

ANALISIS PERBANDINGAN KECERDASAN BUATAN PADA COMPUTER PLAYER DALAM MENGAMBIL KEPUTUSAN PADA GAME BATTLE RPG

Musta'inul Abdi¹⁾, Darlis Herumurti²⁾, dan Imam Kuswardayan³⁾

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Kampus ITS, Jl. Raya ITS, Sukolilo – Surabaya 60111, Telp. +62 31 5939214, Fax. +62 31 5913804
e-mail: m.abdi4444@gmail.com¹⁾, darlis@if.its.ac.id²⁾, imam@its.ac.id³⁾

ABSTRAK

Pemanfaatan kecerdasan buatan telah diimplementasikan ke dalam banyak hal, salah satunya adalah game. Secara umum tujuan dibuatnya game adalah untuk membuat pengguna menjadi terhibur dan merasakan kesenangan ketika sedang atau telah bermain. Kecerdasan buatan di dalam game dibutuhkan untuk meningkatkan tantangan di dalam game dan membuat game menjadi lebih dinamis dan terarah. Sehingga akan menciptakan kesenangan bagi pengguna pada saat dan setelah memainkan game.

Beberapa penerapan kecerdasan buatan di dalam game diantaranya adalah dengan menggunakan metode Support Vector Machine (SVM). Dalam beberapa kasus game ada juga yang menggunakan metode Decision tree yang akan mengatur perilaku computer player di dalam permainan. Metode yang lebih sederhana untuk mengatur perilaku computer player yaitu Rulebase. Pada penelitian ini akan dilakukan perbandingan kecerdasan buatan untuk mengatur perilaku computer player di dalam game Role-Playing Game (RPG). Yang dimaksud computer player pada penelitian ini adalah pemain atau karakter yang dijalankan oleh sistem di dalam game.

Tujuan dilakukannya perbandingan tersebut adalah untuk mengetahui metode kecerdasan buatan manakah yang paling baik diterapkan pada game berjenis battle RPG. Metode yang digunakan untuk menguji kecerdasan buatan yang diterapkan pada game battle RPG ini adalah dengan menggunakan skenario pertandingan.

Berdasarkan analisis yang telah dilakukan didapatkan hasil bahwa kecerdasan buatan dengan menggunakan metode SVM memiliki keunggulan dalam faktor jumlah kemenangan. Hal ini dibuktikan dengan persentase kemenangan metode SVM sebesar 72.5%, Decision tree sebesar 50% dan Rulebase sebesar 25%. Berdasarkan data tersebut dapat disimpulkan bahwa pada penelitian ini metode SVM adalah metode pengambilan keputusan yang paling baik dibandingkan dengan metode decision tree dan Rulebase.

Kata Kunci: Decision tree, Kecerdasan Buatan, Rulebase, RPG, SVM.

ABSTRACT

Artificial intelligence has been implemented in many domain researches, and one of them is game. Artificial intelligence in game is needed to make the game more challenging and make it more dynamic and focused. Hence, it will create enjoyment for users at the time of and after playing the game.

SVM, decision tree and rulebase are some of artificial intelligence methods that have been commonly used. In this study, these artificial intelligence methods will be implemented into a computer player within a game and then compared with each other in order to regulate the behavior of computer player in the Role-Playing Game (RPG). The computer player is a player or character which is run by the game system.

The purpose of this study is finding which artificial intelligence technique is the best to be applied to battle RPG type games. Moreover, artificial intelligence which is applied to this battle-type RPG game is compared by utilizing a match scenario. Regarding to the analysis that has been done, the result shows artificial intelligence using SVM method has an advantage in the number of winning factor. This is evidenced by the percentage of winning methods of SVM, Decision tree and Rulebase, at about 72.5%, 50% and 25% respectively. Therefore, it can be concluded that SVM method is the best decision-making method compared to decision tree and rulebase method when applied as computer player on battle-type RPG games.

Keywords: Artificial Intelligence Decision tree, Rulebase, RPG, SVM.

I. PENDAHULUAN

Perkembangan teknologi komputer yang sangat pesat telah mendorong pengembang game untuk mencari inovasi-inovasi baru di dalam produk game mereka, sehingga akan menarik minat pengguna untuk memainkan game tersebut. Tujuan dari pembuatan game adalah sebagai sarana hiburan bagi penggunanya, untuk itu dalam pembuatan sebuah game diperlukan perancangan agar tujuan tersebut dapat tercapai.

Salah satu jenis game yang populer saat ini adalah game berjenis RPG. Role Playing Game (RPG) adalah sebuah permainan yang memungkinkan pemain untuk terlibat secara langsung di dalam perencanaan strategi dan alur di dalam permainan untuk mencapai suatu tujuan tertentu [1]. Kemajuan teknologi saat ini memungkinkan penggunaan komputer untuk memvisualisasikan permainan RPG sehingga memberikan pengalaman bermain dan

realisme lebih kepada pemain [2]. Karakteristik dan fitur menarik dari *game* berjenis RPG adalah kemungkinan untuk dapat mengembangkan peran karakter yang dipilih yang perkembangan dan keterampilannya melibatkan pengguna secara langsung di dalam permainan, fitur inilah yang membuat *game* RPG menjadi lebih dramatis dan dinamis [3]. Berdasarkan uraian diatas, dapat diambil kesimpulan bahwa *game* berjenis RPG adalah *game* yang memiliki banyak sekali kemungkinan. Untuk itu dalam pembuatan AI (*computer player*) diperlukan sebuah metode atau algoritma yang baik sehingga AI dapat berjalan dengan optimal.

Salah satu elemen yang penting dalam sebuah *game* ialah adanya kecerdasan buatan atau AI (*artificial intelligence*). AI merupakan salah satu unsur yang diperlukan dalam pembentukan *game* yakni untuk membuat permainan lebih dinamis dan terarah [4]. *Game* AI merupakan *game* yang mengubah metode, proses, dan algoritma pada kecerdasan tersebut yang akan diaplikasikan ke dalam pembuatan dan pengembangan *game* [5]. Terdapat tiga panorama dalam *game* AI yakni perspektif metode (komputer), perspektif pengguna (manusia) dan perspektif interaksi pemain. Kecerdasan buatan sering digunakan pada kebanyakan *game* saling memiliki ketergantungan interaksi terhadap pemain. Sehingga AI memiliki peran yang penting untuk meningkatkan ketertarikan pengguna dalam bermain *game* [6]. Dari uraian diatas menunjukkan bahwa penggunaan kecerdasan buatan di dalam *game* dibutuhkan untuk meningkatkan tantangan di dalam *game*. Membuat *game* menjadi lebih dinamis dan dapat menunjang realitas *game*, sehingga akan menciptakan kesenangan bagi pengguna pada saat memainkan *game* dan ketertarikan pengguna untuk memainkan *game* juga akan meningkat.

Dalam kasus *game* RPG yang karakteristiknya memiliki banyak kemungkinan, dibutuhkan metode yang mampu mengambil keputusan yang baik. Beberapa penerapan kecerdasan buatan di dalam *game* diantaranya adalah dengan menggunakan metode Support Vector Machine (SVM) [7], [8]. Dalam penelitian tersebut dijelaskan bahwa metode SVM dapat meningkatkan kecepatan dalam mengambil keputusan. Tetapi kekurangan dari metode ini adalah metode SVM ini sangat bergantung pada banyaknya jumlah data *training*. SVM merupakan sistem logika yang menggunakan data *training* untuk menentukan keputusan yang akan di ambil berikutnya. Data *training* tersebut digunakan sebagai data masukan untuk mencari vektor (*hyperplane*) terbaik sebagai pemisah dua buah kelas, dimana kelas dapat diartikan sebagai keputusan yang akan di ambil. Pada beberapa penelitian tentang *game* lainnya ada juga yang menggunakan metode *Decision tree*, dimana metode tersebut digunakan untuk mengatur perilaku *computer player* di dalam permainan [9]–[11]. Metode *decision tree* menggunakan data *training* untuk membangun pohon keputusan. Pohon keputusan tersebut digunakan sebagai dasar dalam pengambilan keputusan selanjutnya. Metode lain untuk mengatur perilaku *computer player* yaitu menggunakan *rulebase* [12]. Berbeda dengan SVM dan *decision tree*, *rulebase* merupakan sekumpulan aturan/kondisi yang digunakan untuk menentukan aksi. *Rulebase* terdiri dari dua bagian penting yaitu bagian kondisi dan bagian aksi.

Berdasarkan uraian diatas, disebutkan bahwa tujuan utama dari pengembangan *game* adalah sebagai sarana hiburan. Salah satu jenis *game* yang cukup populer adalah *game* berjenis RPG. *Game* RPG memiliki karakteristik banyaknya kemungkinan yang dapat terjadi dalam mencapai tujuan permainan. Salah satu faktor yang menentukan tingkat *entertainment* dalam sebuah *game* adalah adanya AI. AI dalam sebuah *game* akan membuat permainan menjadi lebih dinamis, sehingga akan meningkatkan minat pengguna dalam bermain. Dikarenakan karakteristik dalam RPG yang memiliki banyak kemungkinan, dibutuhkan algoritma AI yang dapat mengambil keputusan dengan baik. Oleh sebab itu pada penelitian ini akan dilakukan perbandingan kecerdasan buatan yang mengatur perilaku *computer player* dalam mengambil keputusan pada *game* RPG, kecerdasan buatan yang akan dikembangkan adalah SVM, *decision tree*, dan *rulebase*. Diharapkan dengan penelitian ini dapat mengetahui metode manakah yang paling sesuai untuk mengatur perilaku *computer player* pada *game* RPG.

II. KAJIAN PUSTAKA

A. *Game*

Pada dasarnya definisi *game* adalah sebuah sistem kontrol yang bebas di mana di dalamnya terdapat suatu pertentangan, dan dibatasi oleh prosedur untuk menghasilkan sebuah tujuan [13]. *Game* adalah sebuah sistem di mana pemain terlibat dalam pertempuran buatan, ditentukan oleh aturan, yang menghasilkan hasil yang terukur [14]. *Game* adalah sebuah kegiatan dengan aturan. *Game* adalah bentuk permainan yang melibatkan sebuah interaksi, baik dengan pemain lain, dengan sistem permainan itu sendiri, ataupun dengan nasib dan keberuntungan [15]. *Game* dapat didefinisikan sebagai suatu bentuk permainan yang memiliki tujuan dan struktur. *Game* menyediakan aktivitas yang menarik minat pemain sekaligus sebagai media hiburan [16]. Layanan komputer yang didukung dengan *game* bertujuan untuk memotivasi dan mendukung aktivitas pelatihan pemain [17]. Berdasarkan pernyataan-pernyataan tersebut dapat disimpulkan bahwa *game* adalah sebuah kegiatan atau aktivitas yang menarik yang didalamnya terdapat aturan untuk mencapai suatu tujuan.

B. RPG

Role Playing Game (RPG) merupakan sebuah *game* yang memungkinkan pemain untuk dapat menentukan perencanaan strategi dan alur di dalam *game* untuk mencapai suatu tujuan tertentu [1]. Karakteristik dari *game* berjenis RPG adalah kemungkinan untuk dapat mengembangkan peran karakter yang dipilih yang perkembangan dan keterampilannya melibatkan pengguna secara langsung di dalam permainan. Fitur inilah yang membuat *game* RPG menjadi lebih dinamis [3]. Kemajuan teknologi saat ini memungkinkan penggunaan komputer untuk memvisualisasikan permainan RPG sehingga memberikan pengalaman bermain dan realisme lebih kepada pemain [2].

C. SVM

Support Vector Machine (SVM) adalah sistem klasifikasi yang menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (*feature space*), dilatih dengan algoritma pembelajaran yang didasarkan pada teori optimasi dengan mengimplementasikan *learning* yang berasal dari teori *statistic* [18]. Teori yang mendasari SVM sendiri sudah berkembang sejak 1960-an, tetapi baru diperkenalkan oleh Vapnik, Boser dan Guyon pada tahun 1992 dan sejak itu SVM berkembang dengan pesat [19]. SVM merupakan salah satu teknik yang relatif baru dibandingkan dengan teknik lain, tetapi memiliki performansi yang lebih baik di berbagai bidang aplikasi seperti bioinformatics, pengenalan tulisan tangan, klasifikasi teks dan lain sebagainya [20].

Di dalam *game*, metode SVM digunakan untuk mengontrol Non-Character Player (NPC) [8], [21]. Dimana didalam penelitian tersebut dijelaskan bahwa metode SVM memiliki properti yang sangat cocok untuk mengontrol NPC di dalam *game*. Dalam penelitian lain yang telah dilakukan oleh [7], mereka merancang sebuah *game* yang dinamakan dengan *Game of Go*. Di dalam *game* tersebut metode SVM digunakan untuk menentukan keputusan yang akan di ambil selanjutnya dalam menjalankan permainan. Pada keputusan yang lebih dari dua atau *multiclass* penggunaan SVM juga telah diterapkan pada penelitian yang di lakukan oleh [8], [21], pengambilan keputusannya yaitu *explore*, *attack*, dan *run away*. Pada penelitian lain yang dilakukan oleh [22] untuk mengatasi masalah *multiclass* dengan metode SVM, dimana metode SVM yang di bangun menggunakan SVM *one against all*. Dalam metode tersebut peneliti membangun k buah model SVM (k adalah jumlah kelas). Setiap model klasifikasi ke-i dilatih dengan menggunakan keseluruhan data, untuk mencari solusi permasalahan.

D. Decision tree

Metode *decision tree* merepresentasikan struktur percabangan pohon dalam membuat strategi keputusan [9], [11]. Deskripsi lain dari metode *decision tree* adalah sebuah metode yang merepresentasikan sebuah hirarki dari beberapa keputusan, pohonnya berawal dengan keputusan yang telah dibuat dan cabang berikutnya berasal dari keputusan yang telah ada [23]. Pada penelitian yang telah dilakukan oleh [10], peneliti menggunakan metode *decision tree* untuk membangun sebuah *game* robot sepakbola. Dalam penelitian tersebut setiap robot diberi kecerdasan buatan berbasis metode *decision tree* sehingga dapat mengatur perilaku setiap agen atau robot di dalam permainan. Penggunaan metode *decision tree* juga diterapkan pada penelitian yang telah dilakukan oleh [23], dalam penelitian tersebut metode *decision tree* digunakan untuk membangun sebuah strategi tak terkalahkan dalam permainan Tic-Tac-Toe.

E. Rulebase

Ada banyak pengertian dari rulebase berdasarkan para peneliti, diantaranya yaitu *Rulebase* merupakan bahasa script yang dirancang untuk memungkinkan ekspresi dari aturan yang terdiri dari pernyataan kondisional optional dan satu tindakan, dimana pernyataan kondisional tersebut terdiri dari satu atau lebih kondisi yang dikombinasikan dengan logika AND dan OR [24], [25]. Pengertian lainnya yaitu *Rulebase* adalah metode yang terdiri dari sekumpulan rule yang digunakan untuk menentukan aksi. *Rulebase* terdiri dari dua bagian penting yaitu bagian kondisi dan bagian aksi [12].

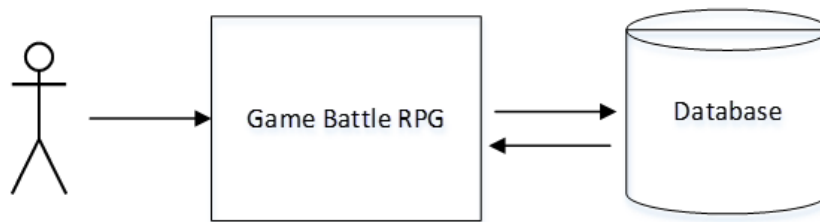
III. METODE PENELITIAN

A. Desain dan Implementasi

Tahap ini adalah tahap menentukan racangan desain dari *game battle* RPG dan bagaimana cara melakukan implementasinya. Desain dan implementasi terdiri dari beberapa tahap, diantaranya:

1) Analisis kebutuhan

Analisis kebutuhan merupakan tahapan awal dari penelitian ini. Tujuannya adalah untuk mengumpulkan informasi kebutuhan perangkat keras, perangkat lunak, serta kebutuhan konten dan sistem di dalam *game*. Kebutuhan yang akan dianalisis diperoleh dengan cara melakukan observasi terhadap pihak-pihak yang



Gambar 1. Proses pembuatan dataset

berhubungan dengan penelitian, dan juga dengan mencari referensi-referensi mengenai *game* yang sejenis.

2) *Desain sistem*

Desain sistem adalah proses dimana kebutuhan yang telah didapatkan akan dibuat menjadi sebuah desain sistem. *Game* yang akan dibuat akan diterapkan kecerdasan buatan yang terdiri dari SVM, decision tree dan rulebase.

3) *Implementasi*

Tahap implementasi adalah tahap dimana *game* akan diimplementasikan ke dalam bahasa pemrograman sesuai dengan rancangan desain sistem yang telah dibuat sehingga semua fungsi dapat dijalankan.

B. *Pengujian*

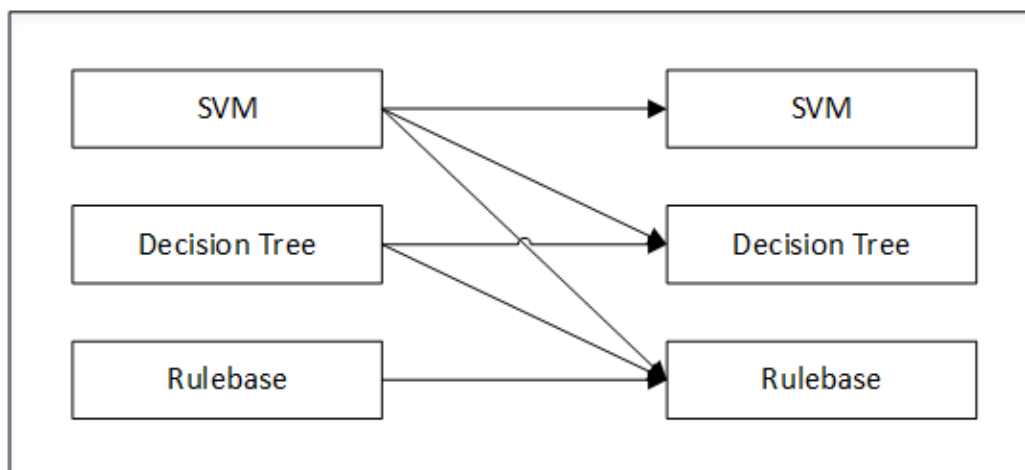
Berikut ini adalah beberapa tahapan didalam pengujian:

1) *Pembuatan dataset*

Diperlukan adanya dataset untuk dipergunakan untuk klasifikasi sebagai data *training*. Pada penelitian ini dataset digunakan untuk *training* metode SVM dan *decision tree*. Dataset dibangun dari percobaan pengguna melawan metode kecerdasan buatan *rulebase*. Dimana pengguna yang dipilih merupakan pakar dalam *game*. Pada setiap proses pengumpulan data, akan disimpan perlakuan pengguna dalam menghadapi musuh didalam *game battle RPG*. Kemudian akan dipilih data-data yang memiliki skenario kemenangan didalam setiap pertandingan. Sehingga akan menghindari skenario kekalahan dalam pertandingan. Gambaran proses pembuatan dataset dapat dilihat pada Gambar 1.

2) *Skenario pengujian*

Berikut ini adalah skenario pengujian yang akan dilakukan: skenario 1: membandingkan antara metode kecerdasan buatan SVM dengan *decision tree*. Skenario 2: membandingkan antara metode kecerdasan buatan SVM dengan *rulebase*. Skenario 3: membandingkan antara metode kecerdasan buatan *decision tree* dengan *rulebase*. Skenario 4: membandingkan antara metode kecerdasan buatan SVM dengan SVM. Skenario 5: membandingkan antara metode kecerdasan buatan *decision tree* dengan *decision tree*. Skenario 6: membandingkan antara metode kecerdasan buatan *rulebase* dengan *rulebase*. Skenario pengujian *game battle RPG* selengkapnya dapat dilihat pada Gambar 2. Setiap skenario akan di ujikan sebanyak 100 kali. Dalam setiap pengujian skenario akan dihitung presentase kemenangan *computer player*. Untuk menentukan giliran awal digunakan metode *random*.



Gambar 2. Skenario pengujian

3) *Analisis hasil uji coba*

Langkah yang dilakukan dalam melakukan analisis hasil pengujian antar metode adalah mencari persentasi kemenangan setiap metode kecerdasan buatan dengan menggunakan persamaan 1,

$$\text{persentase kemenangan} = \frac{\text{jumlah kemenangan}}{\text{jumlah semua pertandingan}} \times 100\% \quad (1)$$

sehingga persentasi kemenangan dapat diskalakan dari 0 hingga 100, dimana jika *computer player* mendapat persentasi mendekati 0 maka dapat diartikan bahwa *computer player* tersebut banyak mengalami kekalahan dalam setiap pertandingan. Sedangkan jika *computer player* mendapat persentasi mendekati 100 maka dapat diartikan bahwa *computer player* tersebut mendapat banyak kemenangan di setiap pertandingannya. Diasumsikan bahwa metode kecerdasan buatan yang memiliki persentase kemenangan tertinggi adalah metode kecerdasan yang paling baik.

IV. HASIL DAN PEMBAHASAN

A. *Hasil Desain dan Implementasi*

Berikut ini adalah hasil dari desain dan implementasi yang telah dilakukan, diantaranya:

1) *Hasil analisis kebutuhan*

Analisi kebutuhan dilakukan dengan dua cara yaitu melakukan observasi dan mencari referensi yang relevan dengan *game battle RPG* yang akan dibuat. Observasi dilakukan terhadap pihak yang berhubungan dengan penelitian yaitu developer, desainer *game* dan pakar kecerdasan buatan. Cara selanjutnya dalam tahapan pengumpulan kebutuhan yaitu dengan mencari referensi yang relevan terhadap *game battle RPG* yang akan dibuat. Berdasarkan analisis kebutuhan yang telah dilakukan, didapatkan sejumlah hasil sebagai berikut:

- a. Desain dan tampilan antar muka pada *game battle RPG* disesuaikan dengan jenis *game battle RPG*.
- b. Dua buah karakter yang akan diujikan harus memiliki atribut yang sama.
- c. Pengambilan keputusan *computer player* pada *game battle RPG* harus dapat dilakukan dengan cepat dan akurat.
- d. Setiap metode kecerdasan buatan memungkinkan untuk dapat mengambil keputusan pada setiap kemungkinan yang akan terjadi.
- e. Perbandingan akan dilakukan terhadap tiga buah metode kecerdasan buatan.

2) *Hasil desain sistem dan implementasi*

Desain sistem adalah tahap dimana hasil dari analisis kebutuhan yang telah diperoleh kemudian dijadikan sebagai acuan dari desain *game battle RPG* yang akan dikembangkan. Perangkat lunak yang digunakan sebagai alat bantu dalam mengimplementasikan sistem adalah sebagai berikut:

- a. *Adobe Flash CS6* sebagai alat bantu penulisan kode program.
- b. *Netbeans 8* sebagai alat bantu penulisan kode program, pembuatan kecerdasan buatan.
- c. *XAMPP* sebagai alat bantu pembuatan database.
- d. *MDM Zinc* sebagai alat bantu koneksi aplikasi dengan database.
- e. *Audacity 2.1.3* sebagai alat bantu dalam pengolahan audio.
- f. *PhotoScape 3.7* sebagai alat bantu dalam pengolahan gambar bertipe bitmap.
- g. *Adobe Illustrator CS6* sebagai alat bantu dalam pengolahan gambar bertipe vector.

Perangkat keras yang digunakan sebagai lingkungan pengembangan dan pengujian *game battle RPG* yaitu seperangkat komputer dengan spesifikasi sistem operasi Microsoft Windows 10 64 bit, processor Core i3-6100 3.7 Ghz, RAM 2666 Mhz 8 GB, *hardisk* 1 TB, dan VGA NVIDIA GTX 1060 GB.

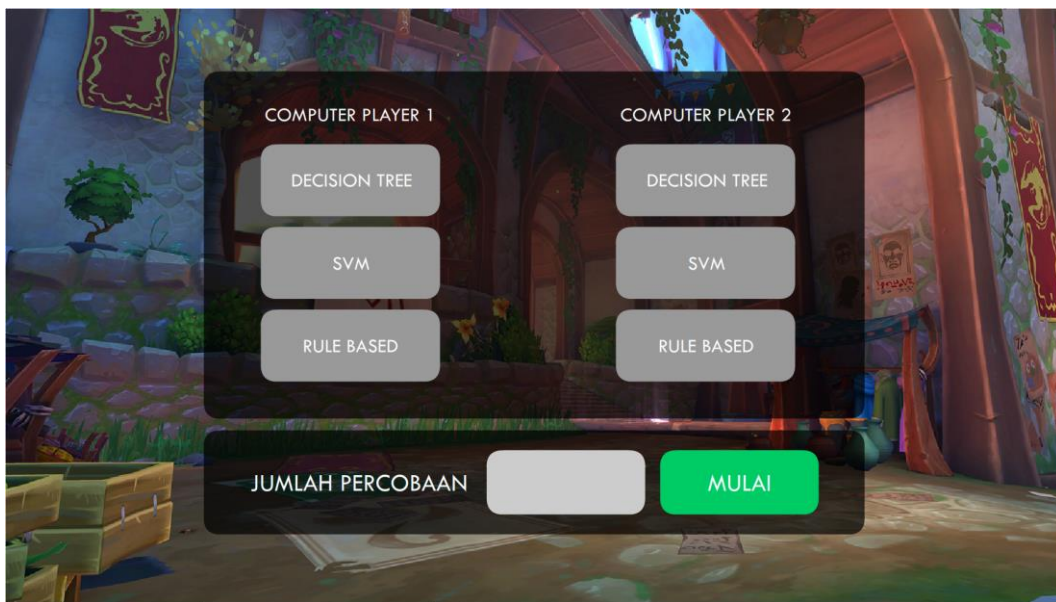
Berdasarkan hasil dari kebutuhan yang diperoleh maka didapatkan hasil dari pembuatan desain sistem *pada game battle RPG* sebagai berikut:

- a. Konten utama pada *game* adalah berupa permainan RPG. Desain permainan RPG yang digunakan adalah tipe *battle* dimana terdapat dua buah *computer player* yang saling menyerang.
- b. Setiap *computer player* memiliki atribut yang terdiri dari *health*, energi, *damage* dan *armor*. *Health* adalah nyawa dari *computer player*, jika *health* yang dimiliki habis maka *computer player* akan mati dan kalah. Energi adalah atribut yang digunakan memnuhi *cost* setiap keputusan dari *computer player*. *Armor* adalah atribut yang digunakan untuk mengurangi serangan musuh. *Damage* adalah atribut yang digunakan untuk menambah besaran serangan terhadap musuh.
- c. Terdapat empat keputusan *ultimate*, *attack*, *defense*, *heal*. *Ultimate* adalah serangan ke musuh dengan besaran serangan lebih besar dari pada serangan *attack*. *Attack* adalah serangan ke musuh dengan besaran

tertentu. *Defense* adalah keputusan untuk bertahan dari serangan musuh dengan mereduksi serangan musuh sebesar *armor* yang dimiliki. *Heal* adalah keputusan untuk menambah jumlah energi dengan besaran tertentu.

- d. Setiap *computer player* memiliki kecerdasan buatan yang berbeda untuk mengambil keputusan. Dalam hal ini kecerdasan buatan SVM, *decision tree* dan *Rulebase*.
- e. Dataset digunakan sebagai data *training* kecerdasan buatan yang akan dipakai oleh kecerdasan buatan untuk mengambil keputusan.
- f. Terdapat indikator untuk mengetahui status *computer player* didalam permainan.
- g. Konten RPG didalam *game* diterapkan dengan adanya pengembangan status atribut pada *computer player*.

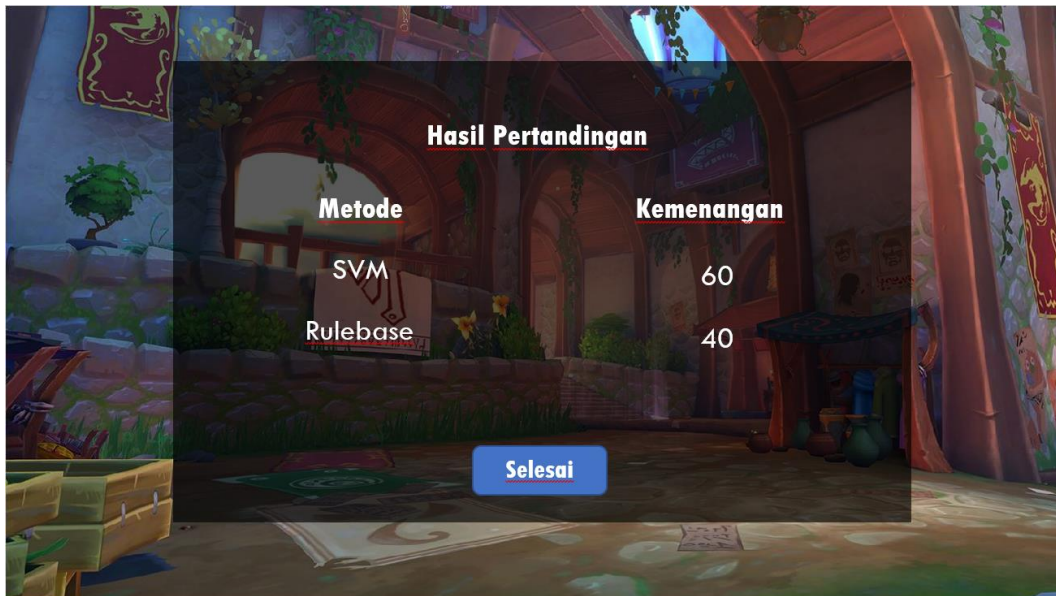
Hasil pembuatan antar muka dari *game battle* RPG dapat dilihat pada Gambar 3 yang merupakan tampilan menu utama, Gambar 4 merupakan tampilan *game battle* RPG dan Gambar 5 adalah tampilan hasil pertandingan.



Gambar 3. Tampilan menu utama



Gambar 4. Tampilan *Game* Battle RPG



Gambar 5. Tampilan hasil pertandingan

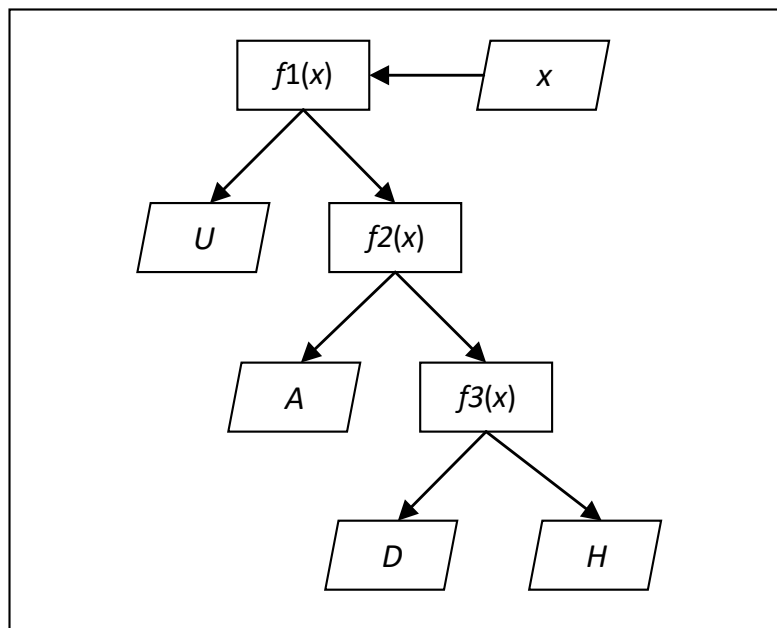
Hasil pembuatan desain sistem adalah sebagai berikut:

a. SVM

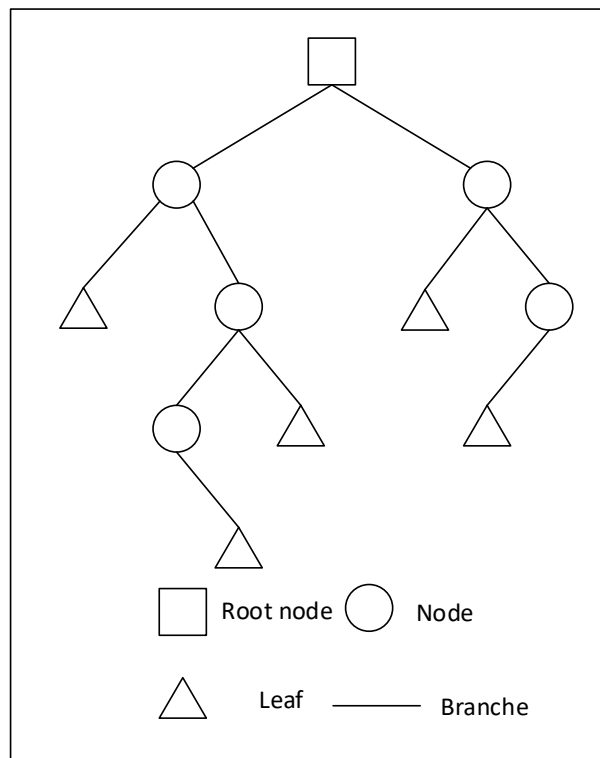
Pada penelitian ini metode SVM yang dipakai adalah SVM *one-against-all*, dimana metode SVM yang dipakai menggunakan fungsi kernel polynomial. Data inputan yang digunakan adalah nilai dari *health*, energi, *enemy health*, *damage* dan *armor*. Kemudian akan dibagi menjadi empat kelas yaitu *ultimate attack*, *attack*, *defense*, *heal*. Hasil pembuatan sistem kecerdasan buatan dengan metode SVM selengkapnya dapat dilihat pada Gambar 6.

b. Decision tree

Pada penelitian ini metode kecerdasan buatan *decision tree* dibangun dari dataset yang telah dibuat sebelumnya. Data inputan yang digunakan adalah nilai dari *health*, energi, *enemy health*, *damage* dan *armor*. Kemudian akan dibagi menjadi empat kelas yaitu *ultimate attack*, *attack*, *defence*, *heal*. Hasil pembuatan sistem kecerdasan buatan dengan metode *decision tree* selengkapnya dapat dilihat pada Gambar 7.



Gambar 6. Metode SVM



Gambar 7. Metode Decision tree

```
function Rulebase()
{
  if (Health <= 20){
    if (Energi >= CostUltimate) Ultimate();
    else if (Energi >= CostAttack && EnemyHealth <= 20) Attack();
    else if (Energi >= CostDefense) Defense();
    else Heal();
  }
  else if (State == "Ultimate" || Energi >= CostUltimate) Ultimate();
  else if (State == "Attack" || Energi >= CostAttack) Attack();
  else if (State == "Defense" || Energi >= CostDefense) Defense();
  else Heal();
}
```

Gambar 8. Pseudo-code Metode Rulebase

c. Rulebase

Hasil pembuatan sistem dengan kecerdasan buatan *rulebase*, metode ini dibangun dari kemungkinan-kemungkinan yang dapat terjadi didalam permainan, diantaranya adalah kemungkinan *computer player* untuk melakukan *attack*, *ultimate*, *defense* dan *heal*. Berbeda dengan dua kecerdasan buatan sebelumnya yang menggunakan dataset sebagai data *training*, metode ini hanya mengacu pada kemungkinan-kemungkinan tersebut. Hasil pembuatan sistem kecerdasan buatan dengan metode *rulebase* selengkapnya dapat dilihat pada Gambar 8.

B. Hasil Pengujian

Berikut ini adalah hasil dari pengujian *game battle RPG* yang telah dilakukan.

1) Hasil pembuatan dataset

Dataset yang digunakan di dalam *game* ini ada 1, dataset tersebut digunakan untuk metode kecerdasan buatan SVM dan metode *decision tree*. Atribut-atribut dari dataset ini adalah *health*, energi, *enemy health*, *damage* dan *armor*. Dataset yang digunakan untuk setiap metode SVM dan *decision tree* berisi data sebanyak 156 data. Nilai maksimum dari atribut *health* adalah sebesar 300, untuk atribut energi memiliki nilai maksimum sebesar 100 dan nilai minimum sebesar 0, untuk atribut *enemy health* memiliki nilai maksimum sebesar 300, untuk atribut *damage* memiliki nilai minimum sebesar 0, dan atribut *armor* memiliki nilai minimum sebesar 0. Contoh dataset

dapat dilihat pada Tabel 1.

TABEL I
CONTOH DATASET

No.	Health	Energi	Enemy Health	Damage	Armor	Decision
1	300	100	300	0	0	U
2	260	60	260	0	2	U
3	220	20	220	0	4	H
4	200	40	220	0	4	U
5	200	0	180	0	6	H
6	200	20	180	0	6	H
7	180	40	180	0	6	A
8	160	20	160	2	6	A
9	160	0	138	2	8	H
10	140	20	138	2	8	A
11	140	0	116	2	10	H
12	118	20	116	2	10	H
13	118	40	116	2	10	U
14	118	0	74	4	10	H
15	118	20	74	4	10	H
16	96	40	74	4	10	U
17	96	0	30	6	10	H
18	74	20	30	6	10	A
19	74	20	30	6	10	A
20	50	0	4	8	10	H
21	50	20	4	8	10	A

TABEL II
HASIL PENGUJIAN SKENARIO 1 HINGGA 3

Metode	Jumlah Kemenangan		
	SVM	Decision tree	Rulebase
SVM	-	42	13
Decision tree	58	-	42
Rulebase	87	58	-

2) Hasil skenario pengujian

Pengujian pertama yaitu metode SVM melawan metode *decision tree*. Dalam 100 kali pertandingan, metode SVM mendapatkan jumlah kemenangan sebanyak 58 kali. Sedangkan untuk metode *decision tree* mendapatkan 42 kali kemenangan. Selanjutnya pengujian kedua yaitu metode SVM melawan metode *rulebase*. Dalam 100 kali pertandingan, metode SVM mendapatkan jumlah kemenangan sebanyak 87. Sedangkan untuk metode *rulebase* mendapatkan 13 kali kemenangan. Pengujian ketiga yaitu metode *decision tree* melawan metode *rulebase*. Dalam 100 kali pertandingan, metode *decision tree* mendapatkan jumlah kemenangan sebanyak 58 kali. Sedangkan untuk metode *rulebase* mendapatkan 42 kali kemenangan. Hasil pengujian skenario 1 hingga 3 dapat dilihat pada Tabel 2.

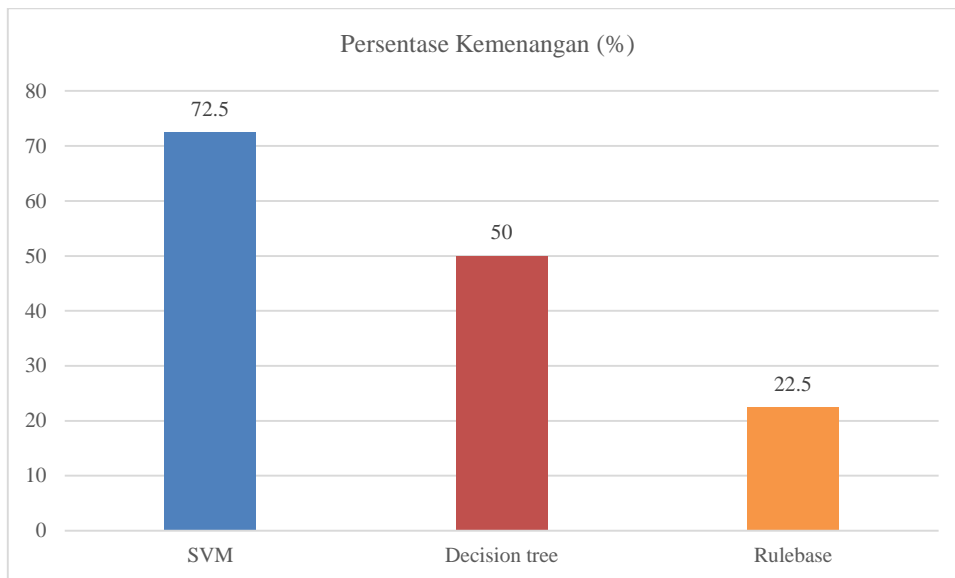
Pengujian keempat yaitu metode SVM melawan metode SVM. Dalam 100 kali pertandingan, metode SVM *computer player 1* mendapatkan jumlah kemenangan sebanyak 48 kali. Sedangkan untuk SVM *computer player 2* mendapatkan 52 kali kemenangan. Pengujian kelima yaitu metode *decision tree* melawan metode *decision tree*. Dalam 100 kali pertandingan, metode *decision tree computer player 1* mendapatkan jumlah kemenangan sebanyak 50 kali. Sedangkan untuk *decision tree computer player 2* mendapatkan 50 kali kemenangan. Pengujian keenam yaitu metode *rulebase* melawan metode *rulebase*. Dalam 100 kali pertandingan, metode *rulebase computer player 1* mendapatkan jumlah kemenangan sebanyak 49 kali. Sedangkan untuk *rulebase computer player 2* mendapatkan 51 kali kemenangan. Hasil pengujian skenario 4 hingga 6 dapat dilihat pada Tabel 3.

TABEL III
HASIL PENGUJIAN METODE YANG SAMA

Metode	Jumlah Kemenangan	
	Computer player 1	Computer player 2
SVM	48	52
Decision tree	50	50
Rulebase	49	51

TABEL IV
HASIL ANALISIS PERBANDINGAN METODE YANG BERBEDA

Metode	Persentase Kemenangan (%)		
	SVM	Decision tree	Rulebase
SVM	-	42	13
Decision tree	58	-	42
Rulebase	87	58	-
Total	145	100	55
Rata-rata	72.5	50	22.5



Gambar 9. Perbandingan metode yang berbeda

3) Hasil analisis uji coba

Berikut ini adalah hasil analisis uji coba perbandingan metode kecerdasan buatan yang berbeda dan analisis uji coba perbandingan metode yang sama.

a. Analisis perbandingan metode yang berbeda

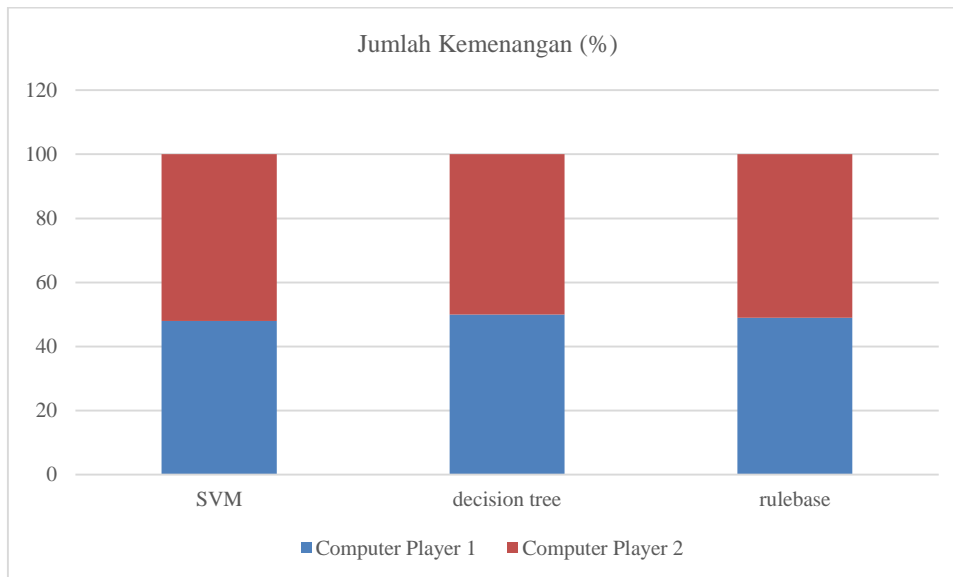
Proses analisis dari hasil pengujian yaitu dengan mencari persentase rata-rata kemenangan setiap metode kecerdasan buatan. Berdasarkan hasil pengujian yang telah dilakukan, dapat diketahui bahwa persentase kemenangan kecerdasan buatan dengan menggunakan metode SVM memiliki persentase kemenangan tertinggi yaitu sebesar 72.5% dibandingkan dengan metode *decision tree* yaitu sebesar 50% dan metode *rulebase* yaitu sebesar 22.5%. dapat disimpulkan bahwa kecerdasan buatan dengan menggunakan metode SVM memiliki keunggulan dalam faktor jumlah kemenangan. Sehingga dapat disimpulkan bahwa metode SVM adalah metode pengambilan keputusan yang paling baik dari pada metode *decision tree* dan *rulebase* pada *game* berjenis battle RPG. Hasil analisis perbandingan metode yang berbeda selengkapnya dapat dilihat pada Tabel 4 dan Gambar 9.

b. Analisis perbandingan metode yang sama

Langkah yang dilakukan dalam melakukan analisis hasil pengujian dengan metode kecerdasan yang sama akan di lihat pada selisih persentase kemenangan pada *computer player 1* dan *computer player 2*. Diasumsikan bahwa semakin kecil selisih persentase faktor-faktor antara kedua *computer player* tersebut maka semakin seimbang kedua metode kecerdasan buatan tersebut. Berdasarkan hasil pengujian yang telah dilakukan, dapat diketahui bahwa selisih persentase kemenangan kecerdasan buatan dengan menggunakan metode SVM dan SVM hanya sebesar 4%, metode *decision tree* dan *decision tree* sebesar 0%, metode *rulebase* dan *rulebase* hanya sebesar 2%. Dapat disimpulkan bahwa untuk kedua *computer player* yang memiliki metode kecerdasan yang sama, bila dipertandingkan maka akan mendapatkan kemiripan hasil jumlah kemenangan. Hal tersebut dikarenakan didalam setiap kondisi pada saat pertandingan akan memiliki kecenderungan kesamaan keputusan yang dihasilkan. Hasil analisis perbandingan metode yang sama selengkapnya dapat dilihat pada Tabel 5 dan Gambar 10.

TABEL V
HASIL ANALISIS PERBANDINGAN METODE YANG SAMA

Metode	Jumlah Kemenangan (%)		Selisih
	Computer player 1	Computer player 2	
SVM	48	52	4
Decision tree	50	50	0
Rulebase	49	51	2



Gambar 10. Perbandingan Metode yang Sama

V. KESIMPULAN DAN SARAN

Pada Bab ini dijelaskan kesimpulan akhir yang didapatkan dari penelitian yang telah dilakukan dan juga dipaparkan saran-saran yang bersifat membangun untuk penelitian selanjutnya di masa yang akan datang.

A. Kesimpulan

Penelitian ini terbagi kedalam beberapa tahapan yaitu: analisis kebutuhan, desain sistem, implementasi dan pengujian. Tujuan dari penerapan kecerdasan buatan pada *game* battle RPG dalam penelitian ini adalah untuk menganalisis tingkat kecerdasan buatan *computer player* dalam pengambilan keputusan menggunakan metode SVM, decision tree dan rulebase untuk mengetahui kecerdasan mana yang paling baik diantara ketiga metode tersebut.

Hasil percobaan untuk metode yang berbeda menunjukkan bahwa metode SVM mendapatkan rata-rata kemenangan sebesar 72.5%, metode decision tree sebesar 50% dan metode rulebase sebesar 22.5%. Dari hasil percobaan, diperoleh kesimpulan bahwa dalam penelitian ini metode SVM adalah metode pengambilan keputusan yang paling baik dibandingkan metode decision tree dan rulebase pada *game* berjenis battle RPG.

Hasil percobaan untuk metode yang sama menunjukkan bahwa selisih kemenangan metode SVM sebesar 4%, decision tree sebesar 0% dan rulebase sebesar 2%. Dari hasil percobaan tersebut dapat disimpulkan bahwa pertandingan dua buah *computer player* dengan metode kecerdasan yang sama, akan mendapatkan hasil jumlah kemenangan yang hampir sama.

B. Saran

Kemajuan teknologi saat ini mendorong inovasi-inovasi baru dalam bidang *game*. Seiring dengan perkembangan tersebut tingkat kompleksitas sistem semakin tinggi, oleh karena itu dibutuhkan penggunaan teknologi kecerdasan buatan *game* terbaru. Memperluas jenis *game* yang tidak hanya terbatas pada *game* berjenis battle RPG, diharapkan nantinya akan diketahui apakah penggunaan metode SVM, decision tree dan rulebase dapat digunakan untuk jenis *game* yang lain.

DAFTAR PUSTAKA

- [1] S. Bedoya-rodriguez, C. Gomez-urbano, A. Uribe-quevedo, and C. Quintero, "Augmented Reality RPG Card-based Game," *Games Media Entertain. (GEM)*, 2014 IEEE, pp. 3–6, 2014.
- [2] W. K. W. J. Patrick Williams, Sean Q. Hendricks, *Gaming As Culture: Essays on Reality, Identity And Experience in Fantasy Games*. McFarland & Company, Inc. Publishers, 2006.
- [3] M. Childress and R. Braswell, "Using Massively Multiplayer Online Role-Playing Games for Online Learning," *Distance Educ.*, vol. 27, no. 2, pp.

- 187–196, 2006.
- [4] M. Urh, G. Vukovic, E. Jereb, and R. Pintar, “The Model for Introduction of Gamification into E-learning in Higher Education,” *Procedia - Soc. Behav. Sci.*, vol. 197, no. February, pp. 388–397, 2015.
- [5] G. N. Yannakakis and J. Togelius, “A Panorama of Artificial and Computational Intelligence in Games,” no. c, 2014.
- [6] K. McGehee and A. T. Abraham, “Real-time team-mate AI in Games Categories and Subject Descriptors,” *ACM*, pp. 124–131, 2010.
- [7] L. Ralaivola, L. Wu, and P. Baldi, “SVM and Pattern-Enriched Common Fate Graphs for the Game of Go,” *Eur. Symp. Artif. Neural Networks Bruges*, vol. 19, pp. 27–29, 2005.
- [8] P. Melendez, *Decision-Making in game characters using Support Vector Machines*, no. Computer Games Technology. University of Abertay Dundee, 2010.
- [9] J. R. Ouinlan, “Decision Trees and Decisionmaking,” *IEEE Trans. Syst. Man Cybern.*, vol. 20, pp. 339–346, 1990.
- [10] P. Stone and M. Veloso, “Using Decision Tree Confidence Factors for Multiagent Control,” *Proc. Second Int. Conf. Auton. agents*, pp. 86–91, 1997.
- [11] A. Narayek, “AI in Computer Games,” *Queue - Game Dev.*, vol. 1, no. 10, p. 58, 2004.
- [12] S. Rostianingsih, G. S. Budhi, and H. K. Wijaya, “Game Simulasi Finite State Machine untuk Pertanian dan Peternakan,” *Konf. Nas. Sist. Informasi, Univ. Kristen Petra*, pp. 2–7, 2013.
- [13] B. S.-S. E.M. Avedon, *The Study of Games*. New York: John Wiley & Sons, Inc, 1981.
- [14] K. Salen and E. Zimmerman, *Rules of Play: Game Design Fundamentals*. 2004.
- [15] B. Brathwaite and I. Schreiber, *Challenges for game designers*. 2009.
- [16] K. Maroney, “My Entire Walking Life,” *The Games Journal | A Magazine About Boardgames*, 2001. .
- [17] J. Hamari and J. Koivisto, “Social motivations to use gamification: an empirical study of gamifying exercise,” *Proc. 21st Eur. Conf. Inf. Syst. Soc.*, no. JUNE, pp. 1–12, 2013.
- [18] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press New York, NY, USA ©2000, 2000.
- [19] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A Training Algorithm for Optimal Margin Classifiers,” *Proc. fifth Annu. Work. Comput. Learn. theory*, pp. 144–152, 1992.
- [20] T. Zhang, “An Introduction to Support Vector Machines and Other Kernel-based Learning Methods,” vol. 22, no. 2, pp. 103–104, 2001.
- [21] P. Melendez, “Controlling non-player characters using Support Vector Machines,” *Proceeding Futur. Play '09 Proc. 2009 Conf. Futur. Play @ GDC Canada*, pp. 33–34, 2009.
- [22] B. Fei and J. Liu, “Binary Tree of SVM: A New Fast Multiclass Training and Classification Algorithm,” *IEEE Trans. NEURAL NETWORKS*, vol. 17, no. 3, pp. 696–704, 2006.
- [23] S. Sriram, R. Vijayarangan, S. Raghuraman, and A. T. Game, “Implementing a No-Loss State in the Game of Tic-Tac-Toe using a Customized Decision Tree Algorithm,” *IEEE Int. Conf. Inf. Autom.*, pp. 1211–1216, 2009.
- [24] P. Spronck, I. Sprinkhuizen-kuyper, E. Postma, and U. M. Ikat, “Online Adaptation of Game Opponent AI in Simulation and in Practice,” *Proc. 4th Int. Conf. Intell. Games Simul.*, pp. 93–100, 2003.
- [25] P. Spronck, I. Sprinkhuizen-kuyper, and E. Postma, “Difficulty Scaling of Game AI,” *Proc. 5th Int. Conf. Intell. Games Simul. (GAME-ON 2004)*, pp. 33–37, 2004.