jika player melakukan *casting spell* maka mana bisa berkurang dan hanya bisa diisi dengan keluar *game* atau menggunakan barang-barang *in-game*.
- d. Experience: merupakan batasan pemain untuk menaikkan level karakter dalam *game*. Jika *experience* pemain sudah mencukupi maka pemain akan naik level dan stats *karakter* akan bertambah sesuai yang sudah ditentukan oleh *game*

#### 6. Conflict

Pemain harus bermain *quest* agar dapat menaikkan level dan stats, agar dapat mengalahkan 3 *bounties* untuk menyelesaikan *game*.

#### 7. Boundaries

Pemain hanya bisa berkeliling pada *quest area* maupun *boss area* yang sudah ditentukan oleh *game* tersebut.

#### 8. Outcome

Pemain akan menang jika pemain sudah mengalahkan 3 *bounties* pada *game*.

Namun tidak hanya *formal elements* yang terdapat dalam *game Evil Frontier*, berikut merupakan *dramatic elements* yang membantu membentuk *game Evil Frontier* meliputi:

##### 1. Challenge

Tantangan dalam *game Evil Frontier* berupa 7 tipe musuh yang memiliki attack points berbeda dan 3 *bounties* yang memiliki kemampuan unik masing-masing.

##### 2. Play

Pemain diberikan kebebasan untuk menjalankan *quest* yang akan selalu dihasilkan secara random, mencoba melawan 3 *bounties*, *upgrade stats*, ataupun melakukan *save progress*.

##### 3. Premise

Seorang *mercenary* yang dibayar untuk mengalahkan 3 *wizard lords* yang jahat, tetapi *mercenary* tersebut harus meningkatkan kemampuannya terlebih dahulu sebelum dapat mengalahkan musuhnya.

##### 4. Character

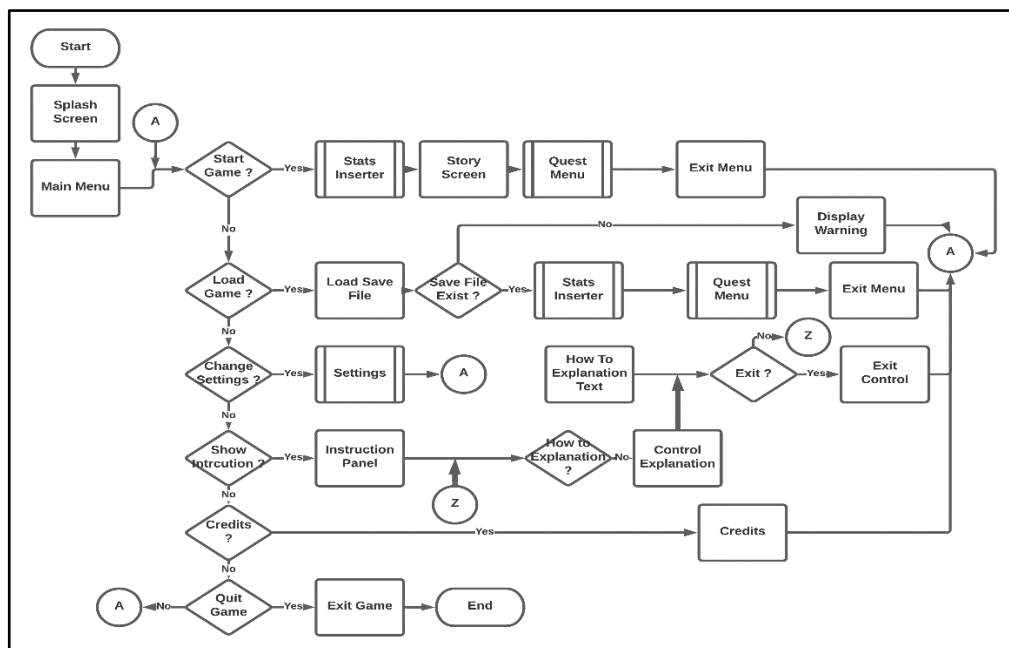
Seorang *mercenary* yang dibayar oleh warga sekitar untuk membasmi kejahatan dalam lingkungan tersebut.

## 5. Story

Sejak kedatangan 3 *wizard lords* di sebuah desa, banyak *monster* yang muncul dan menguasai beberapa wilayah di desa. Hal ini membuat warga desa tidak dapat hidup dengan damai. Mendengar berita seperti itu, seorang *mercenary* datang ke desa untuk membantu para warga membasmi *monster*. Bayaran yang ia terima dari warga desa digunakan untuk meningkatkan kemampuan bertarungnya, agar dapat mengalahkan musuh utamanya, yaitu 3 *wizard lords*.

### 3.2.2 Flowchart

*Flow* permainan yang dijalankan oleh *game Dungeon Crawler Drunkard's Walk* akan digambarkan oleh beberapa *flowchart*. *Flowchart* pertama yang dijalankan adalah *flowchart main menu*, yang tergambar pada gambar 3.1.



Gambar 3.1 Flowchart Main Menu

Proses *flowchart* dimulai dari pemain masuk kedalam *game*. Sebelum pemain masuk kedalam main menu, *game* akan menampilkan sebuah *splash screen* yang berisikan logo studio dan *production*. Pada main menu pemain disajikan beberapa pilihan yang dapat dipilih oleh pemain seperti *start game*, *load game*, *settings*, *instruction*, *credits*, dan *quit*.

Jika pemain ingin memulai permainan dengan *base stats*, dan belum ada *progress* sama sekali maka pemain dapat memilih opsi *start game*. *Start game* akan menjalankan modul *stats inserter* yang bertugas untuk memberikan *stats* utama pemain, dilanjutkan dengan proses *story*, lalu masuk kedalam modul *quest menu*, setelah selesai pada modul *quest menu* pemain akan dikembalikan pada *main menu*.

Jika pemain ingin melanjutkan *progress* yang telah dilakukan maka pemain dapat memilih opsi *load game*. *Load game* akan memeriksa *directory* penyimpanan *savefile* dan memastikan keberadaan *file* tersebut. Jika *savefile* berhasil ditemukan maka proses yang sama pada *start game* akan dijalankan pada *load game*, dengan satu perbedaan berupa *stats* yang dimasukkan pada pemain. Jika *savefile* tidak ditemukan maka pemain akan diberikan *warning message* dan akan langsung dikembalikan pada *main menu*.

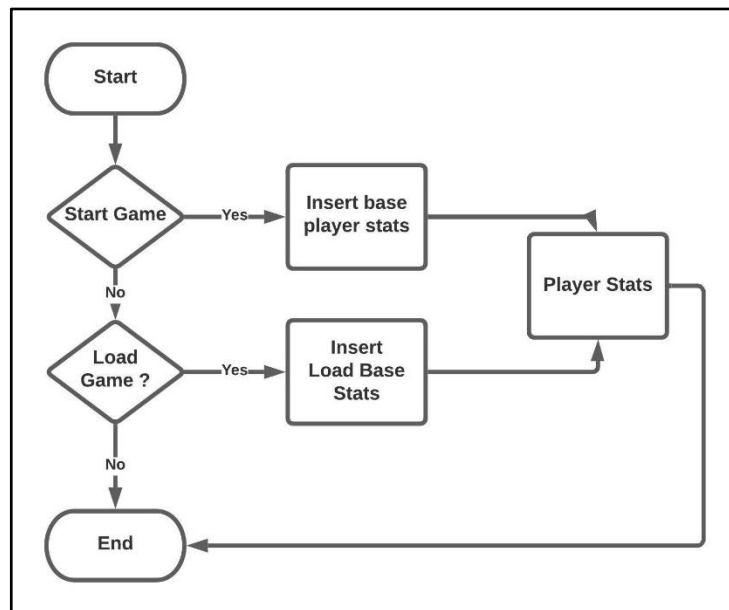
Jika pemain ingin mengubah *settings game* untuk menyesuaikan kenyamanan permainan maka pemain dapat memilih opsi *settings*. *Settings* akan menjalankan modul *settings*, lalu jika pemain selesai pengaturan pemain akan langsung dikembalikan pada *main menu*.

Jika pemain ingin melihat kontrol dan cara bermain maka pemain dapat memilih opsi *instruction*. *Instruction* pertama akan menampilkan kontrol *game*



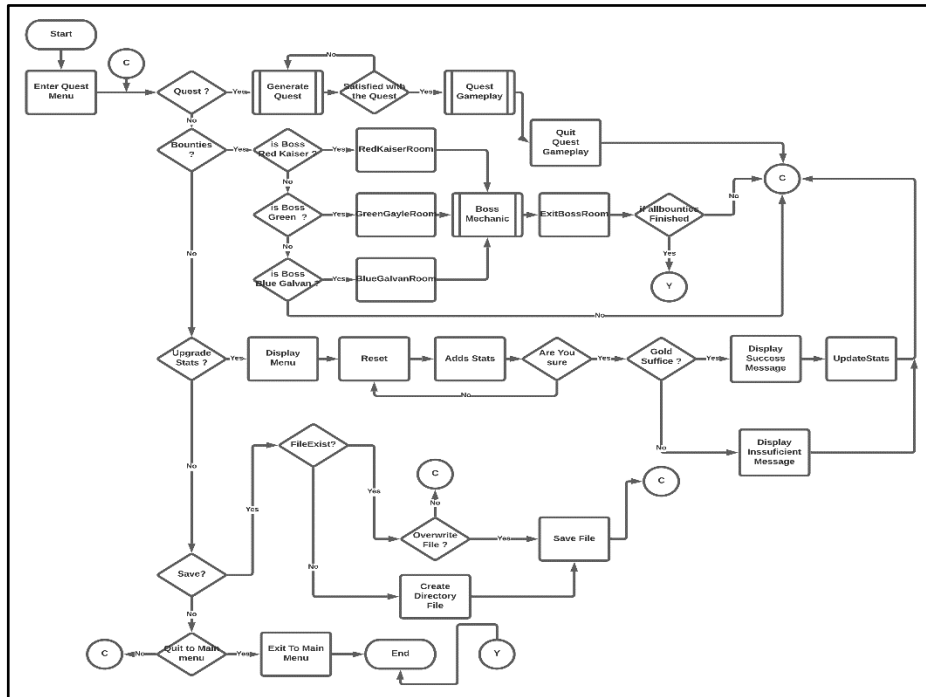
namun pemain diberikan opsi untuk mengganti tampilan teks menjadi cara bermain, dan sebaliknya jika diperlukan. Jika pemain sudah selesai menggunakan *instruction* pemain dapat langsung memilih opsi keluar untuk kembali menuju *main menu*.

Dua opsi terakhir yang dapat dipilih oleh pemain merupakan *credits* dan *quit*. Opsi *credits* akan memasukan pemain kedalam *credits page* dan akan mengembalikan pemain menuju *main menu* apabila credits telah selesai ditampilkan, atau jika pemain memilih opsi *skip credits*. Opsi *quit* akan mengeluarkan pemain dari *game* dan menyelesaikan proses *flow* permainan.



Gambar 3.2 Flowchart Stats Inserter

Pada gambar 3.2 merupakan modul *flowchart stats inserter* yang digunakan pada flowchart main menu pada gambar 3.1. *Stats inserter* bekerja dengan cara menentukan apakah pemain memilih opsi *start game* ataupun *load game*, dan memasukan *stats* yang sesuai dengan kondisi yang dipilih oleh pemain yang bermain saat ini.



Gambar 3.3 Flowchart Quest Menu

Pada gambar 3.3 merupakan gambar modul *flowchart quest menu* yang berada pada *flowchart main menu* pada gambar 3.1 diatas. Pada *quest menu* pemain disajikan beberapa pilihan opsi seperti *quest*, *bounties*, *upgrade*, *save*, dan *quit*.

Jika pemain ingin menjalankan *quest* maka pemain dapat memilih opsi *quest* yang tersedia dalam *quest menu*. *Quest* akan mengaktifkan modul *generate quest* yang akan menghasilkan *quest* secara *random*. Jika pemain tidak menyukai *quest* yang dihasilkan oleh *generator* maka pemain berhak untuk melakukan *generate* ulang *quest* hingga puas dengan hasil *generator*. Bilamana *quest* sudah terpilih pemain akan dimasukan kedalam modul *gameplay* hingga pemain keluar dari modul *gameplay*, dan pemain akan dikembalikan pada *quest menu*.

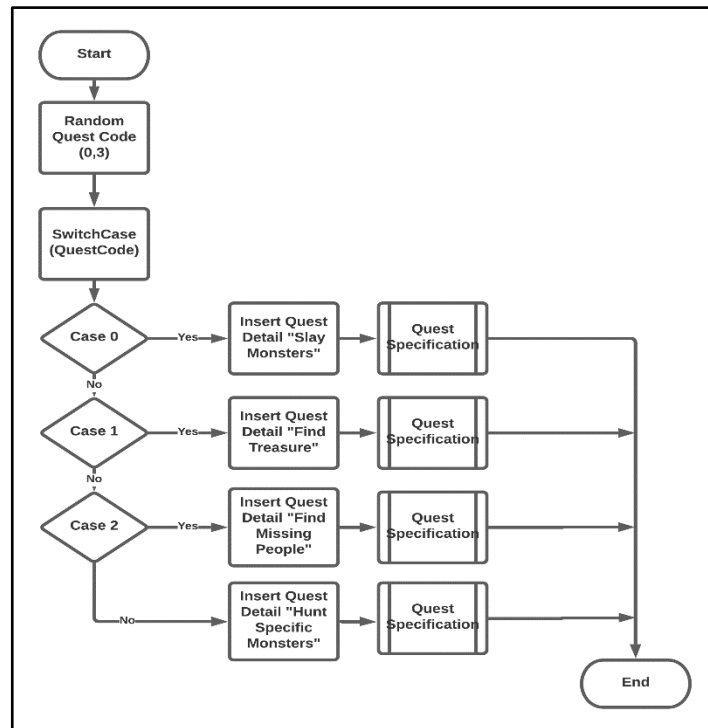
Jika pemain ingin mencoba melawan salah satu *bounties* maka pemain dapat memilih opsi *bounties*. *Bounties* mempersembahkan pemain dengan 3 opsi yang dapat dipilih oleh pemain seperti *red kaiser*, *green gayle*, dan *blue galvan*. Setelah

pemain memilih salah satu opsi *bounties* pemain akan dimasukan kedalam *area bounty* yang sesuai, dimana modul *boss mechanic* dijalankan hingga pemain keluar dari ruangan *bounty* tersebut. Jika pemain telah menyelesaikan semua *bounties* yang disediakan oleh *game* maka pemain sudah menyelesaikan *game*, dimana pemain akan dimasukan menuju *credits* lalu *main menu*. Jika pemain belum menyelesaikan semua *bounties* maka pemain akan dikembalikan menuju *quest menu* awal.

Jika pemain ingin melakukan penambahan *stats* maka pemain dapat memilih opsi *upgrade*. *Upgrade* akan membuka tampilan *upgrade menu* dan melakukan *reset* terhadap *value* sebelumnya, sehingga pemain bisa menambah sesuai keinginannya mulai dari *value* dasar tanpa ada kesalahan. Jika pemain menerima *upgrade* tambahan tersebut *game* akan menentukan apakah *gold* mencukupi, jika mencukupi *stats* akan ditambahkan dan pemain akan dikembalikan pada *quest menu*, jika tidak mencukupi maka *warning message* akan keluar dan pemain akan dikembalikan pada *quest menu*. Jika pemain tidak menerima *upgrade* tersebut *value* akan di *reset* ulang, dan pemain berhak memasukan *value* yang sesuai pada *upgrade menu*.

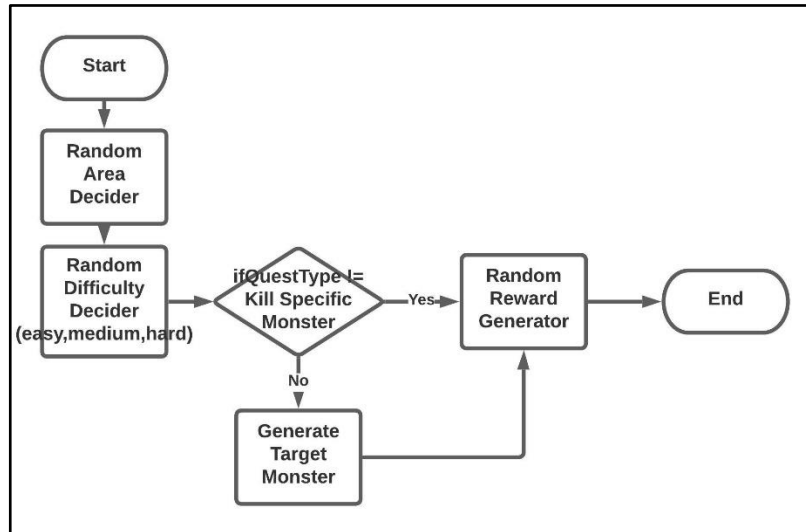
Jika pemain ingin melakukan penyimpanan terhadap *progress* permainan maka pemain dapat memilih opsi *save*. Jika *savefile* dideteksi oleh *game* maka *game* akan diberikan sebuah prompt “Are you sure you want to overwrite your file?”. Jika pemain memilih untuk *overwrite savefile* tersebut maka *progress save* akan dimasukan kedalam *file* tersebut, jika tidak pemain akan dikembalikan menuju *quest menu*. Jika *savefile* tidak ditemukan maka *directory file* akan dibuat, dan *progress* akan disimpan pada *file* tersebut.

Jika pemain ingin kembali menuju *main menu*, pemain dapat memilih opsi *quit*. *Quit* akan mengeluarkan pemain dari *quest menu*, jika tidak maka pemain akan dikembalikan menuju *quest menu*.



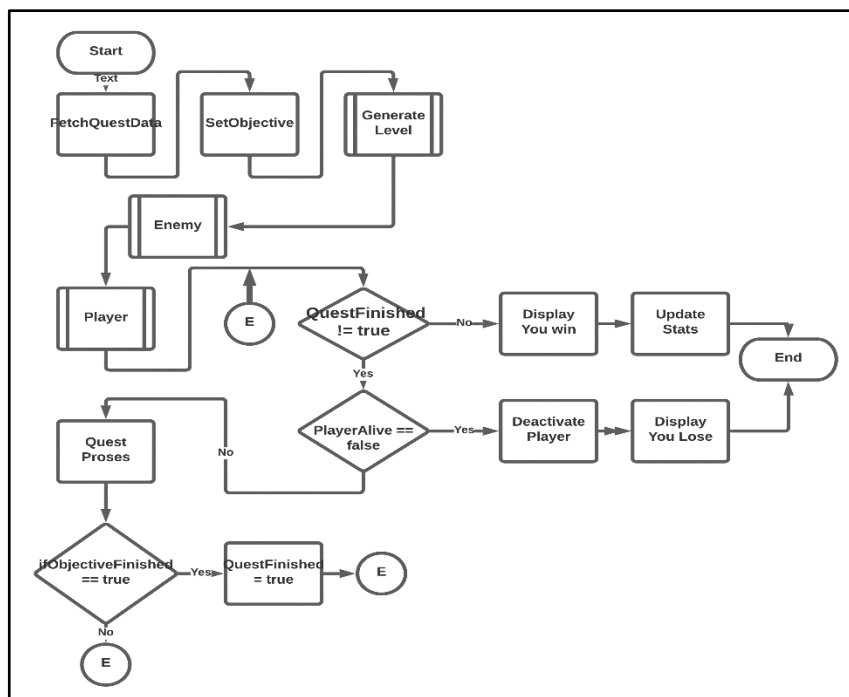
Gambar 3.4 Flowchart Generate Quest

Gambar 3.4 merupakan gambar modul *flowchart generate quest* yang dipanggil pada *flowchart quest menu* pada gambar 3.2 diatas. Modul ini berkerja dengan melakukan *random number* antara 0 sampai 3, yang akan digunakan untuk nilai *switch case*. Setiap *case* akan mendapatkan *detail quest* berbeda yang akan dilanjutkan dengan pengisian spesifikasi *quest* oleh modul *quest specification*.



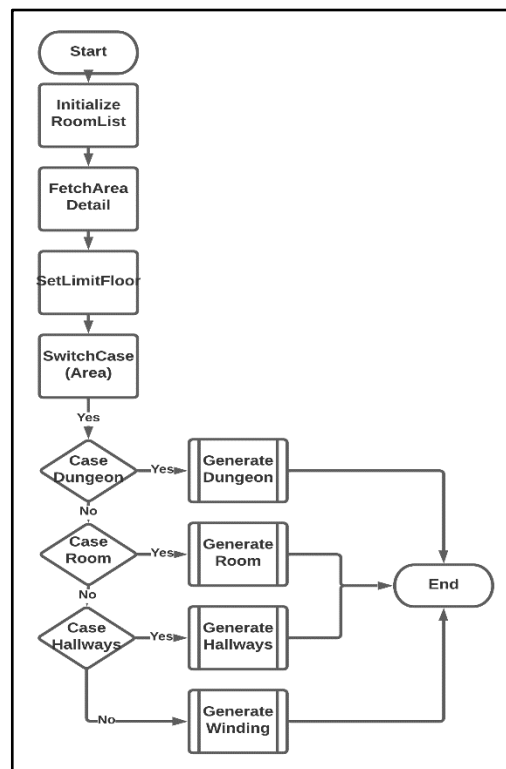
Gambar 3.5 Flowchart Quest Specification

Gambar 3.5 merupakan gambar modul *flowchart quest specification* yang dipanggil pada flowchart *generate quest* pada gambar 3.4 diatas. Modul bermulai dengan melakukan *random area decider* untuk environment, lalu random difficulty. Jika *questType* merupakan *killspecific* maka game akan menentukan target *monster* dan reward, jika bukan *killspecific* maka hanya reward yang ditentukan.



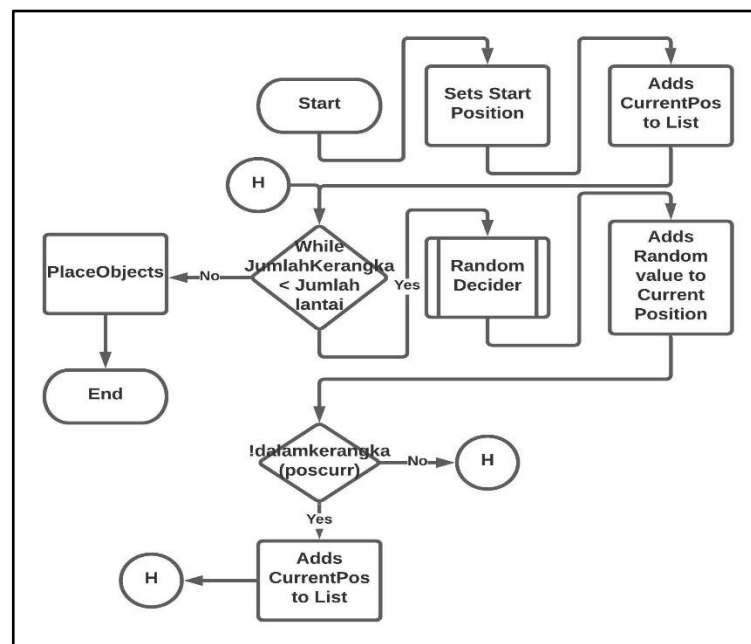
Gambar 3.6 Gambar Flowchart Quest Gameplay

Gambar 3.6 merupakan gambar modul *flowchart quest gameplay*, yang dipanggil oleh *flowchart quest menu* pada gambar 3.4 diatas. *Quest gameplay* dimulai dengan mendapatkan *quest data*, dimana data *quest* yang didapatkan akan menjadi basis objektif untuk sesi *quest* tersebut. Modul *generate level* akan dijalankan setelah *detail quest* didapatkan, dilanjutkan dengan modul *enemy* yang kemudian dilanjutkan oleh modul *player*. Jika *quest* belum selesai maka pemain akan meneruskan proses *flowchart* namun bilamana pemain mati sebelum proses *flowchart* berakhir, modul *player* dimatikan dan panel *you lose* akan muncul. Jika *quest* belum selesai, namun pemain tidak mati sebelum proses *flowchart* berakhir maka proses *flowchart* akan terus berjalan hingga pemain berhasil menyelesaikan objektif terkait. Jika *quest* telah selesai dilakukan maka *panel display you win* akan ditampilkan, dan *stats player* akan ditambahkan sesuai spesifikasi *quest*.



Gambar 3.7 Flowchart Generate Level

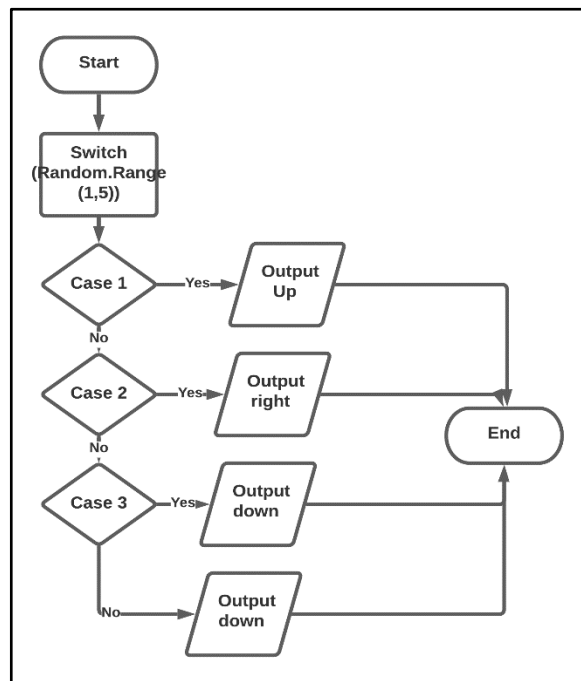
Gambar 3.7 merupakan gambar modul *flowchart* modul *generate level* yang dipanggil dalam *flowchart quest gameplay* pada gambar 3.6 diatas. Modul *generate quest* berjalan dengan membuat *room list*, menerima *quest detail*, mendefinisikan limit lantai yang bisa digenerasi, dan *switch case*. *Switch case* akan menentukan jenis area yang akan dihasilkan berdasarkan nilai area yang meliputi modul *generate dungeon*, *generate room*, *generate hallways*, dan yang terakhir *generate winding*.



Gambar 3.8 Flowchart Generate Dungeon

Gambar 3.8 merupakan gambar modul *flowchart generate dungeon*, yang dipanggil oleh *flowchart generate level* pada gambar 3.7 diatas. *Generate dungeon* merupakan bagian dari *Drunkard's Walk* algorithm, yang digunakan dalam perancangan struktur goa game *Evil Frontier*. *Generate dungeon* memulai proses dengan mendefinisikan posisi awal dan memasukkannya dalam *floorlist*. Jika jumlah lantai masih lebih banyak dibandingkan lantai dalam list maka modul *random decider* akan dijalankan untuk memberikan *value random* pada *current position*.

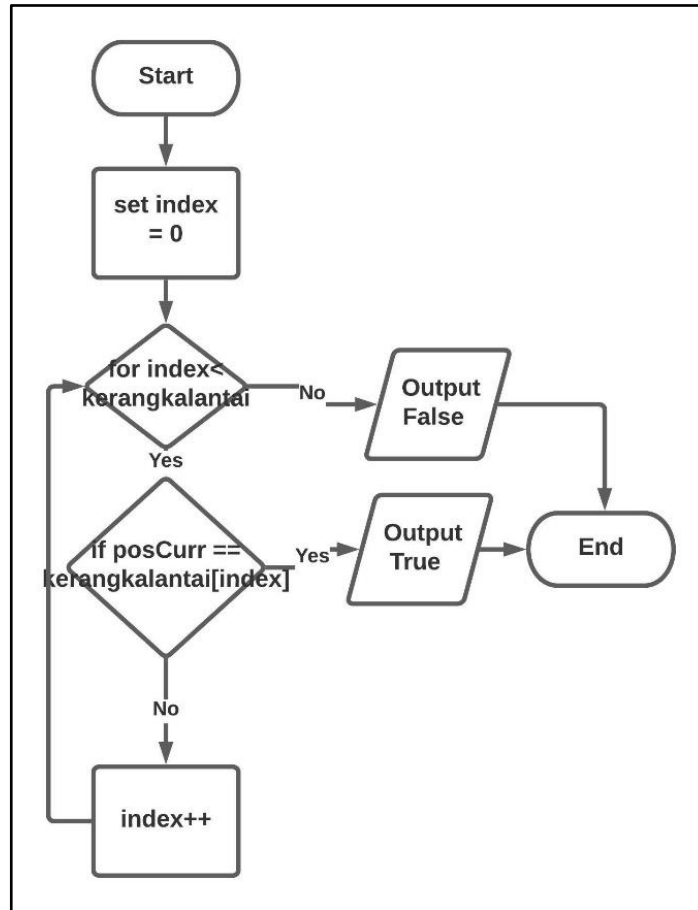
Jika *current position* tidak memiliki *value* yang sama pada list maka *current position* akan dimasukan kedalam list, jika ada *value* yang sama akan langsung kembali menuju *while* utama. Jika list sudah lebih besar dibandingkan dengan jumlah lantai maka proses *generator* konten akan dijalankan oleh *flowchart*.



Gambar 3.9 Flowchart Random Decider

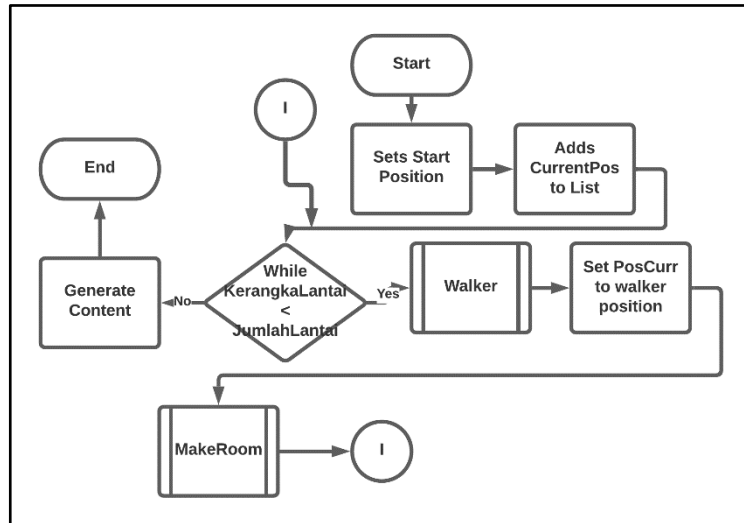
Gambar 3.9 merupakan gambar dari modul *flowchart random decider* yang digunakan dalam keempat modul *generator*. *Random Decider* bertugas untuk memberikan petunjuk dimana *generator* bisa menempatkan konten, dengan menggunakan *value* arah jalan. *Random decider* bekerja dengan melakukan *random number* dan memberikan output berdasarkan *value* yang dimasukan kedalam *switch case*. *Value* arah yang tersedia untuk dikeluarkan adalah atas, bawah, kiri, dan kanan.





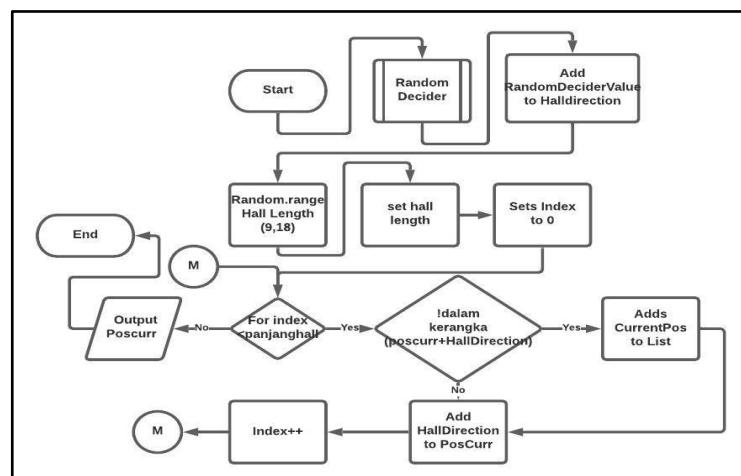
Gambar 3.10 Flowchart Dalam Kerangka

Gambar 3.10 merupakan gambar dari modul *flowchart* dalam kerangka yang digunakan didalam keempat modul *generator*. *Flowchart* dalamkerangka bertujuan untuk memeriksa apabila *current position* memiliki posisi yang sama dengan posisi yang sudah ada dalam list. *Flowchart* dalamkerangka bekerja dengan cara menerima *current position* lalu memasukannya didalam *for function*, dimana *function* tersebut akan dijalankan sebanyak jumlah list lantai. Jika *current position* memiliki koordinat yang sama dengan salah satu list maka *flowchart* akan memberikan *value true*, jika tidak ada yang sama maka akan diberikan *value false*.



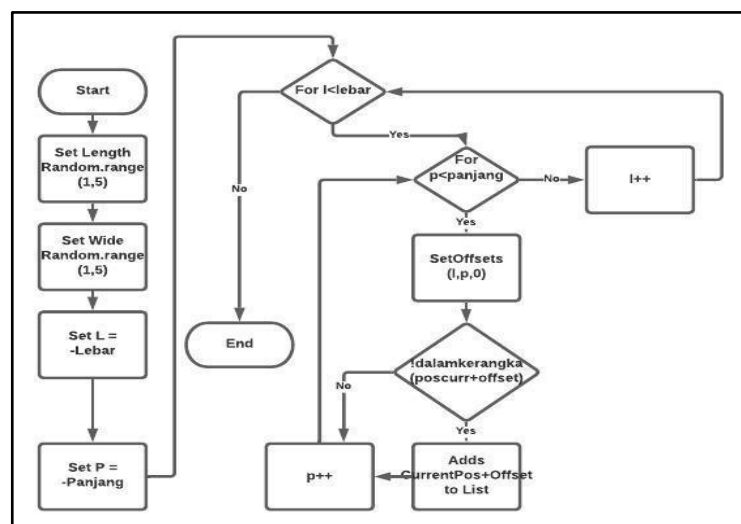
Gambar 3.11 Flowchart Generate Room

Gambar 3.11 merupakan gambar modul *flowchart generate room*, yang dipanggil pada *flowchart generate level* pada gambar 3.7 diatas. *Generate room* merupakan bagian dari *Drunkard's Walk* algorithm, yang digunakan dalam perancangan ruang *game Evil Frontier*. *Generate room* memulai proses dengan mendefinisikan posisi awal dan memasukannya dalam *floorlist*. Jika jumlah lantai masih lebih besar dibandingkan list kerangka lantai maka modul *walker* akan dijalankan, lalu walker akan memberikan koordinat *value* walker pada *current position*.



Gambar 3.12 Flowchart Walker

Gambar 3.12 merupakan gambar modul *flowchart walker* yang digunakan oleh *room generator* dan *winding generator*. *Walker* memiliki tujuan untuk pembentukan lorong pada *game*, yang akan digunakan untuk penyambungan antar ruangan yang dihasilkan oleh *generator*. *Walker* memulai *progress* dengan memanggil modul *random decider* seperti pada gambar 3.9 diatas, dan menambahkan nilai *random decider* pada *hall direction*. Setelah *Hall direction* mendapatkan nilai *random*, *walker* akan memasukan nilai *random* panjang ruangan, memasukan nilai *index*, dan memulai proses *for loop*. Jika *index* lebih kecil dibandingkan dengan *hall length* maka modul dalamkerangka pada gambar 3.9 akan dijalankan. Jika ditemukan bahwa ada list dengan posisi yang sama dengan *current position* maka posisi *current position* hanya akan di tambah dengan *hall direction*, lalu menambahkan nilai *index*. Jika tidak ditemukan list dengan posisi yang sama dengan *current position* maka *current position* akan dimasukan kedalam list, *hall direction* akan ditambahkan pada *current position*, dan *index* ditambhakan. Jika *index* lebih besar dibandingkan panjang *hall length*, *walker* akan memberikan output berupa *current position* terbaru.

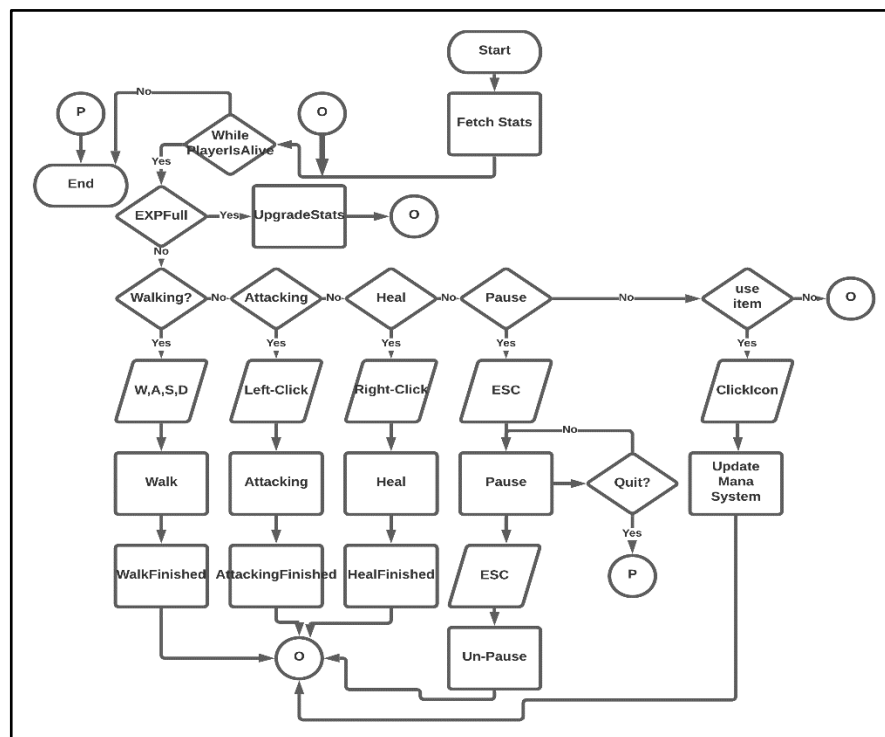


Gambar 3.13 Flowchart Make Room

Gambar 3.13 merupakan gambar modul *flowchart make room* yang digunakan oleh *room generator* dan *winding generator*. *Make room* memiliki tujuan untuk membuat atau menghasilkan ruangan dengan besaran *random*. *Make room* memulai *progress* dengan melakukan ini inisialisasi terhadap panjang dan lebar yang digunakan, dengan menggunakan *function random range*. Setelah nilai lebar dan panjang telah ditentukan *flowchart* akan menjalankan *function nested for*, untuk pembentukan struktur ruangan pada *game*. *For* pertama akan menggunakan parameter lebar dan untuk *for* kedua akan menggunakan parameter panjang dimana *offsets* akan diinisialisasi berdasarkan lebar dan panjang yang tersedia. *Offsets* akan digunakan untuk menambah jarak pada *current position*, dimana *current position* akan diperiksa oleh modul dalamkerangka pada gambar 3.10 untuk kesamaan posisi dengan yang ada dalam list. Jika ada yang sama maka hanya parameter panjang yang ditambahkan, dan proses akan kembali pada *for* lebar untuk penambahan. Jika tidak ada yang sama maka *current position* akan ditambahkan kedalam list, dan proses akan berjalan seperti layaknya jika ada posisi yang sama dalam list. Jika parameter lebar sudah tidak terpenuhi maka *flowchart* diakhiri.



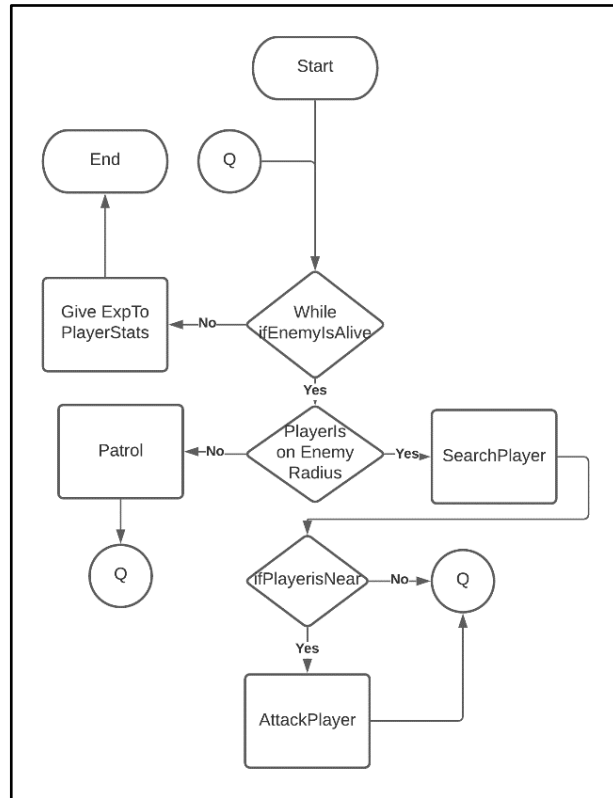
Gambar 3.15 merupakan gambar modul *flowchart generate winding* yang dijalankan pada *flowchart generate level* pada gambar 3.7 diatas. *Generate hallways* merupakan bagian dari *Drunkard's Walk* algorithm, yang digunakan dalam perancangan struktur ruangan *winding game Evil Frontier*. *Generate winding* memiliki jalan proses yang sama dengan *generate room* namun dengan perbedaan berupa inisialisasi *presentage* ruangan, dan *conditional* yang harus dilewati. *Conditional* tersebut dijalankan setelah pemeriksaan *flowchart* dalamkerangka, dan proses dimulai dengan mendapatkan nilai *random integer* ruangan. Jika *random integer* ruangan lebih besar dibandingkan dengan *room presentage* maka ruangan akan dibuat dengan modul *make room* pada gambar 3.13 diatas, jika tidak maka ruangan tidak akan dibuat dan *flowchart* akan berlanjut.



Gambar 3.16 Flowchart Player

Gambar 3.16 merupakan gambar modul *flowchart player* yang dijalankan pada *flowchart gameplay* maupun *boss mechanic*. *Player* memulai proses *flowchart*

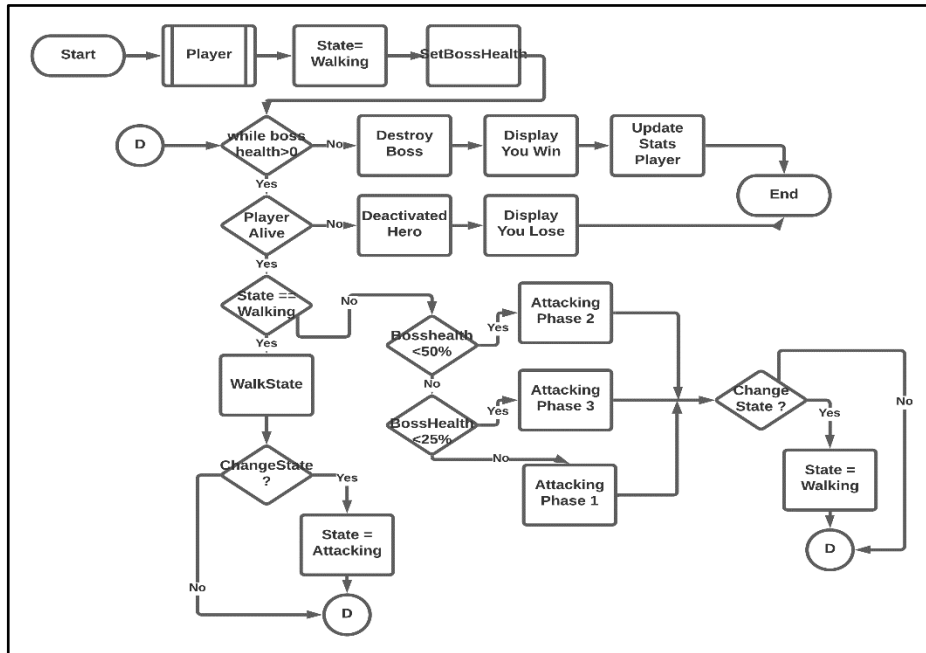
dengan pemeriksaan apakah pemain masih hidup, dan pemeriksaan apakah *experience* penuh. Jika pemain masih hidup maka *flowchart* akan berjalan terus, namun jika pemain mati maka proses *flowchart* akan diakhiri. Jika pemain memiliki *experience* penuh maka pemain akan menjalankan proses *upgrade stats*, dan melanjutkan proses *flowchart*. Beberapa opsi aksi yang bisa dilakukan oleh pemain seperti *walking*, *attacking*, *heal*, *pause game*, dan *use item*. Jika pemain ingin berjalan maka pemain dapat memasukan input berupa w, a, s, d, untuk menjalankan proses berjalan hingga selesai. Jika pemain ingin menyerang maka pemain dapat menekan tombol kiri *mouse* untuk menjalankan proses *attack* hingga selesai. Jika pemain ingin melakukan aksi *heal* maka pemain dapat menekan tombol kanan *mouse* untuk menjalankan proses *heal* hingga selesai. Jika pemain ingin menampilkan *pause menu* maka pemain dapat menekan tombol *escape* pada keyboard, dan player memiliki pilihan untuk keluar dari permainan atau tidak. Jika pemain ingin keluar maka pemain dapat memilih opsi *quit* dan pemain akan dikeluarkan menuju *quest menu*, jika tidak memilih *quit* maka player akan dikembalikan pada sesi *gameplay*. Jika pemain memilih untuk memakai *item* maka pemain bisa langsung untuk menekan gambar terkait pada layar *gameplay* dengan menggunakan *mouse*.



Gambar 3.17 Flowchart Enemy

Gambar 3.17 merupakan modul *flowchart enemy* yang dipanggil dalam *flowchart quest gameplay* pada gambar 3.6 diatas. *Enemy* memulai proses dengan pemeriksaan bahwa *enemy* masih hidup, jika masih hidup maka *flowchart* akan berjalan seperti seharusnya, jika tidak maka proses pemberian *experience* akan dijalankan dan proses *flowchart* akan diakhiri. Jika pemain ada dalam radius musuh maka musuh akan mencari posisi pemain, jika tidak maka musuh akan menjalankan proses *patrol*. Jika pemain dekat dengan musuh maka musuh dapat melakukan proses menyerang terhadap pemain.

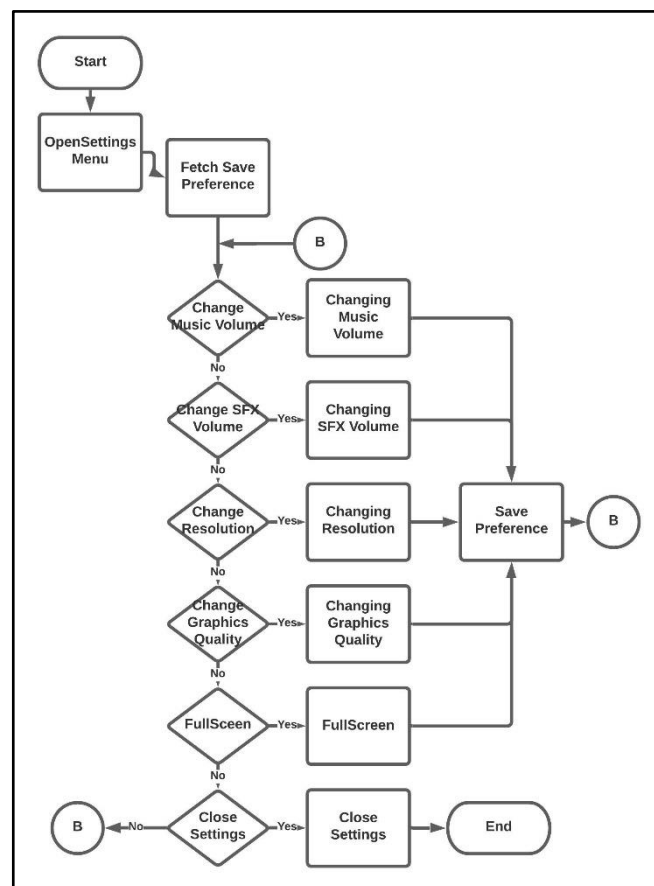




Gambar 3.18 Flowchart Boss

Gambar 3.18 merupakan gambar modul *flowchart boss mechanic* yang dipanggil pada *flowchart quest menu* pada gambar 3.3 diatas. *Flowchart* memulai proses dengan menginisialisasi modul *player*, *bosshealth*, dan *state* yang dilakukan oleh *flowchart*. Jika *bosshealth* lebih besar dari 0 maka proses akan berlanjut pada *conditional* berikutnya, yakni *conditional* pemain masih hidup. Jika pemain masih hidup maka proses akan berlanjut pada *conditional* berikutnya, yakni *conditional* apakah penentu *state*. Jika *state boss* adalah *walking* maka proses *walking state* akan dijalankan hingga proses selesai dijalankan, dan *conditional change state* muncul. Jika boss mengganti *state* maka nilai *state* akan diganti, dan *flowchart* akan mengulang pada *while* utama. Jika *state* bukan *walking* maka *conditional state attacking* yang berdasarkan sisah dari nyawa *boss* akan dijalankan. Jika *bosshealth* dibawah 50% maka proses *attacking phase 2* akan dijalankan, jika *bosshealth* dibawah 25% maka proses *attacking phase 3* akan dijalankan, jika tidak dua-duanya maka *attacking phase 1* akan dijalankan. Jika proses *attacking* sudah selesai

dijalankan *boss* akan memiliki kesempatan untuk mengubah *state*. Jika boss mengganti *state* maka nilai *state* akan diganti, dan flowchart akan mengulang pada *while* utama. Melanjutkan dari *conditional* pemain, jika pemain mati maka pemain akan dinonaktifkan dan panel *you lose* akan ditampilkan. Melanjutkan dari *conditional bosshealth*, jika *bosshealth* kurang dari 0 maka *boss* akan hancur, panel *you win* akan ditampilkan, dan *stats* akan pemain akan di *update*.



Gambar 3.19 Flowchart Settings




Gambar 3.19 merupakan gambar modul *flowchart settings* yang dipanggil pada *flowchart main-menu* pada gambar 3.1 diatas. *Settings* memulai proses dengan membuka panel *settings*, dan melakukan *fetching player preference* untuk *settings*. Pemain memiliki beberapa opsi yang bisa dilakukan seperti mengganti volume music, volume sfx, kualitas grafik, resolusi, dan fullscreen. Jika pemain ingin









mengganti volume music, maka proses penggantian volume akan dilaksanakan lalu di simpan dalam *player preferences*. Jika pemain ingin mengganti volume sfx, maka proses penggantian sfx akan dilaksanakan lalu di simpan dalam *player preferences*. Jika pemain ingin mengganti resolusi layar, maka proses penggantian *resolution* akan dilaksanakan lalu di simpan dalam *player preferences*. Jika pemain ingin mengganti kualitas grafik, maka proses penggantian kualitas grafik akan dilaksanakan lalu di simpan dalam *player preferences*. Jika pemain ingin mengganti *fullscreen*, maka proses *fullscreen* akan dilaksanakan lalu di simpan dalam *player preferences*. Jika pemain ingin keluar dari *settings* maka pemain dapat memilih untuk keluar dan mengakhiri proses *flowchart*.


### 3.2.3 Assets

Berikut merupakan *assets-assets* yang digunakan dalam perancangan dan pembangunan *Evil Frontier*.

Tabel 3.1 Tabel Player dan Monster





No	Nama	Gambar	Deskripsi	Sumber
1	Main Character		Main character yang digunakan oleh pemain	Pita/Itch.io
2	Slime		NPC musuh yang muncul pada <i>difficulty easy</i>	Pita/Itch.io
3	Rat		NPC musuh yang muncul pada <i>difficulty easy</i>	Pita/Itch.io

4	Spider		NPC musuh yang muncul pada <i>difficulty easy</i> dan <i>medium</i>	Pita/Itch.io
5	Bat		NPC musuh yang muncul pada <i>difficulty medium</i> .	Pita/Itch.io
6	Goblin		NPC musuh yang muncul pada <i>difficulty medium</i> dan <i>hard</i>	Pita/Itch-io
7	Orc		NPC musuh yang muncul pada <i>difficulty hard</i>	Pita/Itch-io
8	Troll		NPC musuh yang muncul pada <i>difficulty hard</i>	Pita/Itch-io
9	Red Kaiser		Boss NPC pada bounty <i>Red Kaiser</i>	FinalBossBlue /Itch.io
10	Green Gayle		Boss NPC pada bounty <i>Green Gayle</i>	FinalBossBlue /Itch.io
11	Blue Galvan		Boss NPC pada bounty <i>Blue Galvan</i>	FinalBossBlue /Itch.io

12	Villagers		NPC <i>villager</i> yang ditampilkan pada <i>quest find missing peoples</i>	FinalBossBlue /Itch.io
----	-----------	---	---	------------------------


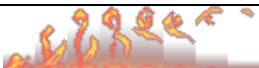


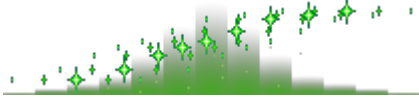

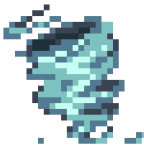
Tabel 3.1 diatas merupakan penjabaran *assets* yang digunakan dalam game *Evil Frontier*, baik sebagai karakter utama, NPC musuh, maupun NPC *boss*.


Tabel 3.2 Tabel Items

No	Nama	Gambar	Deskripsi	Sumber
1	Mana Potion 10%		<i>Item</i> yang memberikan mana kepada pemain sebesar 10% <i>max mana</i> pemain	Admurin/Itch.io
2	Mana Potion 20%		<i>Item</i> yang memberikan mana kepada pemain sebesar 20% <i>max mana</i> pemain	Admurin/Itch.io
3	Mana Potion 30%		<i>Item</i> yang memberikan mana kepada pemain sebesar 30% <i>max mana</i> pemain	Admurin/Itch.io
4	Treasure Chest		<i>Item</i> yang digunakan sebagai objektif pada tipe <i>quest find treasures</i>	Pita/Itch.io

Tabel 3.2 diatas merupakan penjabaran *items* yang digunakan dalam game *Evil Frontier*, dan memiliki kegunaan masing-masing.

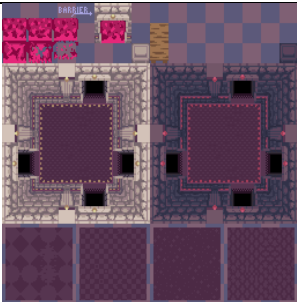
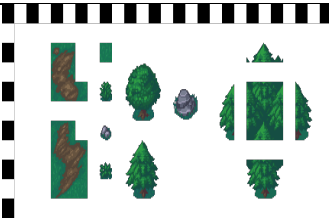
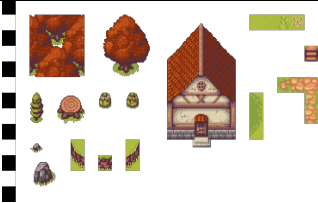
Tabel 3.3 Status Effect dan Attack Effect


No	Nama	Gambar	Deskripsi	Sumber
1	Dark Effect		Dark effect digunakan jika <i>boss</i> dalam keadaan <i>Vulnerable</i>	Henry Software/ Itch.io
2	Fire Effect		Fire Effect digunakan pada <i>boss red Kaiser</i> saat dalam <i>walking state</i>	Henry Software/ Itch.io
3	Death Effect		Death Effect digunakan pada saat <i>boss</i> tidak memiliki health lagi	Henry Software/ Itch.io
4	Explosion		Explosion digunakan pada saat NPC musuh dan <i>boss</i> hancur	Henry Software/ Itch.io
5	Heal Effect		Heal Effect digunakan pada saat pemain melakukan aksi <i>heal</i>	Henry Software/ Itch.io
6	Fire Attack		Fire Attack merupakan serangan yang digunakan oleh <i>Red Kaiser</i>	NYKNCK/ Itch.io
7	Wind Attack		Wind Attack merupakan serangan yang digunakan	NYKNCK/ Itch.io

			oleh <i>Green Gayle</i>	
8	Thunder Attack		Thunder Attack merupakan serangan yang digunakan oleh <i>Blue Galvan</i>	NYKNCK/ Itch.io

Tabel 3.3 diatas merupakan penjabaran *effects* yang digunakan dalam *game Evil Frontier* baik dalam indikasi status *effect*, maupun *attack effect* yang digunakan oleh beberapa NPC.





Tabel 3.4 Tabel Tilesets

No	Nama	Gambar	Deskripsi	Sumber
1	Top-Down Duo Dungeon		<i>Tileset dungeon</i> yang digunakan dalam <i>random generation</i> , dengan kode area 1	Enjl/Itch.io
2	Top-Down Forest		<i>Tileset dungeon</i> yang digunakan dalam <i>random generation</i> , dengan kode area 2	Ansimuz/Itch.io
3	Top-Down Outskirts		<i>Tileset dungeon</i> yang digunakan dalam <i>random generation</i> , dengan kode area 3	Ansimuz/Itch.io

4	Top-Down Dungeon		<i>Tileset Dungeon yang digunakan dalam pembuatan level bounties red Kaiser, green gayle, blue galvan</i>	Pita/Itch.io
---	------------------	---	---	--------------

Tabel 3.4 diatas merupakan penjabaran *tilesets* yang digunakan dalam game *Evil Frontier* baik dalam pembuatan level *quest*, maupun level *bounties*.

Tabel 3.5 Tabel Assets Lanjutan

No	Nama	Gambar	Deskripsi	Sumber
1	UI		<i>Asset yang digunakan untuk pembentukan UI seluruh game.</i>	IDLandGames/Itch.io
2	"Title Theme"		<i>Background music yang digunakan dalam main-menu</i>	xDeviruchi/Itch.io
3	"Mysterious Dungeon"		<i>Background music yang digunakan dalam quest menu, Story, dan sebelum melawan boss</i>	xDeviruchi/Itch.io
4	"Prepare for Battle"		<i>Background music yang digunakan saat melawan boss</i>	xDeviruchi/Itch.io

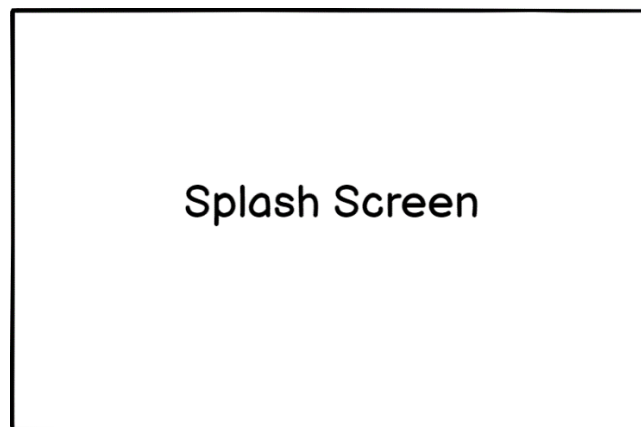


5	“And the Journey Begins”		<i>Background music</i> yang digunakan saat masuk dalam <i>area forest</i>	xDeviruchi/Itch.io
6	“The Icy Cave”		<i>Background music</i> yang digunakan saat masuk dalam <i>area Dark Dungeon</i>	xDeviruchi/Itch.io
7	“Exploring the Unknown”		<i>Background music</i> yang digunakan saat masuk dalam <i>area Outskirts</i>	xDeviruchi/Itch.io
8	“The Final of the Fantasy”		<i>Background music</i> yang digunakan saat <i>Boss</i> berhasil dikalahkan, dan <i>credits</i>	xDeviruchi/Itch.io
9	SFX Game		Beberapa <i>sound effect</i> yang digunakan dalam <i>game</i>	<i>Asset pribadi</i>
10	Background Menu, Quest Menu, Story, Credits		Background yang dipakai dalam <i>scene</i> , yang bukan merupakan <i>gameplay</i>	Analokisa/Itch.io

Tabel 3.5 diatas merupakan penjabaran *assets* lanjutan seperti UI, musik, dan SFX yang digunakan dalam *game Evil Frontier*.

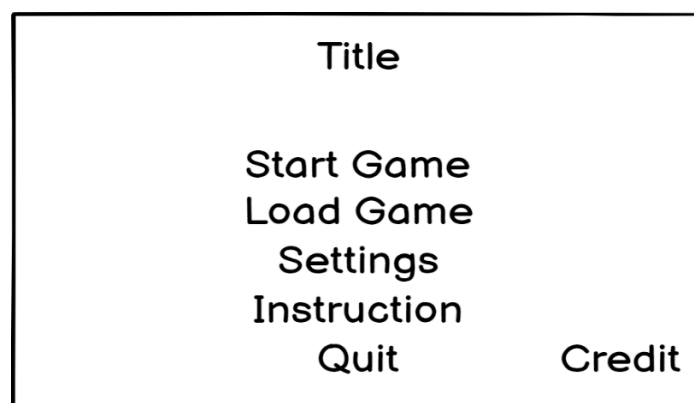
### 3.2.4 Mockup

Berikut merupakan penggambaran bentuk *mockup design* yang digunakan dalam *game*. Gambar 3.20 merupakan tampilan *splash screen* yang menyambut pemain sebelum pemain masuk kedalam *main menu* berisikan logo Unity, logo *personal*, dan logo *production*.



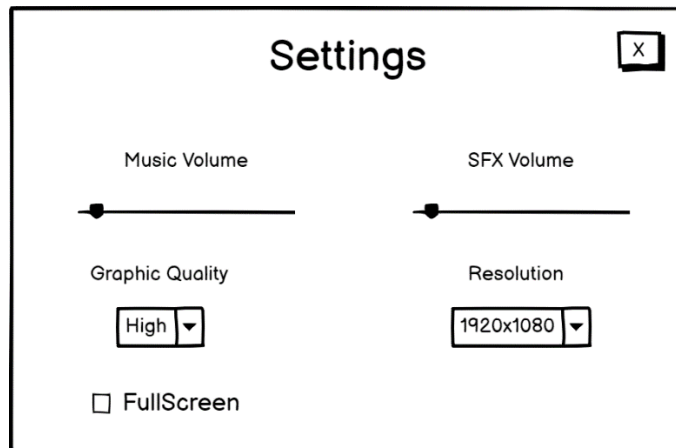
Gambar 3.20 Mockup Splash Screen

Setelah *splash screen* telah dilewati maka pemain akan bertemu dengan *main menu*. Gambar 3.21 merupakan *mockup main menu* yang memiliki opsi untuk pemain memulai permainan dengan *start game*, melakukan *load file* dengan *load game*, mengatur pengaturan dengan *settings*, melihat instruksi dengan *instruction*, melihat kontributor pada pembuatan *game* dalam *credits*, dan keluar dengan *quit*.



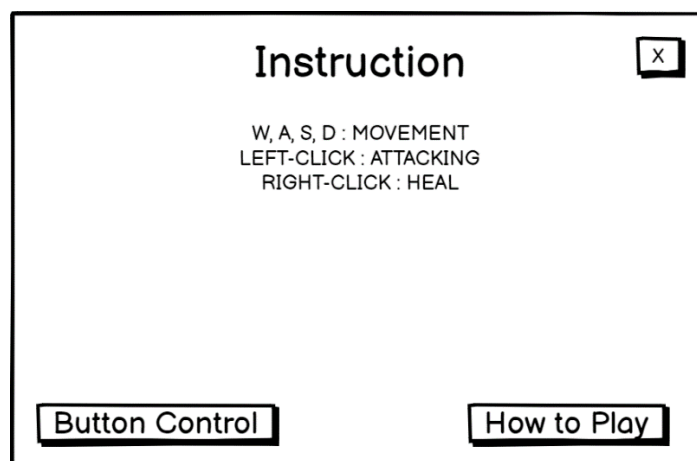
Gambar 3.21 Mockup Main Menu

Jika pemain memilih opsi *settings* dalam *main menu* pemain akan diperlihatkan *settings* menu seperti pada mockup gambar 3.22 dibawah. *Mockup settings* terdiri dari 2 *sliders* yang digunakan untuk volume sfx dan musik, 2 *dropdown* untuk resolusi dan kualitas grafik, *checkboxlist* untuk *fullscreen*, dan *quit button*.



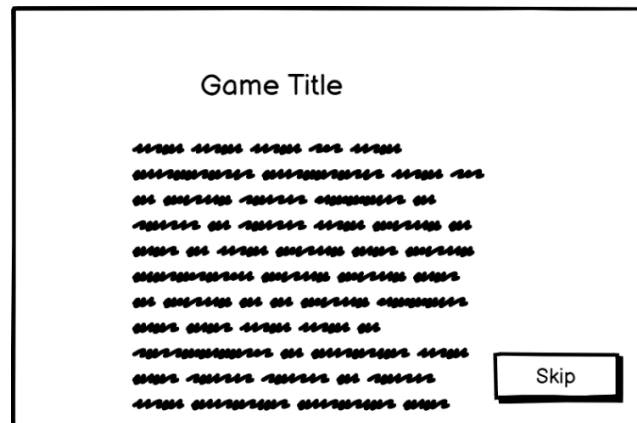
Gambar 3.22 Mockup Settings Menu

Jika pemain memilih opsi *instruction* maka pemain akan ditampilkan *instruction menu*, seperti pada gambar 3.23 dibawah. *Mockup instruction menu* terdiri dari teks instruksi yang bisa berganti, 2 *buttons* yang digunakan untuk mengubah teks instruksi menjadi control atau cara bermain.



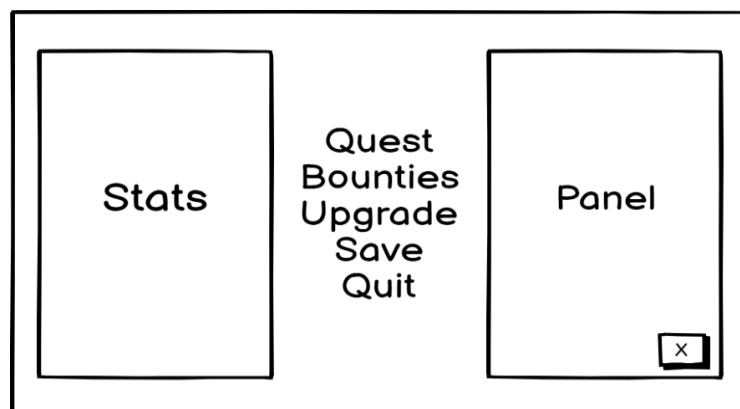
Gambar 3.23 Mockup Instruction Menu

Jika pemain memilih antara *start game* maka pemain akan bertemu dengan *story* seperti pada gambar 3.24 dibawah, sebelum bertemu dengan *quest menu*. *Mockup story* terdiri dari teks yang berisikan premise cerita, dan tombol *skip* yang dapat melewati proses *story* tersebut.



Gambar 3.24 Mockup Story Panel

Setelah pemain melewati *story panel* maka pemain akan ditemukan dengan *quest menu* seperti pada gambar 3.25 dibawah. *Mockup quest menu* terdiri dari *quest* untuk menampilkan *panel quest*, *bounties* untuk menampilkan *panel bounties*, *upgrade* untuk menampilkan *panel upgrade*, *save* untuk menampilkan *panel save*, *quit* untuk keluar menuju *main menu*, *panel stats* yang menampilkan *stats* pemain, dan *panel active menu* terkait.



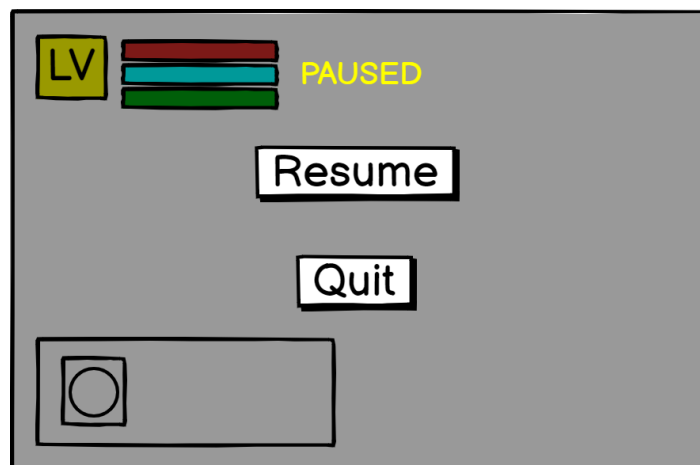
Gambar 3.25 Mockup Quest Menu

Setelah pemain memilih untuk menjalankan *start quest* ataupun *bounties* pemain akan masuk kedalam *gameplay*, dengan *mockup* seperti pada gambar 3.26 dibawah. *Mockup gameplay* terdiri dari *slider health*, *slider mana*, *slider experience*, sebuah panel *item*, dan teks level.



Gambar 3.26 Mockup Gameplay

Gambar 3.27 dibawah merupakan *mockup pause menu* yang bisa diakses oleh pemain saat melakukan proses *gameplay*. *Mockup pause menu* terdiri hanya dengan 2 *buttons* yakni *button resume* untuk melanjutkan permainan, dan *button quit* untuk keluar dari proses *gameplay*.



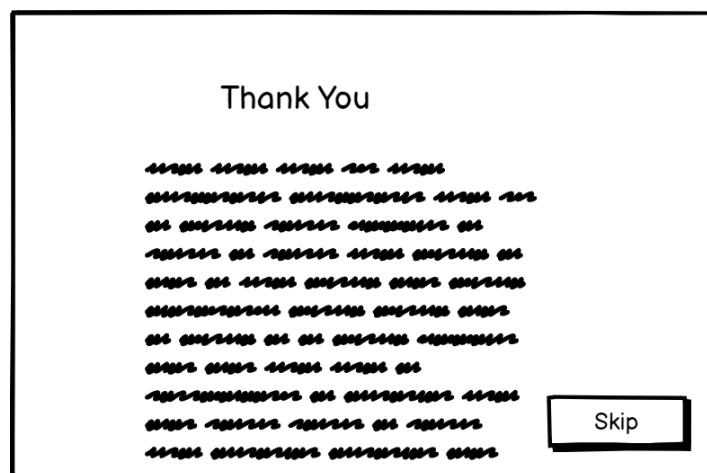
Gambar 3.27 Mockup Pause Menu

Jika pemain memenangkan *gameplay* ataupun kalah dalam *gameplay*, maka panel *game over* akan muncul seperti pada gambar 3.28 dibawah. *Mockup pause menu* terdiri dari teks *game over*, dan *button okay*.



Gambar 3.28 Mockup Game Over

Jika pemain sudah mengalahkan 3 *bounties* yang tersedia atau pemain memilih opsi *credits*, maka pemain akan masuk kedalam *credits screen* seperti pada gambar 3.29 dibawah. *Mockup credits* terdiri dari teks *credits* yang tertulis dengan kontribusi-kontribusi, dan *button skip* untuk melewati bagian *credits* dan kembali pada *main menu*.



Gambar 3.29 Mockup Credits