**4. IMPLEMENTASI SISTEM**

Pada bab ini akan dijelaskan tentang sistem yang digunakan sesuai dengan teori-teori dan desain sistem yang ada pada bab sebelumnya. Implementasi sistem yang akan dibahas meliputi pembuatan AI FSM dan AI GOAP menggunakan *Unreal Engine 4* *Blueprint* beserta potongan / segmen *screenshot* dari program yang dibuat berupa *blueprint*.

**4.1 Implementasi AI FSM pada *game***

Pada tabel 4.1, terdapat daftar-daftar prosedur yang diimplementasi untuk membuat AI FSM di dalam *game* beserta keterangan dan *flowchart* dari prosedur tersebut.

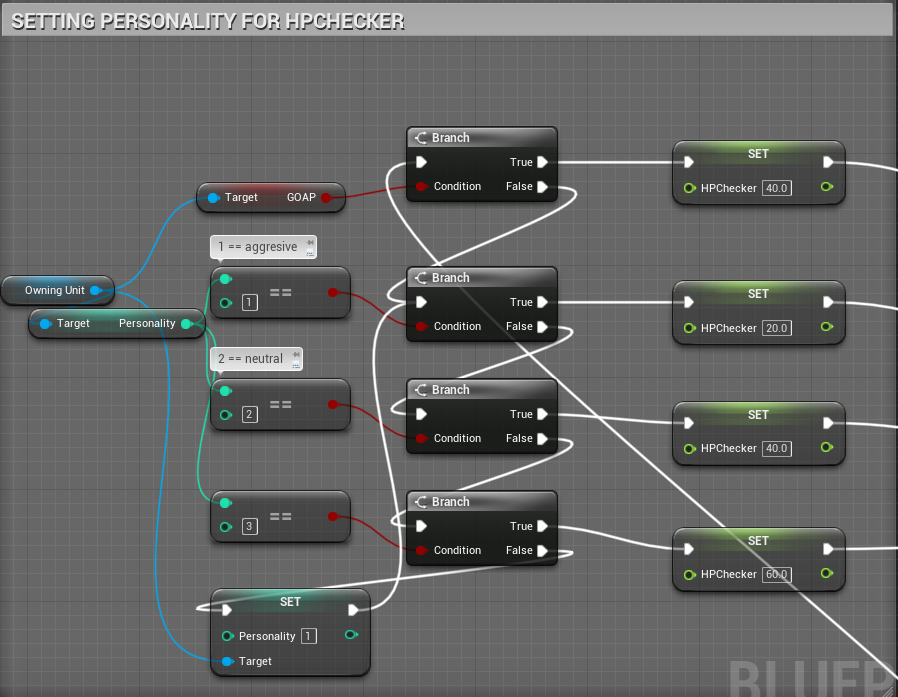
Tabel 4.1 Daftar prosedur yang digunakan untuk AI FSM

|  |  |  |  |
| --- | --- | --- | --- |
| Segmen Program | Nama Prosedur | Keterangan Prosedur | Gambar Flowchart |
| 4.1 | Pengaturan HPCheck untuk tiap personality | Mengatur HPCheck untuk tiap personality AI | - |
| 4.2 | Implementasi AI FSM | Membuat AI FSM sesuai dengan desain yang ada | Flowchart 3.2 |

**4.1.1 Implementasi Personality Karakter AI FSM**

Pada karakter AI FSM, terdapat tiga jenis personality yang digunakan agar alur permainan tidak monoton, yaitu *aggressive, neutral,* dan *non-aggressive*. Yang membedakan setiap *personality* adalah kondisi yang dibutuhkan untuk melakukan aksi “*Run away from enemy*”. Pada personality 1 (*aggressive)*, AI akan menjalankan aksi “*Run away from enemy”*, ketika HP berada di bawah atau sama dengan 20%. Pada personality 2 (*neutral*), AI akan menjalankan aksi “*Run away from enemy”*, ketika HP berada di bawah atau sama dengan 40%. Pada personality 3 (*non-aggressive*), AI akan menjalankan aksi “*Run away from enemy”*, ketika HP berada di bawah atau sama dengan 60%. Segmen program ini dapat dilihat pada Segmen 4.1.

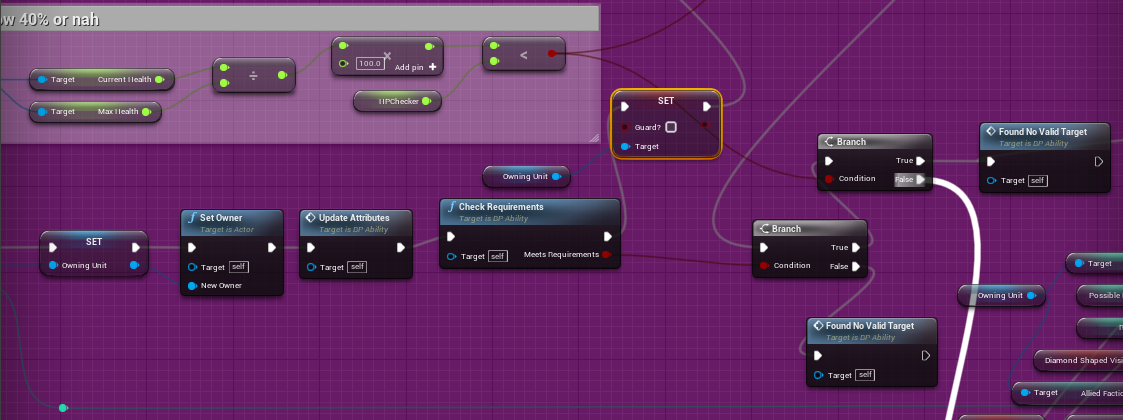
Segmen 4.1 Inisialisasi kondisi HPCheck tiap personality



**4.1.2 Implementasi AI FSM**

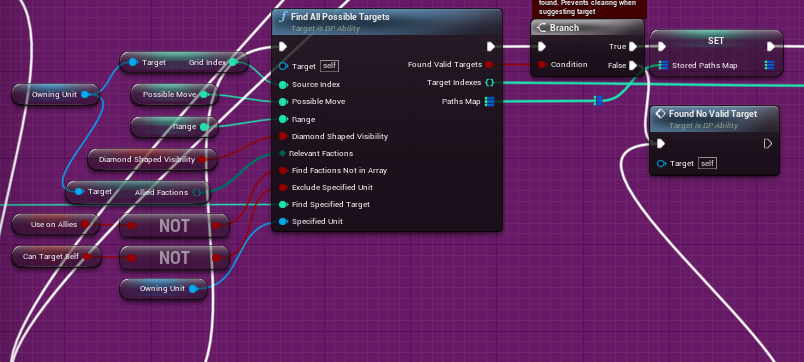
Implementasi AI FSM merupakan sebuah prosedur yang digunakan untuk membuat AI FSM yang bekerja sesuai pada gambar 3.2 yang sudah dibahas sebelumnya. AI FSM disini berfungsi sebagai bahan penelitian yang digunakan untuk melawan AI GOAP nantinya, untuk melihat apakah AI GOAP mampu melawan AI FSM. Segmen program ini dapat dilihat pada Segmen 4.2.

Segmen 4.2 Implementasi AI FSM



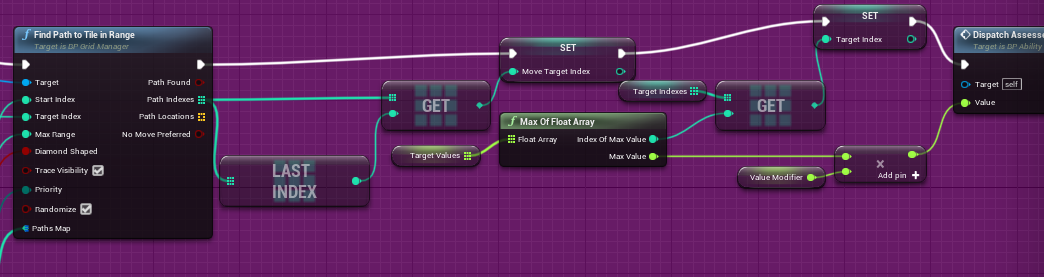
Ketika putaran suatu karakter AI dimulai, maka *state guard* yang dimiliki AI akan dinonaktifkan. Lalu akan dicek apakah HP yang dimiliki AI dibawah HPChecker atau tidak. Jika ya, maka ia akan menjalankan aksi “*Run away from enemy*”.

**Segmen 4.2 Implementasi AI FSM (Lanjutan)**

****

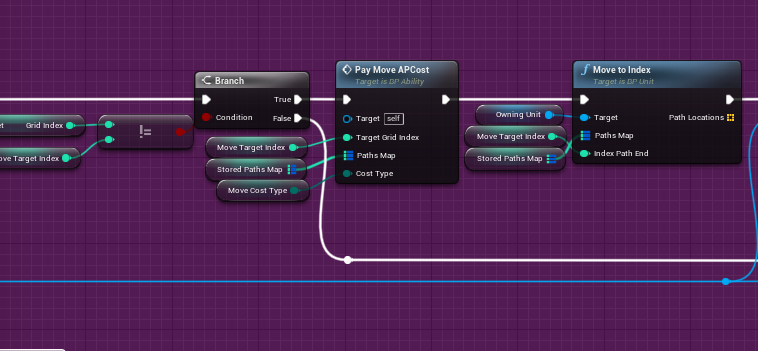
Jika tidak, maka AI akan menjalankan aksi “*Kill enemy”*. AI akan berusaha mencari musuh terdekat dari posisinya.

**Segmen 4.2 Implementasi AI FSM (Lanjutan)**



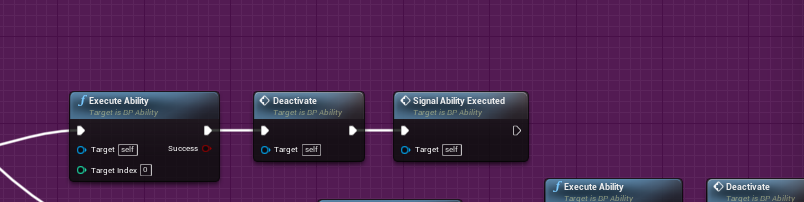
Setelah menemukan musuh terdekat, maka AI FSM akan mencari *tile* yang paling dekat dengan musuh tersebut sesuai *move range* yang dimiliki karakter. Setelah menemukan *tile* yang paling dekat dengan musuh, maka AI akan mengatur tujuan *tile* ke *tile* tersebut.

**Segmen 4.2 Implementasi AI FSM (Lanjutan)**



Setelah mengatur tujuan *tile*, AI akan berjalan ke *tile* tersebut. Apabila AP yang dimiliki AI masih 2, maka AI dapat jalan ke *tile* tersebut.

**Segmen 4.2 Implementasi AI FSM (Lanjutan)**



Apabila *range* *attack* milik AI dapat mencapai musuh, maka AI akan mengeksekusi kemampuannya sesuai dengan *job* karakter masing-masing.

**4.2 Implementasi AI GOAP pada *game***

Pada tabel 4.2, terdapat daftar-daftar prosedur yang diimplementasi untuk membuat AI FSM di dalam *game* beserta keterangan dan *flowchart* dari prosedur tersebut.

Tabel 4.2

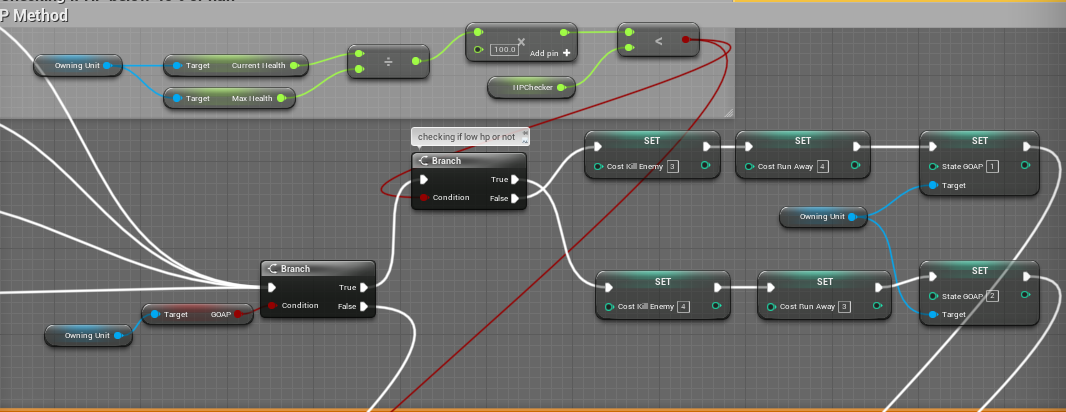
Daftar prosedur yang digunakan untuk AI GOAP

|  |  |  |  |
| --- | --- | --- | --- |
| Segmen Program | Nama Prosedur | Keterangan Prosedur | Gambar Flowchart |
| 4.3 | Mengatur *cost action* | Mengatur *cost* untuk setiap *action* yang ada di dalam AI GOAP, dan mengubah *cost-cost* tersebut di situasi tertentu (*situational costs*) | - |
| 4.4 | Implementasi AI GOAP | Membuat AI GOAP sesuai dengan desain yang ada | Flowchart 3.3 |

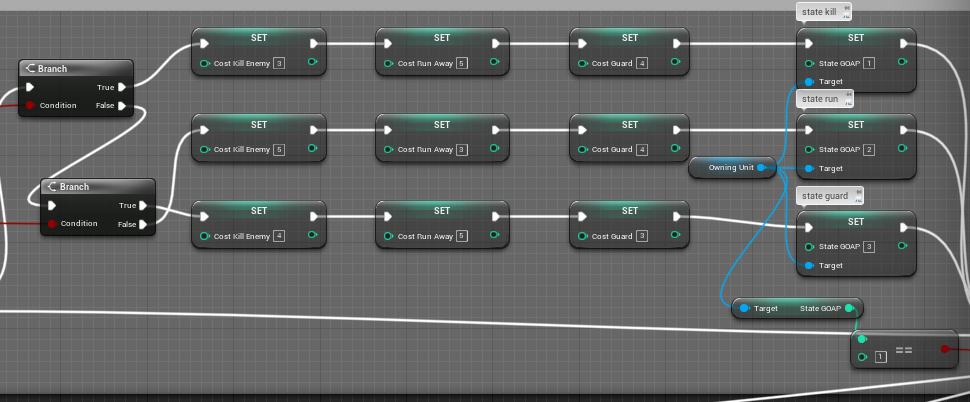
**4.2.1 Implementasi *Cost* pada AI GOAP**

Pada karakter AI GOAP, terdapat empat *action* yang dapat dieksekusi, yaitu *attack, move, guard,* dan *heal*. Ketika HP yang dimiliki AI GOAP diatas 40%, maka AI GOAP akan berusaha menyerang musuh. Dengan begitu, maka AI GOAP akan menjalankan fungsi *move to X* hingga musuh ada di dalam *range attack* AI GOAP. Setelah musuh ada di *range attack* AI GOAP, maka AI GOAP akan menjalankan fungsi *attack*. Tetapi, jika HP yang dimiliki AI GOAP dibawah 40%, maka AI GOAP akan menjalankan fungsi *move to target X* dan *heal*. Semua hal tersebut diatur melalui *cost action* masing-masing sesuai dengan kondisi yang dimiliki AI GOAP. Segmen program ini dapat dilihat pada Segmen 4.3

Segmen 4.3 Mengatur *cost action*



**Segmen 4.3 Mengatur *cost action* (Lanjutan)**

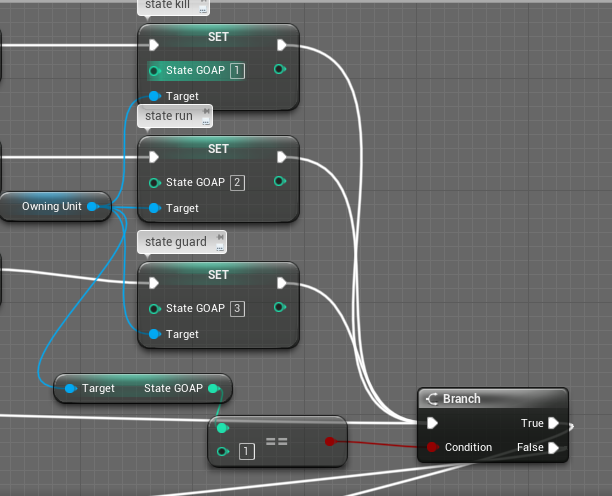


Kemudian, apabila AI GOAP sedang konflik dengan musuh, akan dilakukan pengecekan lagi untuk melakukan *situational costs*. Apabila HP yang dimiliki oleh AI GOAP dibawah 40%, dan HP yang dimiliki oleh AI GOAP lebih besar dari musuh, maka *cost attack* akan direndahkan, sehingga AI GOAP akan melakukan aksi *attack*. Apabila HP yang dimiliki oleh AI GOAP dibawah 40% dan perbedaan HP AI GOAP dan musuh diatas 10, maka AI GOAP akan melakukan *run away from enemy* dan *heal*. Tetapi, apabila perbedaan HP AI GOAP dan musuh dibawah 10, maka AI GOAP akan melakukan *run away* *from enemy* dan *guard*.

**4.2.2 Implementasi AI GOAP**

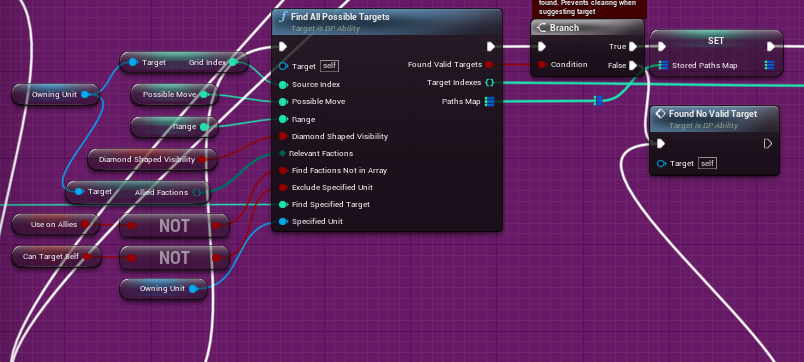
Implementasi AI GOAP merupakan prosedur pembuatan AI GOAP yang digunakan pada penelitian ini untuk menguji apakah AI GOAP mampu melawan AI FSM nantinya. Segmen program ini dapat dilihat di Segmen 4.4

Segmen 4.4 Implementasi AI GOAP

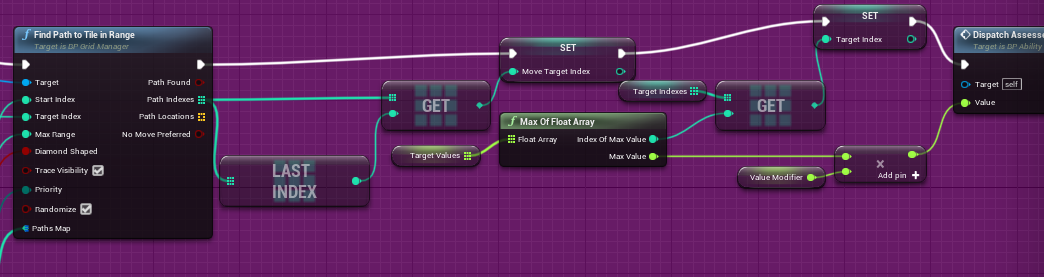


AI GOAP akan menentukan aksi mana yang akan dikerjakan sesuai dengan kondisi AI sekarang.

**Segmen 4.4 Implementasi AI GOAP (Lanjutan)**

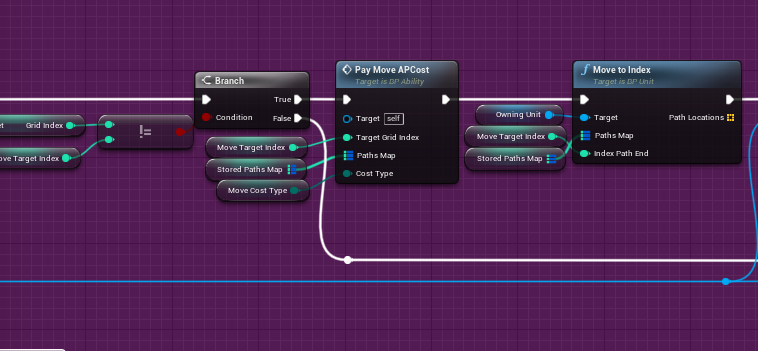
****

Apabila AI GOAP menentukan untuk menyerang musuh, maka AI GOAP akan mencari musuh terdekat dari posisinya.



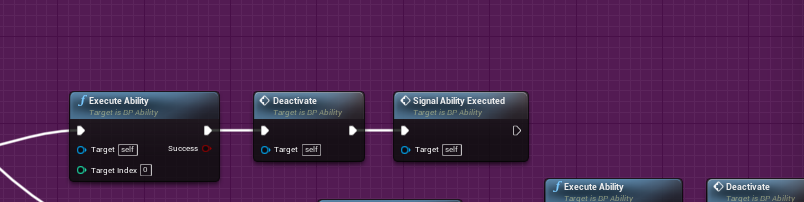
Setelah menemukan musuh terdekat, maka AI GOAP akan mencari *tile* yang paling dekat dengan musuh tersebut sesuai *move range* yang dimiliki karakter. Setelah menemukan *tile* yang paling dekat dengan musuh, maka AI akan mengatur tujuan *tile* ke *tile* tersebut.

**Segmen 4.4 Implementasi AI GOAP (Lanjutan)**



Setelah mengatur tujuan *tile*, AI akan berjalan ke *tile* tersebut. Apabila AP yang dimiliki AI masih 2, maka AI dapat jalan ke *tile* tersebut.

**Segmen 4.4 Implementasi AI GOAP (Lanjutan)**

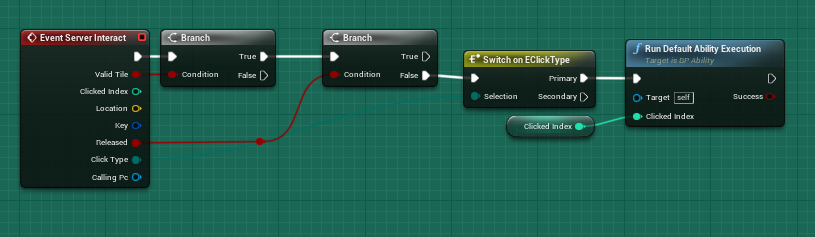


Apabila *range* *attack* milik AI dapat mencapai musuh, maka AI akan mengeksekusi kemampuannya sesuai dengan *job* karakter masing-masing.

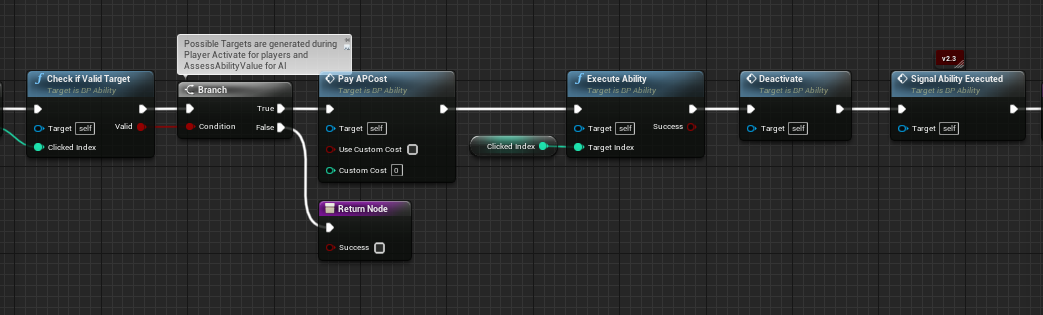
**4.3 Implementasi Player Turn**

Implementasi *Player Turn* digunakan agar *player* bisa menggerakkan karakternya ke *tile* atau musuh yang diinginkan. Segmen program ini dapat dilihat pada Segmen 4.5.

Segmen 4.5 Implementasi Player Turn



**Segmen 4.5 Implementasi Player Turn (Lanjutan)**

****

Ketika *player* memilih *tile* dan kemampuan karakter, maka sistem akan menjalankan karakter tersebut sesuai dengan yang sudah dipilih oleh *player*.