



**AKSI PENYERANGAN *NON-PLAYER CHARACTER*
(NPC) MENGGUNAKAN METODE *NAÏVE BAYES*
PADA *SHOOTER GAME***

Skripsi

**diajukan sebagai salah satu persyaratan untuk memperoleh gelar
Sarjana Pendidikan Program Studi Pendidikan Teknik Informatika dan
Komputer**

Oleh
Edi Siswanto
NIM.5302415002

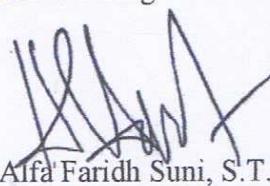
**PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2020**

PERSETUJUAN PEMBIMBING

Nama : Edi Siswanto
NIM : 5302415002
Program Studi : S-1 Pendidikan Teknik Informatika dan Komputer
Judul Skripsi : Aksi Penyerangan *Non-Player Character* (NPC) Menggunakan
Metode *Naïve Bayes* pada *Shooter Game*

Skripsi ini telah disetujui oleh pembimbing untuk diajukan ke panitia ujian
Skripsi Program Studi S-1 Pendidikan Teknik Informatika dan Komputer Fakultas
Teknik Universitas Negeri Semarang.

Semarang, Agustus 2020
Pembimbing



Alfa Faridh Suni, S.T., M.T.
NIP. 198210192014041001

LEMBAR PENGESAHAN

Skripsi dengan judul "Aksi Penyerangan Non-Player Character (NPC) Menggunakan Metode *Naïve Bayes* pada *Shooter Game*" telah dipertahankan dihadapan sidang Panitia Ujian Skripsi Fakultas Teknik UNNES, pada tanggal 19 Agustus 2020.

Disusun oleh :

Nama : Edi Siswanto

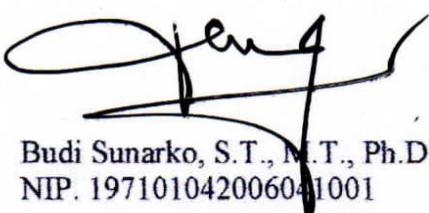
NIM : 5302415002

Panitia

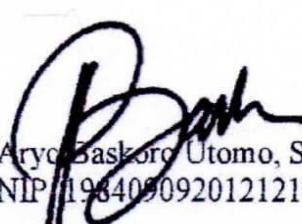
Ketua


Ir. Ulfah Mediaty Arief, M.T., IPM
NIP. 196605051997022001

Sekretaris


Budi Sunarko, S.T., M.T., Ph.D
NIP. 197101042006041001

Pengaji 1


Aryo Baskoro Utomo, S.T., M.T.
NIP. 198409092012121002

Pengaji 2


Anggraini Malwinda, S.T., M.Eng.
NIP. 197812262005012002

Pembimbing


Alfa Faridh Suni, S.T., M.T.
NIP. 198210192014041001

Mengetahui,

Dewan Fakultas Teknik UNNES



Dr. Nur Qudus, M.T., IPM
NIP. 196911301994031001

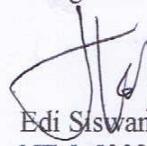
PERNYATAAN KEASLIAN TULISAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, magister, dan/atau doktor), baik di Universitas Negeri Semarang (UNNES) maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan Pembimbing dan masukan Tim Pengaji.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi ini.

Semarang,

Yang membuat pernyataan,


Edi Siswanto
NIM. 5302415002

MOTTO DAN PERSEMBAHAN

Motto :

- Barang siapa yang menghendaki dunia wajib atasnya dengan ilmu, barang siapa menghendaki akhirat maka wajib atasnya dengan ilmu dan barang siapa yang menghendaki kedua-duanya maka wajib atasnya dengan ilmu. (H. R Bukhari)
- *You Don't Have To Be Great To Start, But You Have To Start To Be Great.* –
(Zig Ziglar)

Persembahan :

- a. Terimakasih yang sebesar-besarnya kepada Tuhan Yang Maha Esa.
- b. Ibu, Bapak dan keluarga yang selalu mendukung saya sepenuhnya selama ini.
- c. Bapak Alfa selaku dosen pembimbing yang selalu memberikan semangat, arahan, dan dukungan yang maksimal.
- d. Teman-teman seperjuangan dari PTIK 2015.
- e. Semua pihak yang senantiasa memberikan doa, dukungan, dan semangat.

ABSTRAK

Edi Siswanto. 2020. Aksi Penyerangan *Non-Player Character* (NPC) Menggunakan Metode *Naïve Bayes* pada *Shooter Game*. Alfa Faridh Suni, S.T., M.T. Pendidikan Teknik Informatika dan Komputer. Universitas Negeri Semarang.

Aksi penyerangan pada *non-player character* (NPC) merupakan salah satu substansi penting dalam pembuatan *game*. Dalam melakukan penyerangan diperlukan strategi khusus agar NPC tidak mudah dikalahkan. Salah satu metode yang digunakan untuk mengatur penyerangan NPC adalah *rulebase*. Penggunaan *rulebase* dapat memberikan variasi serangan sesuai kondisi yang terjadi. Namun metode *rulebase* biasanya menghasilkan behaviour yang statis dan tidak adaptif jika terdapat kondisi baru. AI seperti ini akan mudah diprediksi dan repetitif sehingga menurunkan tingkat tantangan bermain *game*. Berdasarkan hal tersebut banyak peneliti yang menggunakan teknik *learning*. Salah satunya menggunakan metode *naïve bayes*. Tujuan dari penelitian ini adalah menerapkan metode *naïve bayes* sebagai strategi penyerangan NPC pada *shooter game*.

Metode *naïve bayes* digunakan untuk keputusan serangan yang diambil NPC. Adapun parameter yang digunakan untuk klasifikasi penyerangan adalah nyawa, jarak, jumlah granat, dan jumlah amunisi yang dimiliki NPC. Sedangkan klasifikasi penyerangan dibagi menjadi serangan tembak, serangan granat, dan serangan pisau. Pengujian dilakukan dengan menghitung nilai akurasi, presisi, dan *recall* untuk mengetahui kinerja NPC dalam memprediksi serangan serta pengujian tingkat kemenangan NPC jika menggunakan metode *naïve bayes* dibandingkan metode *rulebase*.

Hasil penelitian menunjukkan kinerja NPC mampu memprediksi serangan dengan benar jika terdapat kondisi baru dengan akurasi sebesar 91,67%, presisi 91,65%, dan *recall* 91,67%. Penerapan metode *naïve bayes* juga dapat meningkatkan kemenangan NPC dibanding penggunaan *rulebase*. Tingkat kemenangan NPC menggunakan metode *naïve bayes* sebesar 64% sedangkan *rulebase* sebesar 32%.

Kata Kunci: Aksi Penyerangan, *NPC*, *Naïve Bayes*, *Rulebase*.

KATA PENGANTAR

Segala puji dan syukur penulis ucapkan kehadirat Allah SWT yang telah melimpahkan rahmat-Nya sehingga penulis dapat menyelesaikan Skripsi yang berjudul “Aksi Penyerangan *Non-Player Character* (NPC) Menggunakan Metode *Naïve Bayes* pada *Shooter Game*”. Skripsi ini disusun sebagai salah satu persyaratan meraih gelar Sarjana Pendidikan. Penyelesaian skripsi ini tidak lepas dari bantuan berbagai pihak, oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih serta penghargaan kepada:

1. Dr. Nur Qudus, M.T.,IPM. selaku Dekan Fakultas Teknik Universitas Negeri Semarang.
2. Ir. Ulfah Mediaty Arief, M.T.,IPM. selaku Ketua Jurusan Teknik Elektro UNNES yang telah memberikan arahan dan motivasi selama menempuh studi.
3. Alfa Faridh Suni, S.T.,M.T. selaku dosen pembimbing yang penuh kesabaran dalam membimbing, memberi arahan, dan motivasi sehingga skripsi ini dapat selesai.
4. Aryo Baskoro Utomo, S.T., M.T. selaku penguji I dan Anggraini Mulwinda, S.T., M.Eng. selaku penguji II sekaligus dosen wali yang telah memberikan arahan dan masukkan berupa saran, ralat, perbaikan, dan komentar sehingga menambah kualitas nilai dari karya tulis ini.
5. Keluarga tercinta yang selalu memberi motivasi baik moral maupun material serta doa restu dalam menyelesaikan skripsi ini.

Penulis menyadari banyak keterbatasan yang dimiliki sehingga masih banyak kekurangan dan kesalahan, semoga karya tulis ini dapat dipahami sebaik-baiknya dan bermanfaat.

Semarang, September 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
Persetujuan Pembimbing	ii
LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	iv
MOTTO DAN PERSEMBERAHAN	v
ABSTRAK	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah	5
1.3 Batasan Masalah.....	6
1.4 Rumusan Masalah	6
1.5 Tujuan Penelitian.....	7
1.6 Manfaat Penelitian.....	7
BAB II KAJIAN PUSTAKA DAN LANDASAN TEORI	9
2.1 Kajian Pustaka.....	9
2.2 Landasan Teori.....	14

2.2.1 Permainan atau <i>Game</i>	14
2.2.2 <i>Shooter Game</i>	15
2.2.3 Kecerdasan Buatan.....	15
2.2.4 <i>Non-Player Character</i>	17
2.2.5 <i>Finite State Machine</i>	18
2.2.6 <i>Dataset</i>	19
2.2.7 <i>Naïve Bayes</i>	21
2.2.8 <i>Confusion Matrix</i>	25
BAB III METODOLOGI PENELITIAN.....	29
3.1 Waktu dan Tempat Penelitian	29
3.2 Desain Penelitian	29
3.2.1 Studi Literatur	30
3.2.2 Penentuan Parameter <i>Naïve Bayes</i>	31
3.2.3 Perancangan Simulasi <i>Game</i>	35
3.2.4 Pembuatan Simulasi <i>Game</i>	43
3.2.5 Pengujian Simulasi <i>Game</i>	44
3.2.6 Hasil dan Analisis.....	46
3.3 Alat dan Bahan Penelitian	46
3.4 Parameter Penelitian.....	47
3.5 Teknik Pengumpulan Data	47
3.6 Teknis Analisis Data	48
BAB IV HASIL DAN PEMBAHASAN	49
4.1 Hasil Penelitian	49

4.1.1 Hasil Simulasi <i>Game</i>	49
4.1.2 Hasil Pengujian Kinerja NPC.....	55
4.1.3 Hasil Tingkat Kemenangan NPC	62
4.1.3.1 Hasil Tingkat Kemenangan NPC pada Metode <i>Naïve Bayes</i>	62
4.1.3.2 Hasil Tingkat Kemenangan NPC pada <i>Rulebase</i>	64
4.1.3.3 Hasil Perbandingan Tingkat Kemenangan NPC pada Metode <i>Naïve Bayes</i> dan <i>Rulebase</i>	75
4.2 Pembahasan.....	76
4.2.1 Hasil Simulasi <i>Game</i>	76
4.2.2 Hasil Pengujian Simulasi <i>Game</i>	79
BAB V PENUTUP.....	84
5.1 Kesimpulan.....	84
5.2 Saran.....	85
DAFTAR PUSTAKA	86
LAMPIRAN	91

DAFTAR TABEL

Tabel 2.1 Contoh data <i>training</i>	20
Tabel 2.2 Contoh data <i>testing</i>	20
Tabel 2.3 <i>Confusion matrix multiclass</i> (Iskandar dan Suprapto, 2015).....	25
Tabel 3.1 Kriteria parameter nyawa.....	32
Tabel 3.2 Kriteria parameter jarak	34
Tabel 3.3 Kriteria parameter jumlah granat	34
Tabel 3.4 Kriteria parameter jumlah amunisi	35
Tabel 3.5 <i>Damage</i> setiap senjata.....	36
Tabel 3.6 Komponen pada <i>game</i>	42
Tabel 3.7 Alat penelitian	46
Tabel 3.8 Rekapitulasi hasil pengujian tingkat kemenangan NPC menggunakan metode <i>naïve bayes</i> dan <i>rulebase</i>	48
Tabel 4.1 Hasil pengujian data testing oleh sistem	55
Tabel 4.2 Nilai <i>confusion matrix</i>	59
Tabel 4.3 Hasil uji coba menggunakan metode <i>naive bayes</i>	62
Tabel 4.4 Hasil uji coba menggunakan metode <i>rulebase</i>	73
Tabel 4.5 Perbandingan hasil uji coba menggunakan metode <i>naïve bayes</i> dan <i>rulebase</i>	75
Tabel 4.6 Waktu komputasi perhitungan metode <i>naïve bayes</i>	77
Tabel 4.7 Perbandingan hasil dengan penelitian terdahulu.....	82

DAFTAR GAMBAR

Gambar 2.1 Model <i>game</i> AI	16
Gambar 2.2 Tahap perilaku NPC	17
Gambar 2.3 Hierarki gerak perilaku NPC.....	18
Gambar 2.4 Contoh alur FSM	19
Gambar 3.1 Diagram alir penelitian.....	30
Gambar 3.2 Medan arena <i>game</i>	32
Gambar 3.3 Blok diagram perilaku NPC	37
Gambar 3.4 Radius serang NPC	38
Gambar 3.5 <i>Flowchart naive bayes</i> pada NPC	40
Gambar 4.1 Tampilan awal <i>game</i>	49
Gambar 4.2 Penerapan data <i>training</i>	50
Gambar 4.3 Tampilan status NPC.....	51
Gambar 4.4 Tampilan serangan tembak NPC.....	52
Gambar 4.5 Tampilan serangan granat NPC.....	52
Gambar 4.6 Tampilan serangan pisau NPC	53
Gambar 4.7 Tampilan NPC kalah	54
Gambar 4.8 Tampilan NPC menang	54
Gambar 4.9 Perhitungan <i>naive bayes</i> pada sistem	57
Gambar 4.10 Perhitungan <i>naive bayes</i> pada sistem (lanjutan)	58
Gambar 4.11 Respon NPC ketika kondisi baru	58

DAFTAR LAMPIRAN

Lampiran 1 Daftar <i>asset game</i>	91
Lampiran 2 Rekapitulasi hasil klasifikasi manual oleh pemain.....	93
Lampiran 3 <i>Dataset</i> penelitian hasil <i>preprocessing</i>	100
Lampiran 4 Data <i>training</i>	104
Lampiran 5 Data <i>testing</i>	107
Lampiran 6 Daftar pemain uji coba <i>game</i>	109
Lampiran 7 Dokumentasi uji coba <i>game</i>	110
Lampiran 8 Formulir usulan topik skripsi.....	112
Lampiran 9 Surat penetapan dosen pembimbing skripsi	113
Lampiran 10 Formulir usulan judul skripsi.....	114
Lampiran 11 Surat tugas penguji seminar proposal.....	115
Lampiran 12 Daftar hadir peserta seminar proposal	116
Lampiran 13 Berita acara seminar proposal.....	117

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan kecerdasan buatan dari tahun ke tahun semakin meningkat. Penelitian yang dirilis oleh Accenture (NYSE: ACN) memprediksi bahwa kecerdasan buatan atau *Artificial Intelligence* (AI) dapat mencapai dua kali lipat pertumbuhannya hingga tahun 2035. Pemanfaatan kecerdasan buatan telah diimplementasikan ke dalam banyak bidang salah satunya dalam bidang *game*. Kecerdasan buatan memiliki peran penting untuk memberikan ketertarikan pengguna dan meningkatkan tantangan dalam bermain *game* (McGee dan Abraham, 2010). Rachman, *et al.*, (2019) dalam penelitiannya menyebutkan bahwa pengguna *game* di Indonesia pada tahun 2019 sebesar 34 juta dan diperkirakan tahun 2020 mencapai 100 juta. Menurut laporan *Newzoo*, pada tahun 2019 industri *game* global akan mencapai angka 2,15 kuadriliun, meningkat 9,6 % dibanding tahun sebelumnya (Wijman, 2019). Berdasarkan hal tersebut membuktikan bahwa pengguna *game* semakin meningkat dari tahun ke tahun dan memberikan prospek yang baik sebagai lapangan pekerjaan. Seiring bertambahnya pengguna *game*, maka harapan dan ekspektasi pengguna terhadap kualitas *game* juga terus berkembang (Aghlara dan Tamjid, 2011).

Salah satu *game* yang saat ini banyak dikembangkan adalah *shooter game* (Frosi dan Cristina, 2018; Syahputra, *et al.*, 2019). Dalam memainkan *shooter game* pemain diharuskan melakukan aksi cepat dalam merespon keadaan yang terjadi

(Frosi dan Cristina, 2018). Sehingga dalam memainkan *shooter game* memiliki tantangan tersendiri bagi pengguna. *Shooter game* memiliki *subgenre* yaitu *First Person Shooter* (FPS) dan *Third Person Shooter* (TPS). Perbedaan keduanya hanya terletak pada sudut pandang pemain. Pada *genre* FPS menggunakan sudut pandang orang pertama seolah-olah menjadi karakter utama, sedangkan TPS memakai sudut pandang orang ketiga sehingga bisa melihat karakter dari belakang (punggung) (Bimantika, 2017).

Salah satu komponen pada *shooter game* yang diterapkan AI adalah karakter *Non-Player Character* (NPC). NPC adalah objek atau karakter yang tidak dikendalikan oleh pemain tapi dijalankan oleh komputer (Liarokapis, *et al.*, 2014). Penerapan AI tersebut umumnya terdiri dari perencanaan *path*, menggunakan *item*, dan berperang (Hong dan Cho, 2005). Dalam berperang NPC biasanya berperilaku seperti mengejar, menghindar maupun menyerang lawan (Asmiatun dan Hendrawan, 2016).

Metode dasar kecerdasan buatan yang sering diterapkan untuk mengatur serangan pada NPC adalah metode *Finite State Machine* (FSM) (Frosi dan Cristina, 2018). FSM merupakan alur untuk menggambarkan perpindahan *state* perilaku NPC, seperti dari *state* mengejar ke *state* menyerang (Lee and Lee, 1998). Ada beberapa penelitian seperti Syahputra, *et al.*, (2019) yang menerapkan FSM untuk mengatur serangan pada NPC. NPC dapat melakukan serangan berupa tembakan jika berada pada jarak tertentu. Kusumah, *et al.*, (2018) juga menerapkan FSM sebagai model penyerangan pada NPC *zombie*. NPC *zombie* dapat menyerang pemain jika pada jarak tertentu. Pada penelitian tersebut FSM menghasilkan sebuah

kecerdasan buatan yang dapat membuat perilaku NPC menyerang sesuai aturan yang dibuat. Permasalahan yang muncul pada saat menyerang adalah bagaimana membuat NPC agar tidak mudah dikalahkan. Tanpa adanya strategi khusus dalam melakukan serangan, NPC akan mudah dikalahkan. Ada beberapa penelitian yang dilakukan pada NPC dalam perilaku penyerangannya, hasil penelitian menunjukkan NPC tanpa adanya strategi dalam melakukan serangan akan mudah dikalahkan oleh pemain (Ahmad, *et al.*, 2018).

Meninjau permasalahan tersebut, beberapa inovasi telah dilakukan pada agen NPC dari segi *behaviour* sebagai solusi untuk menciptakan NPC yang lebih agresif. Salah satunya adalah memberikan variasi serangan pada agen NPC (Wicaksono, *et al.*, 2013). Adanya variasi serangan pada NPC membuat agen NPC lebih agresif dan sulit dikalahkan (Majid, *et al.*, 2010).

Yunanta (2017) melakukan penelitian dengan menerapkan FSM dan *rulebase* untuk mengatur serangan NPC. FSM digunakan untuk perpindahan *state* perilaku NPC sedangkan *rulebase* untuk menentukan keputusan serangan yang diambil sesuai kondisi NPC. Pada penelitian tersebut NPC dapat melakukan keputusan penyerangan dari jarak dekat maupun dari jarak jauh sesuai kondisi yang telah ditentukan. *Rulebase* merupakan teknik untuk merancang *behaviour* agen AI melalui kombinasi aturan atau *rule* yang dimodelkan dengan kalimat *if-then* (Roastiningsih, *et al.*, 2013). Namun metode *rulebase* biasanya menghasilkan *behaviour* yang statis dan tidak adaptif jika terdapat kondisi baru. AI seperti ini akan mudah diprediksi dan repetitif sehingga menurunkan tingkat tantangan bermain *game* (Wang dan Tang, 2015). Selain itu, penerapan *rulebase* sebagai

kecerdasan buatan memberikan tingkat kemenangan NPC yang rendah yaitu sebesar 22% sampai 25% (Abdi, *et al.*, 2017; Putera dan Herumurti, 2018).

Berdasarkan hal tersebut banyak peneliti yang memanfaatkan teknik *learning* sebagai kecerdasan buatan. Teknik *learning* memungkinkan NPC dapat belajar dari data atau pengalaman yang sudah ada. Dengan demikian memungkinkan NPC adaptif terhadap kondisi baru yang terjadi. Beberapa penelitian menunjukkan bahwa teknik *learning* memberikan kecerdasan buatan yang baik pada *game* (Abdi, *et al.*, 2017; Putera dan Herumurti, 2018; Sanjaya, *et al.*, 2019). Teknik *learning* tersebut diterapkan menggunakan algoritma klasifikasi seperti *K-Nearest Neighbor* (K-NN), *decision tree*, *Support Vector Machine* (SVM), dan *naïve bayes*. Dari beberapa algoritma klasifikasi tersebut metode *naïve bayes* relatif lebih unggul (Ashari, *et al.*, 2013; F.Osisanwo, *et al.*, 2017).

Metode *naïve bayes* merupakan klasifikasi probabilitas yang menggunakan teorema bayes dengan asumsi bahwa fitur-fitur yang mendeskripsikan objek secara statistik tidak terikat satu sama lain atau independen (Wang dan Shang, 2017). Salah satu contoh penelitian Sanjaya, *et al.*, (2019) yang menerapkan metode *naïve bayes* sebagai AI pada *racing game* untuk keputusan penggereman (*braking decision*). Hasil penelitian menunjukkan metode *naïve bayes* mampu memberikan keputusan penggereman dengan baik hasil dari pembelajaran data *training*. Pada penelitian tersebut dijelaskan bahwa metode *naïve bayes* memiliki kelebihan cepat dalam memproses data serta tidak memakan *resources* komputasi yang besar. Selain itu metode *naïve bayes* dapat dilatih dengan *dataset* kecil (Kaviani dan Dhotre, 2017).

Meninjau latar belakang tersebut diperlukan penerapan kecerdasan buatan yang baik pada *shooter game* agar NPC lebih cerdas dalam melakukan serangan sehingga membuat *game* lebih menarik dan menantang. Dalam penelitian ini metode *naïve bayes* diusulkan untuk diterapkan sebagai kecerdasan buatan dalam aksi penyerangan NPC pada *shooter game*. Metode *naïve bayes* digunakan sebagai strategi untuk pengambilan keputusan penyerangan berdasarkan parameter-parameter tertentu. Selain itu akan dihitung pula tingkat kemenangan NPC dalam strategi penyerangan jika menggunakan metode *naïve bayes* dibandingkan menggunakan metode *rulebase*. Penerapan metode *naïve bayes* diharapkan efektif dapat dijadikan kecerdasan buatan NPC pada *shooter game* dengan performa akurat dan efisien.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang yang telah dipaparkan, masalah yang muncul adalah sebagai berikut:

1. NPC mudah dikalahkan tanpa adanya strategi khusus dalam melakukan serangan.
2. Penerapan *rulebase* biasanya menghasilkan *behaviour* yang statis dan tidak adaptif jika terdapat kondisi baru.
3. Tingkat kemenangan NPC menggunakan *rulebase* masih rendah yaitu 22% sampai 25%.
4. Diperlukan metode kecerdasan yang baik untuk mengatur penyerangan NPC seperti metode *naïve bayes*.

1.3 Batasan Masalah

Dalam penelitian skripsi ini diberikan batasan masalah agar tidak meluas dan penjelasannya menjadi lebih terarah. Berikut batasan masalah penelitiannya:

1. Simulasi *game* ini akan didesain dengan 3 dimensi (*3D*).
2. Pembuatan simulasi *game* menggunakan *game engine Unity 3D*.
3. Jenis *game* yang dibuat simulasi adalah *shooter game*.
4. Metode yang digunakan dalam mengatur serangan NPC adalah *naïve bayes*.
5. Parameter yang digunakan untuk menghitung *naïve bayes* adalah nyawa, jarak, jumlah granat, dan jumlah amunisi dari NPC.
6. *Dataset* penelitian ini diperoleh dari data kondisi NPC yang berisi nilai-nilai parameter, kemudian akan diberikan klasifikasi serangan NPC secara manual oleh *user*.
7. Klasifikasi aksi penyerangan dibagi menjadi tiga yaitu serangan tembak, serangan granat, dan serangan pisau.

1.4 Rumusan Masalah

Dari uraian latar belakang masalah, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana membuat strategi penyerangan NPC menggunakan metode *naïve bayes* pada *shooter game*?
2. Apakah metode *naïve bayes* dapat memberikan tingkat kemenangan NPC yang lebih baik dalam strategi penyerangan NPC dibandingkan menggunakan *rulebase*?

1.5 Tujuan Penelitian

Berdasarkan rumusan masalah, maka tujuan yang hendak dicapai pada penelitian ini adalah:

1. Membuat strategi penyerangan NPC berupa pengambilan keputusan penyerangan secara otonom berdasarkan perubahan kondisi yang telah diklasifikasikan menggunakan metode *naïve bayes*.
2. Mengetahui perbandingan tingkat kemenangan NPC dalam strategi penyerangan menggunakan metode *naïve bayes* dengan metode sebelumnya (*rulebase*) pada *shooter game*.

1.6 Manfaat Penelitian

Manfaat dari penelitian ini antara lain:

1. Manfaat Teoritis

Hasil penelitian ini diharapkan menjadi dokumen akademik sebagai acuan bagi penelitian yang akan datang khususnya dalam bidang *game* dan kecerdasan buatan.

2. Manfaat Praktis

- a. Dapat memberikan gambaran tentang penerapan kecerdasan buatan menggunakan metode *naïve bayes* sebagai strategi penyerangan NPC pada *shooter game* agar lebih menarik dan menantang.
- b. Dapat menambah wawasan dan kreativitas dalam menyelesaikan permasalahan dalam strategi penyerangan pada NPC menggunakan teknik

learning dengan memberikan inovasi terhadap penerapan suatu model AI yang digunakan berdasarkan keilmuan yang dimilikinya.

BAB II

KAJIAN PUSTAKA DAN LANDASAN TEORI

2.1 Kajian Pustaka

Dalam suatu penelitian diperlukan adanya kajian dari penelitian yang sudah ada untuk dijadikan referensi metode, cara, dan hasil penelitian untuk memberikan wacana dalam bentuk perbaikan dan pembaharuan yang lebih inovatif. Berikut adalah penelitian terdahulu yang berkaitan dengan penelitian ini.

Asmiatun, *et al.*, (2013) melakukan penelitian dengan menerapkan strategi penyerangan NPC menggunakan metode *naïve bayes*. Teknik *learning* pada metode *naïve bayes* memungkinkan NPC dapat belajar dari data atau pengalaman yang sudah ada. Dalam melakukan penyerangan, perilaku NPC diklasifikasikan menjadi beberapa variasi seperti mengecoh, menerkam, dan menggigit. Namun penelitian tersebut belum diimplementasikan pada *tools game* sehingga belum diketahui realitas strategi penyerangan NPC.

Asmiatun dan Hendrawan (2016) melakukan penyempurnaan pada penelitian sebelumnya (Asmiatun, *et al.*, 2013) dengan mengimplementasikan strategi penyerangan NPC pada *tools game engine Unity3D*. Adapun klasifikasi perilaku NPC adalah memukul dan menggigit. Sedangkan parameter untuk melakukan klasifikasi penyerangan adalah kekuatan serang, nyawa, dan jarak. Penelitian tersebut mampu membuat NPC menyerang jika terdapat kondisi baru. Namun strategi penyerangan hanya sebatas jarak dekat sehingga NPC hanya bisa

menyerang dari jarak dekat saja. Selain itu pada penelitian tersebut penerapan metode *naïve bayes* belum diketahui tingkat kemenangan NPC.

Penelitian oleh Fathurochman, *et al.*, (2014) menggunakan metode *naïve bayes* untuk keputusan penyerangan NPC. Penerapan metode tersebut diimplementasikan pada *game turn based*. Untuk menghitung metode *naïve bayes* maka dibutuhkan suatu parameter. Adapun parameter yang digunakan untuk menghitung probabilitas keputusan aksi penyerangan adalah kekuatan serang dan kekuatan bertahan dari karakter NPC. Sedangkan aksi penyerangan dibagi menjadi tiga, yaitu serangan tembak, serangan granat, dan serangan pisau. Pada penelitian tersebut NPC mampu merespon secara otonom jika terjadi perubahan keadaan dari parameter tersebut. Namun penelitian tersebut penerapan metode *naïve bayes* belum diketahui tingkat kemenangan NPC.

Abdi, *et al.*, (2017) melakukan penelitian untuk membandingkan kecerdasan buatan pada *game battle RPG*. Adapun metode yang digunakan untuk perbandingan adalah *Support Vector Machine* (SVM), *decision tree*, dan *rulebase*. Pengujian dilakukan dengan melakukan pertandingan *battle* antara NPC melawan NPC untuk masing-masing metode. Hasilnya metode SVM memiliki tingkat kecerdasan paling tinggi yaitu 72,5%. Sedangkan metode *decision tree* dan *rulebase* secara berurutan memiliki tingkat kecerdasan 50% dan 25%. Namun performa metode SVM sangat bergantung pada banyaknya data *training*.

Yunanta (2017) melakukan penelitian pada *shooter game* menggunakan metode FSM dan *rulebase*. FSM digunakan untuk perpindahan *state* perilaku NPC sedangkan *rulebase* digunakan untuk keputusan serangan NPC. *Rulebase* dibuat

sesuai *fuzzy rule* dengan parameter nyawa dan jarak. Sedangkan variasi serangan NPC dibagi menjadi pukul dekat, pukul jauh, dan tembak jauh. Pada penelitian tersebut NPC dapat menyerang sesuai kondisi sesuai parameter tersebut.

Lample dan Chaplot (2017) melakukan penelitian yang berjudul “*Playing FPS Games with Deep Reinforcement Learning*”. Teknik *learning* yang digunakan adalah *reinforcement learning*. Dalam bermain *game* pasti terdapat *agent* seperti musuh datang tiba-tiba atau sesuatu yang tidak pernah ada sebelumnya. Penelitian tersebut metode *deep reinforcement learning* digunakan untuk meningkatkan kecepatan dan kinerja NPC dalam mengenali lingkungan *game*. Hasilnya metode tersebut dapat meningkatkan kecepatan dan kinerja NPC serta pengalaman dalam mengenali lingkungan *game*.

Penelitian Putera dan Herumurti (2018) menerapkan teknik *learning* menggunakan metode *K-Nearest Neighbor* (K-NN) berbobot. Metode tersebut diimplementasikan pada *game* pertarungan (*fighting*). NPC dapat melakukan aksi seperti *heal*, *attack*, *defence*, dan *ultimate*. Sedangkan atribut yang digunakan yaitu *energy*, *damage*, *armor*, dan *health*. Masing-masing atribut berpengaruh terhadap aksi yang dilakukan. Pembobotan dilakukan untuk memberikan pengaruh setiap atribut dengan bobot disesuaikan dengan aksi pemain. Dari hasil evaluasi yang dilakukan terhadap 50 kali pertandingan, metode K-NN berbobot mampu menghasilkan tingkat kemenangan NPC sebesar 51%. Sedangkan metode sebelumnya yaitu K-NN tanpa bobot hanya menghasilkan tingkat kemenangan NPC sebesar 38% dan metode *rulebase* sebesar 25%.

Qaimkhani, *et al.*, (2018) melakukan penelitian yang berjudul “*Game Based Learning Using First Person Person – Battle of IICT*”. Pada penelitian tersebut dikembangkan *shooter game 3D* seperti *game* “*counter strike*” dengan menggunakan *game engine Unity*. Metode yang digunakan untuk menentukan tindakan apa yang dilakukan oleh NPC adalah berbasis *finite state machine* (FSM). NPC mampu berperilaku seperti *walk*, *run*, dan *shoot*. Namun penyerangan NPC hanya sebatas menembak saja, belum terdapat variasi serangan.

Frosi dan Cristina (2018) melakukan penelitian yang berjudul “*Building Bots for Shooter Game based on the Bartle’s Player Types and Finite State Machine: A Battling Behaviour Analysis*” pada penelitian tersebut NPC dibuat menjadi beberapa tipe perilaku yaitu *killer*, *achiever*, *explorer*, dan *beginner*. Setiap tipe perilaku NPC dimodelkan dengan diagram *finite state machine* (FSM). Tujuan dibuatnya beberapa tipe perilaku tersebut adalah untuk memberikan kesan pengalaman pemain. Dari hasil penelitian tersebut dikatakan bahwa perlunya pengaturan perilaku NPC agar bisa adaptif dan diharapkan ada inovasi lain dalam perilaku NPC.

Penelitian lain pada *shooter game* yang menerapkan FSM sebagai kecerdasan buatan pada NPC adalah penelitian oleh Syahputra, *et al.*, (2019) dengan judul “*Historical Theme Game Using Finite State Machine for Actor Behaviour*”. Kecerdasan buatan yang diterapkan pada NPC adalah metode FSM. Penerapan FSM membuat NPC mampu melakukan penyerangan sesuai aturan yang dibuat. Namun penyerangan NPC hanya bisa menembak saja, belum ada variasi serangan.

Sanjaya, *et al.*, (2019) melakukan penelitian berjudul “Penerapan *Naïve Bayes* untuk *NPC Braking Decision* pada *Racing Game*”. Pada penelitian tersebut metode yang digunakan adalah *naïve bayes*. Perhitungan *naïve bayes* menggunakan tiga fitur untuk masukan dan dua kelas keluaran yang data latihnya diperoleh dari pemain. Tiga fitur tersebut yaitu kecepatan, jarak dengan tikungan, dan sudut tikungan. Sedangkan kelas keluaran bernilai 0 (jika terjadi penggereman) dan kelas 1 (jika tidak terjadi penggereman). Hasil pengujian menunjukkan bahwa hasil *braking decision* dari *naïve bayes* mampu memberikan keputusan penggereman dengan baik tanpa menabrak dinding hasil dari pembelajaran data *training*.

Berdasarkan beberapa penelitian tersebut dapat disimpulkan bahwa teknik *learning* seperti metode K-NN, SVM, dan *naïve bayes* dapat dijadikan kecerdasan buatan untuk perilaku NPC dengan performa yang baik. Pada penelitian Sanjaya, *et al.*, (2019) dijelaskan bahwa metode *naïve bayes* memiliki kelebihan cepat dalam memproses data serta tidak memakan *resources* komputasi yang besar. Pada penelitian Asmiyatun dan Hendrawan (2016) penerapan metode *naïve bayes* hanya untuk penyerangan NPC pada jarak dekat sehingga NPC hanya bisa menyerang dari jarak dekat saja. Sementara penelitian Fathurochman, *et al.*, (2014) metode *naïve bayes* diterapkan pada *shooter game turn base*. Penyerangan NPC bisa dilakukan pada jarak dekat maupun dari jarak jauh. Namun penelitian keduanya, baik Asmiyatun dan Hendrawan (2016) serta Fathurochman, *et al.*, (2014) belum diketahui tingkat kemenangan NPC seperti penelitian yang dilakukan oleh Abdi, *et al.*, (2017) dan Putera dan Herumurti (2018). Maka dalam penelitian ini akan menerapkan metode *naïve bayes* sebagai strategi penyerangan NPC seperti

penelitian yang dilakukan oleh Asmiatun dan Hendrawan (2016) dengan memberikan aksi serangan seperti penelitian Fathurochman, *et al.*, (2014) agar NPC dapat menyerang dari jarak dekat maupun dari jarak jauh. Selain itu akan dihitung pula tingkat kemenangan NPC untuk mengetahui persentase kecerdasan buatan menggunakan metode *naïve bayes* dibanding menggunakan metode yang sudah ada yaitu *rulebase*.

2.2 Landasan Teori

2.2.1 Permainan atau *Game*

Game memiliki beberapa definisi diantaranya: *game* adalah sebuah kegiatan yang melibatkan pengguna atau *user* yang memiliki sebuah tujuan dan aturan tertentu (Suits, 2005). Definisi lain *game* adalah aktivitas yang dapat menarik minat pemain sekaligus sebagai media hiburan (*entertainment*) (Maroney, 2001). Dari beberapa definisi tersebut dapat diartikan bahwa *game* adalah kegiatan permainan sebagai media hiburan yang memiliki aturan-aturan tertentu serta terdapat tujuan yang harus dicapai. Dalam sebuah *game* terdapat pertempuran antara pemain dan pemain lain atau antara pemain dengan objek lain dari bagian sistem *game*. Dengan begitu akan menimbulkan tantangan bagi pemain itu sendiri saat memainkan *game*. Tantangan tersebut akan menjadi penghalang bagi pemain untuk mencapai misi tertentu. Selain sebagai media hiburan *game* juga dapat melatih kecerdasan seseorang karena terdapat berbagai jenis *game* yang menuntut pemain untuk berpikir kritis seperti *game* kuis, teka-teki silang, dan *game* catur.

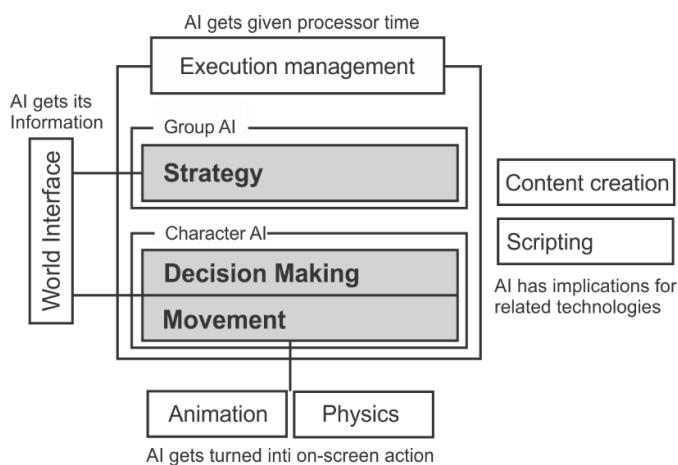
2.2.2 *Shooter Game*

Istilah *shooter game* mengacu pada permainan senjata api dimana pemain harus mengendalikan pemain dan menghancurkan musuh (Frosi dan Cristina, 2018). *Shooter game* memiliki *subgenre* yaitu *first person shooter* (FPS) dan *third person shooter* (TPS). Perbedaan keduanya hanya terletak pada sudut pandang pemain. Pada *genre* FPS menggunakan sudut pandang orang pertama seolah-olah menjadi karakter utama, sedangkan TPS memakai sudut pandang orang ketiga sehingga bisa melihat karakter dari belakang (punggung) dan terdapat kursor pemain sedikit di atas mereka. (Bimantika, 2017). TPS memberikan permainan yang cenderung lebih realistik, tidak hanya grafis tapi juga dari segi *gameplay*. TPS memungkinkan pemain untuk melihat wilayah sekitarnya dengan avatar yang lebih jelas (Bimantika, 2017). Dalam memainkan *shooter game* pemain diharuskan bermain cepat dan tepat dalam merespon keadaan yang terjadi seperti dalam melakukan tembakan harus tepat sasaran dan perlu taktik khusus dalam melakukan serangan dengan memaksimalkan jumlah senjata yang dimiliki.

2.2.3 Kecerdasan Buatan

Kecerdasan buatan atau lebih dikenal *Artificial Intelligence* (AI) merupakan salah satu unsur yang berperan penting dalam pembuatan sebuah *game*. AI diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dilakukan manusia. Menurut Millington dan Funge (2009:4) AI merupakan ilmu tentang bagaimana membangun sistem komputer yang memiliki kecerdasan seperti kecerdasan manusia. Teknologi AI telah berkembang

tidak hanya di bidang kesehatan dan bidang industri saja tetapi sudah banyak diimplementasikan pada bidang *game*. *Game* yang diterapkan AI biasanya disebut *game AI*. Millington dan Funge (2009) dalam bukunya yang berjudul “*Artificial Intelligence for Games*”, Konsep *game AI* memiliki struktur dan model seperti ditunjukkan pada gambar 2.1.



Gambar 2.1 Model *game AI*

Sumber : Millington dan Funge (2009:9)

Pada gambar 2.1 secara garis besar pembagian model *game AI* dibagi menjadi tiga yaitu strategi (*strategy*), pengambilan keputusan (*decision making*), dan pergerakan (*movement*).

1. Strategi (*strategy*)

Model strategi mengacu pada pendekatan keseluruhan oleh sekelompok karakter. *Game AI* yang menerapkan model ini biasanya *game* berbasis strategi seperti *Clash of Clans* (COC), *game* catur, dan *Plants vs Zombies*.

2. Pengambilan Keputusan (*Decision Making*)

Model AI pengambilan keputusan biasanya terdapat pada karakter yang memiliki serangkaian perilaku berbeda yang bisa mereka pilih untuk dilakukan.

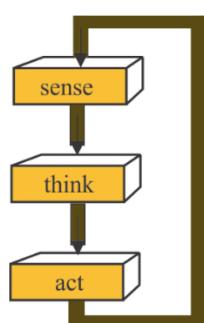
Sistem pengambilan keputusan akan mencari tahu perilaku yang paling tepat pada saat kondisi tertentu.

3. Pergerakan (*Movement*)

Pergerakan pada *game AI* mengacu pada algoritma yang digunakan. Algoritma tersebut akan mengubah keputusan menjadi semacam gerakan. Salah satu algoritma yang sering digunakan sebagai dasar pergerakan adalah algoritma A star, algoritma *dijkstra*, dan algoritma *D* Lite*.

2.2.4 Non-Player Character

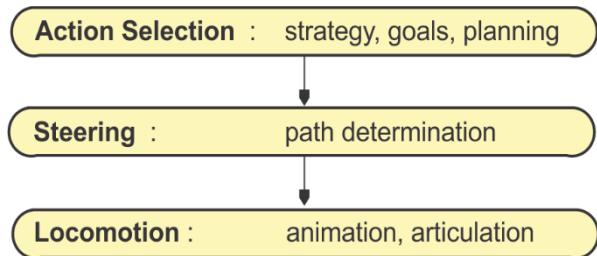
Non-Player Character (NPC) merupakan objek atau karakter yang tidak dikendalikan oleh pemain tapi dijalankan oleh komputer (Liarokapis, *et al.*, 2014). NPC juga disebut sebagai karakter otonom yang terdapat pada media interaktif seperti *games* dan *virtual reality*. Dikatakan karakter otonom karena NPC dapat melakukan tindakan sendiri atas perilakunya. Menurut Kim, *et al.*, (2006) perilaku NPC pada dasarnya akan berperilaku dengan cara mengulangi tiga tahap yaitu *sense*, *think*, dan *act* seperti ditunjukkan pada gambar 2.2. Perilaku NPC tersebut dilakukan secara otonom yang secara hierarkis dibagi menjadi tiga lapisan.



Tahap	Aktivitas
<i>sense</i>	Memahami situasi yang terjadi saat itu pada sistem
<i>think</i>	Menentukan aksi apa yang tepat sesuai dengan aturan dan situasi
<i>act</i>	Memerintahkan sistem untuk melakukan perbuatan atau aksi

Gambar 2.2 Tahap perilaku NPC

Sumber : Kim, *et al.*, (2006)



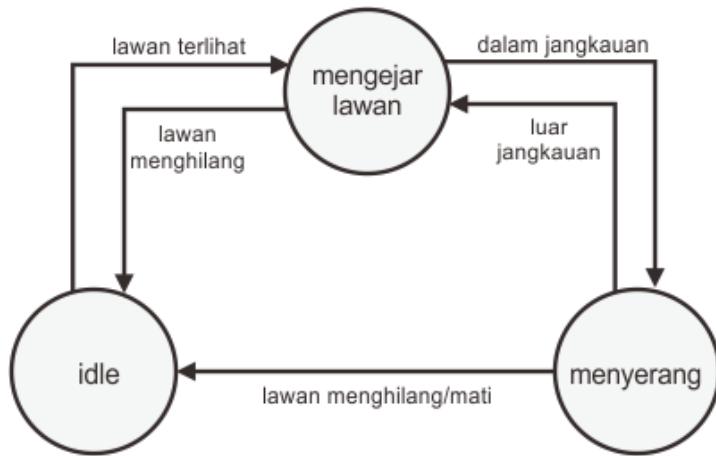
Gambar 2.3 Hierarki gerak perilaku NPC

Sumber : Reynolds (1999)

Gambar 2.3 menunjukkan pembagian perilaku karakter otonom menjadi tiga lapisan yaitu: *action selection*, *steering*, dan *locomotion*. *Action selection* lebih mengarah ke bagaimana strategi dan tujuan yang dicapai, *steering* lebih mengacu ke *path* area pada *game*, sedangkan *locomotion* lebih ke arah pergerakan karakter (animasi). Pada penelitian ini difokuskan bagian *action selection* yang didalamnya terdapat tentang strategi penyerangan pada NPC.

2.2.5 Finite State Machine

Finite state machine (FSM) merupakan metode yang paling sering digunakan untuk memodelkan perilaku NPC dalam sebuah *game*. Hal ini karena kesederhanaan dan kemudahan FSM untuk diimplementasikan (Asmiatun, *et al.*, 2013). Struktur FSM terdiri dari tiga hal yaitu keadaan (*state*), kejadian (*event*), dan aksi (*action*). Pada suatu saat sistem akan berada pada salah satu *state* yang aktif. Sistem mampu berpindah menuju *state* lain ketika mendapatkan *input* atau *event* tertentu. Perpindahan ini umumnya akan disertai aksi yang dilakukan oleh sistem sesuai *input* tersebut. Aksi yang dilakukan dapat berupa aksi yang sederhana atau melibatkan serangkaian proses yang kompleks (Bimantika, 2017). Contoh alur FSM dapat dilihat pada gambar 2.4.



Gambar 2.4 Contoh alur FSM

Sumber: Bimantika (2017)

Gambar 2.4 menunjukkan FSM memiliki alur yaitu ketika sistem mulai dihidupkan, sistem akan bertransisi menuju *state idle*, pada *event* ini *sistem* akan menghasilkan *action* mengejar musuh ketika terjadi *state* musuh tampak, sedangkan jika terjadi *state* musuh dalam jangkauan maka akan menghasilkan *action* menyerang. Ketika terjadi *state* musuh menghilang maka akan menghasilkan *action* *idle* dan seterusnya.

2.2.6 Dataset

Himpunan data atau *dataset* merupakan kumpulan data yang berisi variabel atau atribut tertentu dan objek keluarannya (Anggraini *et al.*, 2019). Atribut adalah parameter yang menyebabkan *class* atau label terjadi. Sedangkan *class* adalah objek yang akan dijadikan target atau keluaran. *Dataset* dapat digunakan untuk mencari informasi penting dari apa yang kita butuhkan. Dalam *dataset* terkadang ada informasi yang tidak relevan pada kenyataan, oleh karena itu untuk mengevaluasi *dataset* dilakukan pembagian jenis data yaitu data *training* dan data *testing*.

1. Data *Training*

Data *training* adalah data yang digunakan sebagai data pembelajaran untuk menghitung probabilitas dari data baru. Data *training* merupakan data yang sudah ada sesuai dengan fakta. Tabel 2.1 merupakan contoh data *training*.

Tabel 2.1 Contoh data *training*

Sumber: (https://www.saedsayad.com/naive_bayesian.htm)

No	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play golf</i>
1	<i>Rainy</i>	<i>Hot</i>	<i>High</i>	<i>False</i>	<i>No</i>
2	<i>Rainy</i>	<i>Hot</i>	<i>High</i>	<i>True</i>	<i>No</i>
3	<i>Overcast</i>	<i>Hot</i>	<i>High</i>	<i>False</i>	<i>Yes</i>
4	<i>Sunny</i>	<i>Mild</i>	<i>High</i>	<i>False</i>	<i>Yes</i>
5	<i>Sunny</i>	<i>Cool</i>	<i>Normal</i>	<i>False</i>	<i>Yes</i>
6	<i>Sunny</i>	<i>Cool</i>	<i>Normal</i>	<i>True</i>	<i>No</i>
7	<i>Overcast</i>	<i>Cool</i>	<i>normal</i>	<i>True</i>	<i>Yes</i>
8	<i>Rainy</i>	<i>Mild</i>	<i>High</i>	<i>False</i>	<i>No</i>
9	<i>Rainy</i>	<i>Cool</i>	<i>Normal</i>	<i>False</i>	<i>Yes</i>
10	<i>Sunny</i>	<i>Mild</i>	<i>Normal</i>	<i>False</i>	<i>Yes</i>
11	<i>Rainy</i>	<i>Mild</i>	<i>Normal</i>	<i>True</i>	<i>Yes</i>
12	<i>Overcast</i>	<i>Mild</i>	<i>High</i>	<i>True</i>	<i>Yes</i>
13	<i>Overcast</i>	<i>Hot</i>	<i>Normal</i>	<i>False</i>	<i>Yes</i>
14	<i>Sunny</i>	<i>Mild</i>	<i>High</i>	<i>True</i>	<i>No</i>

2. Data *Testing*

Data *testing* merupakan data yang akan dipergunakan sebagai bahan uji untuk mengukur sejauh mana keberhasilan dan ketepatan sistem melakukan klasifikasi dengan benar. Contoh data *testing* dapat dilihat pada tabel 2.2.

Tabel 2.2 Contoh data *testing*

Sumber: (https://www.saedsayad.com/naive_bayesian.htm)

No	<i>Outlook</i>	<i>Temperature</i>	<i>Humidity</i>	<i>Windy</i>	<i>Play golf</i>
1	<i>Rainy</i>	<i>Cool</i>	<i>High</i>	<i>True</i>	?

2.2.7 Naïve Bayes

Naïve bayes merupakan metode untuk klasifikasi suatu objek. Nama *bayes* berasal dari nama penemunya yaitu Thomas Bayes pada tahun 1950. Teori *naïve bayes* memiliki kemampuan klasifikasi yang serupa dengan *decision tree* dan *neural network* bahkan metode *naïve bayes* memiliki kecepatan yang tinggi saat diaplikasikan ke dalam *database* dengan data yang besar (F.Osisanwo, *et al.*, 2017; Anggraini, *et al.*, 2019). Teorema *bayes* memprediksi peluang dimasa depan berdasarkan pengalaman di masa sebelumnya. Teorema tersebut dikombinasikan dengan *naïve* dimana diasumsikan kondisi antar atribut saling bebas atau independen (Wang dan Shang, 2017). Dengan kata lain *naïve bayes* mengasumsikan bahwa keberadaan sebuah atribut (variabel) tidak ada kaitannya dengan keberadaan atribut lain. Secara umum metode *naïve bayes* dapat didefinisikan sesuai persamaan 2.1 (Anggraini, *et al.*, 2019).

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \quad (2.1)$$

Keterangan:

X = data atribut (parameter)

C = *class* yang diberikan

$P(C|X)$ = probabilitas suatu *class* C bersyarat atribut X (*posterior probability*)

$P(C)$ = probabilitas suatu *class* C (*prior probability*)

$P(X|C)$ = probabilitas atribut X bersyarat suatu *class* C (*likelihood probability*)

$P(X)$ = probabilitas atribut X

Rumus *naïve bayes* pada persamaan 2.1 menjelaskan bahwa peluang masuknya atribut tertentu dalam kelas C (*posterior*) adalah peluang munculnya kelas C (sebelum masuknya atribut, disebut *prior*) dikali dengan peluang

kemunculan atribut tertentu pada kelas C (disebut *likelihood*) dibagi dengan peluang kemunculan atribut tertentu secara global (disebut *evidence*). Rumus pada persamaan 2.1 dapat pula ditulis secara sederhana dalam persamaan 2.2.

$$\text{Posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}} \quad (2.2)$$

Nilai *evidence* selalu tetap untuk setiap kelas pada suatu atribut. Nilai *posterior* tersebut nantinya akan dibandingkan dengan nilai-nilai *posterior* kelas lain untuk menentukan kelas apa yang akan diklasifikasikan. Penjabaran lanjut rumus *naïve bayes* tersebut dilakukan dengan menjabarkan $P(C|X_1, \dots X_n)$ menggunakan aturan perkalian pada persamaan 2.3.

$$\begin{aligned} P(C|X_1, \dots X_n) &= P(C) \cdot P(X_1, \dots X_n | C) = P(C)P(X_1|C)P(X_2, \dots X_n|C, X_1) \\ &= P(C)P(X_1|C)P(X_2|C, X_1)P(X_3, \dots X_n|C, X_1, X_2) \\ &= P(C)P(X_1|C)P(X_2|C, X_1)P(X_3|C, X_1, X_2)P(X_4, \dots X_n|C, X_1, X_2, X_3) \\ &= P(C)P(X_1|C)P(X_2|C, X_1)P(X_3|C, X_1, X_2), \dots P(X_n|C, X_1, X_2, X_3, \dots X_{n-1}) \end{aligned} \quad (2.3)$$

Keterangan:

C = *class* yang diberikan

X_1, X_2, X_3, X_n = atribut (parameter)

$P(C)$ = probabilitas suatu *class* C

$P(C|X_1, \dots X_n)$ = probabilitas suatu *class* C bersyarat beberapa atribut X

Persamaan 2.3 menunjukkan hasil penjabaran persamaan *naïve bayes* yang memiliki banyak persyaratan atribut. Persamaan *naïve bayes* tersebut akan semakin kompleks karena terdapat faktor-faktor syarat yang mempengaruhi nilai probabilitas yang hampir tidak mungkin dianalisis satu per satu. Akibatnya perhitungan tersebut menjadi sulit untuk dilakukan. Disinilah digunakan asumsi independensi yang tinggi (naif) bahwa masing-masing atribut saling bebas

(independen) satu sama yang lain. Dengan asumsi antar atribut yang saling bebas maka dapat dilakukan penyederhanaan rumus sehingga perhitungan menjadi mungkin untuk dilakukan. Oleh karena itu penjabaran $P(C|X_1, \dots, X_n)$ dapat disederhanakan seperti ditunjukkan pada persamaan 2.4.

$$P(C|X_1, \dots, X_n) = P(C)P(X_1|C)P(X_2|C)P(X_3|C), \dots, P(X_n|C) \quad (2.4)$$

Keterangan:

C = *class* yang diberikan

X_1, X_2, X_3, X_n = atribut (parameter)

$P(C)$ = probabilitas suatu *class* C

$P(C|X_1, \dots, X_n)$ = probabilitas suatu *class* C bersyarat beberapa atribut X

Persamaan 2.4 merupakan rumus *naïve bayes* yang selanjutnya akan digunakan dalam proses perhitungan.

Berikut contoh perhitungan *naïve bayes* untuk menentukan apakah bermain *golf* atau tidak. Semua atribut saling bebas satu sama lain dan tidak terikat. Adapun atribut atau parameter yang dipakai untuk penghitungan ini adalah cuaca (*outlook*), suhu (*temperature*), kelembaban (*humidity*), dan angin (*windy*). Sedangkan *class* yang akan diklasifikasikan adalah *yes* dan *no*. Adapun data *training* yang digunakan dapat dilihat pada tabel 2.1. Sedangkan data *testing* pada tabel 2.2. Selanjutnya dalam menghitung *naïve bayes* digunakan persamaan 2.4.

1. Perhitungan *prior*

$P(\text{play golf} = \text{yes}) = \frac{9}{14}$, jumlah banyaknya kelas *yes* pada data *training* dibagi banyaknya data *training*.

$P(\text{play golf} = \text{no}) = \frac{5}{14}$, jumlah banyaknya kelas *no* pada data *training* dibagi banyaknya data *training*.

2. Perhitungan *likelihood*

Probabilitas *likelihood* merupakan probabilitas dengan nilai tertentu dari suatu parameter terhadap kelas.

a. *Likelihood* kelas yes

$$P(outlook = rainy|yes) = \frac{2}{9}$$

$$P(temperature = cold|yes) = \frac{3}{9}$$

$$P(humidity = high|yes) = \frac{3}{9}$$

$$P(windy = true|yes) = \frac{3}{9}$$

b. *Likelihood* kelas no

$$P(outlook = rainy|no) = \frac{3}{5}$$

$$P(temperature = cold|no) = \frac{1}{5}$$

$$P(humidity = high|no) = \frac{4}{5}$$

$$P(windy = true|no) = \frac{3}{5}$$

3. Perhitungan *posterior*

a. *Posterior* kelas yes

$$P(play\ golf = yes) = P(outlook = rainy|yes) \times P(temperature = cold|yes) \times$$

$$P(Humidity = high|yes) \times P(windy = true|yes) \times P(yes)$$

$$P(play\ golf = yes) = \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} = 0,00529$$

b. *Posterior* kelas no

$$P(play\ golf = no) = P(outlook = rainy|no) \times P(temperature = cold|no) \times$$

$$P(Humidity = high|no) \times P(windy = true|no) \times P(no)$$

$$P(\text{play golf} = \text{no}) = \frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} = 0,02057$$

Dari hasil perhitungan diperoleh probabilitas kelas *yes* = 0,00529 sedangkan probabilitas kelas *no* = 0,02057. Maka kesimpulannya jika terdapat kondisi sesuai data *testing* pada tabel 2.2 maka keputusan bermain *golf* adalah *no*.

2.2.8 Confusion Matrix

Confusion matrix merupakan suatu metode untuk memberikan evaluasi terhadap kinerja suatu sistem. *Confusion matrix* pada dasarnya memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem terhadap klasifikasi sebenarnya. Pengukuran terhadap kinerja suatu sistem klasifikasi merupakan hal yang penting. Kinerja sistem klasifikasi menggambarkan seberapa baik sistem dalam mengklasifikasikan data. Berdasarkan jumlah keluaran kelasnya, *Confusion matrix* memiliki beberapa bentuk seperti berdimensi 2 (terdiri dari 2 kelas) dan *confusioon matrix* berdimensi 3 (*multiclass*) (Witten dan Frank, 2005). Pada penelitian ini akan digunakan model *confusion matrix* yang terdiri dari 3 kelas yaitu untuk klasifikasi serangan tembak, granat, dan pisau. Adapun model *confusion matrix* dimensi 3 dapat dilihat pada tabel 2.3.

Tabel 2.3 *Confusion matrix multiclass* (Iskandar dan Suprapto, 2015)

<i>Actual class</i>	<i>Predicted class</i>		
	<i>Class A</i>	<i>Class B</i>	<i>Class C</i>
<i>Class A</i>	AA	AB	AC
<i>Class B</i>	BA	BB	BC
<i>Class C</i>	CA	CB	CC

Keterangan:

Predicted class = Kelas yang menunjukkan hasil klasifikasi sistem

Actual class = Kelas yang menunjukkan klasifikasi sebenarnya

Class A = Kelas dengan nama A

Class B = Kelas dengan nama B

Class C = Kelas dengan nama C

AA = Jumlah data dengan nilai sebenarnya A dan diprediksi tepat sebagai kelas A

AB = Jumlah data dengan nilai sebenarnya A dan diprediksi sebagai kelas B

AC = Jumlah data dengan nilai sebenarnya A dan diprediksi sebagai kelas C

BA = Jumlah data dengan nilai sebenarnya B dan diprediksi sebagai kelas A

BB = Jumlah data dengan nilai sebenarnya B dan diprediksi tepat sebagai kelas B

BC = Jumlah data dengan nilai sebenarnya B dan diprediksi sebagai kelas C

CA = Jumlah data dengan nilai sebenarnya C dan diprediksi sebagai kelas A

CB = Jumlah data dengan nilai sebenarnya C dan diprediksi sebagai kelas B

CC = Jumlah data dengan nilai sebenarnya C dan diprediksi tepat sebagai kelas C

Tabel *confusion matrix* memberikan informasi nilai dari klasifikasi sebenarnya (*actual class*) dan klasifikasi yang dilakukan oleh sistem (*predicted class*). Dari informasi tersebut untuk menghitung kinerja dari suatu algoritma dapat dihitung berdasarkan nilai akurasi, presisi, dan *recall*.

1. Akurasi

Akurasi merupakan keakuratan nilai dari semua data yang diprediksi dengan benar. Dengan kata lain nilai akurasi merupakan perbandingan antara data yang terkласifikasi benar dengan keseluruhan jumlah data. Nilai akurasi menggambarkan keseluruhan hasil klasifikasi. Nilai akurasi dapat ditentukan sesuai persamaan 2.5.

$$\text{Akurasi} = \frac{\text{AA} + \text{BB} + \text{CC}}{\text{AA} + \text{AB} + \text{AC} + \text{BA} + \text{BB} + \text{BC} + \text{CA} + \text{CB} + \text{CC}} \times 100\% \quad (2.5)$$

2. Presisi

Presisi dapat diartikan sebagai kepersisan atau kecocokan antara permintaan informasi dengan jawaban terhadap permintaan tersebut. Presisi dapat dikatakan perbandingan data yang diklasifikasi benar pada data tertentu terhadap jumlah data yang diklasifikasi pada data tersebut. Nilai presisi dapat ditentukan sesuai persamaan berikut.

$$\text{Presisi class A} = \frac{\text{AA}}{\text{AA} + \text{BA} + \text{CA}} \times 100\% \quad (2.6)$$

$$\text{Presisi class B} = \frac{\text{BB}}{\text{BB} + \text{AB} + \text{CB}} \times 100\% \quad (2.7)$$

$$\text{Presisi class C} = \frac{\text{CC}}{\text{CC} + \text{AC} + \text{BC}} \times 100\% \quad (2.8)$$

$$\bar{X}_{\text{Presisi}} = \frac{\text{Presisi class A} + \text{Presisi class B} + \text{Presisi class C}}{\text{Jumlah kelas}} \quad (2.9)$$

3. Recall

Recall merupakan tingkat keberhasilan sistem dalam menemukan kembali sebuah informasi. *Recall* dapat ditentukan dengan membandingkan data yang diprediksi benar pada data tertentu terhadap jumlah data sebenarnya pada data tersebut. Nilai *recall* dapat ditentukan sesuai persamaan berikut.

$$\text{Recall class A} = \frac{\text{AA}}{\text{AA} + \text{AB} + \text{AC}} \times 100\% \quad (2.10)$$

$$\text{Recall class B} = \frac{\text{BB}}{\text{BB} + \text{BA} + \text{BC}} \times 100\% \quad (2.11)$$

$$\text{Recall class C} = \frac{\text{CC}}{\text{CC} + \text{CA} + \text{CB}} \times 100\% \quad (2.12)$$

$$\bar{X}_{\text{Recall}} = \frac{\text{Recall class A} + \text{Recall class B} + \text{Recall class C}}{\text{Jumlah kelas}} \quad (2.13)$$

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian dan implementasi metode *naïve bayes* pada *game shooter* maka diperoleh simpulan bahwa:

1. Implementasi metode *naïve bayes* sebagai kecerdasan buatan untuk strategi penyerangan NPC diawali dengan menentukan parameter. Adapun parameter untuk perhitungan *naïve bayes* adalah nyawa, jarak, jumlah granat, dan jumlah amunisi yang dimiliki NPC. Sedangkan klasifikasi penyerangan NPC dibagi menjadi serangan tembak, serangan granat, dan serangan pisau. Setiap parameter akan diberikan *range* nilai yang dikategorikan dalam beberapa kriteria. Selanjutnya dilakukan pengumpulan *dataset* dari nilai parameter tersebut dengan memberikan klasifikasi manual oleh pemain. *Dataset* tersebut akan dibagi menjadi 2 *subset* data yaitu data *training* dan data *testing*. Data *training* digunakan sebagai data pembelajaran NPC sedangkan data *testing* digunakan sebagai data uji jika NPC dalam kondisi baru. Pengujian dilakukan dengan menghitung kinerja NPC dalam memprediksi serangan yang akan diambil. Hasil pengujian diperoleh bahwa NPC mampu memprediksi serangan dengan benar sesuai data *testing* dengan akurasi sebesar 91,67%, presisi 91,65%, dan *recall* 91,67%.

2. Metode *naïve bayes* memberikan tingkat kemenangan NPC yang lebih baik dibanding menggunakan *rulebase*. Metode *naïve bayes* mendapat tingkat kemenangan sebesar 64%, sedangkan *rulebase* sebesar 32%.

5.2 Saran

Beberapa saran dapat dipertimbangkan dalam mengembangkan penelitian lebih lanjut adalah sebagai berikut:

1. Penambahan strategi dalam berperang misalnya taktik bersembunyi.
2. Penambahan parameter dari kondisi pemain sebagai masukan sistem dalam perhitungan *naïve bayes*.
3. Perlu dilakukan penelitian dengan metode pembanding lain untuk mendapatkan kecerdasan AI yang lebih optimal.

DAFTAR PUSTAKA

- Abdi, M., Herumurti, D., & Kuswardayan, I. (2017). Analisis Perbandingan Kecerdasan Buatan pada Computer Player dalam Mengambil Keputusan pada Game Battle RPG. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 15(2), 226-237. <https://doi.org/10.12962/j24068535.v15i2.a671>
- Accenture. (2016). *Artificial Intelligence is the Future of Growth*. <https://www.accenture.com/us-en/insight-artificial-intelligence-futuregrowth>. oriana.branon@accenture.com. 19 November 2019 (20:17).
- Aghlara, L., & Tamjid, N. H. (2011). The Effect of Digital Games on Iranian Children's Vocabulary Retention in Foreign Language Acquisition. *Procedia - Social and Behavioral Sciences*, 29, 552–560. <https://doi.org/10.1016/j.sbspro.2011.11.275>
- Anggraini, R. A., Widagdo, G., Budi, A. S., & Qomaruddin, M. (2019). Penerapan Data Mining Classification untuk Data Blogger Menggunakan Metode Naïve Bayes. *Jurnal Sistem Dan Teknologi Informasi (JUSTIN)*, 7(1), 47. <https://doi.org/10.26418/justin.v7i1.30211>
- Ashari, A., Paryudi, I., & Min, T. A. (2013). Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool. *International Journal of Advanced Computer Science and Applications*, 4(11), 33–39. <https://doi.org/10.14569/IJACSA.2013.041105>
- Asmiyatun, S., & Hendrawan, A. (2016). Implementasi Klasifikasi Bayesian Untuk Strategi Menyerang Jarak Dekat Pada Npc (Non Player Character) Menggunakan Unity 3D. *Jurnal Transformatika*, 13(2), 42. <https://doi.org/10.26623/transformatika.v13i2.139>
- Asmiyatun, S., Hermawan, L., & Daryatni, T. (2013). Strategi Menyerang Jarak Dekat Menggunakan Klasifikasi Bayesian Pada NPC (Non Player Character). *Semantik*, 3(1), 351–357.
- Audio, S.E.G. 2015. *Post Apocalypse Guns Demo*. <https://assetstore.unity.com/packages/audio/sound-fx/weapons/post-apocalypse-guns-demo-33515>. spinear@soudearth.kr. 21 Juli 2020 (21:03).
- A3D. (2017). *Medieval Sword*. <https://assetstore.unity.com/packages/3d/medieval-sword-73359>. a3deve@gmail.com. 14 Juli 2020 (20:01).

- Bimantika, R. (2017). Pengembangan Game The Galaxy Menggunakan Metode FSM (Finite State Machine). *Jurnal Mahasiswa Teknik Informatika*, 1(1), 180–187.
- Blacksmith, P. (2016). *Toon Soldiers Demo*. <https://assetstore.unity.com/packages/3d/characters/toon-soldiers-demo-69684>. polygonblacksmith@gmail.com. 14 Juli 2020 (11:42).
- . (2017). *Toon Soldiers WW2 Demo*. <https://assetstore.unity.com/packages/3d/characters/humanoids/toon-soldiers-ww2-demo-85702>. polygonblacksmith@gmail.com. 14 Juli 2020 (14:03).
- F, Osisanwo., J.E.T, Akinsola., O, Awodele., J. O, Himnikaiye., O, Olakanmi., & J, Akinjobi. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, 48(3), 128–138. <https://doi.org/10.14445/22312803/ijctt-v48p126>
- Fathurochman, D., Witanti, W., & Yuniarti, R. (2014). Perancangan Game Turn Based Strategy Menggunakan Logika Fuzzy Dan Naive Bayes Classifier. *Seminar Nasional Informatika 2014*, 2014(Game Development), 44–51.
- Fauziah, D. A., Maududie, A., & Nuritha, I. (2018). Klasifikasi Berita Politik Menggunakan Algoritma K-nearest Neighbor. *Jurnal Sistem Informasi Universitas Jember*, 6(2), 8.
- Fibrianda, M. F., & Bhawiyuga, A. (2018). Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 2(9), 3112–3123.
- Frosi, F. O., & Cristina, I. S. (2018). Building Bots for Shooter Games based on the Bartle ' s Player Types and Finite State Machines : A Battling Behaviour Analysis. *Proceedings of SBG Games 2018*, 631–634.
- Hong, J. H., & Cho, S.-B. (2005). Evolving reactive NPCs for the real-time simulation game. *IEEE 2005 Symposium on Computational Intelligence and Games, CIG'05*.
- Iskandar, D., & Suprapto, Y. K. (2015). Perbandingan Akurasi Klasifikasi Tingkat Kemiskinan Antara Algoritma C 4.5 dan Naive Bayes. *Jurnal Ilmiah NERO*, 2(1), 37–43.
- Kaviani, P., & Dhotre, S. (2017). Short Survey on Naive Bayes Algorithm. *International Journal of Advance Engineering and Research Development*, 4(11).

- Kim, C., Jeong, S., Hur, G., & Kim, B. (2006). Verification of FSM using Attributes Definition of NPCs Models. *International Journal of Computer Science and Network Security*, 6(7), 168–174. http://paper.ijcsns.org/07_book/200607/200607A25.pdf
- Kusumah, F. S. F., Fajri, H., & Barake, M. Al. (2018). Game 3D “Zombie Attack” dengan Menerapkan AI Pada NPC. *Kreatif: Jurnal Teknik Informatika*, 7(1), 20. <https://doi.org/10.32832/kreatif.v7i1.2044>
- Lample, G., & Chaplot, D. S. (2017). Playing FPS Games with Deep Reinforcement Learning. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2140–2146.
- Lee, B., & Lee, E. A. (1998). Interaction of finite state machines and concurrency models. *Conference Record of the Asilomar Conference on Signals, Systems and Computers*, 2(November), 1715–1719. <https://doi.org/10.1109/acssc.1998.751618>
- LemmoLab. (2018). *Grenades Lowpoly*. <https://assetstore.unity.com/packages/3d/props/grenades-lowpoly-120047>. lemokzz@gmail.com. 14 Juli 2020 (19:04).
- Liarakapis, F., Debattista, K., Vourvopoulos, A., Petridis, P., & Ene, A. (2014). Comparing Interaction Techniques for Serious Games through Brain-Computer Interfaces : A User Perception Evaluation Study. *Entertainment Computing*. <https://doi.org/10.1016/j.entcom.2014.10.004>
- Maroney, K. (2001). My Entire Walking Life. *The Games Journal*. <http://www.thegamesjournal.com/articles/MyEntireWakingLife.shtml>. 17 Juli 2020 (14:17).
- McGee, K., & Abraham, A. T. (2010). Real-time team-mate AI in games: A definition, survey, & critique. *FDG 2010 - Proceedings of the 5th International Conference on the Foundations of Digital Games*, 124–131. <https://doi.org/10.1145/1822348.1822365>
- Millington, I., & Funge, J. (2009). Artificial Intelligence for Games. In *Elsevier* (second). Morgan Kaufmann.
- Putera, M. I. A., & Herumurti, D. (2018). Peningkatan Kecerdasan Computer Player Pada Game Pertarungan Berbasis K-Nearest Neighbor Berbobot. *JUTI: Jurnal Ilmiah Teknologi Informasi*, 16(1), 90. <https://doi.org/10.12962/j24068535.v16i1.a710>

- Qaimkhani, F. M., Bhatti, Z., Technology, C., & Introduction, I. (2018). Game Based Learning using First Person Shooter – Battle of IICT. *International Journal of Computer Science and Emerging Technologies*, 2(December), 6–10. <http://ijcet.salu.edu.pk/index.php/IJCET/article/view/24>
- Rachman, A., Prasetyo, B. E., Arief, R., Ferdiansyah, M. A., & Sulistyowati. (2019). Pengembangan Aplikasi Game Pembelajaran Matematika “ Momon Math Run ” Berbasis Desktop Menggunakan Model Waterfall. *Seminar Nasional Sains Dan Teknologi Terapan VII 2019*, 433–438.
- Reynolds, C. (1999). Steering Behaviour For Autonomous Character. *Game Developer Conference*, 763–782. <https://doi.org/10.4404/hystrix-27.2-11895>
- Rostianingsih, S., Gregorius, S. B., & Wijaya, H. K. (2013). Game Simulasi Finite State Machine Untuk Pertanian Dan Peternakan. *Jurnal DKV Adiwarna*, 5, 2–7.
- Sanjaya, S. W., Akbar, M. A., & Afirianto, T. (2019). Penerapan Naïve Bayes untuk NPC Braking Decision pada Racing Game. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 3(4), 3252–3257.
- Suits, B. (2005). The Grasshopper: Games, Life and Utopia. In T. Hurka (Ed.), *Poetry & Prose: With Thomas Sprat's Life and Observations by Dryden, Addison, Johnson and others* (third). Broadview Press. <https://doi.org/10.1093/oseo/instance.00010341>
- Syahputra, M. F., Arippa, A., Rahmat, R. F., & Andayani, U. (2019). Historical Theme Game Using Finite State Machine for Actor Behaviour. *Journal of Physics: Conference Series*, 1235(1). <https://doi.org/10.1088/1742-6596/1235/1/012122>
- Talib, F. Z., Mayadewi, R. P., & Rosely, E. (2016). Aplikasi Prediksi Peminatan SMAN 8 Bandung Menggunakan Metode Klasifikasi Dengan Algoritma ID3. *Proceedings of Applied Science*, 2(3), 1053–1061.
- Technologies, U. (2019). *Unity Particle Pack 5.x*. <https://assetstore.unity.com/packages/essentials/asset-packs/unity-particle-pack-5-x-73777>. support@unity3d.com. 21 Juli 2020 (21:20).
- Testmobile. (2018). *UAA - Business - Paintbooth*. <https://assetstore.unity.com/packages/3d/environments/uaa-business-paintbooth-120410>. rythmic_chips@yahoo.com.br. 14 Juli 2020 (21:11).

- Wang, D., & Tan, A. H. (2015). Creating Autonomous Adaptive Agents in a Real-Time First-Person Shooter Computer Game. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(2), 123–138. <https://doi.org/10.1109/TCIAIG.2014.2336702>
- Wang, K., & Shang, W. (2017). Outcome prediction of DOTA2 based on Naïve Bayes classifier. *Proceedings - 16th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2017*, 1994, 591–593. <https://doi.org/10.1109/ICIS.2017.7960061>
- Wicaksono, A., Hariadi, M., & N, S. M. Supeno. (2013). Strategi Menyerang NPC Game FPS Menggunakan Fuzzy Finite State Machine. *Seminar Nasional Teknologi Informasi Dan Multimedia*, 25–30.
- Wijman, T. (2019). *The Global Games Market Will Generate \$152.1 Billion in 2019 as the U.S. Overtakes China as the Biggest Market*. <https://newzoo.com/insights/articles/the-global-games-market-will-generate-152-1-billion-in-2019-as-the-u-s-overtakes-china-as-the-biggest-market/>. 22 Februari 2020 (22:08).
- Witten, I., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (second). Morgan Kaufmann Publisher.
- Yunanta, D. P. (2017). Game Rescue Idol Dengan Menggunakan Metode Finite State Machine (Fsm). *Jurnal Mahasiswa Teknik Informatika*, 1(2), 96–103.