



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### METODE PENELITIAN

#### 3.1. Objek Penelitian

Objek penelitian untuk diselesaikan pada skripsi ini adalah *AI* untuk *hero Zeus* pada *game* Dota 2. Ada dua pertimbangan utama dalam pemilihan *Zeus* sebagai objek penelitian untuk *dynamic scripting*.

Pertimbangan pertama *Zeus* adalah *hero* yang mampu memiliki banyak peran dalam permainan. Pertama, *Zeus* mampu berperan sebagai *nuker* (bab II, sub bab 2.6) yaitu peran dimana *Zeus* menggunakan *skill-skill* miliknya untuk terus menerus mengurangi *health point (hp)* lawannya. *Zeus* juga dapat berperan menjadi *semi-carry*. *Carry* sendiri merupakan peran dimana suatu *hero* menjadi senjata utama di dalam timnya untuk mengalahkan *hero-hero* pada tim musuh. *Semi-carry* merupakan peran yang sama seperti *carry* hanya saja perlu *item-item* khusus untuk bermain sebagai peran ini (bab II, sub bab 2.6). *Zeus* juga mampu bermain sebagai *support* untuk membantu *carry* pada tim agar *carry* tidak mudah didekati oleh lawan.

Pertimbangan kedua adalah *Zeus* memiliki banyaknya kondisi yang harus dihadapi selama pertandingan. Salah satu kondisi tersebut adalah kondisi saat *mana Zeus* sudah habis yang mengakibatkan *Zeus* tidak mampu menggunakan *skill-nya*, dan jika *Zeus* tidak mampu menggunakan *skill* maka ia rentan saat

bertarung melawan musuh mengingat *Zeus* adalah *hero* yang kekuatannya terdapat pada *skill* aktifnya (bab II, sub bab 2.6). *Mana* sendiri adalah sebuah atribut yang dimiliki semua *hero* dan berguna untuk mengeluarkan *skill* (bab II, sub bab 2.6). Jika sudah memasuki sebuah kondisi seperti di atas maka diperlukan sebuah solusi, yaitu dengan memilih aksi yang dapat memecahkan masalah dalam kondisi tersebut.

Dengan banyaknya peran dan kondisi yang dapat dialami oleh *Zeus* maka banyak aksi-aksi yang dapat ia lakukan, oleh karena itu *Zeus* dipilih sebagai objek penelitian ini. Penelitian skripsi bagian pertama ini akan membuat *AI Zeus* yang mampu memiliki performa yang lebih tinggi daripada *AI* standar pada tingkat *unfair*. Skripsi bagian pertama ini lalu dilanjutkan dengan peningkatan kemampuan *AI* agar mampu mengalahkan pemain manusia pada bagian kedua. Keduanya menggunakan format 1 vs 1 *solo mid only* seperti yang dilakukan oleh OpenAI, yaitu pertandingan dimana hanya *lane* atau jalur tengah saja yang aktif karena dioptimisasi untuk pertandingan satu lawan satu.

### **3.2. Tools dan fungsinya dalam desain AI**

#### **1. Visual Studio Code**

Penelitian ini menggunakan visual studio code sebagai *text editor* untuk *scripting AI*. Visual studio code memiliki *plugin* untuk melakukan *highlight syntax* lua sehingga dapat mempermudah proses *scripting*. Lua merupakan bahasa pemrograman yang

digunakan untuk keperluan *scripting* dan dikembangkan dengan menggunakan ANSI C. Dota 2 sendiri menggunakan bahasa pemrograman lua untuk mengembangkan *addons* maupun *AI*-nya.

## 2. Python

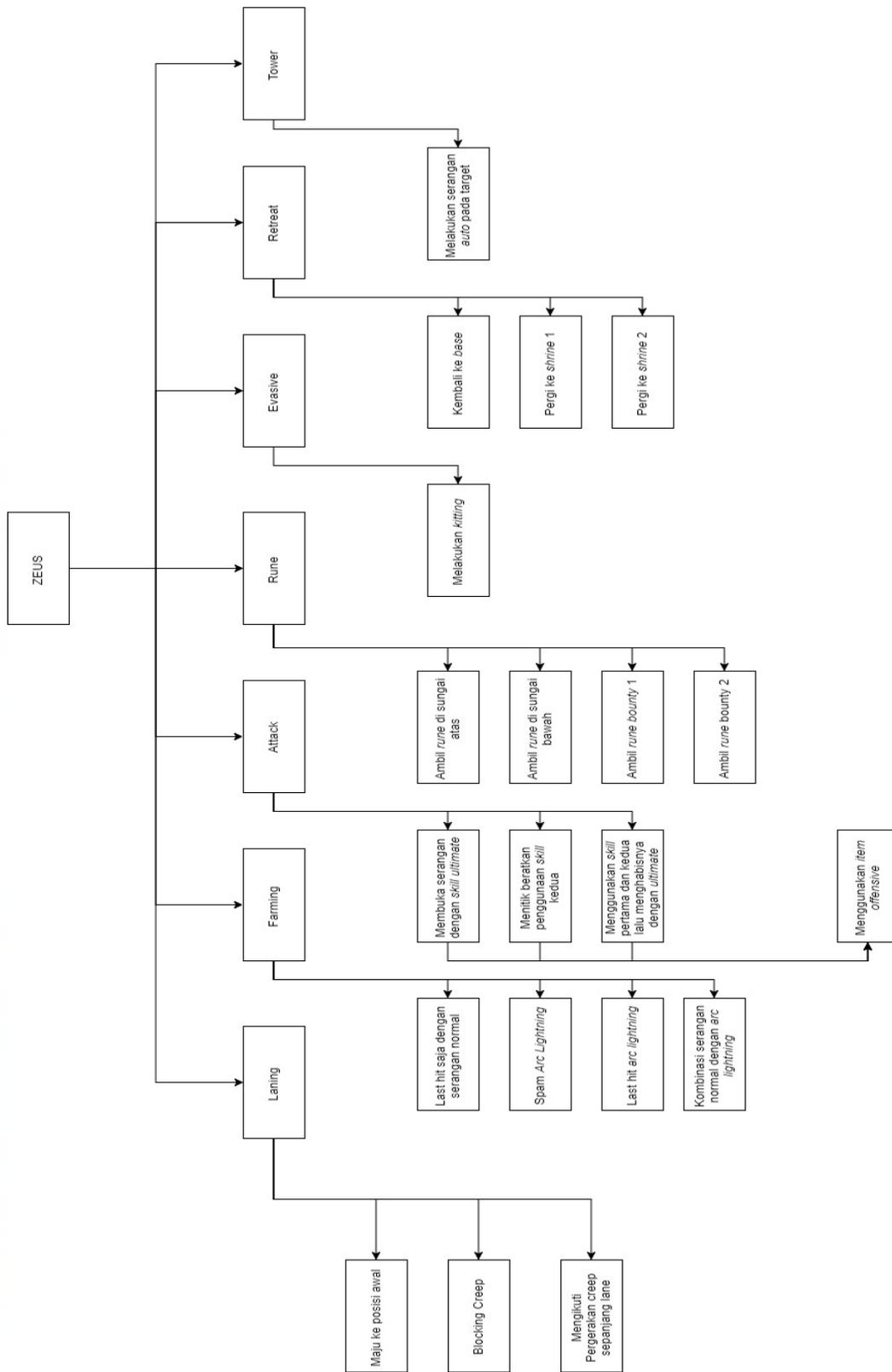
Python digunakan untuk membangun sebuah *service* yang dapat digunakan sebagai API agar *scripting AI* dengan menggunakan lua mampu menjalankan fungsi penulisan *weight* ke dalam sebuah *file*.

## 3. Hammer

Hammer merupakan *tools* untuk mengedit *map* pada Dota 2. Dengan menggunakan *tools* ini koordinasi x, y, dan z suatu tempat pada *map* dapat diketahui sehingga titik koordinasi tersebut berguna untuk menentukan pergerakan *AI* dari satu titik ke titik yang lainnya.

## 4. Dota 2 Workshop Tool

Dota 2 Workshop Tool merupakan *software* yang sudah disediakan oleh Valve selaku *developer* bagi para *modder* untuk melakukan modifikasi atau menambah *addons* pada Dota 2. *Tools* ini digunakan untuk meng-*upload script AI* ke dalam Dota 2 agar dapat digunakan.



Gambar 3.1.1. Diagram Aksi

### 3.3. Pembagian *Mode*

Dalam penelitian ini *AI Zeus* akan dibagi ke dalam beberapa *mode* dan terdapat aksi-aksi yang sesuai di dalam *mode* tersebut. Tiap-tiap *mode* memiliki karakteristik dan tanggung jawab yang berbeda-beda dalam menentukan pergerakan *AI*. Di dalam *mode* tersebut akan terdapat aksi -aksi. Secara umum *mode* dan aksinya dapat dilihat di gambar 3.1. Setiap aksi di dalam *mode* ini memiliki probabilitas untuk aktif. Nilai probabilitas ini dipengaruhi oleh reward dan punishment yang mampu menambah atau mengurangi probabilitas aksi sehingga dapat dipelajari oleh *AI*. Menurut Silva, dan Chaimowicz (2017) ada 7 perilaku utama dalam bermain *game* dengan *genre MOBA*, yaitu *farming*, *attacking*, *laning*, *evasive*, *retreat*, *rune/powerup*, dan *building push/tower*, maka dari itu *AI Zeus* dibagi ke dalam 7 *mode* yang memiliki perilaku berbeda-beda sebagai berikut :

#### 1. *Laning*

Sesuai namanya *mode* ini bertugas untuk menentukan dimana *AI* akan ditempatkan dan menggerakkan *AI* untuk berjalan dari satu point ke point yang lain di sepanjang *lane* atau jalur. Ada tiga jalur utama di Dota 2, yaitu TOP(jalur atas), MID(jalur tengah), dan BOT(jalur bawah) seperti diilustrasikan pada gambar 3.2.



Gambar 3.2. Tiga Lane Utama pada Dota 2

Sumber : (Drachen, A., Yancey, M., Maguire, J., dan Klabajan, D., 2014)

## 2. Farming

Mode ini berisi aksi-aksi untuk melakukan *farming*, yaitu aksi untuk melakukan serangan terhadap *creep* untuk mendapatkan *gold* dan *experience point*. *Creep* adalah pasukan kecil milik suatu tim yang muncul berkelompok untuk menekan *lane* musuh. *Creep* ini muncul setiap 20 detik sekali dari markas dan bergerak lurus mengikuti alur *lane*.

## 3. Attacking

Mode ini berisi aksi-aksi untuk melakukan serangan terhadap *hero* musuh, dan *skill* apa saja yang harus digunakan. AI menentukan apakah akan menyerang musuh dengan serangan biasa atau dengan

menggunakan *skill*. Bila mengeluarkan *skill mode* ini mengatur urutan *skill* yang dikeluarkan.

#### 4. *Evasive*

*Mode* ini berisi aksi-aksi *AI* untuk menghindari kemungkinan-kemungkinan serangan yang dapat membahayakan *AI*. Tugas dari *mode* ini adalah untuk menjaga agar *AI* tidak dikalahkan oleh musuh dalam pertarungan.

#### 5. *Retreat*

*Mode* ini berisi aksi untuk melarikan diri dari *lane* dimana *AI* berada untuk mengisi *hp* atau *mana*. Tugas dari *mode* ini adalah menangani *AI* yang sedang dalam keadaan kritis, dan perlu mengisi *health point* serta *mana*-nya.

#### 6. *Rune*

*Mode* ini berisikan aksi-aksi *AI* untuk mengambil *rune* di tempat yang berbeda-beda. *Rune* ini berguna sebagai *temporary power up*, yaitu penambah atau memodif *status* untuk sementara waktu. Ada 7 jenis *rune* di Dota 2, dan masing-masing memiliki efek yang unik.

#### 7. *Tower*

*Mode* ini bertugas untuk memerintahkan *AI* menyerang *Tower* musuh, dan menghancurkannya. *Tower* merupakan bangunan pertahanan yang menjaga *lane* dari arus serangan musuh (bab II, sub bab 2.5).

Masing-masing tim memiliki 3 *tower* yang menjaga *lane* dari serangan musuh dan menghambat pergerakan mereka. Gambar 3.3 menunjukkan lokasi *tower* di masing-masing tim yang ditandai dengan balok.



**Gambar 3.3. Lokasi *Tower* Pada Peta**

**Sumber : (Learn Dota, t. thn.)**

### **3.4. Mendefinisikan Aksi dalam *Mode***

Pada tiap *mode*, AI memiliki aksi yang mencirikan *mode* tersebut. Mengikuti ciri dari tiap *mode* yang unik maka pendefinisian aksi adalah sebagai berikut :

#### **1. *Laning***

- a. Maju ke posisi awal, yaitu ke daerah *tower* ketiga. *Tower* ketiga adalah *tower* yang berada di pintu masuk markas suatu tim, dan merupakan pertahanan terakhir sebelum tim musuh dapat masuk ke dalam markas.

- b. Melakukan *blocking creep* atau menghalangi jalan *creep*. *Blocking creep* merupakan suatu teknik untuk menghalang pergerakan *creep* di tim sendiri sampai sebelum *tower* pertama dengan tujuan agar *creep* musuh masuk ke daerah *tower* pertama suatu tim. Karena *damage* tower yang besar, pada awal permainan dapat menghabisi *hero* dan *creep* musuh dalam sekejap.
- c. Maju di sepanjang *lane* mengikuti pergerakan *creep*. Bertujuan agar *AI* mampu bergerak di sepanjang *lane* mengikuti *creep* dan ikut dalam memberikan tekanan pada pertahanan musuh. Ini adalah aksi utama dalam *mode laning*.

## 2. *Farming*

- a. Melakukan serangan normal *last hit* pada *creep* musuh. *Last hit* sendiri merupakan suatu teknik untuk mendapatkan tambahan *gold* dan *experience point*. *Last hit* dilakukan dengan memberi pukulan terakhir kepada *creep* musuh.
- b. Menyerang terus-menerus *creep* hanya dengan menggunakan *skill* pertama *Zeus*. *Skill* pertama *Zeus* adalah *arc lightning*, yaitu *skill* yang dapat mengenai banyak target sekaligus (bab II, sub bab 2.6). Aksi ini boros penggunaan *mana* tetapi sangat berguna untuk menghabisi *creep* musuh dengan sangat cepat.
- c. Melakukan *last hit* terhadap *creep* yang memiliki *HP* rendah dengan menggunakan *skill* pertama *Zeus*. Penggunaan *skill* pertama pada

*creep* yang memiliki *hp* rendah adalah untuk mendapatkan *last hit* sekaligus mengurangi *hp creep* yang lain.

- d. Melakukan *last hit* terhadap *creep* yang memiliki *HP* rendah tetapi diselingi dengan serangan normal. Aksi ini ditujukan untuk menghemat mana dengan mendapatkan *last hit* menggunakan skill pertama dan mengurangi *hp creep lain* dengan serangan normal. Aksi ini juga berguna untuk menghabisi *creep* di *lane* musuh dengan cepat.

### 3. *Attacking*

- a. Menyerang *hero* musuh dengan menggunakan *ultimate skill* milik *Zeus* sebagai pembuka, lalu dilanjutkan dengan *skill kedua*, dan bila *skill kedua* sedang dalam proses *cooldown*, maka gunakan *skill pertama*. Aksi ini sering digunakan untuk memulai perang antar tim, karena dengan menggunakan *skill* ini semua anggota tim musuh akan berkurang *hp*-nya.
- b. Menyerang *hero* musuh dengan menggunakan *skill kedua*, lalu diikuti *skill pertama* dan disertai dengan serangan normal. Bila *HP hero* musuh mencapai 50% gunakan *ultimate skill* lalu kembali gunakan *skill kedua*. Aksi serangan tipe ini berguna untuk mengejar ketertinggalan *hp* saat bertarung melawan musuh.
- c. Menyerang *hero* musuh dengan menggunakan *skill kedua*, lalu diikuti *skill pertama* dan disertai dengan serangan normal. Bila *HP*

*hero* musuh mencapai 20% gunakan *ultimate skill*. Aksi ini berguna bila ada musuh yang sudah hampir kalah lalu melarikan diri. *Skill ultimate Zeus* mampu menjangkau seluruh area.

d. Jika *Zeus* membeli *item* penunjang serangan maka ketiga aksi tersebut menambahkan penggunaan *item* tersebut sebelum menggunakan *skill* untuk mendapat *output damage* yang lebih besar.

#### 4. *Evasive*

Melakukan *kitting*, yaitu aksi menjauhi musuh sampai jarak aman lalu maju kembali bila sudah dalam jarak aman. *Kitting* berguna untuk keluar dari area serangan musuh sehingga musuh tidak dapat menyerang *target*.

#### 5. *Rune*

Terdapat 4 lokasi *rune* yang dapat dihampiri oleh aksi di *mode* ini. 2 terletak di sungai penghubung antara *lane* atas dengan tengah dan sungai penghubung *lane* bawah dengan tengah. 2 *rune* lagi terletak di hutan masing-masing tim. Lokasi *rune* dapat dilihat pada gambar 3.4 yang ditandai dengan titik biru.

U  
M  
M  
N



**Gambar 3.4. Lokasi *Rune* (Titik Biru)**

**Sumber : (Gamepedia, t. thn.)**

- a. Mengambil *rune* di sungai bagian atas.
- b. Mengambil *rune* di sungai bagian bawah.
- c. Mengambil di hutan bagian bawah milik tim *Zeus* berada.
- d. Mengambil di hutan bagian atas milik tim *Zeus* berada.

#### 6. *Tower*

Melakukan serangan normal ke *tower* musuh. Aksi ini dilakukan bersama-sama dengan *creep*. Jadi jika *creep* berhenti menyerang *tower*, *AI* akan berhenti menyerang *tower* dan berpindah *mode*.

#### 7. *Retreat*

- a. Kembali ke markas untuk mengisi *HP*. *AI* akan mengisi *HP* di *fountain* timnya. *Fountain* sendiri adalah sebuah bangunan yang

terletak di dalam markas suatu tim, dan memiliki kemampuan untuk meregenerasi *HP* dan *mana* dengan sangat cepat.

- b. Pergi ke *shrine* yang terdapat di hutan bagian bawah untuk mengisi *HP*. *Shrine* adalah sebuah bangunan yang dapat meregenerasi *HP* dan *mana* selama 5 detik. Setelah diaktifkan *shrine* tidak dapat digunakan lagi dalam kurun waktu 5 menit.
- c. Pergi ke *shrine* yang terdapat di hutan bagian atas untuk mengisi *HP*.

### 3.5. Menentukan *Item Build Zeus*

*AI Zeus* ini memiliki set *item build* atau *item-item* apa sajakah yang harus dibeli. *Item* ini terdiri dari *item* awal pada permulaan permainan, *item* penting yang harus segera dimiliki, *item* untuk permainan jangka panjang. Untuk menentukan *item set* Zeus maka dilakukan observasi pada reka ulang permainan para pemain profesional dalam menggunakan Zeus. Melalui observasi dari permainan yang dilakukan oleh beberapa pemain profesional yang memainkan Zeus maka didapatkan beberapa daftar-daftar dari *item-item* tersebut pada tabel 3.1 yang seringkali digunakan dalam pertandingan. Kegunaan dari *item* tersebut dapat dilihat pada bab II, sub bab 2.6.

**Tabel 3.1. Daftar *Item* yang Dibeli**

<i>Item Awal</i>	<i>Item Penting</i>	<i>Item Akhir</i>
<i>Tango, clarity potion, dan mantle</i>	<i>Null Talisman, Bottle, arcane boots, dan soul ring.</i>	<i>Force staff, ultimate scepter dan scythe of vyse</i>

### 3.6. Menentukan Skenario Pergerakan Awal Zeus

Menurut *Spronck et. al.* dalam penelitiannya “*Adaptive Game AI with Dynamic Scripting*” pada keadaan awal *AI* diberikan aksi yang konkrit yang maksudnya adalah bukan aksi yang *random*. Maka dari itu pada awal pertandingan *AI Zeus* diberikan aksi yang konkrit, yaitu pertama-tama melakukan pengecekan terhadap *item-item* yang harus dibeli pada *script item*. Setelah membeli *item* awal *Zeus* pergi ke *tower* ketiga pada *lane* mid untuk menunggu *creep* tim *Dire* muncul. Gambar 3.5 menunjukkan posisi awal *Zeus* di *tower* ketiga. Pada saat *creep* muncul maka *Zeus* akan melakukan *blocking* pada *creep* untuk menghambat lajunya sampai bertemu dengan *creep* musuh. Setelah bertemu maka mekanisme pemilihan aksi pada *rulebase* akan bekerja dan menentukan aksi yang akan digunakan oleh *Zeus*.



Gambar 3.5. Posisi awal Zeus

### 3.7. Perhitungan *Reward* dan *Punishment-Rulebase* Aksi

Semua aksi-aksi *Zeus* dimasukkan ke dalam *rulebase*, dan aksi-aksi tersebut memiliki properti nilai yang dapat berkurang maupun bertambah sesuai dengan kondisinya apakah aksi itu dinilai berhasil atau tidak. Pada awal semua aksi memiliki nilai yang sama yaitu, 100 sehingga masing-masing aksi memiliki probabilitas yang sama untuk dikeluarkan. *Reward* akan diberikan untuk aksi yang berhasil memenuhi *parameter* keberhasilan, dan akan di-*punish* jika gagal. *Reward* atau *punishment* yang diberikan adalah maksimal 100 untuk *reward* dan maksimal 75 untuk *punishment*.

Penilaian untuk menentukan *reward* dan *punishment* dilakukan pada *mode farming, attacking, rune, dan evasive*. Pada keempat *mode* tersebut terdapat aksi-aksi yang dilakukan oleh *AI* yang diatur oleh mekanisme pemilihan aksi berdasarkan nilai *weight*-nya, aksi yang memiliki nilai *weight* lebih tinggi daripada aksi lainnya akan memiliki probabilitas dipilih untuk menjadi aksi aktif lebih tinggi pula. Untuk ketiga *mode* lainnya yaitu *laning, retreat, dan tower* memiliki aksi yang terbatas terutama dalam *environment testing 1 vs 1 solo mid* yang akan dilakukan sehingga pada ketiga *mode* ini tidak diatur oleh mekanisme pemilihan aksi. Semua *mode* tersebut membentuk suatu *script* utuh untuk dieksekusi.

### 3.8. Proses Learning AI

Pada setiap awal *AI* memasuki sebuah *mode* (*farming*, *attacking*, *rune*, *retreat*) maka ia akan memilih aksi dimana aksi yang memiliki nilai *weight* yang tinggi dibandingkan dengan yang lainnya memiliki peluang lebih besar untuk terpilih. Lalu setelah *AI* keluar dari *mode* tersebut maka aksi yang dilakukan pada *mode* tersebut akan dievaluasi dan nilai *weight* dari aksi tersebut akan berubah. Parameter keberhasilan aksi pada tiap-tiap *mode* berbeda, sesuai dengan karakteristik *mode* tersebut :

#### 1. Farming

Jumlah *last hit creep*. Nilai *weight* pada aksi akan bertambah jika jumlah *last hit* bertambah pada saat *AI* berupaya melakukan *last hit*, tetapi *weight* berkurang apabila *AI* berupaya melakukan *last hit* tetapi jumlahnya tidak bertambah.

#### 2. Attacking

Jumlah *hp* musuh yang berkurang, dan jumlah *hp AI* yang berkurang. Jika jumlah *hp* musuh yang berkurang lebih banyak dibandingkan dengan jumlah *hp AI* yang berkurang maka aksi berhasil.

#### 3. Rune

*Rune* berhasil diambil, atau berhasil disimpan. Jika *AI* berhasil *rune* terlebih dulu dibandingkan dengan musuhnya maka aksi dianggap berhasil.

#### 4. Evasive

Parameter penilaiannya adalah *HP AI* setelah *AI* mengeksekusi aksi *evasive*. Jika *AI* berhasil menghindari dari musuh dengan jumlah *HP* sebanyak mungkin maka aksi yang dipilih untuk *mode evasive* ini dianggap berhasil.

### 3.9. Skenario Testing

Proses *testing* dilakukan dalam 2 tahap. *Testing* pertama mempertandingkan *adaptive AI* dengan *AI* standar Dota 2 dengan tingkat kesulitan tertinggi (*unfair*). *Testing* kedua adalah untuk melakukan pengujian pada *adaptive AI* setelah diperkuat melalui analisis dari hasil *testing* pertama dengan mempertandingkannya melawan pemain manusia.

*Test* pertama adalah *AI* baru dihadapkan dengan *AI* standar dari Dota 2 dengan tingkat kesulitan tertinggi, yaitu *unfair*. *Test* pertama ini dilakukan sebanyak 50 kali pertandingan. Pemilihan *hero* lawan dilakukan secara *random*. Format pertandingan yang digunakan adalah 1 vs 1 *solo mid*, yaitu pertandingan yang difokuskan untuk 1 lawan 1. *Adaptive AI* yang digunakan adalah *adaptive AI* dengan nilai *weight* awal (*AI* yang belum pernah belajar). Setelah melewati 50 pertandingan maka *adaptive AI* akan dievaluasi, parameter yang digunakan untuk evaluasi pada *testing* pertama ini adalah *win rate*, rasio *k/d*, *xpm*, *gpm*, dan *item build*. Hasil dari evaluasi tersebut akan digunakan sebagai dasar untuk melanjutkan

pengembangan *AI* yang akan digunakan pada *testing* kedua. Penjelasan parameter – parameter ini terdapat pada sub bab 3.10

*Test* kedua merupakan lanjutan dari hasil *test* pertama, dimana *adaptive AI* yang digunakan sebelumnya akan melalui proses *tuning* dan *training* kembali (Nilai *weight* dilanjutkan dari *test* sebelumnya) untuk melawan pemain manusia. Pada tahap *testing* ini dikumpulkan 6 orang subjek dan *AI* standar dari Dota 2 sendiri sebanyak 100 kali. Subjek yang dipilih terdiri dari 3 kategori keahlian, yaitu pemain yang belum pernah bermain Dota 2 sebelumnya, pemain Dota 2 pemula, dan pemain Dota 2 yang sudah cukup berpengalaman bermain Dota 2. Mereka yang disebut pemain Dota 2 pemula ialah mereka yang baru saja berhasil mendapatkan *rank* awal pada *online ranked match* (*rank Herald*), sedangkan pemain yang sudah memiliki pengalaman bermain Dota 2 ialah pemain yang sudah memiliki *rank Guardian* sampai dengan *Immortal* bermain (melewati *rank Herald*). Pada *test* kedua setiap subjek, yaitu pemain akan bermain masing – masing sebanyak 5 kali melawan *AI* standar dan 5 kali melawan *adaptive AI*. Untuk pemilihan *hero*, subjek menggunakan *hero* yang dipilih secara *random*. Format pertandingan adalah sama dengan tahap *test* pertama, yaitu 1 vs 1 solo mid. Setelah pertandingan maka dilakukan evaluasi menggunakan *win rate*, rata – rata dari jumlah *last hit*, *xpm*, dan *gpm*. Selain melakukan analisis kuantitatif, *test* kedua ini juga menggunakan analisis subjektif yaitu menggunakan survey dengan skala *Likert* dan publikasi pada *steam workshop* (portal untuk berbagi pada *platform steam*).

### 3.10. Variabel Evaluasi

Analisis dilakukan menjadi 2 tahap, yaitu tahap pertama analisis dari hasil *dynamic AI* melawan *AI* standar Dota 2 dan tahap kedua analisis dari hasil *dynamic AI* melawan subjek sebagai pemainnya. Berikut adalah penjelasan dari parameter – parameter yang digunakan:

1. Variabel jumlah menang dan kalah (*Win Ratio*)

Variabel jumlah menang dan kalah merupakan variabel yang menunjukkan berapa kali *AI* memenangkan pertandingan dan berapa kali kalah dalam pertandingan. Variabel ini digunakan sebagai ukuran kemampuan *AI* dalam memenangkan pertandingan (*win rate*) melawan musuhnya.

2. Variabel rasio *kill* dan *death*

Variabel *kill* merupakan jumlah seberapa banyak *hero* mengalahkan musuh di dalam satu pertandingan. Sedangkan variabel *death* merupakan jumlah berapa kali *hero* dikalahkan oleh musuh dalam satu pertandingan.

Variabel ini digunakan sebagai ukuran seberapa mampu *AI* berhasil atau gagal mendominasi musuhnya. Variabel ini memiliki sub variabel untuk yang mempengaruhi *AI* memperoleh *kill* dan *death*, diantara lain :

3. Variabel *last hit*

Merupakan jumlah berapa kali suatu *hero* mendapatkan pukulan terakhir terhadap creep lawan. Variabel ini merupakan salah satu faktor yang mempengaruhi pendapatan *gold* dan *experience point*, karena dengan

melakukan last hitting *hero* akan mendapatkan *gold* dan *experience point* tambahan.

4. Variabel *gold per minute*

Variabel ini merujuk pada jumlah *gold* atau uang di dalam *game* ini yang didapatkan dalam waktu satu menit. Variabel ini berguna untuk mengukur pendapatan *AI* dalam satuan menit, seberapa cepat *AI* mengumpulkan *gold*. Semakin cepat *AI* mengumpulkan *gold* maka akan semakin cepat *AI* mendapatkan *item-item* yang diperlukannya.

5. Variabel *xpm (experience point per minute)*

Variabel ini adalah jumlah *experience point* yang didapatkan dalam satu menit. *Experience point* akan meningkatkan *level hero* dan jika *level* naik maka suatu *hero* akan mendapatkan peningkatan status dan *skill* baru.

6. Variabel *Item build*

Variabel *item build* adalah *item -item* apa saja yang telah dibeli oleh *Zeus* dalam pertandingan. *Item* ini juga memiliki peran penting dalam menentukan performa suatu *hero*. Pembelian *item* yang tepat akan meningkatkan performa permainannya.

Data dari hasil *testing* akan dianalisis secara kuantitatif dan dibuat dalam sebuah tabel dan grafik untuk menggambarkan secara jelas bagaimana perkembangan dan performa *AI* dari satu pertandingan ke pertandingan yang lainnya dalam tiap variabel analisis yang telah ditetapkan. Variabel analisis yang digunakan pada pertandingan melawan manusia adalah variabel *win rate* atau

tingkat kemenangan dan dilakukan pengumpulan survey untuk mendapatkan penilaian subjektif dari para subjek yang telah dipilih untuk mengikuti skenario *test* kedua sebagai pendukung pada hasil analisis. Penilaian ini menggunakan skala *Likert* dan menggunakan *form* yang terdiri dari 5 pernyataan yang dan setiap pernyataan memiliki poin 1(sangat tidak setuju), 2(tidak setuju), 3(netral), 4(setuju), 5(sangat setuju). Maksimal poin yang bisa didapatkan dari 1 subjek adalah 25 dan poin minimum adalah 5. Tabel 3.2 menunjukkan contoh skala *Likert* yang digunakan. *Form* ini disebarakan pada para subjek test dengan menggunakan *Google Form*.

**Tabel 3.2. Skala *Likert* yang digunakan**

Subjek #					
#	1	2	3	4	5
Kemampuan <i>last hit AI</i> sudah baik					
Kemampuan <i>AI</i> dalam melakukan <i>harrass</i> sudah baik					
Kemampuan <i>Zeus</i> untuk mendapatkan <i>kill</i> sudah baik					
<i>Zeus</i> konsisten dalam menjaga <i>lane</i> -nya					
<i>AI adaptive Zeus</i> lebih baik daripada <i>AI</i> standar yang disediakan oleh <i>Valve</i>					

Poin-poin dari hasil respon subjek akan dihitung untuk menghasilkan sebuah nilai yang menggambarkan performa *AI* menggunakan rentang nilai seperti berikut ini :

- Sangat buruk (0%-19,99%)
- Buruk (20% -39,99%)
- Netral (40%-59,99%)
- Baik (60%-79,99%)
- Sangat baik (80%-100%)