



**BUKU TESIS – KI142502**

**MENINGKATKAN KECERDASAN ADAPTIF  
*COMPUTER PLAYER* PADA GAME  
PERTARUNGAN BERBASIS *K-NEAREST  
NEIGHBOR* BERBOBOT**

**M. IHSAN ALFANI PUTERA  
NRP. 5115201027**

**DOSEN PEMBIMBING  
Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.**

**PROGRAM MAGISTER  
BIDANG KEAHLIAN INTERAKSI GRAFIKA DAN SENI  
DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2018**

*[Halaman ini sengaja dikosongkan]*

Tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar  
Magister Komputer (M. Kom.)  
di  
Institut Teknologi Sepuluh Nopember Surabaya


oleh:  
M. IHSAN ALFANI PUTERA  
NRP. 5115201027

Dengan judul:  
MENINGKATKAN KECERDASAN ADAPTIF COMPUTER PLAYER PADA  
GAME PERTARUNGAN BERBASIS K-NEAREST NEIGHBOR BERBOBOT

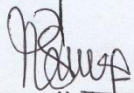
Tanggal Ujian : 12-1-2018  
Periode Wisuda : 2017 Gasal

Disetujui oleh:

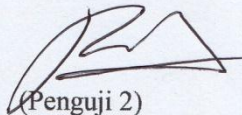
Dr. Eng. Darlis Heru Murti, S. Kom, M. Kom  
NIP. 19771217 2003121001

  
(Pembimbing 1)

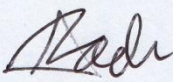
Dr. Eng. Nanik Suciati, S. Kom, M. Kom  
NIP. 19710428 1994122001

  
(Penguji 1)

Royyana Muslim I, S.Kom, M.Kom, Ph.D  
NIP. 19770824 2006041001

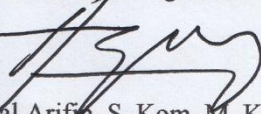
  
(Penguji 2)

Dr.Eng. Radityo Anggoro, S.Kom, M.Sc  
NIP. 19841016 20081210002

  
(Penguji 3)



Dekan Fakultas Teknologi Informasi,

  
Dr. Agus Zainal Arifin, S. Kom, M. Kom.  
NIP. 19720809 199512 1 001

*[Halaman ini sengaja dikosongkan]*

# **MENINGKATKAN KECERDASAN ADAPTIF *COMPUTER* PLAYER PADA GAME PERTARUNGAN BERBASIS K- NEAREST NEIGHBOR BERBOBOT**

Nama Mahasiswa : M. Ihsan Alfani Putera  
NRP : 5115201027  
Pembimbing : Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.

## **ABSTRAK**

Salah satu produk dari teknologi komputer yang berkembang dan perubahannya cukup pesat adalah *game*. Tujuan dibuatnya *game* adalah sebagai sarana hiburan dan untuk memberikan kesenangan bagi penggunanya. Salah satu contoh elemen dalam pembuatan *game* yang penting adalah adanya tantangan yang terukur dan seimbang sesuai level. Dalam hal ini, adanya kecerdasan buatan atau AI merupakan salah satu unsur yang diperlukan dalam pembentukan *game*. Penggunaan AI yang statis dan tidak beradaptasi ke strategi lawan akan mudah diprediksi dan repetitif. Sebaliknya, jika AI terlalu pintar maka *player* akan kesulitan dalam memainkan *game* tersebut. Dengan keadaan seperti itu akan menurunkan tingkat *enjoyment* dari pemain. Oleh karena itu, dibutuhkan suatu metode *adaptif* AI yang dapat beradaptasi dengan kemampuan dari *player* yang bermain. Sehingga tingkat kesulitan yang dihadapi dapat diatur secara otomatis mengikuti kemampuan pemainnya dan pengalaman *enjoyment* ketika bermain *game* terus terjaga. Terkait dengan metode K-NN yang digunakan, metode ini sudah sering digunakan pada penelitian sebelumnya khususnya pada *game* berjenis pertarungan. Namun metode tersebut menganggap semua atribut dalam *game* adalah sama sehingga hal ini mempengaruhi *learning* AI menjadi kurang akurat.

Penelitian ini mengusulkan metode untuk *adaptif* AI dengan menggunakan metode K-NN berbobot pada *game* berjenis pertarungan. Dimana, pembobotan tersebut dilakukan untuk memberikan pengaruh setiap atribut dengan perubahan bobot sesuai dengan aksi *player*. Dari hasil evaluasi yang dilakukan terhadap 50 kali pertandingan pada 3 skenario uji coba, metode yang diusulkan yaitu K-NN berbobot mampu menghasilkan tingkat *adaptif* AI dengan akurasi sebesar 72%. Sedangkan, metode sebelumnya yaitu K-NN hanya menghasilkan tingkat *adaptif* AI sebesar 38% dan metode *random* menghasilkan tingkat *adaptif* AI sebesar 25%.

**Kata Kunci:** *Game, Adaptive Artificial Intelligence, K-NN Berbobot, Computer Player, Game* Pertarungan.

*[Halaman ini sengaja dikosongkan]*

# IMPROVING ADAPTIVE INTELLIGENCE OF COMPUTER PLAYERS ON THE BATTLE GAME BASED ON WEIGHTED K-NEAREST NEIGHBOR METHOD

Name : M. Ihsan Alfani Putera  
Student Identity Number : 5115201027  
Advisor : Dr. Eng. Darlis Herumurti, S.Kom.,  
M.Kom.

## ABSTRACT

One of the computer technology products that develops and changes quite rapidly is game. The purpose of game creation is as an entertainment facility which gives pleasure to its users. One example of the important element in game creation is a measurable and balance challenge by level. In this case, the existence of artificial intelligence or AI is one of the elements which is needed in game formation. The static and unadaptive AI use will be easily predicted by the opponent. Moreover, the game will be repetitive. Conversely, if AI is too smart, the player will have difficulty in playing the game. Consequently, it will reduce the level of the players' enjoyment. Therefore, it needs an adaptive AI method that can adapt the capabilities of the players. So that the difficulty level can be arranged automatically by following the player's ability and enjoyment experience during the continuous play. Related to the previous studies about K-NN method use, this method had been frequently used in many studies, particularly in game battle type. However, the method considers that all the attributes in the game are similar so it affects the learning of AI which can be less accurate.

This study proposed a method for adaptive AI using the weighted K-NN method on game battle type. In this study, the weighting was done by giving an effect to each attribute with weight changes based on the player action. Based on the evaluation results of 50 times competition on 3 trial scenario, the proposed method, weighted K-NN was capable to result AI Adaptive level with the accuracy level about 72%. Meanwhile, the previous method of K-NN only resulted adaptive AI level about 38%, while the random method resulted an adaptive level of AI about 25%.

**Keywords:** *Game, Adaptive AI (Artificial Intelligence), Weighted K-NN, Computer Player, Battle game.*

*[Halaman ini sengaja dikosongkan]*



## KATA PENGANTAR

*Alhamdulillahirabbil'amin.* Puji dan syukur penulis panjatkan kehadiran Allah SWT atas berkat, rahmat dan hidayah-Nya, sehingga tesis ini dapat diselesaikan. Tesis ini dibuat sebagai salah satu syarat dalam menyelesaikan Program Studi Magister di Institut Teknologi Sepuluh November Surabaya. Penulis menyadari bahwa Tesis ini dapat diselesaikan karena dukungan dari berbagai pihak, baik dalam bentuk dukungan moral dan material.

Melalui kesempatan ini dengan kerendahan hati penulis mengucapkan terima kasih dan penghargaan setinggi-tingginya kepada semua orang untuk semua bantuan yang telah diberikan, antara lain kepada:

1. Ayahanda tercinta Taurus Alfani dan Ibunda tercinta Maeda Siti Arbaya untuk semua doa, pengorbanan dan usaha yang tak kenal lelah telah mendidik dan membimbing dengan penuh ketulusan untuk keberhasilan penulis.
2. Annisa Hatfiana Alfani P. serta seluruh keluarga yang selalu berdoa dan memberikan dukungan.
3. Bapak Dr. Eng. Darlis Heru Murti, S.Kom, M.Kom selaku pembimbing yang senantiasa memberikan arahan dan bimbingan kepada penulis. Semoga Allah SWT senantiasa merahmati Ibu dan keluarga.
4. Ibu Dr. Eng. Nanik Suciati, S. Kom, M. Kom, Bapak Royyana Muslim I, S.Kom, M.Kom, Ph.D, dan Bapak Dr.Eng. Radityo Anggoro, S.Kom, M.Sc selaku dosen penguji yang telah banyak memberikan ilmu, arahan, perbaikan dan saran pada tesis ini.
5. Seluruh dosen S2 Teknik Informatika yang telah memberikan ilmu dan pengetahuan kepada penulis selama menempuh studi.
6. Teman seperjuangan, Deny P., Andik Ampuh dan teman seangkatan lainnya yang tidak dapat disebutkan satu persatu, terima kasih atas bantuan dan motivasi yang telah diberikan.
7. Keluarga, kerabat, rekan dan teman dekat yang tidak dapat disebutkan satu-persatu.

Akhirnya dengan segala kerendahan hati penulis menyadari masih banyak terdapat kekurangan pada Tesis ini. Oleh karena itu, kritik dan saran dari pembaca dibutuhkan untuk memperbaiki dan mengembangkan penelitian ini. Penulis berharap semoga perbuatan baik dari semua orang yang dengan tulus memberikan kontribusi terhadap penyusunan Tesis ini mendapatkan pahala dari Allah. *Aamiin Alluhamma Aamiin.*

Surabaya, Januari 2018

M. Ihsan Alfani Putera

*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

LEMBAR PENGESAHAN THESIS .....	iii
ABSTRAK .....	v
ABSTRACT .....	vii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xv
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Perumusan Masalah .....	4
1.3 Tujuan dan Manfaat .....	5
1.4 Kontribusi penelitian .....	5
1.5 Hipotesis Penelitian .....	5
1.6 Batasan Masalah .....	6
BAB 2 KAJIAN PUSTAKA DAN DASAR TEORI .....	7
2.1 Kajian Pustaka .....	7
2.2 Game .....	7
2.3 <i>Battle Game</i> .....	8
2.4 Adaptive Artificial Intelligence .....	8
2.5 Kecerdasan buatan (Artificial Intelligence) .....	10
2.6 <i>K- Nearest Neighbour</i> .....	11
BAB 3 METODE PENELITIAN .....	13
3.1 Studi Literatur .....	13
3.2 Desain dan Implementasi .....	14
3.2.1 Analisis Kebutuhan .....	14
3.2.2 Desain Model Game .....	15
3.2.3 Desain Kecerdasan Buatan .....	17
3.2.4 Lingkungan Pengembangan .....	21
3.3 Pengujian .....	22
3.3.1 Pembuatan Dataset .....	22
3.3.2 Contoh Perhitungan Pengujian .....	23

3.3.3 Skenario Pengujian.....	24
3.4 Analisis Hasil.....	24
<b>BAB 4 UJI COBA DAN ANALISIS HASIL .....</b>	<b>27</b>
4.1 Hasil Desain dan Implementasi .....	27
4.1.1 Hasil Analisis Kebutuhan.....	28
4.1.2 Hasil Desain Sistem .....	29
4.1.3 Hasil Implementasi.....	29
4.1.3.1 Hasil Pembuatan Antarmuka ( <i>User Interface</i> ) .....	29
4.1.3.2 Hasil Pembuatan Sistem .....	33
4.2 Hasil Pengujian.....	34
4.2.1 Hasil Pembuatan Dataset.....	34
4.2.2 Hasil Skenario Pengujian .....	35
4.3 Analisis Hasil Pengujian.....	42
4.3.1 Analisis Hasil Pengujian Terhadap Tingkat Adaptif AI .....	42
4.3.2 Analisis Hasil Pengujian Terhadap Pembobotan Adaptif .....	43
<b>BAB 5 KESIMPULAN DAN SARAN.....</b>	<b>47</b>
5.1 Kesimpulan .....	47
5.2 Saran .....	47
<b>DAFTAR PUSTAKA.....</b>	<b>49</b>
<b>LAMPIRAN 1 .....</b>	<b>51</b>
<b>BIOGRAFI PENULIS .....</b>	<b>55</b>

## DAFTAR GAMBAR

Gambar 2.1 Langkah-langkah Algoritma K-NN .....	11
Gambar 3.1 Studi Literatur .....	13
Gambar 3.2 Game Pertarungan ( <i>Battle</i> ).....	16
Gambar 3.3 Rancangan Desain Kecerdasan Buatan pada Game.....	17
Gambar 3.4 Rancangan Desain Kecerdasan Buatan Menggunakan KNN-Bobot	18
Gambar 3.5 Alur Nilai Pembobotan .....	19
Gambar 3.6 Pembuatan Dataset .....	22
Gambar 3.7 Alur Skenario Pengujian pada <i>game battle</i> .....	24
Gambar 4.1 Tampilan UI Pemilihan Karakter <i>Player</i> .....	30
Gambar 4.2 Tampilan UI Pemilihan <i>Background Game</i> .....	31
Gambar 4.3 Menu Utama <i>Game</i> .....	31
Gambar 4.4 Tampilan Menu Tambahan <i>Damage</i> dan <i>Armor</i> .....	32
Gambar 4.5 Tampilan <i>History</i> Hasil Aksi yang dilakukan .....	32
Gambar 4.6 Grafik Hasil Pengujian dari 3 Metode yang Digunakan .....	32
Gambar 4.7 Grafik Persentase <i>Adaptive AI</i> terhadap <i>Player</i> .....	33
Gambar 4.8 Grafik Hasil Pembobotan pada Metode K-NN Berbobot .....	44

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 3.1 Contoh Nilai bobot masing-masing atribut.....	20
Tabel 3.2 Contoh Contoh Perhitungan dataset ke-41 .....	23
Tabel 3.3 Hasil Normalisasi.....	23
Tabel 3.4 Hasil Perhitungan Jarak pada K-NN dengan Pembobotan .....	23
Tabel 4.1 Hasil Pengujian Skenario <i>Player</i> – K-NN Berbobot ( <i>Enemy</i> ).....	35
Tabel 4.2 Hasil Pengujian Skenario <i>Player</i> – K-NN ( <i>Enemy</i> ).....	37
Tabel 4.3 Hasil Pengujian Skenario <i>Player</i> – <i>Random</i> ( <i>Enemy</i> ) .....	39
Tabel 4.4 Rangkuman Data Hasil Pengujian 3 Skenario .....	41

*[Halaman ini sengaja dikosongkan]*



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Di era globalisasi, banyak sekali inovasi dan kemajuan teknologi seiring berjalannya waktu. Salah satu teknologi yang tingkat perkembangannya sangat pesat yaitu komputer. Hampir semua aspek kehidupan manusia sangat bergantung dengan teknologi komputer. Dengan kemajuan teknologi, hampir semua konten, proses kerja dan interaksi fisik dapat dilakukan melalui proses digital. Salah satu produk dari teknologi komputer yang berkembang dan perubahannya cukup pesat adalah *game*. Industri *game* merupakan salah satu industri yang memiliki pengguna yang cukup banyak. Pada tahun 2015, lebih dari 150 juta orang di Amerika Serikat menjadi pengguna *game*. Dari jumlah tersebut, 42% memainkan *game* minimal 3 jam dalam satu pekan. Jumlah pengguna yang cukup banyak berdampak pada sisi ekonomi, pada tahun 2014, 22 miliar US Dollar dihasilkan dari industri ini dengan penjualan lebih dari 135 juta buah *game* (Entertainment Software Association, 2015). Fakta tersebut menggambarkan bahwa industri *game* adalah industri yang cukup besar dari sisi pengguna maupun penghasilan yang didapatkan oleh para pelaku industri *game*.

Secara umum tujuan dibuatnya *game* adalah sebagai sarana hiburan dan untuk memberikan kesenangan bagi penggunanya. Hal tersebut merupakan salah satu faktor yang mendorong tingkat perkembangan industri *game*. Berdasarkan laporan *Entertainment Software Association* (ESA Entertainment Software Association, 2014) disebutkan bahwa lebih dari 150 juta orang di Amerika Serikat menjadi pengguna *game*. 63% dari jumlah tersebut memainkan *game* minimal 3 jam tiap pekan. 65% dari jumlah tersebut memiliki perangkat untuk bermain *game* di rumah mereka. Hal tersebut membuktikan bahwa antusias masyarakat terhadap *game* cukup tinggi.

Dalam pembuatan *game*, pasti memerlukan suatu perancangan yang sesuai dengan tujuan pembuat. Menurut Marko (Urh *et al.*, 2015), salah satu

contoh elemen yang penting ialah adanya tantangan yang terukur dan seimbang sesuai level. Dalam hal ini, adanya kecerdasan buatan atau AI (*Artificial intelligence*) merupakan salah satu unsur yang diperlukan dalam pembentukan *game*. Saat ini juga banyak sekali *game* yang memiliki kecerdasan buatan (AI) dan kecerdasan komputasional didalam suatu permainan. Menurut Georgios (Yannakakis and Togelius, 2015), *game* AI merupakan *game* yang mengubah metode, proses, dan algoritma pada kecerdasan tersebut yang akan diaplikasikan ke pembuatan dan pengembangan *game*. Dia menyebutkan terdapat tiga panorama dalam *game* AI yakni perspektif metode (komputer), perspektif pengguna (manusia) dan perspektif interaksi pemain. Dalam *game* AI, Dagstul (Lucas, et al. 2012) juga menyebutkan terdapat sepuluh jenis *game* AI. Hal ini menunjukkan bahwa kecerdasan buatan dan kecerdasan komputasional sering digunakan pada kebanyakan *game* saling memiliki ketergantungan interaksi terhadap pemain. Sehingga AI memiliki peran yang penting untuk meningkatkan ketertarikan pengguna dalam bermain *game*.

Terdapat pengembang *game* yang masih menggunakan AI berdasarkan *rule based system*. Metode yang digunakan biasanya akan menghasilkan *behaviour* yang statis dan tidak beradaptasi ke strategi lawan. AI dengan kondisi yang seperti itu akan terlalu mudah diprediksi dan repetitif. Jika pemain mampu menghafalkan perilaku dari AI *game*, pemain akan kehilangan minatnya setelah bermain beberapa *game* dikarenakan tantangan yang kurang menarik (Wang And Tan, 2015). Hal seperti ini yang nantinya dapat menurunkan tingkat *enjoyment* pemain dalam bermain *game*. Berdasarkan hal tersebut, gagasan AI adaptif muncul di tengah perkembangan AI pada *game* saat ini. Dengan memanfaatkan teknik *learning* serta jenis teknik lainnya para peneliti mencoba untuk menciptakan agen AI yang adaptif. *Adaptive Artificial Intelligence* (AAI) mengacu pada karakter *Non Player Character* (NPC) dinamis dimana komputer mampu mengadaptasi *game behaviour*-nya dalam merespon musuhnya baik ketika sesi bermain *game* atau diantara sesi tersebut (Munajat and Toto, 2014). Dalam kasus tertentu, penskalaan tingkat kesulitan pada *game* dinamis menggunakan AAI untuk secara otomatis mengadaptasi parameter *game* dan

*behavior* pada kondisi *real time* menurut tingkat keahlian pemain pada *game* (Dahlbom, 2004).

Beberapa penelitian terkait *game* AAI dilakukan oleh Jiawei Li, dimana peneliti melakukan penelitian tersebut menghasilkan kecerdasan buatan *game* strategi yang adaptif sehingga menghasilkan *co-player* yang dapat beradaptasi dengan kebiasaan *player* (Li and Kendall, 2015). Pada penelitian Takafumi Nakamichi dan Takeshi Ito juga meneliti tentang adaptif *game* yang diterapkan pada *game* shogi (Nakamichi and Ito, 2016). Kecerdasan buatan yang dibuat bisa mendapatkan pengalaman baru untuk *co-player* sehingga *player* pun dapat bermain dengan pengalaman yang baru juga. Penelitian lain juga dilakukan Satrio (Munajat and Toto, 2014), dimana peneliti menggunakan metode berbasis adaptif pada *game* *Capture The flag*. *Game* tersebut merupakan salah satu *game* yang kompleks dan adversarial, dengan pemain yang saling berkompetisi & tujuan yang membutuhkan keputusan untuk dibuat di level permainan yang berbeda. *Game* tersebut bertujuan untuk menghindar atau mengejar serta menyerang dan mempertahankan bendera. Tujuan utama dari penerapan AAI dalam berbagai jenis *game* adalah untuk memberikan kondisi yang *enjoyment* terhadap pemain. Berdasarkan teori flow (Csikszentmihalyi, 1997) kondisi flow diperlukan pengaturan tingkat kesulitan. *Enjoyment* dapat terjadi apabila tercipta keseimbangan antara tantangan dalam *game* dan kemampuan yang dimiliki pemain. Tingkat kesulitan yang diatur secara otomatis apabila melihat teori flow adalah sebagai salah satu faktor yang dapat menjaga pengalaman kenikmatan di dalamnya. Hal tersebut dikarenakan ketika seseorang bermain *game* tingkat kesulitan diatur secara otomatis mengikuti kemampuan pemainnya sehingga pengalaman *enjoyment* ketika bermain *game* terus terjaga hingga pemain tersebut terus berulang-ulang memainkannya.

Pada penelitian yang telah dilakukan oleh (Muliawan, 2014) menggunakan metode fuzzy takagi sugeno pada *game* berjenis Action. Metode tersebut digunakan pada untuk mengontrol perilaku *Enemy* (musuh) agar menjadi dinamis. Hasil yang didapatkan menunjukkan bahwa dengan perilaku musuh yang lebih dinamis membuat *game* menjadi lebih menantang dan menarik untuk dimainkan.

Penelitian yang dilakukan oleh Thawonmas (Thawonmas, 2014) menggunakan metode *k-nearest neighbor* pada *game* berjenis *action fighting*. Peneliti menggunakan metode tersebut untuk memprediksi gerakan *player* secara efektif agar pada *game fighting* tersebut tidak membosankan dan lebih menantang untuk dimainkan. Pada penelitian ini K-NN telah diuji coba melawan tiga AI terbaik yaitu T, SejongAI, dan Kaiju. Hasil percobaan menunjukkan bahwa akurasi K-NN mengungguli tiga AI tersebut.

Pada penelitian dari Emigh, telah dilakukan penerapan pembobotan *nearest neighbor* pada *Q-learning* yang diterapkan pada *game* berjenis *arcade* (Emigh *et al.*, 2014). Tujuannya yaitu untuk meningkatkan akurasi *Q-learning*. Dari hasil penelitian ini didapatkan bahwa pembobotan mempengaruhi akurasi pada AI dari *game*.

Berdasarkan uraian di atas, K-NN sering diterapkan pada *computer player game*. Tetapi, K-NN menganggap bahwa semua atribut dalam *game* tersebut berbobot sama. Sedangkan pada penelitian (Emigh *et al.*, 2014) dibuktikan bahwa pembobotan atribut mempengaruhi akurasi AI. Oleh karena itu pada penelitian ini diusulkan pembobotan pada K-NN dengan tujuan untuk dapat mengatasi *learning* AI menjadi lebih akurat, sehingga perilaku AI (selaku *enemy*) dapat menyesuaikan (beradaptasi) dengan kemampuan dari *player* yang bermain (*user experience*).

## 1.2 Perumusan Masalah

Dalam sub-bab ini menjelaskan rumusan masalah yang akan diangkat dalam penelitian ini, yaitu:

1. Bagaimana merancang dan menerapkan kecerdasan buatan pada *game* berjenis pertarungan dengan menggunakan metode K-NN berbobot?
2. Bagaimana melakukan analisa untuk mengetahui keberhasilan *computer player* dalam meningkatkan kecerdasan adaptif dengan menerapkan metode K-NN berbobot pada kecerdasan buatan *game* berjenis pertarungan?

### 1.3 Tujuan dan Manfaat

#### Tujuan

Tujuan yang ingin dicapai dalam penelitian ini diantaranya adalah:

1. Merancang dan menerapkan kecerdasan buatan pada *game* berjenis pertarungan dengan menggunakan metode K-NN berbobot
2. Melakukan analisa untuk mengetahui keberhasilan penerapan metode K-NN berbobot pada kecerdasan buatan *game* berjenis pertarungan.

#### Manfaat

Beberapa manfaat yang dapat diperoleh dari penelitian ini adalah:

1. Bagi Pengembang *Game*  
Sebagai referensi untuk mengembangkan kecerdasan buatan *game* dalam menentukan pengembangan jenis *game* yang efektif.
2. Bagi Peneliti Lain  
Sebagai referensi bagi peneliti lain yang akan melakukan penelitian mengenai kecerdasan buatan pada *game*.

### 1.4 Kontribusi penelitian

Kontribusi penelitian ini adalah mengusulkan suatu sistem *game* yang memiliki kecerdasan buatan yang adaptif dengan melakukan pembobotan untuk memberikan pengaruh di setiap atribut dari *computer player*.

### 1.5 Hipotesis Penelitian

- Ha : *Computer player* pada *game* berjenis pertarungan dapat meningkatkan kecerdasan secara adaptif terhadap perilaku *player* dengan menerapkan metode K-NN berbobot.
- Ho : *Computer player* pada *game* berjenis pertarungan tidak dapat meningkatkan kecerdasan secara adaptif terhadap perilaku *player* dengan menerapkan metode K-NN berbobot.

## 1.6 Batasan Masalah

Dalam sebuah penelitian perlu diberikan adanya pembatasan masalah agar topik dan tujuan penelitian lebih terarah dan juga menghindari meluasnya permasalahan. Berikut adalah batasan masalah pada penelitian ini:

1. *Game* diimplementasikan dan diujikan pada *platform desktop*.
2. Jenis *game* yang dikembangkan berjenis pertarungan.
3. Kecerdasan buatan yang dibangun pada *Game* berjenis pertarungan yaitu berupa metode K-NN berbobot.
4. Adaptif *game* pada penelitian ini adalah adaptif terhadap *user experience*.

## **BAB 2**

### **KAJIAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Kajian Pustaka**

Kajian pustaka merupakan rangkuman singkat yang komprehensif tentang semua materi terkait yang terdapat di dalam berbagai referensi, berikut ini adalah kajian pustaka dalam penelitian ini:

#### **2.2 *Game***

Definisi dari *game* telah banyak dibahas oleh akademisi dan berbagai publikasi artikel. Setiap definisi yang dijabarkan, memiliki berbagai kesamaan, dan juga memiliki sejumlah perbedaan (Purkiss and Khaliq, 2016). Beberapa definisi tentang *game* diantaranya: *Game* adalah sebuah kegiatan yang melibatkan pengguna kedalam sebuah tujuan yang terikat oleh aturan-aturan (B. Suits, 1978). *Game* diciptakan melalui aturan permainan, yang bergantung pada tindakan pemain (Consalvo, 2009). Pada dasarnya definisi dari *game* adalah sebuah sistem kontrol yang bebas di mana didalamnya terdapat suatu pertentangan, dan dibatasi oleh prosedur untuk menghasilkan sebuah tujuan (E.M. Avedon, 1981). *Game* adalah sebuah sistem di mana pemain terlibat dalam pertempuran buatan, ditentukan oleh aturan, yang menghasilkan hasil yang terukur (Salen and Zimmerman, 2004). *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 (Brathwaite and Schreiber, 2009). *Game* dapat didefinisikan sebagai suatu bentuk permainan yang memiliki tujuan dan struktur. *Game* menyediakan aktivitas yang menarik minat pemain sekaligus sebagai media hiburan (Maroney, 2001). Layanan komputer yang didukung dengan *game* bertujuan untuk memotivasi dan mendukung aktivitas pelatihan pemain (Hamari and Koivisto, 2013).

Berdasarkan pernyataan-pernyataan tersebut, aspek-aspek umum dan definisi *game* dapat dengan mudah diidentifikasi, yaitu *game* adalah permainan

atau kegiatan yang memiliki seperangkat sistem aturan dan tujuan. Aspek lain yang menonjol di dalam *game* adalah adanya konflik atau pertempuran antara pemain dengan pemain lain maupun antara pemain dan sistem *game* itu sendiri. Berdasarkan pendapat yang dikemukakan (Purkiss and Khaliq, 2016), *video game* adalah sebuah kegiatan yang memanfaatkan layar video digital dengan cara tertentu, yang dibatasi oleh sistem aturan di mana seorang pemain dapat berinteraksi dengan pemain lain, maupun dengan sistem permainan itu sendiri, untuk mencapai suatu tujuan dan hasil yang diinginkan.

### **2.3 Battle Game**

*Video game* berjenis *battle* bisa disebut juga *role playing* adalah *game* yang diharuskan mengenali karakter, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam permainan, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain (biasanya menjadi semakin hebat, semakin kuat, semakin berpengaruh, dll) dalam berbagai parameter yang biasanya ditentukan dengan naiknya *level*, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun makhluk peliharaan. Secara kebudayaan, pengembang *game* Jepang biasanya membuat *Role Playing Game (RPG)* ke arah cerita linear yang diarahkan seolah karakter kita adalah tokoh dalam cerita itu, seperti *Final Fantasy*, *Dragon Quest* dan *Xenogears*. Sedangkan pengembang *game RPG* Eropa, cenderung membuat karakter kita bebas memilih jalan cerita sendiri secara non-linear, seperti *Ultima*, *Never Winter Nights*, *Baldur's Gate*, *Elder Scroll*, dan *Fallout*.

### **2.4 Adaptive Artificial Intelligence**

*Adaptive Artificial Intelligence (AAI)* mengacu ke non-player character (NPC) dinamis dimana komputer mampu mengadaptasi *game* behavior-nya dalam merespon musuhnya, baik ketika sesi bermain *game*, atau dalam diantara sesi tersebut. Dalam kasus tertentu, penskalaan tingkat kesulitan pada *game* dinamis menggunakan AAI untuk secara otomatis mengadaptasi parameter *game* dan behavior pada kondisi real time menurut tingkat keahlian pemain pada *game* (Tan, C.H., et al. 2011). Dalam paper berjudul 'Artificial Intelligence for Adaptive



bermain, dan tujuan pemain memainkan *game* Computer Games (Mehta, Ram and Onta, 2007) mengatakan bahwa tantangan untuk AI tersebut. *Game* adalah :

1. *Complex Decision Spaces*

jika sebuah *game* mampu beradaptasi sendiri, pihak pengembang *game* membutuhkan usaha yang lebih AI pada *game* umumnya menggunakan kodingan manual dari programmer *game*, sehingga kadang AI tersebut minimal untuk memprediksi segala kemungkinan yang akan terjadi. repetitive, dan pemain bisa dengan mudah mencari celah, dan mengakali AI tersebut. AAI mampu menangani kemungkinan pengambilan keputusan yang besar.

2. *Knowledge Engineering*

Pembuatan domain knowledge memerlukan usaha yang besar dalam membangunnya. Pengembang *game* harus membuat secara manual kodingan yang ada untuk sebuah spesifik domain (baik untuk memperoleh behavior strategi atau behavior yang cocok untuk RPG).

3. *Authoring Support*

*Behavior* yang dikembangkan secara manual adalah kode perangkat lunak di sebuah bahasa pemrograman yang kompleks, rentan terhadap kesalahan manusia. Behavior error bisa dalam bug program, atau tidak mencapai hasil yang diinginkan. Tools dibutuhkan untuk mendukung keinginan pengembang *game* yang pada umumnya tidak ahli dalam bidang kecerdasan buatan.

4. *Unanticipated Situations*

Hampir tidak memungkinkan untuk mengantisipasi semua kemungkinan situasi dan strategi pemain saat *game* berlangsung. Ini membuat AI sulit untuk dibuat se-alami mungkin untuk merespon setiap sikap yang pemain lakukan.

5. *User-specific Adaptation*

Pemain mempunyai strategi yang berbeda untuk bermain (untuk *game* RPG), atau cara yang berbeda dalam penyampaian alur cerita (untuk *game* drama interaktif, misalnya Visual Novel), tingkah karakter dan interaksi yang berbeda. Setiap *game* designer mulai untuk mendesain kemampuan setiap pengguna, strategi AI dan behavior nya harus bisa beradaptasi berdasarkan profil pengguna-nya.

#### 6. *Replayability and variability*

Pemain akan mengalami kebosanan saat bermain *game* yang ke-*n* kali, dan menjumpai strategi yang sama yang digunakan oleh AI pada *game* tersebut. Walaupun variasi yang simpel bisa diperoleh melalui seleksi stokastik dari behavior atau strategi dari repository yang besar.

#### 7. *Rhetorical Objectives*

*Behavior* atau strategi yang dikembangkan secara manual biasanya tidak mencapai tujuan *game* secara tepat. Terutama dalam aplikasi atau domain dalam skala yang cukup besar. Tujuan *game* bisa berkisar dari hiburan hingga edukasi, dsb. Sehingga, setiap *game* tersebut harus menyadari bahwa tujuannya tidak tercapai di basis perpenggunaan, dan beradaptasi sendiri. Sebagai contoh, beberapa user akan bosan, atau tidak mempelajari pelajaran yang dimaksud. Dan efek implementasi AAI pada *game* adalah :

- Meningkatkan pengalaman bermain, karena AAI bisa beradaptasi ke tiap individu untuk menyesuaikan cara bermain, dan tujuan pemain memainkan *game* tersebut.
- Mengurangi waktu dan biaya pengembangan, jika sebuah *game* mampu beradaptasi sendiri, pihak pengembang *game* membutuhkan usaha yang lebih minimal untuk memprediksi segala kemungkinan yang akan terjadi.

### 2.5 Kecerdasan buatan (**Artificial Intelligence**)

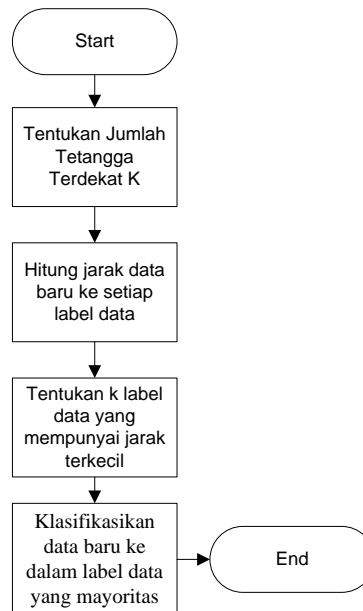
Kecerdasan buatan atau AI (**Artificial Intelligence**) merupakan ilmu tentang bagaimana membangun suatu sistem komputer yang menunjukkan kecerdasan dalam berbagai cara (Millington & Funge 2009). AI merupakan area penelitian yang dinamis dalam topik riset ilmu komputer. Sampai saat ini, telah banyak penelitian mengenai perkembangan AI diantaranya *decision tree*, *neural network*, *evolutionary computing*, *machine learning*, *natural language processing*, dan *object oriented programming* (Millington & Funge 2009). Millington (Millington & Funge 2009) mengungkapkan bahwa terdapat enam model pada *game* AI yaitu pergerakan, pengambilan keputusan, strategi, infrastruktur, *agent-*

based AI, dan *in the book. Game AI* memiliki tujuan untuk mendapatkan sebuah sistem yang dapat menunjukkan perilaku intelektual seperti perilaku manusia.

## 2.6 K-Nearest Neighbor

*Nearest Neighbor* merupakan metode untuk mengklasifikasikan suatu data baru berdasarkan similaritas atau kemiripan dengan *labeled data*. Similaritas menggunakan metrik jarak dengan satuan *Euclidian*. *K- Nearest Neighbor* adalah metode pengembangan dari NN (*Nearest Neighbor*). Dimana, algoritma ini termasuk ke dalam algoritma *supervised learning* dimana hasil dari *instance* yang baru diklasifikasikan berdasarkan mayoritas dari kategori K-tetangga terdekat. Tujuan algoritma ini adalah untuk mengklasifikasikan obyek baru berdasarkan atribut dan sampel-sampel dari data training. Algoritma *K- Nearest Neighbor* menggunakan *neighborhood classification* sebagai nilai prediksi dari nilai *instance* yang baru.

*K- Nearest Neighbor* bekerja berdasarkan jarak minimum dari data baru ke data training sampel untuk menentukan K tertangga terdekat. Selanjutnya kita dapatkan nilai mayoritas sebagai hasil prediksi dari data yang baru tersebut. Berikut Gambar 2.1 langkah-langkah dalam metode K-NN.



Gambar 2.1 Langkah-langkah Algoritma K-NN

Kelebihan dari Algoritma K-NN yaitu:

- Robust terhadap data yang *noisy*.
- Efektif jika training data dalam jumlah banyak.

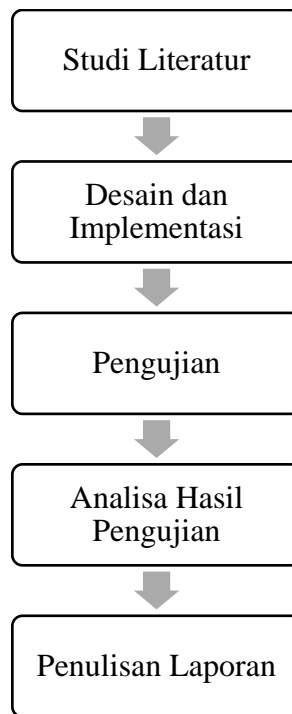
Kekurangan dari Algoritma K-NN yaitu:

- Perlu menunjukkann parameter K
- Berdasarkan perhitungan nilai jarak tidak jelas perhitungan mana yang sebaiknya digunakan dan atribut mana yang memberikan hasil yang baik.
- Nilai komputasi tinggi karena kita perlu menghitung jarak dari nilai baru ke semua data yang ada pada data training. Beberapa cara pengindexan dapat digunakan untuk mereduksi komputasi.

## **BAB 3**

### **METODE PENELITIAN**

Pada bab ini menjelaskan mengenai desain, metode atau pendekatan yang digunakan untuk menjawab permasalahan penelitian / studi untuk mencapai tujuan penelitian, serta tahapan penelitian secara rinci. Alur penelitian yang dilakukan pada penelitian ini yaitu : (1) studi literatur, (2) desain dan implementasi, (3) evaluasi, (4) analisa hasil pengujian dan (5) penulisan laporan, seperti terlihat pada Gambar 3.1.



Gambar 3. 1 Diagram alur penelitian

#### **3.1 Studi Literatur**

Pada tahap ini dilakukan studi literatur terhadap beberapa penelitian yang dianggap berhubungan dan relevan terhadap penelitian ini. Studi literatur dilakukan untuk mengetahui perkembangan penelitian lain dalam kurun waktu beberapa tahun terakhir pada pembahasan atau topik yang sama dengan topik penelitian yang akan dilakukan.

Pada penelitian ini dilakukan pencarian referensi atau literatur yang berkaitan dengan *game Battle*, *AI game*, metode adaptif. Dari hasil pencarian literatur yang telah dilakukan, didapatkan hasil sebagai berikut:

1. *Game* merupakan sebuah teknologi yang dibuat dan dikembangkan dengan tujuan untuk memberikan kesenangan dan sebagai media hiburan bagi pengguna.
2. AI adalah salah satu elemen dalam *game* yang dapat menambah nilai entertainment dalam *game*, karena dengan penerapan *AI game* akan menjadi lebih dinamis.
3. Adaptif *game* adalah metode-metode yang dapat diterapkan untuk menentukan perilaku AI dalam mengambil keputusan.

Berdasarkan dari hasil studi literatur yang telah dilakukan, hal yang dapat diangkat menjadi sebuah penelitian adalah analisa kecerdasan buatan berbasis metode adaptif pada *game battle enjoyment*.

## **3.2 Desain dan Implementasi**

Pada sub bab ini akan dijelaskan mengenai langkah-langkah dalam pembuatan dan pengembangan *game battle* yang meliputi: analisis kebutuhan, desain sistem dan implementasi.

### **3.2.1 Analisis Kebutuhan**

Analisis kebutuhan adalah suatu tahap pengumpulan informasi yang dapat dijadikan sebagai acuan dalam pengembangan *game battle*. Pengumpulan informasi berupa kebutuhan perangkat keras dan perangkat lunak yang digunakan, serta informasi mengenai materi yang akan dijadikan sebagai konten dalam pembuatan *game battle*. Oleh karena pengumpulan kebutuhan akan dilakukan dengan cara observasi baik terhadap pengguna yang akan menjadi target penelitian, ahli dalam bidang yang bersangkutan maupun sumber-sumber relevan yang dapat dijadikan sebagai referensi dalam pengembangan *game battle*. Tahap ini dilakukan agar nantinya *game battle* yang dikembangkan sesuai dengan harapan dan kebutuhan pengguna.

Arsitektur dan desain sistem dari *game* yang akan dibangun adalah *game* berjenis *battle*. *Game* ini akan melakukan pertarungan antara pemain dan *computer player*. Karakter memiliki sejumlah atribut yang bersifat dinamis, dan juga memiliki sejumlah kemampuan.

### 3.2.2 Desain Model *Game*

Desain model *game* merupakan rancangan dari hasil analisis kebutuhan. Dalam pengembangan *game* ini, tingkat kemampuan *Computer player* akan dibuat secara adaptif untuk menyesuaikan kemampuan *player*. *Game* yang akan dibangun memiliki genre yang bertema *Battle*. Pada awal pemain akan masuk ke halaman *record data*. Pada tampilan *record data* adalah dimana kita dapat melihat dataset awal dan data aksi pada permainan sebelumnya. Selanjutnya pada menu *next*, akan ditampilkan pemilihan karakter yang akan dipilih oleh *player*. Setelah memilih karakter, *player* akan memilih *background* dimana *player* akan menentukan tempat pertarungan pada *game* tersebut berlangsung. Setelah memilih *background*, pemain akan masuk ke halaman *battle*. Halaman *battle* ini adalah halaman utama dari permainan ini dimana *player* akan memainkan sebuah karakter dan akan bertarung menghadapi karakter *Computer Player*. Setiap karakter memiliki empat atribut yaitu *health*, *energy*, *armor*, dan *damage*. *Health* diibaratkan sebagai nyawa yang dimiliki karakter, apabila jumlah *health* habis maka karakter akan mati atau kalah. *Energy* diibaratkan sebagai tenaga pada karakter untuk melakukan serangan. *Damage* adalah sejumlah nilai serangan yang digunakan untuk mengurangi *health* yang dimiliki lawan. *Armor* adalah nilai pertahanan untuk mengurangi nilai serangan yang diberikan oleh lawan. Selain atribut, karakter juga mempunyai empat kemampuan yaitu *attack*, *super attack (ultimate)*, *defense* dan *energy restore (healing)*. *Attack* adalah sebuah kemampuan untuk menyerang lawan. *Super attack (ultimate)* adalah kemampuan untuk menyerang dengan serangan yang lebih besar daripada *attack*. *Defense* adalah sebuah kemampuan untuk bertahan dari serangan lawan. *Energy restore (healing)* berfungsi untuk menambah *energy*.



Gambar 3.2 *Game* Pertarungan (*Battle*)

Rancangan aturan permainan *game* pertarungan (*battle*) ini adalah sebagai berikut.

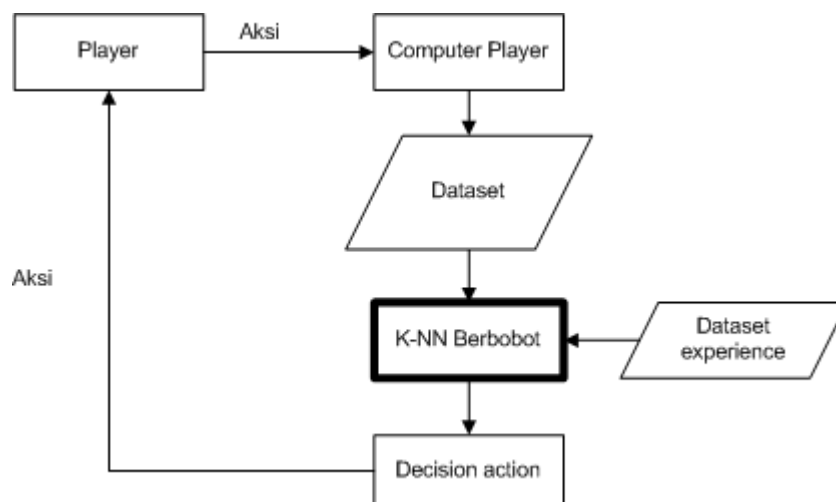
1. Mempertandingkan antara pemain dengan computer *player* dengan menggunakan metode AI K-NN Berbot, K-NN dan *random*.
2. Setiap karakter memiliki atribut *health*, *energy*, *damage*, *armor*, dan setiap karakter memiliki aksi *skill ultimate*, *skill attack*, *skill healing*, dan *skill defense*.
3. Jumlah *health* sebesar 200.
4. Jumlah *energy* sebesar 100.
5. Jumlah *damage* dan *armor* akan bertambah sebanyak hit poin.
6. Setiap karakter memiliki kesempatan untuk menyerang atau bertahan secara bergantian.
7. *Skill ultimate* untuk menyerang musuh dengan nilai *damage* sebesar 20 dengan membutuhkan *cost* sebesar 30 *energy*.
8. *Skill attack* untuk menyerang musuh dengan nilai *damage* 10 dengan membutuhkan *cost* sebesar 20 *energy*.
9. *Skill healing* bertujuan untuk menambah *energy* sebesar 15.



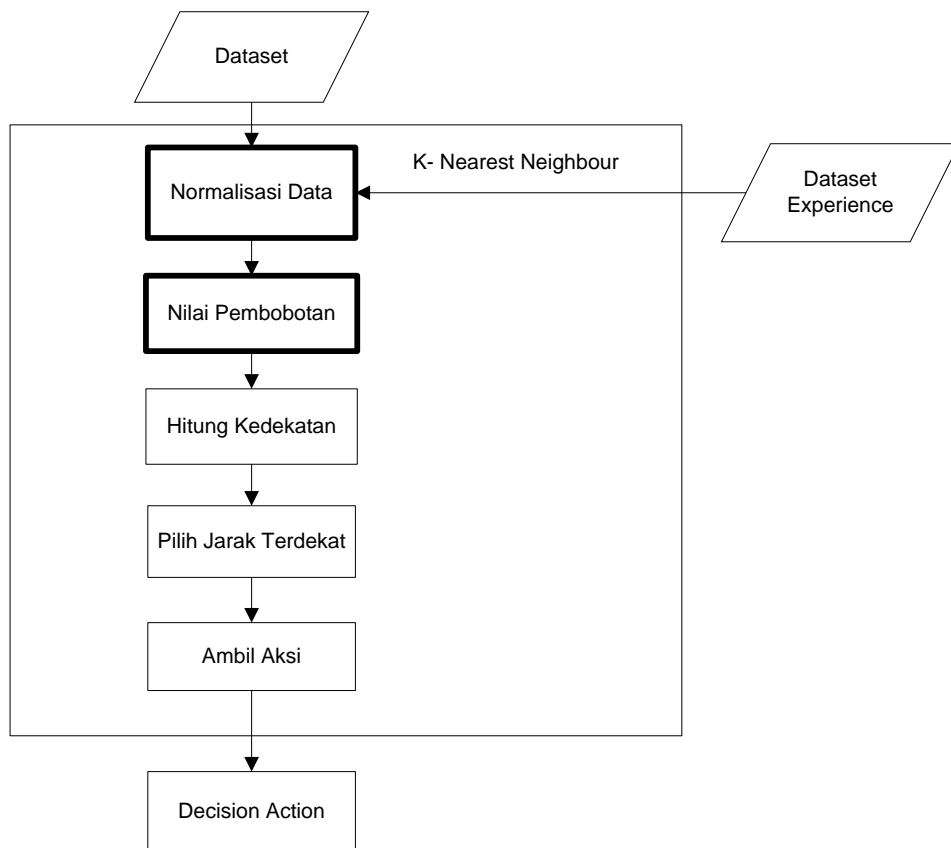
10. *Skill defense* untuk menghindar atau menahan dari serangan musuh membutuhkan *cost* sebesar 10 *energy*.
11. *Energy* akan berkurang sesuai dengan nilai *cost skill* yang digunakan.
12. Jika terkena serangan maka *health* akan berkurang sebesar *damage skill* serangan musuh.
13. *Skill* tidak bisa digunakan jika *cost energy* lebih besar dari pada *energy* yang dimiliki saat itu (*energy* tidak mencukupi).
14. Permainan selesai jika *health* salah satu dari *computer player* habis (mencapai 0 *health*).
15. *Computer player* dikatakan menang apabila berhasil mengalahkan lawan. Indikator kekalahan adalah apabila *health* mencapai 0.

### 3.2.3 Desain Kecerdasan Buatan

Perancangan desain kecerdasan buatan merupakan bagian dari desain model *game*. Perancangan kecerdasan buatan ini merupakan fokus dari penelitian ini, kecerdasan buatan dirancang untuk dapat mengambil keputusan menyerang atau bertahan. Di dalam *game* ini akan dibangun kecerdasan buatan dengan menggunakan *Nearest neighbour*. Desain kecerdasan buatan selengkapnya dapat dilihat pada Gambar 3.3 dan Gambar 3.4.



Gambar 3. 3 Rancangan Desain Kecerdasan Buatan Pada Game.



Gambar 3. 4 Rancangan Desain Kecerdasan Buatan Menggunakan K-NN Berbobot.

Pada gambar 3.4 ketika *player* melakukan aksi kepada *computer player*, maka aksi tersebut akan masuk ke dalam dataset. Dataset tersebut akan dilakukan perhitungan dengan metode K-NN untuk mengambil keputusan aksi apa yang akan diambil. Selanjutnya *computer player* akan melakukan aksi kepada *player* yang sudah ditentukan. Pada gambar 3.5 proses perhitungan dalam metode K-NN akan diawali dengan normalisasi data sampai proses ambil aksi.

Atribut yang digunakan untuk pengambilan keputusan metode K-NN adalah *health*, *energy*, *damage*, *health* dan *armor*.

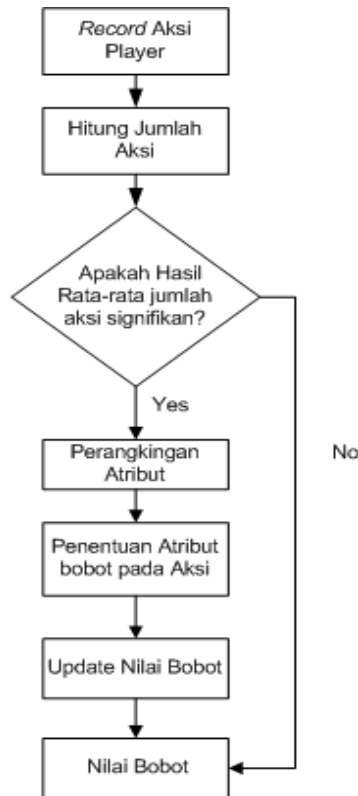
#### 1. Normalisasi data

Proses tersebut dilakukan untuk mengoptimisasi data yang diperoleh. Persamaan normalisasi data ditulis dengan menggunakan persamaan (3.1) sebagai berikut.

$$X_{norm} = \frac{X_i - X_{min}}{X_{max} - X_{min}} \quad (3.1)$$

Hasil yang didapat dari proses normalisasi di atas adalah mengubah semua nilai dari range 0 sampai 1, serta dapat memudahkan proses pembobotan pada tahap selanjutnya.

## 2. Nilai Pembobotan



Gambar 3.5 Alur Proses Menentukan Nilai Pembobotan

Pada Gambar 3.5 dijelaskan bahwa nilai bobot yang diberikan berdasarkan aksi yang diberikan oleh *player*. Setelah di *record* jumlah hasil dari aksi *player*, lalu hasil tersebut di rata-rata. Jika hasil dari nilai rata-rata tidak mengalami perubahan atau mengalami perubahan tapi tidak signifikan, maka tidak akan mengubah nilai bobotnya. Jika nilai rata-rata mengalami perubahan yang signifikan, maka akan dilakukan perankingan kembali pada tiap atribut dan dilakukan penentuan nilai bobot. Ada 4 atribut game pada game pertarungan yang dibangun yaitu *energy*, *damage*, *armor*, dan *health*. Masing-masing atribut memiliki bobot yang sama Pada setiap aksi dalam game mempengaruhi masing-masing atribut. Tabel 3.1 menjelaskan pengaruh atribut pada setiap aksi.

Tabel 3.1. Pengaruh Atribut Pada Setiap Aksi

Aksi	Atribut
<i>Heal</i>	<i>Energy</i>
<i>Attack</i>	<i>Energy, Damage</i>
<i>Defense</i>	<i>Health, Armor</i>
<i>Ultimate</i>	<i>Energy, Damage</i>

Pada Tabel 3.1 di atas ditunjukkan bahwa aksi *heal* berpengaruh dengan atribut *energy*, aksi *attack* berpengaruh pada atribut *energy* dan *damage*, aksi *defense* berpengaruh dengan atribut *health* dan *armor*, dan aksi *ultimate* berpengaruh pada atribut *energy* dan *damage*.

Aksi *player* yang sudah di *record* selanjutnya aksi tersebut di masukkan menjadi jumlah masing-masing atribut dan dilakukan perankingan sesuai dengan aksi mana yang lebih banyak dipakai ketika *player* bermain. Syarat dalam nilai pembobotan yaitu jika dijumlahkan hasilnya adalah 1. Jadi, awal dari nilai setiap atribut memiliki jumlah sebesar 0,25. Dalam perankingan, pada bobot tersebut akan diubah nilainya sesuai ranking masing-masing. Pada perubahan nilai disesuaikan dengan jumlah pertandingan. Pada penelitian ini nilai yang digunakan jika ranking pertama nilai pembobotan ditambah 0,022, ranking kedua nilai pembobotan ditambah 0,011, ranking ketiga nilai pembobotan dikurang 0,011, dan ranking keempat nilai pembobotan kurang 0,011. Tabel 3.2 menjelaskan contoh perhitungan dan nilai bobot masing-masing atribut.

Tabel 3.2. Contoh Perhitungan Dan Nilai Bobot Masing-Masing Atribut

Atribut	Rangking	Bobot ( $\beta$ )
<i>Energy</i>	1	$0,25 + 0,022 = \mathbf{0,272}$
<i>Damage</i>	2	$0,25 + 0,011 = \mathbf{0,261}$
<i>Armor</i>	3	$0,25 - 0,011 = \mathbf{0,239}$
<i>Health</i>	4	$0,25 - 0,022 = \mathbf{0,228}$

Pada Tabel 3.2 di atas menjelaskan bahwa setiap atribut memiliki pengaruh. Atribut yang diutamakan adalah atribut *energy* pada urutan pertama memiliki nilai bobot 0,272. Dilanjutkan dengan atribut *damage* dengan nilai bobot 0,261 dan yang terakhir adalah atribut *armor* dengan nilai bobot 0,239 dan *health* yang memiliki rangking terakhir dengan nilai 0,228. Semakin tinggi pengaruh atribut terhadap tingkat kesulitan maka akan semakin tinggi nilai bobot atribut tersebut.

Untuk nilai pembobotan dari masing-masing atribut didapatkan dengan menggunakan persamaan 3.2 sebagai berikut.

$$\alpha = 1 - \beta \quad (3.2)$$

### 3. Menghitung Kedekatan

Setelah mendapatkan nilai data baru untuk semua data, selanjutnya dihitung jarak kedekatan antar data. Perhitungan tersebut menggunakan persamaan *euclidian distance* sebagai berikut.

$$d_{ij} = \sqrt{\sum_{k=1}^n (a_{ik} - a_{jk})^2} \quad (3.3)$$

Dimana  $n$  adalah jumlah atribut dan  $d_{ij}$  adalah jarak antar data. Semakin kecil nilai jarak yang didapat maka semakin mirip kedua data tersebut. Data yang dihitung nilai kedekatannya adalah data *training* (dataset *experience*) dan data *testing* (data kondisi AI terakhir).

### 4. Memilih Jarak Terdekat dan Ambil Aksi

Setelah dihitung semua nilai kedekatan antar data, selanjutnya akan dipilih nilai kedekatan dengan nilai yang terkecil. Nilai terdekat akan digunakan untuk menentukan keputusan AI yang akan diambil. Sehingga AI akan melakukan aksi diantara aksi yang tersedia pada *player*.

Aksi hanya ditentukan berdasarkan jarak terdekat/terkecil saja. Dimana, artinya K-NN hanya memiliki  $K=1$ . Dikarenakan *multi class* atau aksinya bervariasi. Proses K-NN akan diulangi terus-menerus sampai permainan berakhir.

## 3.2.4 Lingkungan Pengembangan

Pengembangan *game* dalam penelitian ini akan dilakukan pada komputer desktop dengan spesifikasi berikut:

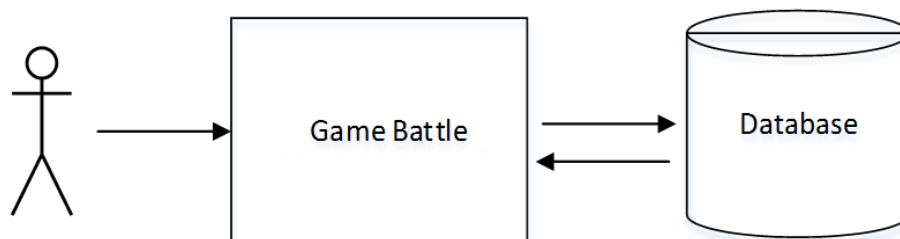
1. Prosesor Intel Core i5 3,7 GHz
2. VGA Nvidia 1050ti 4GB
3. RAM 8GB
4. Harddisk 1TB
5. Sistem operasi Windows 10 (64 Bit)

### 3.3 Pengujian

Pada sub bab ini akan dijelaskan tentang pengujian terhadap *game* yang akan dikembangkan yang meliputi : pembuatan dataset, skenario pengujian dan analisis hasil uji coba.

#### 3.3.1 Pembuatan Dataset

Dataset diperlukan untuk klasifikasi sebagai *training*. Diperlukan adanya dataset untuk dipergunakan untuk klasifikasi sebagai data *training*. Pada penelitian ini dataset digunakan untuk *training* metode *K-NN*. Dataset tersebut berisi nilai-nilai dari atribut dataset yaitu *health*, *energy*, *damage*, dan *armor*. Dataset dibangun dari percobaan *user* melawan metode kecerdasan buatan *rulebase*. Dimana *user* yang dipilih merupakan pakar dalam *game*. Pada setiap proses pengumpulan data, akan disimpan perlakuan user dalam menghadapi musuh didalam *game battle*. Selanjutnya data tersebut dihitung mengetahui layak tidaknya data tersebut digunakan dalam penelitian. Karena, pada dasarnya data tersebut juga mempengaruhi hasil pertandingan. Diharapkan dengan data yang sesuai dapat menghasilkan hasil optimal dan tentunya dapat juga menghindari *noise* pada data. Pada Gambar 3.6 adalah bagan dari pembuatan dataset dan dataset awal akan ditunjukkan pada Lampiran 1.



Gambar 3.6 Pembuatan dataset

### 3.3.2 Contoh Perhitungan Pengujian

Menentukan aksi yang akan diambil pada dataset ke-41. Dimana diketahui nilai masing-masing atribut sebagai berikut dengan ditunjukkan Tabel 3.2.

Tabel 3.2. Contoh Perhitungan dataset ke-41

No.	<i>Health</i>	Energi	<i>Damage</i>	<i>Armor</i>	<i>Decision</i>
50	50	40	5	5	?

Selanjutnya, untuk mendapatkan *decision* dilakukan proses *training* dengan cara menghitung jarak pada tiap data. Sebelumnya, data *training* dan data testing dilakukan normalisasi dengan persamaan 3.1 sehingga hasil normalisasi dapat ditunjukkan pada Tabel 3.3.

Tabel 3.3. Hasil Normalisasi

No.	<i>Health</i>	Energi	<i>Damage</i>	<i>Armor</i>	<i>Decision</i>
1	1	1	0	0	S
2	0.8	0.6	0	0	S
No.	<i>Health</i>	Energi	<i>Damage</i>	<i>Armor</i>	<i>Decision</i>
....					
41	0.12	0.4	0.6	0.5	?

Selanjutnya dilakukan perhitungan untuk nilai K-NN dengan pembobotan. Proses perhitungan dapat dilihat pada Tabel 3.4 berikut.

Tabel 3.4. Hasil Perhitungan Jarak pada K-NN dengan Pembobotan

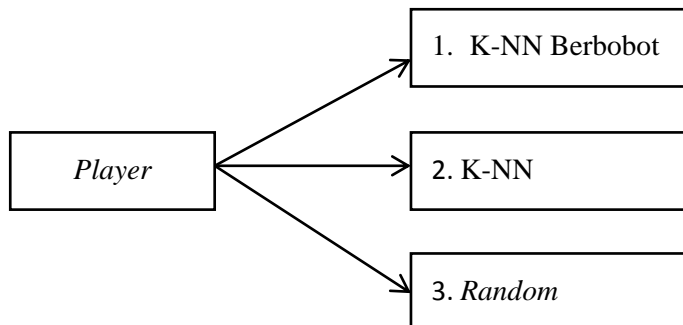
No.	Proses Perhitungan
1.	$(1-0.12)^2 \cdot (1-0.25) + (1-0.4)^2 \cdot (1-0.25) + (0-0.6)^2 \cdot (1-0.25) + (0-0.5)^2 \cdot (1-0.25) = 2.27$
2.	$(0.8-0.12)^2 \cdot (1-0.25) + (0.6-0.4)^2 \cdot (1-0.25) + (0-0.6)^2 \cdot (1-0.25) + (0-0.5)^2 \cdot (1-0.25) = 1.91$
....	
41	<i>Hasil = Jarak Terkecil</i>

Dari hasil perhitungan jarak di atas, *decision* yang akan diambil untuk data ke-41 yaitu data yang memiliki hasil perhitungan jarak yang terkecil. Sehingga aksi yang akan dipilih oleh AI sesuai dengan dataset yang memiliki jarak terkecil.

### 3.3.3 Skenario Pengujian

Pada tahap uji coba terhadap metode yang diusulkan akan dilakukan 3 skenario uji coba. Pertama menguji cobakan secara langsung *player* dengan metode K-NN berbobot. Kedua, menguji cobakan secara langsung *player* dengan metode K-NN. Ketiga, menguji cobakan secara langsung *player* dengan metode *random*. Pada masing-masing uji coba yang dilakukan akan memperhatikan seberapa banyak aksi yang dilakukan pada masing-masing *player*. Dimana, aksi yang dilakukan terdapat 4 poin yaitu, *ultimate*, *attack*, *defence*, dan *heal*.

Setelah dilakukan pengujian, selanjutnya hasil dari aksi yang dilakukan antara *player* dan *enemy* akan dihitung seberapa banyak kemenangan dan kekalahan dari *player* terhadap *enemy* pada masing-masing skenario uji coba. Hal ini perlu dilakukan untuk dapat mengetahui seberapa besar *presentase* tingkat *adaptif* dari kecerdasan AI terhadap *player* pada *game* tersebut. Uji coba yang dilakukan pada masing-masing skenario pengujian sebanyak 50 kali percobaan. Berikut Gambar 3.8 ditunjukkan alur skenario pengujian pada *game battle*.



Gambar 3.7 Alur Skenario Pengujian pada *Game Battle*

### 3.4 Analisis Hasil

Berdasarkan uji coba dalam penelitian ini, evaluasi untuk uji coba adalah sebagai berikut.

Evaluasi perhitungan *presentase* tingkat *adaptif* dari kecerdasan buatan AI terhadap *player* diketahui dari perbandingan jumlah kekalahan dan kemenangan yang dihasilkan pada setiap skenario pengujian yang dilakukan. Evaluasi perhitungan *presentase* tingkat *adaptif* mengikuti Persamaan 3.4 apabila jumlah kemenangan lebih besar dan Persamaan 3.5 apabila jumlah kekalahan lebih besar.



$$\text{Persentase Tingkat Adaptif} = \frac{\text{jumlah kekalahan}}{\text{jumlah kemenangan}} \times 100\% \quad (3.4)$$

$$\text{Persentase Tingkat Adaptif} = \frac{\text{jumlah kemenangan}}{\text{jumlah kekalahan}} \times 100\% \quad (3.5)$$

Persentase tingkat *adaptif* didapatkan dengan membandingkan jumlah kekalahan dan kemenangan *player* terhadap AI dengan diskalakan dari 0 sampai 100. Jika nilai persentase mendekati angka 0 maka dapat dikatakan bahwa tingkat adaptif *game* tersebut adalah rendah. Begitu juga sebaliknya jika nilai persentase mendekati angka 100 maka dapat dikatakan bahwa tingkat adaptif *game* tersebut adalah tinggi.

*[Halaman ini sengaja dikosongkan]*

## **BAB 4**

### **UJI COBA DAN ANALISIS HASIL**

Pada bab ini dipaparkan hasil pembuatan dan pengembangan, pengujian, analisis data serta pembahasan. Analisis data didapat berdasarkan hasil uji coba yang telah dilakukan sesuai dengan skenario yang telah dijelaskan pada bab 3.

#### **4.1 Hasil Desain dan Implementasi**

Pada sub bab ini akan dijelaskan mengenai hasil pembuatan dari *game* pertarungan (*battle*) yang terdiri dari hasil analisis kebutuhan, hasil desain sistem, dan hasil implementasi.

##### **4.1.1 Hasil Analisis Kebutuhan**

Analisis kebutuhan ini dilakukan untuk mengetahui gambaran mengenai aturan permainan, skenario, dan desain *game* yang akan dibuat. Dua cara yang dilakukan untuk melakukan analisis kebutuhan tersebut. Pertama dilakukan observasi terhadap pihak yang dapat membantu dalam mendesain sekaligus membantu dalam pembuatan *game*. Observasi dilakukan terhadap pihak desainer *game* dan dilakukan dengan wawancara.

Kedua, mencari referensi yang relevan terkait *game* pertarungan (*battle*) yang akan dibuat. Dalam hal ini sumber referensi yang akan digunakan adalah referensi mengenai fungsionalitas dan desain *game*. Referensi mengenai desain pada *game* yang akan dikembangkan berasal dari *game-game* dengan jenis yang sama atau hampir sama yang sudah ada sebelumnya dipasaran. Desain *game* yang diamati dari *game-game* tersebut meliputi: fungsionalitas, aturan main, skenario dan tampilan antar muka. Pada penelitian ini referensi *game* diambil dari sejumlah *game* yang sudah ada di Google Playstore. Tujuan dari melihat dan mengamati desain pada *game-game* tersebut adalah untuk mempermudah dalam pembuatan desain dari *game* pertarungan (*battle*). Referensi mengenai kecerdasan buatan pada *game* yang akan dikembangkan berasal dari penelitian-penelitian serupa yang membahas tentang kecerdasan buatan *game*.

Dari proses analisis kebutuhan yang sudah dilakukan didapatkan kebutuhan fungsionalitas dalam pembuatan *game* pertarungan (*battle*) sebagai berikut.

1. Peneliti dapat melihat *gameplay* yang disajikan dalam *game* pertarungan (*battle*).
2. Desain dan tampilan antar muka pada *game* pertarungan (*battle*) disesuaikan dengan jenis *game* pertarungan (*battle*).
3. Pengambilan keputusan pada *computer player* pada *game* pertarungan (*battle*) harus dapat dilakukan dengan cepat dan akurat.
4. Kecerdasan buatan pengambil keputusan pada *game* pertarungan (*battle*) adalah KNN-Berbobot dan KNN.

#### **4.1.2 Hasil Desain Sistem**

Desain sistem adalah tahapan lanjutan dari hasil analisis kebutuhan yang telah dihasilkan, kemudian melakukan representasi dari *game* pertarungan (*battle*) yang akan dikembangkan. Berdasarkan hasil dari kebutuhan yang diperoleh maka didapatkan hasil dari pembuatan desain sistem pada *game* pertarungan (*battle*) sebagai berikut.

1. Konten utama pada *game* adalah berupa permainan pertarungan (*battle*), dimana dalam permainan terdapat dua buah karakter yang saling menyerang, satu sebagai *player* dan satu sebagai *enemy*.
2. *Enemy* dalam *game* akan dilakukan oleh *computer player* yang memiliki kecerdasan buatan yang berbeda untuk mengambil keputusan. Dalam hal ini kecerdasan buatan yang digunakan adalah KNN-berbobot dan KNN.
3. Dataset digunakan sebagai data *training* kecerdasan buatan yang akan dipakai oleh kecerdasan buatan untuk mengambil keputusan.
4. Terdapat *icon* untuk menunjukkan jumlah *energy* dan *health* yang masih tersisa.
5. Terdapat *icon* untuk menunjukkan aksi yang dapat dipilih oleh *player* pada saat permainan berlangsung.

### 4.1.3 Hasil Implementasi

Setelah desain dari *game* pertarungan (*battle*) telah dibuat, tahap selanjutnya dilakukan implementasi desain *game* menjadi sebuah aplikasi perangkat lunak. Implementasi dilakukan dengan bantuan sejumlah perangkat lunak diantaranya:

1. *Adobe Flash CS6* sebagai alat bantu penulisan kode program, pembuatan animasi dan pembuatan antar muka. Dalam hal ini *game* pertarungan (*battle*) yang dikembangkan dibuat dengan menggunakan bahasa pemrograman *Action Script 2*.
2. *XAMPP* sebagai alat bantu pembuatan database.
3. *PhotoScape* versi 3.7 sebagai alat bantu dalam pengolahan gambar bertipe bitmap.
4. *Adobe Illustrator CS6* sebagai alat bantu dalam pengolahan gambar bertipe vector.

Selain menggunakan bantuan perangkat lunak, pengembangan *game* pertarungan (*battle*) juga dibuat dengan bantuan perangkat keras yaitu seperangkat komputer dengan spesifikasi: Sistem Operasi Microsoft Windows 10 64bit, Processor Core i5 3.7 Ghz, RAM 8GB, Hardisk 1TB, dan VGA Nvidia 1050ti 4GB. Perangkat ini digunakan sebagai lingkungan pengembangan dan pengujian *game* pertarungan (*battle*). Sejumlah konten yang ada pada *game* tersebut juga berasal dari beberapa sumber, salah satunya adalah situs [www.freepik.com](http://www.freepik.com) untuk pengambilan sejumlah konten vector dan gambar. *game* pertarungan (*battle*) dibuat dalam bentuk dua dimensi dan diimplementasikan pada *platform desktop* yaitu windows. Hasil implementasi *game* yang telah dikembangkan akan dijelaskan dalam dua bagian yaitu: hasil pembuatan antar muka dan hasil pembuatan sistem.

#### 4.1.3.1 Hasil Pembuatan Antarmuka (*User Interface*)

Hasil pembuatan antarmuka *game* akan ditunjukkan pada Gambar 4.1, Gambar 4.2, Gambar 4.3, Gambar 4.4 dan Gambar 4.5. Gambar 4.1 merupakan gambar dari tampilan awal dari *game* pertarungan (*battle*) ini, dimana *player* terlebih dahulu memilih satu dari tiga karakter yang disediakan. Karakter yang

akan dipilih tersebut akan melawan karakter dari *computer player* yang berlaku sebagai *enemy*.



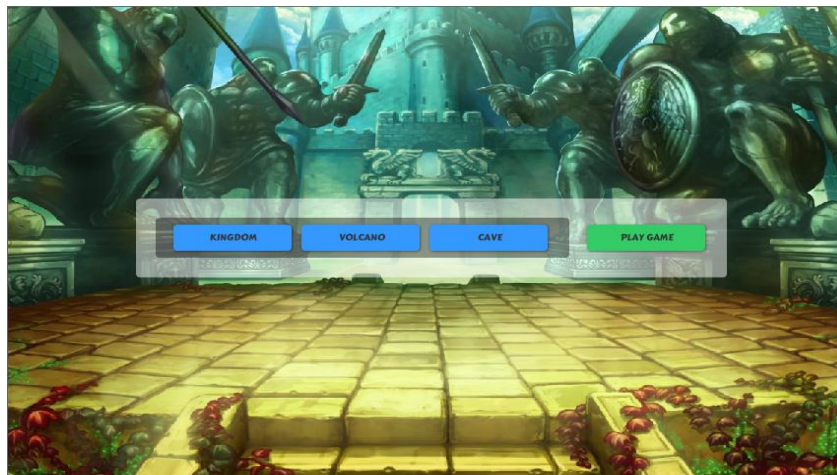
Gambar 4.1. Tampilan UI Pemilihan Karakter *Player*

Selanjutnya, *player* melakukan pemilihan latar belakang suasana pertandingan yang akan diikuti. Berikut ditunjukkan Gambar 4.2 tampilan pemilihan *background game*. Kemudian dilanjutkan lagi dengan masuk pada menu utama dari *game* pertarungan (*battle*) ini yang ditunjukkan pada Gambar 4.3. *Player* akan menghadapi *enemy* yang sudah disediakan oleh *computer player*. Di dalam menu utama ini terdapat *icon* (komponen) pendukung permainan. Berikut penjabaran fungsi dari masing-masing komponen yang terdapat dalam menu utama.

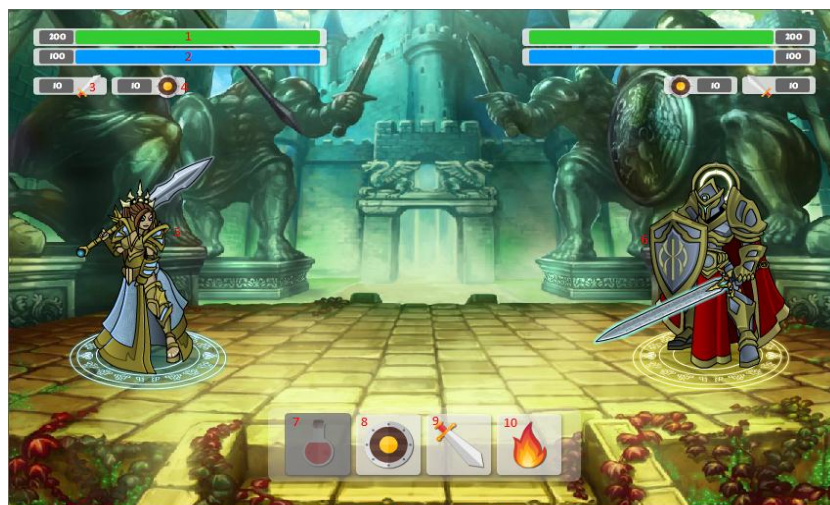
1. *Icon* yang menunjukkan jumlah *health* dari masing-masing *player* yang divisualisasikan dalam bentuk bar dan angka. *Player* akan terus bermain sampai nilai *health* ini adalah 0.
2. *Icon* yang menunjukkan jumlah *energy* yang dikeluarkan pada setiap aksi yang dilakukan dan ditunjukkan dalam bentuk bar dan angka.
3. *Icon* yang menunjukkan jumlah *damage* yang dilakukan pada masing-masing *player*.
4. *Icon* yang menunjukkan jumlah *armor* yang dilakukan pada masing-masing *player*.
5. Karakter *player* yang dimainkan oleh *user*.

6. Karakter *enemy* yang dimainkan oleh *computer player* dengan disertai kecerdasan berupa metode K-NN berbobot, K-NN, dan *random*.
7. Tombol untuk melakukan *skill healing*.
8. Tombol untuk melakukan *skill defense*.
9. Tombol untuk melakukan *skill attack*.
10. Tombol untuk melakukan *skill ultimate*.

Selanjutnya pada Gambar 4.4 ditunjukkan tampilan dari menu penambahan *damage* dan *armor* selama pertandingan berlangsung. Pada Gambar 4.5 ditunjukkan kondisi ketika player memenangkan permainan dan pada Gambar 4.6 ditunjukkan kondisi ketika player kalah bermain. Kemudian Gambar 4.7 ditunjukkan *history* dari aksi yang dilakukan selama pertandingan berlangsung.

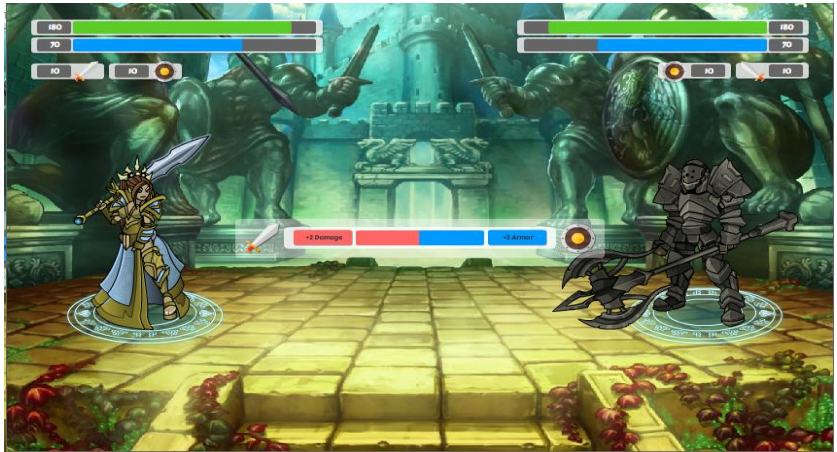


Gambar 4.2. Tampilan UI Pemilihan *Background Game*

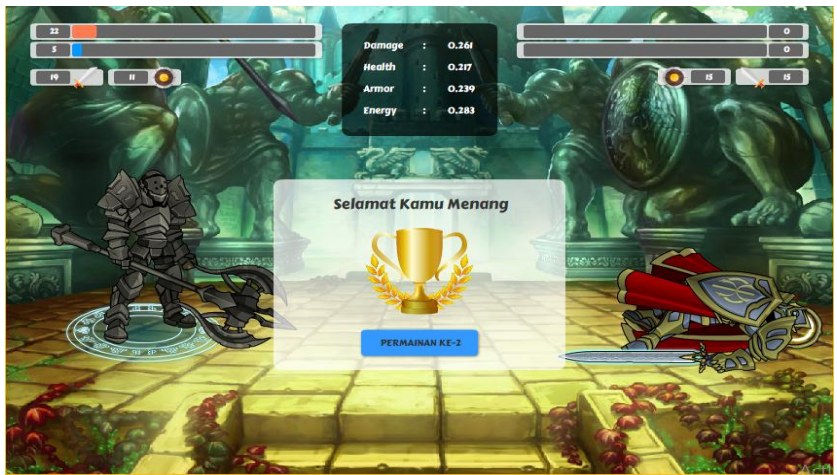


Gambar 4.3. Menu Utama *Game*

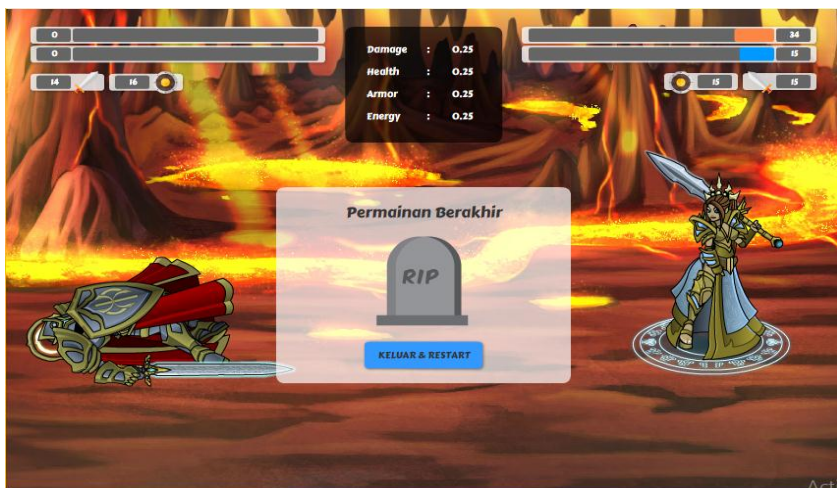




Gambar 4.4. Tampilan Menu Tambah *Damage and Armor*



Gambar 4.5 Tampilan Kondisi Ketika *Player* Memenangkan Permainan.



Gambar 4.6 Tampilan Kondisi Ketika *Player* Kalah Dalam Permainan.



Dataset	Total	Ultimate	Attack	Defense	Heal	Total	Damage	Armor
Main	0	0	0	0	0	0	0	0

Gambar 4.5. Tampilan *History* Hasil Aksi yang Dilakukan

#### 4.1.3.2 Hasil Pembuatan Sistem

Tahapan pembuatan sistem merupakan tahap dimana representasi dari hasil analisis kebutuhan yang dilakukan. Dalam hal ini hasil rancangan sistem berupa gambaran umum dari metode K-NN berbobot dan K-NN. Perbedaan dari kedua metode tersebut adalah letak penambahan bobot. Atribut yang digunakan dalam metode K-NN berbobot akan diberikan nilai bobot yang berbeda pada tiap atributnya, tergantung dari pengaruh atribut. Terdapat 4 atribut yang digunakan dalam pembuatan sistem ini, diantaranya adalah *Health*, *Energy*, *Damage*, dan *Armor*.

Pada proses awal metode K-NN data inputan yang digunakan adalah nilai dari *health*, *energy*, *damage* dan *armor*. Nilai yang diinput akan di proses oleh metode K-NN sehingga menghasilkan sebuah keputusan. Keputusan yang di dapatkan akan dicek terlebih dahulu apakah keputusan tersebut memenuhi jumlah energi. Jika dalam kondisi *attack* atau *ultimate* terpenuhi, akan di cek kembali apakah *enemy flag defense* bernilai *true* atau *false*. Jika nilai *true* terpenuhi maka akan mengurangi nilai *enemy health* sebesar *damage attack* atau *ultimate* di tambah *damage* dan dikurangi *enemy armor*. Jika nilai *enemy health* hasil operasi tersebut lebih kecil dari nol maka akan di normalisasi menjadi nol. Kemudian nilai energi juga akan dikurangi sebesar *cost attack*, dan mengubah *enemy flag defense* menjadi *false*. Jika tidak memenuhi maka nilai *enemy health* hanya akan dikurangi sebesar *damage attack* di tambah *damage* saja. Dan jika nilai *enemy*

*health* hasil operasi tersebut lebih kecil dari nol maka akan di normalisasi menjadi nol. Kemudian nilai energi juga akan dikurangi sebesar *cost attack* atau *ultimate*.

Setelah mengambil keputusan *attack* atau *ultimate*, selanjutnya dicek apakah nilai *damage* sebesar 15 atau nilai *armor* sebesar 15. Jika *damage* bernilai 15, maka akan hanya akan aktif tombol untuk menginput menambah *armor*. Sebaliknya, Jika *armor* bernilai 15, maka akan hanya akan aktif tombol untuk menginput menambah *damage*. Jika dalam kondisi nilai *damage* dan *armor* kurang dari 15, maka dapat menginput menambahkan *damage* atau *armor*.

Kondisi selanjutnya yaitu apakah keputusan tersebut adalah *defense* dan energi cukup untuk memenuhi *cost defense*, jika memenuhi maka *flag defense* akan dirubah menjadi *true* dan energi akan dikurangi sebesar *cost defense*. Jika semua kondisi diatas tidak memenuhi maka sistem akan langsung memutuskan *heal* dengan menambah energi sebesar 20 dan jika nilai energi melebihi 100 maka akan dinormalisasi menjadi 100.

## **4.2 Hasil Pengujian**

Pada sub-bab ini akan dijelaskan tentang hasil dari pengujian *game* yang telah dibuat. Hasil dari pengujian ini mencakup hasil pembuatan dataset yang digunakan dalam pengujian, hasil dari skenario pengujian yang telah dilakukan dan hasil dari analisis hasil uji coba yang telah didapatkan.

### **4.2.1 Hasil Pembuatan Dataset**

Dataset yang digunakan di dalam *game* ini telah ditentukan oleh peneliti yang didapat berdasarkan hasil uji coba. Dari hasil uji coba tersebut kemudian dilakukan pengujian terhadap dataset yang akan digunakan, hal ini dilakukan untuk mengetahui apakah data tersebut layak atau tidak digunakan sebagai dataset awal dalam pengujian. Dataset yang digunakan sebanyak 132 data dengan melihat 5 atribut yaitu, *health*, *energy*, *damage*, *armor*, dan *decision*.

Dataset yang telah dibuat selanjutnya akan dilakukan pengujian untuk melihat tingkat akurasi dari dataset yang akan digunakan. Pengujian dilakukan dengan menggunakan K-NN dengan metode *10-cross validation* menghasilkan akurasi sebesar 70,58%. Dataset awal yang akan digunakan dalam penelitian ditunjukkan pada Lampiran 1.

#### 4.2.2 Hasil Skenario Pengujian

Pada sub bab ini dilakukan pembahasan mengenai hasil uji coba untuk mengetahui berapa besar tingkat adaptif dari kecerdasan buatan yang digunakan. Ada 3 skenario uji coba yang dilakukan. Pertama menguji cobakan secara langsung *player* dengan metode K-NN berbobot. Kedua, menguji cobakan secara langsung *player* dengan metode K-NN. Ketiga, menguji cobakan secara langsung *player* dengan metode *random*. Pada masing-masing uji coba yang dilakukan akan memperhatikan seberapa banyak aksi yang dilakukan pada masing-masing *player*. Aksi yang dilakukan terdapat 4 poin yaitu, *ultimate*, *attack*, *define*, *heal*. Pengujian akan dilakukan sebanyak 50 kali percobaan pada masing-masing pertandingan. Berikut ditunjukkan Tabel 4.1 Hasil pengujian pada skenario antara *player* dengan *enemy* menggunakan K-NN berbobot, tabel 4.2 hasil pengujian pada skenario antara *player* dengan *enemy* menggunakan K-NN, dan tabel 4.3 hasil pengujian pada skenario antara *player* dengan *enemy* menggunakan metode *random*.

Tabel 4.1 Hasil Pengujian Skenario *Player* – K-NN Berbobot (*Enemy*)

No.	Hasil	Sisa_Health	Bobot				Aksi Enemy				Aksi Hero			
			AP	HP	DP	EP	U	A	D	H	U	A	D	H
1	m	11	0.25	0.25	0.25	0.25	5	4	2	10	7	3	0	12
2	m	43	0.261	0.217	0.239	0.283	5	3	1	10	7	1	2	10
3	m	49	0.272	0.184	0.228	0.316	4	4	2	8	7	2	0	10
4	m	78	0.283	0.151	0.217	0.349	3	5	3	10	8	1	1	12
5	m	23	0.294	0.118	0.206	0.382	3	15	13	29	8	15	5	33
6	m	10	0.305	0.085	0.195	0.415	3	13	8	24	2	12	11	24
7	k	0	0.316	0.052	0.14	0.44	3	11	13	28	1	12	13	29
8	k	0	0.316	0.052	0.14	0.44	2	10	9	20	4	8	10	20
9	k	0	0.316	0.052	0.14	0.44	4	9	4	16	5	7	1	17
10	m	68	0.316	0.052	0.14	0.44	3	6	6	14	8	2	1	13
11	m	74	0.25	0.25	0.25	0.25	3	6	5	13	7	3	1	12
12	m	8	0.261	0.217	0.239	0.283	9	8	5	22	14	4	1	21
13	m	39	0.272	0.184	0.228	0.316	3	6	3	11	6	4	0	11
14	m	64	0.283	0.151	0.217	0.349	4	5	4	14	8	1	2	12
15	k	0	0.294	0.184	0.206	0.316	5	9	7	20	7	6	2	23
16	m	24	0.294	0.184	0.206	0.316	5	4	2	11	7	2	0	11
17	m	22	0.305	0.085	0.195	0.415	6	5	9	21	8	7	3	21
18	m	21	0.316	0.052	0.14	0.44	5	4	5	14	8	2	0	13

No.	Hasil	Sisa_Health	Bobot				Aksi Enemy				Aksi Hero			
			AP	HP	DP	EP	U	A	D	H	U	A	D	H
19	m	16	0.338	0.118	0.162	0.382	4	9	12	24	6	10	7	25
20	k	0	0.371	0.085	0.129	0.415	7	3	0	15	5	10	4	14
21	m	33	0.371	0.085	0.129	0.415	3	7	4	13	6	5	0	12
22	m	39	0.382	0.052	0.118	0.448	3	9	8	17	7	7	1	18
23	m	14	0.393	0.019	0.107	0.481	6	3	2	11	14	2	1	11
24	k	0	0.404	0.008	0.096	0.492	3	9	7	17	7	6	1	17
25	k	0	0.404	0.008	0.096	0.492	5	5	2	16	6	5	1	15
26	k	0	0.404	0.008	0.096	0.492	8	3	4	16	8	3	2	15
27	k	0	0.404	0.008	0.096	0.492	6	6	4	17	9	3	1	16
28	k	0	0.404	0.008	0.096	0.492	5	8	6	19	9	5	0	19
29	m	20	0.404	0.008	0.096	0.492	6	7	7	20	11	2	1	19
30	m	16	0.404	0.008	0.096	0.492	7	4	3	16	8	2	3	14
31	m	26	0.25	0.25	0.25	0.25	5	7	7	18	5	7	5	18
32	k	0	0.261	0.217	0.239	0.283	4	9	6	18	4	10	3	19
33	k	0	0.261	0.217	0.239	0.283	6	4	2	12	5	3	0	14
34	m	35	0.261	0.217	0.239	0.283	7	3	1	14	5	5	3	13
35	m	1	0.283	0.183	0.21	0.317	5	9	9	24	4	11	6	25
36	m	51	0.294	0.14	0.206	0.36	3	10	8	20	5	7	7	21
37	m	13	0.305	0.107	0.195	0.393	6	8	11	29	3	12	11	28
38	k	0	0.316	0.074	0.184	0.426	4	9	9	20	4	6	12	20
39	m	5	0.316	0.074	0.184	0.426	5	7	7	18	4	10	2	20
40	k	0	0.327	0.041	0.173	0.459	5	5	20	26	6	6	10	27
41	m	54	0.327	0.041	0.173	0.459	4	10	4	23	5	5	11	21
42	m	10	0.338	0.008	0.162	0.492	4	10	8	23	4	9	8	23
43	k	0	0.338	0.008	0.162	0.492	6	7	8	22	5	6	8	21
44	k	0	0.338	0.008	0.162	0.492	5	6	2	14	5	5	2	13
45	k	0	0.338	0.008	0.162	0.492	8	2	4	17	6	2	7	15
46	k	0	0.338	0.008	0.162	0.492	7	7	4	17	6	4	4	16
47	m	22	0.338	0.008	0.162	0.492	6	2	4	12	8	2	0	12
48	k	0	0.338	0.008	0.162	0.492	7	3	3	15	5	5	3	14
49	k	0	0.338	0.008	0.162	0.492	5	6	1	13	7	3	0	12
50	k	0	0.338	0.008	0.162	0.492	8	0	3	13	9	0	0	12
<b>Win Rate Player</b>	<b><u>kalah : 21 = 42%</u></b>						<b><u>menang : 29 = 58%</u></b>							

Keterangan Tabel 4.1 :

m : hasil jika *player* menang terhadap *enemy*

k : hasil jika *player* kalah terhadap *enemy*

DP : *Damage Player*

HP : *Health Player*

AP : *Armor Player*

EP : *Energy Player*

U : *Ultimate*

A : *Attack*

D : *Damage*

H : *Heal*

Tabel 4.2 Hasil Pengujian Skenario *Player – K-NN (Enemy)*

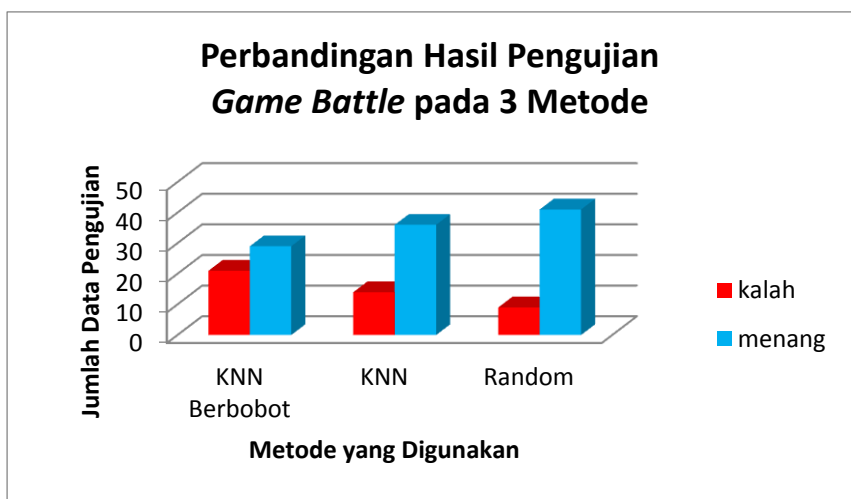
No.	Hasil	Sisa_Health	Aksi Enemy				Aksi Hero			
			U	A	D	H	U	A	D	H
1	m	54	6	3	0	10	7	3	0	10
2	m	72	4	6	1	20	6	6	1	20
3	k	0	3	9	2	25	9	10	5	24
4	m	77	7	1	2	12	6	4	1	12
5	m	49	2	4	3	22	3	5	3	22
6	k	0	5	9	8	14	6	8	8	26
7	m	57	6	8	11	23	3	9	11	29
8	m	72	7	0	0	9	7	1	0	9
9	m	38	4	3	0	19	8	3	0	19
10	m	66	5	3	0	14	7	4	0	14
11	k	0	9	11	7	20	9	11	5	24
12	m	100	4	9	2	25	10	9	0	27
13	m	44	1	7	4	15	6	4	3	15
14	m	73	3	6	3	10	6	3	3	9
15	m	47	6	4	12	18	5	6	2	20
16	k	0	4	1	12	19	6	13	1	17
17	k	0	8	9	14	24	8	12	6	13
18	m	80	8	4	4	25	3	8	0	26
19	m	46	3	6	8	10	9	8	9	12
20	m	37	3	5	2	14	9	4	6	14
21	m	83	3	9	4	16	2	5	3	16
22	m	18	3	8	2	14	4	8	2	14

No.	Hasil	Sisa_Health	Aksi Enemy				Aksi Hero			
			U	A	D	H	U	A	D	H
23	m	60	2	10	2	16	3	8	5	15
24	m	51	4	7	4	15	5	6	5	15
25	m	78	3	6	1	11	5	5	1	11
26	K	0	4	9	3	17	3	8	5	17
27	M	126	4	7	2	15	4	7	4	14
28	K	0	6	5	2	17	4	5	7	16
29	K	0	6	9	14	28	3	12	13	14
30	M	60	3	5	10	27	2	11	13	27
31	M	63	1	6	11	29	2	16	12	29
32	K	0	8	3	7	20	4	2	11	19
33	m	24	3	9	1	15	2	11	1	15
34	k	0	1	10	4	20	0	16	4	20
35	m	35	5	8	6	23	4	9	7	23
36	m	67	9	2	7	21	6	3	10	21
37	k	0	5	9	4	22	3	8	8	21
38	m	12	5	5	2	15	5	6	2	15
39	k	0	8	5	2	19	3	7	7	17
40	m	33	6	2	3	15	7	2	4	14
41	m	8	7	0	0	10	7	1	0	10
42	m	32	6	1	0	8	7	0	1	8
43	m	41	6	2	1	11	7	1	2	11
44	m	11	6	2	0	10	6	3	0	10
45	m	63	3	5	3	10	8	1	1	12
46	k	0	7	5	3	13	3	7	7	14
47	m	34	3	6	3	11	6	4	0	11
48	m	36	3	6	5	11	10	1	0	15
49	k	0	5	5	1	11	3	6	1	11
50	m	38	7	1	1	12	8	1	1	12
<b>Win Rate Player</b>			<b><u>kalah : 14 = 28%</u></b>				<b><u>menang : 36 = 72%</u></b>			

Tabel 4.3 Hasil Pengujian Skenario *Player – Random (Enemy)*

No.	Hasil	Sisa_Health	Aksi Enemy				Aksi Hero			
			U	A	D	H	U	A	D	H
1	m	60	5	3	11	16	5	8	4	19
2	k	0	4	6	19	26	3	12	14	25
3	m	74	4	7	18	25	2	16	11	26
4	m	77	4	7	13	21	5	11	7	23
5	m	49	3	9	9	19	5	11	4	21
6	k	0	7	7	14	26	2	15	8	29
7	m	57	4	6	11	19	4	13	3	21
8	m	72	3	6	9	16	4	11	2	18
9	k	0	3	14	1	19	4	7	8	17
10	m	66	4	6	14	21	5	9	9	23
11	m	121	2	7	14	16	10	5	20	12
12	m	100	2	4	15	17	6	11	0	22
13	m	44	4	6	4	13	7	5	1	15
14	m	73	3	5	9	13	6	7	2	16
15	m	47	2	9	8	16	5	11	1	19
16	k	0	7	5	3	13	3	7	7	14
17	k	0	8	3	8	19	5	9	4	20
18	m	80	3	4	11	14	9	4	2	18
19	m	46	5	4	9	15	6	8	2	18
20	m	37	6	3	10	16	6	9	2	19
21	m	100	3	3	8	10	9	2	0	14
22	m	124	3	6	5	11	10	1	0	15
23	m	140	3	5	3	10	8	1	1	12
24	m	132	3	4	5	13	14	3	0	16
25	m	168	1	6	5	12	10	1	0	15
26	m	109	4	2	5	9	8	0	2	11
27	m	126	5	1	7	15	16	0	4	19
28	m	106	3	3	8	12	13	0	0	16
29	m	132	2	3	10	9	10	0	1	14
30	m	60	5	3	11	16	5	8	4	19
31	k	0	6	7	8	22	5	8	9	21
32	m	81	3	5	6	12	7	5	0	15
33	m	75	1	7	9	12	7	6	0	17
34	m	116	2	4	20	17	8	10	2	24
35	m	20	3	7	15	22	8	11	3	26
36	k	0	5	10	12	25	7	14	2	29

No.	Hasil	Sisa_Health	Aksi Enemy				Aksi Hero			
			U	A	D	H	U	A	D	H
37	m	73	2	6	4	10	6	5	0	12
38	m	43	6	2	6	12	7	4	1	15
39	m	68	4	5	10	16	7	7	3	19
40	m	101	3	3	8	10	8	3	0	14
41	m	83	1	7	11	13	7	8	0	18
42	m	44	5	4	8	15	6	7	3	13
43	m	83	2	6	5	9	7	3	1	12
44	m	60	3	4	15	15	6	10	2	20
45	m	39	3	8	13	19	5	11	6	22
46	m	69	3	5	18	18	4	16	1	24
47	k	0	6	6	17	25	2	16	9	27
48	k	0	4	7	8	16	4	11	1	19
49	m	78	2	5	6	11	8	3	0	14
50	m	37	3	7	7	14	8	5	1	18
<i>Win Rate Player</i>	<b><u>kalah : 9 = 18%</u></b>					<b><u>menang : 41 = 82%</u></b>				



Gambar 4.6 Grafik Perbandingan Hasil Pengujian *Game Battle* pada 3 Metode yang Digunakan (Hasil *Player* Terhadap AI)

Data hasil uji coba di atas didapatkan melalui pengujian yang dilakukan dengan melibatkan 5 orang. Pada tabel 4.1 didapatkan hasil dari 50 kali ujicoba dengan metode K-NN berbobot *player* mendapatkan 29 kali kemenangan dan 21 kali kekalahan yang jika di persentase sebanyak 58% kemenangan dan 42% kekalahan. Pada tabel 4.1 juga menunjukkan bahwa nilai dari pembobotan K-NN



berbobot sudah adaptif. Telah dibuktikan bahwa pada perubahan nilai pembobotan pada setiap pertandingan sesuai dengan cara bermain *player*. Sebagai contoh pada data ke-1 *player* banyak melakukan *Heal* dan *Ultimate* sehingga pada atribut *Attack* dan *Energy* bertambah, sedangkan atribut *Defense* dan *Health* berkurang sesuai dengan perangnya dan penilaian bobot masing-masing sehingga pada permainan berikutnya *player* akan bermain dengan nilai bobot yang berbeda. Pada tabel 4.2 didapatkan hasil dari 50 kali ujicoba dengan metode K-NN *player* mendapatkan 36 kali kemenangan dan 14 kali kekalahan yang jika di persentase sebanyak 72% kemenangan dan 28% kekalahan. Sedangkan pada tabel 4.3 didapatkan hasil dari 50 kali ujicoba dengan metode *random*, *player* mendapatkan 41 kali kemenangan dan 9 kali kekalahan yang jika di persentase sebanyak 82% kemenangan dan 18% kekalahan. Rangkuman keseluruhan data hasil pengujian ditunjukkan pada Tabel 4.4. Selanjutnya, pada Gambar 4.6 juga ditunjukkan grafik hasil pengujian dari masing-masing metode.

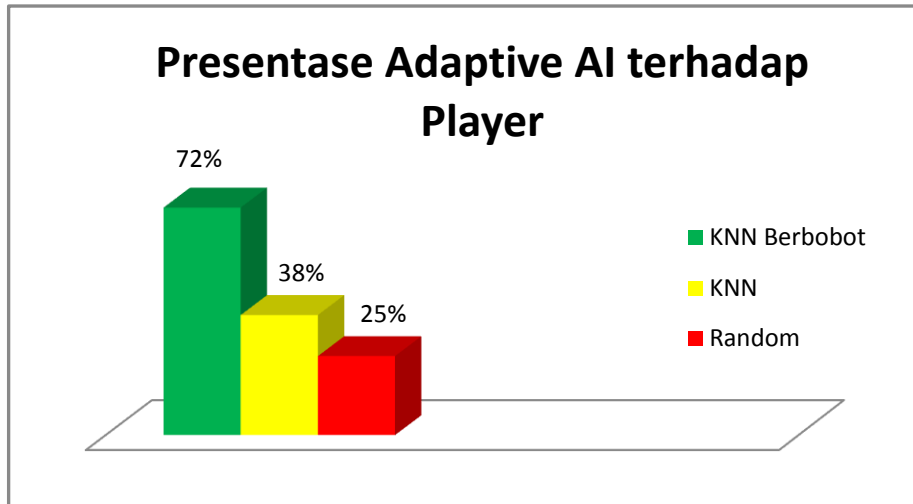
Tabel 4.4 Rangkuman Data Hasil Pengujian 3 Skenario

Pengujian <i>Player vs</i>	Hasil		Hasil (%)	
	Kalah	Menang	Kalah	Menang
<b>K-NN Berbobot</b>	21	29	42%	58%
<b>K-NN</b>	14	36	28%	72%
<b><i>Random</i></b>	9	41	18%	82%

Dari hasil uji coba di atas dengan melihat Tabel 4.4. dan Gambar 4.6 dapat diketahui bahwa rasio kekalahan dan kemenangan antara *player* dan *enemy* (AI) yang hampir seimbang (selisih paling kecil) adalah skenario pengujian pertama, yaitu *player* dengan metode K-NN berbobot. Sedangkan untuk rasio kekalahan dan kemenangan dengan selisih paling banyak ditunjukkan pada skenario pengujian yang ketiga, yaitu *player* dengan metode *random*.

Selanjutnya, dari data yang didapatkan dihitung persentase tingkat *adaptif* AI dengan menggunakan persamaan 3.4. Berikut Gambar 4.7 ditunjukkan hasil tingkat *adaptif* AI pada setiap skenario pengujian. Diasumsikan bahwa jika

semakin kecil selisih antara jumlah kekalahan dan kemenangan *player* terhadap AI maka semakin tinggi tingkat *adaptif game* tersebut.



Gambar 4.7 Grafik Persentase *Adaptive AI* terhadap *Player*

Pada Gambar 4.7 ditunjukkan bahwa tingkat adaptif AI yang paling tinggi dihasilkan pada skenario pengujian pertama yang ditandai dengan grafik berwarna hijau, yaitu antara *player* dengan metode K-NN berbobot sebesar 72%. Selanjutnya, tingkat adaptif AI posisi kedua dihasilkan pada skenario pengujian kedua, yaitu antara *player* dengan metode K-NN dan ditandai dengan grafik berwarna kuning sebesar 38%. Selanjutnya, tingkat adaptif AI paling rendah dihasilkan pada skenario pengujian ketiga, yaitu antara *player* dengan metode *random* yang ditandai dengan warna merah sebesar 25%.

### 4.3 Analisis Hasil Pengujian

#### 4.3.1 Analisis Hasil Pengujian Terhadap Tingkat *Adaptif AI*

Pada hasil pengujian yang ditunjukkan pada Tabel 4.4, selisih antara jumlah kekalahan dan kemenangan yang dihasilkan *player* terhadap *enemy* (AI) mampu menghasilkan tingkat *adaptif AI* yang lebih tinggi dibandingkan dengan 2 pengujian yang lainnya yaitu sebesar 72%. Dimana metode yang diusulkan yaitu K-NN berbobot. Metode tersebut memiliki titik berat pada pembobotan atribut yang dibuat adaptif. Setiap pembobotan atribut ditentukan dari aksi pada *game* tersebut. Awal pembobotan atribut adalah sama dengan jumlah keseluruhan adalah 1. Sehingga nilai bobot awal masing-masing atribut adalah 0.25 dari

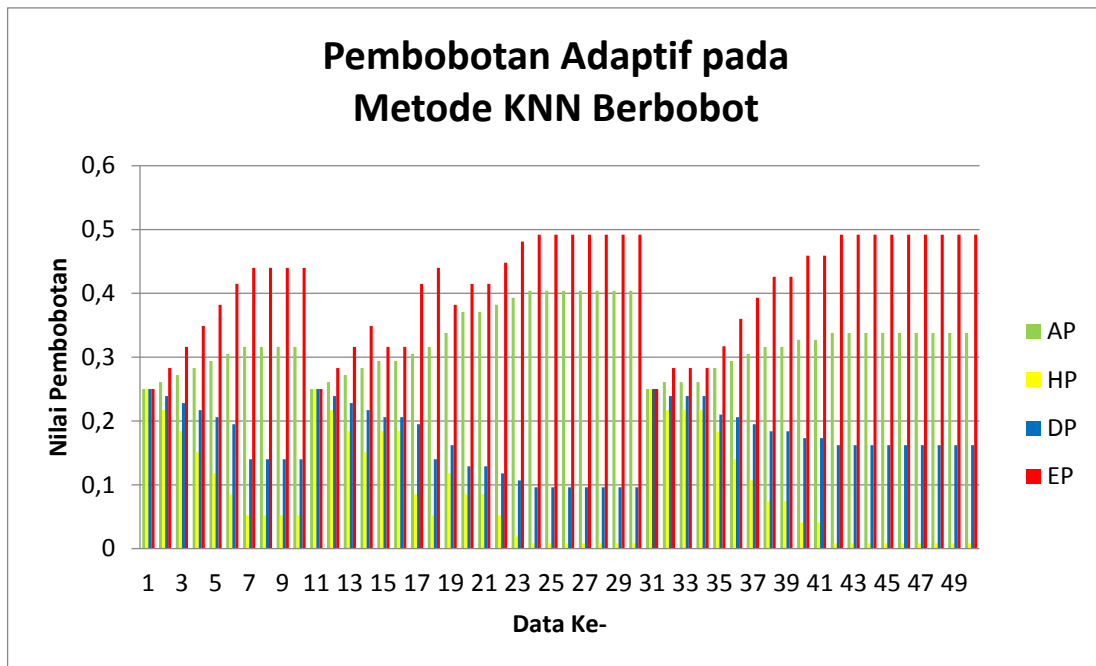
jumlah keseluruhan atribut ada 4. Dimana 4 atribut yang dilakukan perubahan bobot yaitu *energy*, *damage*, *armor*, dan *health*. Perubahan bobot masing-masing atribut ditentukan oleh aksi yang paling berpengaruh terhadap *player*. Perubahan bobot atribut akan terus berubah sampai mendapatkan nilai yang optimal. Selama bobot atribut tersebut berubah maka akan semakin banyak perilaku yang dipelajari oleh AI. Sehingga AI akan mampu menyesuaikan perilaku *player* dalam permainan. Jika nilai bobot atribut sudah mencapai nilai optimal, maka *player* akan mengalami kesulitan dan mengalami kekalahan. Nilai optimal yang dihasilkan pada tiap *game* berbeda tergantung dari permainan *player*.

Selanjutnya untuk skenario pengujian kedua, yaitu *player* dengan AI (K-NN). Dari hasil data pada Tabel 4.4 dan Gambar 4.7 AI masih kurang bisa beradaptasi dengan *player*, dimana tingkat adaptif AI hanya mencapai 38%. Pada dasarnya K-NN menganggap semua bobot atribut dalam permainan adalah sama, selain itu pengambilan keputusan terhadap aksi yang dilakukan hanya mempertimbangkan dari jarak yang terdekat sehingga aksi yang diambil oleh AI belum sepenuhnya akurat. Hal tersebut menjadi faktor rendahnya tingkat adaptif AI yang dihasilkan oleh metode K-NN.

Pada skenario pengujian terakhir, yaitu *player* dengan *random* tingkat adaptif yang dihasilkan hanya mencapai 25%. Pengambilan keputusan aksi yang diambil pada *enemy* yang *random* tidak berdasarkan apapun, hanya mempertimbangkan *energy* yang masih dimiliki pada *computer player (enemy)*.

#### **4.3.2 Analisis Hasil Pengujian Terhadap Pembobotan *Adaptif* pada Metode K-NN Berbobot**

Pada Tabel 4.1 selain ditunjukkan hasil win rate dari metode K-NN Berbobot juga ditunjukkan hasil pembobotan yang dihasilkan dari masing-masing atribut. Terdapat 4 atribut yang diukur yaitu AP (*ArmorPlayer*), DP (*Damage Player*), HP (*Health Player*), dan EP (*Energy Player*). Berikut Gambar 4.8 ditunjukkan grafik nilai pembobotan *adaptif* pada metode K-NN berbobot.



Gambar 4.8 Grafik Nilai Pembobotan adaptif pada metode K-NN berbobot

Pada Gambar 4.8 menunjukkan bahwa bobot EP (*Energy Player*) yang ditandai dengan grafik batang berwarna merah mengalami perubahan kenaikan dari bobot awal yang ditentukan yakni 0.25. Atribut AP (*Armor Player*) yang ditandai dengan grafik batang berwarna hijau juga terus mengalami perubahan kenaikan dari bobot awal, namun tidak lebih besar kenaikan nilainya jika dibandingkan dengan atribut EP. Selanjutnya untuk atribut DP (*Damage Player*) yang ditandai dengan grafik batang berwarna biru perubahan nilai bobot atribut mengalami penurunan, begitu juga dengan atribut HP (*Health Player*) yang ditandai dengan grafik batang berwarna kuning juga mengalami penurunan terhadap perubahan nilai bobot yang dihasilkan.

Perubahan nilai bobot yang mengalami kenaikan dan penurunan mengindikasikan adanya pengaruh atau tidaknya atribut tersebut dalam permainan. Semakin besar hasil bobot nilai yang dihasilkan atau perubahan nilai bobot mendekati nilai 0.5 maka bisa dikatakan bahwa atribut tersebut memiliki pengaruh besar terhadap aksi yang dilakukan selama permainan berlangsung. Begitu juga sebaliknya, semakin kecil nilai bobot yang dihasilkan atau perubahan nilai bobot mendekati nilai 0 maka bisa dikatakan bahwa atribut tersebut memiliki

pengaruh kecil terhadap aksi yang dilakukan. Oleh karena itu bisa dikatakan bahwa atribut EP dan AP memiliki pengaruh besar terhadap aksi yang dilakukan sebaliknya atribut DP dan HP kurang memiliki pengaruh terhadap aksi yang dilakukan selama permainan berlangsung.

*[Halaman ini sengaja dikosongkan]*

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan beserta saran untuk pengembangan lebih lanjut.

#### **5.1 Kesimpulan**

Berdasarkan hasil uji coba dan analisis yang dilakukan pada penelitian ini dapat disimpulkan beberapa hal.

- Metode K-NN berbobot dapat diterapkan dalam AI untuk meningkatkan kecerdasan adaptif pada game berjenis pertarungan.
- Dari analisa hasil pengujian dijelaskan bahwa nilai tingkat adaptif pada metode K-NN berbobot sebesar 72%, sedangkan nilai tingkat adaptif pada metode K-NN sebesar 38%, dan nilai tingkat adaptif pada metode *random* sebesar 28%. Dari ketiga metode yang telah diuji coba, K-NN berbobot memiliki nilai adaptif tertinggi. Hal ini menunjukkan bahwa K-NN berbobot berhasil meningkatkan kecerdasan adaptif *computer player* pada game berjenis pertarungan. Hal ini dikarenakan metode yang diusulkan memperhatikan pembobotan di setiap atribut.
- Dari pengujian yang dilakukan terhadap adaptif AI (K-NN berbobot) dapat dilihat bahwa nilai bobot dari keempat atribut yang digunakan yang memiliki pengaruh besar pada game pertarungan adalah atribut *energy* dan *damage*.

#### **5.2 Saran**

Pada penelitian ini, metode yang diusulkan untuk meningkatkan adaptif AI khususnya pada *game battle* secara umum sudah menghasilkan tingkat adaptif AI yaitu dengan metode K-NN berbobot dibandingkan dengan 2 metode lainnya. Salah satu penentu penting pada metode ini terletak pada pembobotan, keakuratan dalam menentukan nilai optimal pada pembobotan menjadi penentu penting dihasilkannya sebuah AI yang mampu beradaptasi dengan *player*. Sehingga *player* dapat tertarik pada *game* tersebut karena kesulitan yang didapatkan *player* akan menyesuaikan dengan kemampuan *player*. Oleh karena itu, pada penelitian

selanjutnya diharapkan mampu menentukan nilai optimal yang sesuai dari *game* yang dibuat. Untuk variasi *game* yang dibuat untuk penelitian baik yang berkaitan dengan teknologi maupun *genre game* agar dapat ditingkatkan sesuai dengan perkembangan jaman dan teknologi yang ada saat ini.



## DAFTAR PUSTAKA

B. Suits (1978) *The Grasshopper: Games, Life and Utopia*. Toronto: University of Toronto Press.

Brathwaite, B. and Schreiber, I. (2009) *Challenges for game designers*. 1st edn. Rockland, MA, USA: Course Technology.

Consalvo, M. (2009) 'There is No Magic Circle', *Games and Culture*, 4(4), pp. 408–417. doi: 10.1177/1555412009343575.

Dahlbom, A. (2004) 'An adaptive AI for real-time strategy games'.  
E.M. Avedon, B. S.-S. (1981) *The Study of Games*. New York: John Wiley & Sons, Inc.

Emigh, M. S. *et al.* (2014) 'Reinforcement Learning in Video Games using Nearest Neighbor Interpolation and Metric Learning', 32611(c), pp. 1–20. doi: 10.1109/TCIAIG.2014.2369345.

ESA Entertainment Software Association (2014) *Industry Facts - The Entertainment Software Association*.

Hamari, J. and Koivisto, J. (2013) 'Social motivations to use gamification: an empirical study of gamifying exercise', *Proceedings of the 21st European Conference on Information Systems SOCIAL*, (JUNE), pp. 1–12. doi: 10.1016/j.chb.2015.07.031.

Li, J. and Kendall, G. (2015) 'A hyper-heuristic methodology to generate adaptive strategies for games', *IEEE Transactions on Computational Intelligence and AI in Games*, PP(99), p. 1. doi: 10.1109/TCIAIG.2015.2394780.

Maroney, K. (2001) *My Entire Walking Life, The Games Journal | A Magazine About Boardgames*.

Mehta, M., Ram, A. and Onta, S. (2007) 'Artificial Intelligence for Adaptive Computer Games', (Ccl).

Muliawan, F. (2014) 'Enemy Speed Control on Shoot em Up Game with Fuzzy Takagi Sugeno Method', pp. 87–90.

Munajat, B. and Toto, A. (2014) 'Implementasi Adaptive Artificial Intelligence pada game Capture The Flag dengan metode Dynamic Scripting', 1(1), pp. 695–705.

Nakamichi, T. and Ito, T. (2016) 'Implementation and qualitative analysis of an adaptive computer Shogi program by producing seesaw *game*', *TAAI 2015 - 2015 Conference on Technologies and Applications of Artificial Intelligence*, pp. 453–460. doi: 10.1109/TAAI.2015.7407107.

Purkiss, B. and Khaliq, I. (2016) 'A study of interaction in idle *games* & perceptions on the definition of a *game*', *2015 IEEE Games Entertainment Media Conference, GEM 2015*. doi: 10.1109/GEM.2015.7377233.

Salen, K. and Zimmerman, E. (2004) *Rules of Play: Game Design Fundamentals*, *Nihon Ronen Igakkai zasshi. Japanese journal of geriatrics*. doi: 10.1093/intimm/dxs150.

Thawonmas, R. (2014) 'Deduction of Fighting-*Game* Countermeasures Using the k-Nearest Neighbor Algorithm and a *Game* Simulator', pp. 0–4.

Urh, M. *et al.* (2015) 'The Model for Introduction of Gamification into E-learning in Higher Education', *Procedia - Social and Behavioral Sciences*. Elsevier B.V., 197(February), pp. 388–397. doi: 10.1016/j.sbspro.2015.07.154.

Yannakakis, G. N. and Togelius, J. (2015) 'A Panorama of Artificial and Computational Intelligence in *Games*', *IEEE Transactions on Computational Intelligence and AI in Games*, 7(4), pp. 317–335. doi: 10.1109/TCIAIG.2014.2339221.

## LAMPIRAN 1

Lampiran ini berisi data dari hasil uji coba awal yang selanjutnya dijadikan dataset awal pada metode K-NN. Dataset awal yang digunakan sebanyak 132.

No	Health1	Health2	Energy	Damage	Armor	Aksi
1	200	200	100	10	10	U
2	180	180	70	12	10	U
3	160	156	40	14	10	A
4	148	142	20	15	10	A
5	135	127	0	15	11	H
6	135	127	15	15	11	H
7	135	127	30	15	11	U
8	107	97	0	15	13	H
9	107	97	15	15	13	H
10	107	97	30	15	13	U
11	79	67	0	15	15	H
12	79	67	15	15	15	H
13	79	67	30	15	15	U
14	51	37	0	17	13	H
15	51	37	15	17	13	H
16	51	37	30	17	13	U
17	23	3	0	19	11	H
18	23	3	15	19	11	H
19	23	3	30	19	11	A
20	190	190	80	11	10	U
21	170	168	50	13	10	U
22	146	142	20	15	10	A
23	132	127	0	15	11	H
24	132	127	15	15	11	H
25	132	127	30	15	11	H
26	132	127	45	15	11	H
27	132	127	60	15	11	H
28	132	127	75	15	11	H
29	132	127	90	15	11	U
30	102	97	60	15	13	D
31	102	97	50	15	14	H
32	102	97	65	15	14	U
33	86	81	35	15	15	H

34	86	81	50	15	15	A
35	71	66	30	16	14	U
36	43	34	0	18	12	H
37	43	34	15	18	12	H
38	43	34	30	18	12	U
39	190	200	90	10	11	D
40	179	200	80	10	12	D
41	169	200	70	11	12	A
42	169	189	50	12	12	H
43	169	189	65	12	12	U
44	169	177	35	14	12	U
45	145	149	5	15	12	H
46	145	149	20	15	12	H
47	145	149	35	15	12	H
48	145	149	50	15	12	H
49	145	149	65	15	12	H
50	145	149	80	15	12	U
51	145	119	50	15	14	U
52	145	89	20	15	15	D
53	145	89	10	16	14	H
54	145	89	25	16	14	D
55	145	89	15	17	13	H
56	145	89	30	17	13	H
57	128	89	45	17	13	A
58	113	85	25	18	12	A
59	98	67	5	19	11	H
60	98	67	20	19	11	H
61	98	67	35	19	11	H
62	98	67	50	19	11	H
63	98	67	65	19	11	U
64	68	29	35	21	9	A
65	51	8	15	22	8	H
66	51	8	30	22	8	U
67	190	200	90	11	10	U
68	168	178	60	13	10	U
69	142	152	30	15	10	U
70	142	122	0	15	12	H
71	142	122	15	15	12	H
72	142	122	30	15	12	D
73	142	122	20	15	13	H
74	142	122	35	15	13	U
75	125	105	5	15	15	H

76	125	105	20	15	15	H
77	125	105	35	15	15	H
78	125	105	50	15	15	A
79	110	90	30	16	14	U
80	78	58	0	14	16	H
81	78	58	15	14	16	D
82	78	58	5	13	17	H
83	78	58	20	13	17	H
84	78	58	35	13	17	H
85	78	58	50	13	17	H
86	78	58	65	13	17	U
87	61	45	35	15	15	U
88	23	15	5	17	13	H
89	23	15	20	17	13	A
90	190	180	70	12	10	U
91	170	156	40	14	10	U
92	146	128	10	15	10	H
93	146	128	25	15	10	H
94	146	128	40	15	10	H
95	146	128	55	15	10	H
96	146	128	70	15	10	U
97	118	98	40	15	12	A
98	103	83	20	15	13	A
99	88	68	0	15	14	H
100	88	68	15	15	14	H
101	88	68	30	15	14	U
102	58	38	0	15	15	H
103	58	38	15	15	15	H
104	58	38	30	15	15	U
105	28	8	0	17	13	H
106	28	8	15	17	13	H
107	28	8	30	17	13	A
108	200	200	90	11	10	D
109	200	200	80	12	10	H
110	200	200	95	12	10	U
111	200	187	65	14	10	D
112	200	187	55	15	10	H
113	200	187	70	15	10	D
114	186	187	60	15	11	U
115	160	168	30	15	13	U
116	130	138	0	15	15	H
117	130	138	15	15	15	H

118	130	138	30	15	15	H
119	130	138	45	15	15	H
120	130	138	60	15	15	U
121	100	108	30	17	13	U
122	70	74	0	19	11	H
123	70	74	15	19	11	H
124	70	74	30	19	11	H
125	70	74	45	19	11	H
126	70	74	60	19	11	H
127	70	74	75	19	11	H
128	70	74	90	19	11	D
129	70	74	80	20	10	H
130	70	74	95	20	10	U
131	48	48	65	22	8	U
132	12	4	35	24	6	A

## BIOGRAFI PENULIS



M. Ihsan Alfani Putera dilahirkan di Banjarmasin, Kalimantan Selatan pada tanggal 30 Agustus 1992 dan merupakan anak pertama dari dua bersaudara dari pasangan Taurus Alfani dan Maeda Siti Arbayah. Penulis menempuh pendidikan formal yaitu di TK Tumbuh Kembang Surabaya (1997-1999), SD Negeri 2 Tanjung (1999-2005), lalu melanjutkan pendidikan menengah pertama di SMP Negeri 3 Marabahan pada (2005-2008), dan pendidikan menengah atas di SMA Negeri 1 Marabahan (2008-2011). Penulis melanjutkan jenjang Diploma 3 jurusan Teknik Informatika Informatika di Politeknik Negeri Banjarmasin (2011-2014), dan Diploma 4 Jurusan Teknik Informatika di Politeknik Elektronika Negeri Surabaya (2014-2015). Setelah itu penulis melanjutkan jenjang Magister Jurusan Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya (2015-2018) .

Penulis mempunyai hobi yaitu bermain *game* (desktop dan mobile) dan berolahraga. Penulis dapat dihubungi melalui email yaitu: [m.ihsan.ap@gmail.com](mailto:m.ihsan.ap@gmail.com) atau [ihsanalfani@gmail.com](mailto:ihsanalfani@gmail.com).