TUGAS AKHIR- TE 141599

SCORING BERBASIS AGEN CERDAS MENGGUNAKAN LOGIKA FUZZY PADA PERMAINAN (*GAME*) *REAL TIME STRATEGY* (RTS)

Kartika Dwi Handini
NRP 2212106016

Dosen Pembimbing
Mochamad Hariadi, ST., M.Sc., Ph.D.
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh November
Surabaya 2015

TUGAS AKHIR– TE 141599

SCORING BASED INTELLIGENT AGENT USING FUZZY LOGIC IN REAL TIME STRATEGY(RTS) GAME

Kartika Dwi Handini
NRP 2212106016

Supervisor
Mochamad Hariadi, ST., M.Sc., Ph.D.
Dr. Supeno Mardi Susiki Nugroho, ST., MT.

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Industrial Technology
Institut Teknologi Sepuluh November
Surabaya 2015

SCORING BERBASIS AGEN CERDAS MENGGUNAKAN LOGIKA *FUZZY* PADA PERMAINAN (*GAME*) *REAL TIME STRATEGY* (RTS)

TUGAS AKHIR

Diajukan guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar sarjana Teknik
Pada
Bidang Studi Teknik Komputer dan Telematika
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I	Dosen Pembimbing II
Mochamad Hariadi, S.T.,M.Sc.,Ph.D	Dr. Supeno Mardi S N, ST., MT.
NIP. 19691209 1997031 002	NIP. 19700313 1995121 001

SURABAYA
JUNI, 2015

SCORING BERBASIS AGEN CERDAS MENGGUNAKAN LOGIKA _FUZZY_ PADA PERMAINAN (_GAME_) _REAL TIME STRATEGY_ (RTS)

Nama : Kartika Dwi Handini
Pembimbing : 1. Mochamad Hariadi, ST.,M.Sc.,Ph.D.
2. Dr. Supeno Mardi S N, ST., MT.

ABSTRAK

Salah satu bagian dari *game* yang dapat menarik minat pemain adalah penilaian (*scoring*) yang diterapkan. Pemanfaatan properti dari setiap unit pada *game* RTS dapat digunakan untuk pemodelan *scoring* yang sesuai dengan keadaan pada saat permainan diberakhir. Dengan menggunakan logika *fuzzy*, beberapa properti dapat digabungkan untuk mendapatkan hasil nilai (*score*) yang dapat digunakan untuk keperluan dalam permainan. Perubahan kondisi dari properti setiap unit yang terdapat pada *game* RTS digunakan sebagai masukan yang kemudian diolah dengan logika *fuzzy*. Nilai standar deviasi (variasi data) untuk s*coring* menggunakan logika *fuzzy* sebesar 0.7378. sedangkan pada *scoring* statis nilainya sebesar 0.7017. Logika *fuzzy* mampu menghasilkan nilai(*score*) yang lebih bervariasi sehingga hasil yang didapat lebih dinamis. 3. Besar persentase hasil bintang yang sama antara penghitungan statis dan *fuzzy* sebesar 60%. Sisanya sebesar 40% adalah hasil perolehan bintang yang berbeda.

Kata kunci: *Fuzzy, Real Time Strategy, Scoring, game*

SCORING BERBASIS AGEN CERDAS MENGGUNAKAN LOGIKA _FUZZY_ PADA PERMAINAN (_GAME_) _REAL TIME STRATEGY_ (RTS)

Nama : Kartika Dwi Handini
Pembimbing : 1. Mochamad Hariadi, ST.,M.Sc.,Ph.D.
2. Dr. Supeno Mardi S N, ST., MT.

ABSTRAK

Salah satu bagian dari *game* yang dapat menarik minat pemain adalah penilaian (*scoring*) yang diterapkan. Pemanfaatan properti dari setiap unit pada *game* RTS dapat digunakan untuk pemodelan *scoring* yang sesuai dengan keadaan pada saat permainan diberakhir. Dengan menggunakan logika *fuzzy*, beberapa properti dapat digabungkan untuk mendapatkan hasil nilai (*score*) yang dapat digunakan untuk keperluan dalam permainan. Perubahan kondisi dari properti setiap unit yang terdapat pada *game* RTS digunakan sebagai masukan yang kemudian diolah dengan logika *fuzzy*. Nilai standar deviasi (variasi data) untuk s*coring* menggunakan logika *fuzzy* sebesar 0.7378. sedangkan pada *scoring* statis nilainya sebesar 0.7017. Logika *fuzzy* mampu menghasilkan nilai(*score*) yang lebih bervariasi sehingga hasil yang didapat lebih dinamis. 3. Besar persentase hasil bintang yang sama antara penghitungan statis dan *fuzzy* sebesar 60%. Sisanya sebesar 40% adalah hasil perolehan bintang yang berbeda.

Kata kunci: *Fuzzy, Real Time Strategy, Scoring, game*

--Halaman ini sengaja dikosongkan--

SCORING BASED INTELLIGENT AGENT USING FUZZY LOGIC IN REAL TIME STRATEGY(RTS) GAME

Name **: Kartika Dwi Handini**
Supervisor **: 1. Mochamad Hariadi, ST.,M.Sc.,Ph.D.**
2. Dr. Supeno Mardi S N, ST., MT.

ABSTRACT

One part of the game that can attract players is assessment (scoring) is applied. Exploiting the property of each unit in RTS games can be used for modeling scoring appropriate to the circumstances at the time of the game diberakhir. By using fuzzy logic, some properties can be combined to get the value (score) that can be used for the purpose in the game. Changes in the condition of the property each unit contained in RTS games are used as inputs are then processed with fuzzy logic. Standard deviation (variation data) for scoring using fuzzy logic at 0.7378. whereas the static scoring value by 0.7017. Fuzzy logic is able to generate value (score) is more varied so that the results are more dynamic. 3. Large percentage of the same star between the static and fuzzy calculation by 60%. The rest of 40% is the result of the acquisition of different stars.

Keyword*: Fuzzy, Real Time Strategy, Scoring, game*

--Halaman ini sengaja dikosongkan--

KATA PENGANTAR

Puji dan Syukur kehadirat Allah SWT atas segala limpahan berkah, rahmat serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul : ***Scoring* Berbasis Agen Cerdas Menggunakan Logika *Fuzzy* pada Permainan (G*ame*) *Real Time Strategy* (RTS).**

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Jurusan Teknik Elektro ITS, Bidang Studi Teknik Komputer dan Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S-1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, danBapak yang telah memberikan dorongan spiritual dan material serta seluruh kerabat dan kolega penulis yang banyak membantu proses dalam menyelesaikan buku penelitian ini.
2. Bapak Dr. Tri Arief Sardjono, S.T., M.T. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember
3. Secara khusus penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Mochamad Hariadi, ST.,M.Sc.,Ph.D. dan Bapak Dr. Supeno Mardi S N, ST., MT. atas bimbingan selama mengerjakan penelitian.
4. Bapak-ibu dosen pengajar Bidang Studi Teknik Komputer dan Telematika, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh teman-teman Lintas Jalur Elektro Genap 2012 serta teman-teman Bidang Studi Teknik Komputer dan Telematika.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun serta menghatur maaf atas segala kekurangan yang ada dalam penulisan buku ini. Semoga penelitian ini dapat memberikan manfaat bagi kita semua.Amin.

Surabaya, Juni 2015

Penulis

--Halaman ini sengaja dikosongkan--

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR ..... **..i**
LEMBAR PENGESAHAN ..... **iii**
ABSTRAK ..... **…v**
ABSTRACT ..... **.vii**
KATA PENGANTAR ..... **...ix**
DAFTAR ISI ..... **...xi**
DAFTAR GAMBAR ..... **..xv**
DAFTAR TABEL ..... **.xvii**
BAB I PENDAHULUAN **1**
1.1. Latar Belakang 1
1.2. Permasalahan 2
1.3. Tujuan 2
1.4. Batasan Masalah 2
1.5. Metodologi 2
1.6. Sistematika Penulisan 4
BAB 2 TEORI PENUNJANG **5**
2.1. *Fantasy Chronicles* 5
2.2. *Game Real Time Strategy* 6
2.3. *Scoring* pada Permainan 7
2.4. Logika *Fuzzy* 8
BAB 3 PERANCANGAN SISTEM **21**
3.1. Desain Sistem 22
3.2. Perancangan Variable Masukan 22
3.2.1. Parameter Kayu 23

3.2.2. Parameter Poin 23
3.2.3. Paramener Pasukan 23
3.2.4. Parameter Markas 23
3.2.5. Parameter Predikat 24
3.3. Perancangan Fungsi Keanggotaan Kayu 24
3.3.1. Fungsi Keanggotaan Parameter Kayu Kondisi “sedikit” 24
3.3.2. Fungsi Keanggotaan Parameter Kayu Kondisi “sedang” 25
3.3.3. Fungsi Keanggotaan Parameter Kayu Kondisi “banyak” 25
3.4. Perancangan Fungsi Keanggotaan Poin 26
3.4.1. Fungsi Keanggotaan Parameter Poin Kondisi “sedikit” 26
3.4.2. Fungsi Keanggotaan Parameter Poin Kondisi “sedang” 27
3.4.3. Fungsi Keanggotaan Parameter Poin Kondisi “banyak” 28
3.5. Perancangan Fungsi Keanggotaan Pasukan 28
3.5.1. Fungsi Keanggotaan Parameter Pasukan Kondisi “sedikit” . 28
3.5.2. Fungsi Keanggotaan Parameter Pasukan Kondisi “sedang” 29
3.5.3. Fungsi Keanggotaan Parameter Pasukan Kondisi “banyak”30
3.6. Perancangan Fungsi Keanggotaan Markas 30
3.6.1. Fungsi Keanggotaan Parameter Markas Kondisi “hancur” .. 31
3.6.2. Fungsi Keanggotaan Parameter Markas Kondisi “sedang” . 31
3.6.3. Fungsi Keanggotaan Parameter Markas Kondisi “utuh” 32
3.7. Perancangan Fungsi Keanggotaan Predikat 32
3.7.1. Fungsi Keanggotaan Parameter Predikat Kondisi “poor” 33
3.7.2. Fungsi Keanggotaan Parameter Predikat Kondisi “fair” 33
3.7.3. Fungsi Keanggotaan Parameter Predikat Kondisi “good” ... 34
3.7.4. Fungsi Keanggotaan Parameter Predikat Kondisi “excelent”34

3.8. Desain Aturan *Fuzzy* (*Rules Base*) 35
BAB 4 PENGUJIAN DAN ANALISA 39
4.1. Skenario Pengujian 39
4.2. Pengujian Implemantasi Parameter 39
4.3. Pengujian Sistem *Scoring* 40
4.3.1. Pengujian Hasil Akhir Kondisi “poor” 41
4.3.2. Pengujian Hasil Akhir Kondisi “fair” 42
4.3.3. Pengujian Hasil Akhir Kondisi “good” 43
4.3.4. Pengujian Hasil Akhir Kondisi “excelent” 44
4.4. Pengujian Metode 45
4.4.1. Pengujian Hasil Akhir Kondisi Menang 45
4.4.2. Pengujian Hasil Akhir Kondisi Kalah 47
BAB 5 PENUTUP 51
5.1. Kesimpulan 51
5.2. Saran 51
DAFTAR PUSTAKA 53
LAMPIRAN 55
RIWAYAT HIDUP 67

--Halaman ini sengaja dikosongkan--

DAFTAR TABEL

Tabel 3.4: Rule based fuzzy Predikat 35
Tabel 4.1: Hasil Pengujian kondisi menang 46
Tabel 4.2: Hasil Pengujian kondisi kalah 48

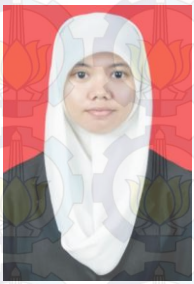
--Halaman ini sengaja dikosongkan--

DAFTAR GAMBAR

Gambar 1.1 Metodologi penelitian 3
Gambar 2.1 Tampilan Awal Permainan *Fantasy Chronicles* 3D 6
Gambar 2.2 Game RTS Command & Conquer (techxav.com) 7
Gambar 2.3 Contoh pemetaan *input output* (Gelley,2000) 8
Gambar 2.4 Himpunan *fuzzy* pada variable temperatur[3] 10
Gambar 2.5 Representasi linier naik 11
Gambar 2.6 Representasi linier turun 12
Gambar 2.7 Representasi Kurva Segitiga 12
Gambar 2.8 Representasi Kurva Trapesium 13
Gambar 2.9 Komposisi Aturan *Fuzzy* : Metode *Max*[3] 16
Gambar 2.10 Proses Defuzzyfikasi 17
Gambar 3.1 Desain *game Fantasy Chronicles 3D* 21
Gambar 3.2 Tahapan desain sistem 22
Gambar 3.3 FK parameter kayu kondisi “sedikit” 24
Gambar 3.4 FK parameter kayu kondisi “sedang” 25
Gambar 3.5 FK parameter kayu kondisi “banyak” 26
Gambar 3.6 FK parameter poin kondisi “sedikit” 27
Gambar 3.7 FK parameter poin kondisi “sedang” 27
Gambar 3.8 FK parameter poin kondisi “banyak” 28
Gambar 3.9 FK parameter pasukan kondisi “sedikit” 29
Gambar 3.10 FK parameter pasukan kondisi “sedang” 29
Gambar 3.11 FK parameter pasukan kondisi “banyak” 30
Gambar 3.12 FK parameter markas kondisi “hancur” 31
Gambar 3.13 FK parameter markas kondisi “sedang” 31
Gambar 3.14 FK parameter markas kondisi “utuh” 32
Gambar 3.15 FK parameter predikat kondisi “poor” 33

Gambar 3.16 FK parameter predikat kondisi “fair” 33
Gambar 3.17 FK parameter predikat kondisi “good”..... 34
Gambar 3.18 FK parameter predikat kondisi “excelent” 35
Gambar 4.1 Implementasi variabel masukan 39
Gambar 4.2 Hasil Akhir Kondisi “poor”..... 41
Gambar 4.3 Hasil Akhir Kondisi “fair” 42
Gambar 4.4 Hasil Akhir Kondisi “Good” 43
Gambar 4.5 Hasil Akhir Kondisi “excelent”..... 44
Gambar 4.5 pengujian metode hasil permainan menang, predikat “excelent” dan “poor” 45
Gambar 4.6 pengujian metode hasil permainan kalah predikat “poor”. 47

RIWAYAT HIDUP



Kartika Dwi Handini. lahir di Tegal pada 11 April 1991. Anak kedua dari dua bersaudara ini menyelesaikan pendidikan dasar di SDN Randugunting 3 Tegal. Kemudian menempuh jalur pendidikan di SMP Negeri 10 Tegal. Setelah itu melanjutkan pendidikan di SMK Telkom Sandhy Putra Purwokerto dengan mengambil jurusan Rekayasa Perangkat Lunak (RPL) pada tahun 2006 hingga tahun 2009. Melanjutkan pendidikan ke jenjang perkuliahan di Jurusan D3 Teknik Elektro Politeknik Negeri Semarang (POLINES) prodi Teknik Informatika dari tahun 2009 sampai 2012. Dan melanjutkan kuliah S1 di Lintas Jalur ITS jurusan Teknik Elektro prodi Teknik Komputer dan Telematika. (kartikadwihandini@gmail.com).

BAB 1
PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi yang semakin pesat khususnya dalam bidang komputer membuat komputer saat ini bukan hanya sebagai perangkat yang digunakan untuk mengolah data saja, tetapi juga dapat digunakan sebagai sarana hiburan. Pemanfaatan komputer sebagai sarana hiburan dapat di implementasikan dalam berbagai macam bentuk seperti permainan komputer (*Game*). *Game* dengan alur cerita yang menarik serta gambar yang berkualitas baik akan lebih diminati oleh pengguna sehingga membuat pemain tidak merasa cepat bosan dan ingin melanjutkan *game* hingga selesai. Sebuah *game* memiliki 5 komponen penting yaitu fitur, *gameplay*, *interface*, aturan, dan desain [6]. Salah satu bagian yang termasuk dalam aturan sebuah *game* adalah aturan *scoring* yang akan digunakan. *Scoring* bertujuan untuk memberikan nilai akhir pada sebuah permainan dan merupakan komponen penting yang dapat menjaga minat pemain untuk tetap bermain [1].

Dalam *game* RTS memiliki dua aspek penilaian yaitu aspek ekonomi yang berupa sumber daya yang dimiliki dan aspek militer yang dapat dilihat dari pasukan dan *basecamp* yang ada [5]. Pada beberapa *game* yang ada saat ini, sumber daya biasanya digunakan untuk membeli / meningkatkan keterampilan pemain / senjata . Sedangkan pasukan dan *basecamp* tidak begitu diperhitungkan dalam *scoring* hasil akhir permainan. Seperti dalam permainan *Clash Of Clans*, sumber daya digunakan untuk membeli perlengkapan *basecamp* dan pasukan tetapi untuk menentukan hasil akhir permainan hanya melihat dari seberapa besar bagian dari bangunan yang berhasil dihancurkan tanpa melihat kondisi pasukan dan sumber daya yang didapat dan digunakan. Sehingga diperlukan sebuah sistem *scoring* yang dapat mencakup semua aspek penilaian dalam *game* RTS.

Dalam Tugas Akhir ini akan dibuat *scoring* yang dapat menentukan hasil akhir permainan dengan melibatkan semua aspek penilaian yang ada pada permainan. Karena parameter *scoring* yang digunakan beragam maka *scoring* akan diolah menggunakan logika *fuzzy*. Logika *fuzzy* adalah cabang dari sistem kecerdasan buatan (*Artificial Intelegent*) yang mengemulasi kemampuan manusia dalam berfikir ke dalam bentuk algoritma[3]. Logika *fuzzy* juga dapat digunakan dalam berbagai aplikasi

pemrosesan data yang tidak dapat direpresentasikan dalam bentuk biner. Selain itu logika *fuzzy* memiliki beberapa kelebihan diantaranya mampu beradaptasi dengan perubahan, mudah diimplementasikan, dan didasarkan pada bahasa alami. Dalam permainan komputer (*game*) logika *fuzzy* adalah teknik kecerdasan lainnya yang dapat digunakan untuk meningkatkan kinerja *game*. *Fuzzy* mampu menangani permasalahan yang kompleks dengan. komputasi yang rendah[4].

1.2. Permasalahan

Masalah yang diharapkan untuk ditemukan solusinya melalui tugas akhir ini adalah penghitungan *scoring* hasil akhir pada suatu permainan peperangan memiliki kecenderungan tidak memperhitungkan nilai – nilai dari seluruh aspek yang terdapat pada permainan.

1.3. Tujuan

Tujuan dan capaian yang diharapkan tercapai setelah selesainya tugas akhir ini adalah sebagai berikut:

1. Menghasilkan pemodelan *scoring* menggunakan logika *fuzzy* yang sesuai dengan kondisi keadaan saat permainan berakhir.
2. Menentukan hasil akhir permainan dengan logika *fuzzy* berdasarkan nilai yang diperoleh dari parameter yang terdapat pada permainan.

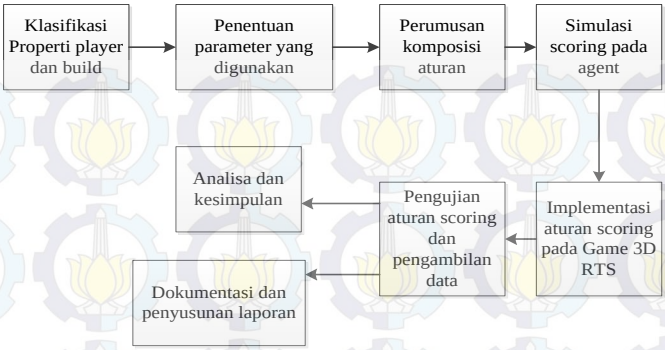
1.4. Batasan Masalah

Batasan masalah dari Tugas Akhir ini adalah:

1. Metode *fuzzy* yang digunakan adalah *fuzzy* Mamdani dengan fungsi keanggotaan segitiga dan trapesium.
2. Permainan yang dibuat adalah permainan *Real Time Strategy* yang bersifat *Single player*.
3. Permainan yang dibuat berbasis *Desktop Game.*

1.5. Metodologi

Metodologi yang digunakan pada pengerjaan tugas akhir ini dijabarkan dengan blok diagram pada Gambar 1.1.



Gambar 1.1 Metodologi penelitian

Penjelasan dari blok diagram Gambar 1.1 sebagai berikut:

1. Klasifikasi Properti *Player* dan *building*
   Tahap awal adalah pengklasifikasian properti *game* yang terdapat pada *player* dan *building* (bangunan).
2. Penentuan parameter yang digunakan
   Pada bagian ini dilakukan pengumpulan properti - properti yang efisien yang berhubungan dengan *player* dan *building* yang dapat dimanfaatkan dalam membuat pemodelan *scoring*. Parameter masukan ini digunakan sebagai variabel acuan yang akan diolah nantinya
3. Perumusan komposisi aturan
   Sebelum melakukan simulasi yang akan digunakan perlu ditentukan komposisi aturan *fuzzy* yang akan digunakan dalam *scoring*. Tahapan ini dilakukan untuk menentukan aturan yang diterapkan dalam *scoring* agar pemodelan *scoring* yang dibuat sesuai yang diharapkan.
4. Simulasi *scoring* pada agen.
   Simulasi pada *scoring* dilakukan pada *player* dan *building*. Pada *player* akan ditentukan beberapa kondisi sesuai komposisi aturan yang sudah dibuat. Jika terdapat komposisi aturan yang kurang sesuai dapat terlihat hasilnya pada tahapan ini.
5. Implementasi aturan *scoring* pada *Game* 3D RTS
   Setelah melakukan simulasi, selanjutnya hasilnya akan diterapkan pada *game* 3D RTS. *Tools* yang digunakan adalah Unity3D. Implementasi diterapkan pada *game* pertempuran RTS.

6. Pengujian aturan *scoring* dan pengambilan data
   Setelah implementasi dilakukan , maka proses pengukuran dapat dilakukan. Pada tahapan ini, dilakukan pengumpulan data untuk berbagai kondisi parameter yang telah ditentukan untuk pengujian. Tujuan akhirnya adalah menentukan komposisi aturan mana yang tepat untuk diterapkan.
7. Analisa dan Kesimpulan
   Analisa data dan pengambilan kesimpulan mengacu pada data hasil pengukuran.
8. Dokumentasi dan Penyusunan Laporan
   Tahapan ini meliputi pembuatan laporan tugas akhir dan penulisan jurnal ilmiah. Pembuatan laporan tugas akhir untuk beberapa bagian dilakukan bersesuaian dengan pengerjaan tahapan-tahapan diatas. Sedangkan jurnal ilmiah dilakukan setelah laporan tugas akhir selesai.

1.6. Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini terbagi dalam lima bab, masing-masing bab diuraikan sebagai berikut:

1. Bab 1 merupakan bab pendahuluan yang berisi latar belakang, permasalahan, tujuan tugas akhir, batasan masalah, metodologi, sistematika penulisan dan relevansi
2. Bab 2 merupakan bab yang membahas tentang teori penunjang dan literatur yang digunakan dalam penyusunan tugas akhir
3. Bab 3 membahas tentang perancangan sistem dan implementasi sistem
4. Bab 4 berisi tentang pengujian dari sistem yang telah dibangun
5. Bab 5 merupakan bab penutup laporan yang berisi tentang kesimpulan yang diperoleh dari pembuatan tugas akhir, serta saran-saran untuk pengembangan sistem lebih lanjut.

BAB 2
TEORI PENUNJANG

Untuk mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi dengan demikian penelitian ini lebih terarah

Pada penelitian [2] menunjukkan bahwa parameter yang efisien dapat digunakan sebagai acuan dalam aturan pembuatan *score* sederhana. Dengan menggunakan parameter yang efisien maka dihasilkan probabilitas aturan yang relevan dengan hasil yang diinginkan. Kemudian penelitian [1] membahas mengenai sistem *scoring* dalam permainan menggunakan pendekatan statistik dan pendekatan logika *fuzzy*. Hasil yang didapat menjelaskan bahwa pendekatan logika *fuzzy* jauh lebih fleksibel dibandingkan dengan statistik. Dengan mengacu pada hasil pertandingan dan perbedaan nilai antara pemain, hasil akhir peringkat pemain dapat berubah secara fleksibel sesuai parameter acuannya.

2.1 *Fantasy Chronicles*

Fantasy Chronicles 3D merupakan permainan ber*genre Real Time Strategy* yang dikembangkan oleh tim beranggotakan 5 orang dari lintas jalur teknik komputer dan telematika. Permainan ini memiliki latar belakang terjadi pertempuran antara 2 ras utama yaitu *White Warrior* (NPC *player*) dan *Black Warrior* (NPC musuh). Kedua ras tersebut memiliki 4 kelompok yang sama yang terdiri dari penyerangan, pertahanan, bangunan, dan pekerja. Dalam permainan ini kedua ras saling berhadapan untuk memperebutkan kekuasaan dan saling menghancurkan. Di kelompok penyerangan ras *White Warrior* memiliki 6 *macam* (NPC *player*) yang berperang menggunakan pedang dan sihir api. Sedangkan di ras *Black Warrior* memiliki 6 *macam* (NPC musuh) yang berperang menggunakan senjata (pedang, kapak,tongkat,gada) dan sihir api hitam. Permainan *Fantasy Chronicles* 3D ini meliputi 5 bagian penting yaitu perilaku NPC, *flocking*, strategi penyerangan, *scoring* dan *Gameplay*. Kelima unsur tersebut nantinya akan dijadikan satu dan diujicobakan ke *game* yang sebenarnya. Gambar 2.1 merupakan tampilan awal dari permainan *Fantasy Chronicles* 3D.

Gambar 2.1 Tampilan Awal Permainan *Fantasy Chronicles* 3D

2.2 *Game Real Time Strategy*

Real time strategy (RTS) Merupakan salah satu jenis permainan komputer yang mencakup *game* peperangan yang terjadi secara "*real time*". "*Real time"* berarti bahwa adanya interaksi langsung pada permainan antara komputer dan pengguna. Pada permainan RTS terdapat 2 aspek yaitu aspek ekonomi dan aspek militer sehingga permainan jenis RTS termasuk jenis permainan yang sulit untuk dikuasai. Dalam permainan RTS biasanya pemain diminta untuk mengumpulkan sumber daya, membangun tentara, dan berperang untuk mengalahkan musuh. Saat ini sudah banyak permainan RTS yang telah dirilis, antara lain yang terkenal adalah "StarCraft" , "Warcraft"," Command & Conquer " , " The Age of Empires, dan masih banyak lagi. Contoh lain dari permainan RTS adalah SimCity yang dirilis oleh EA Games dimana permainan ini tidak memiliki bagian perperangan tetapi fokus pada perencanaan kota dan ekonomi. *Platform* yang paling populer untuk memainkan permainan ini adalah komputer pribadi, tetapi beberapa permainan tersebut dirilis dalam bentuk lain seperti *Sony Playstation.* Sebuah survei yang dibuat oleh *The Entertainment Software Association* di Amerika Serikat menyatakan bahwa hampir

30.8% dari penjualan permainan untuk komputer pribadi adalah permainan RTS[5].

Gambar 2.2 Game RTS Command & Conquer (techxav.com)

Pada awal permainan yang sedang dimainkan , pemain biasanya memiliki sedikit sumber daya dan unit , tetapi memiliki kemungkinan memperoleh lebih banyak sumber daya . Tugasnya membangun atau memperluas markas, dengan tujuan mengumpulkan sumber daya dan menghabiskan sumber daya untuk memproduksi tentara . Pemain juga harus memastikan apakah akan mampu bertahan pada tahap awal dengan unit-unit produksi atau struktur defensif. Beberapa *game* memiliki cerita dengan tujuan tertentu yang harus dipenuhi , tetapi secara umum adalah tentang kontrol sumber daya, menempatkan unit untuk penggunaan yang baik dan mencapai kekuatan militer yang unggul . Permainan ini biasanya berakhir ketika seorang pemain telah menghancurkan semua musuhnya , atau ketika seorang pemain mencapai beberapa tujuan yang sudah ditentukan. Dalam permainan tujuan ini biasanya ditentukan dalam misi yang diletakan pada *storyline* dan biasanya dapat diakses saat bermain *game*.

2.3 *Scoring* pada Permainan

Dalam sebuah permainan, nilai biasanya mengacu pada kuantitas dari seorang pemain atau sebuah tim. Biasanya diukur berdasarkan poin yang terkumpul dan setiap kejadian yang ada dalam permainan dapat menambah atau mengurai poin dalam nilai. Pada kebanyakan *game* indikator yang digunakan adalah poin dari tingkat keberhasilan dalam permainan atau dalam permainan kompetisi *goal* untuk mendapatkan nilai yang lebih dari lawan juga dapat digunakan sebagai indikator.

Pada masa permaianan *video game* pemain akan terus bermain untuk mendapatkan nilai yang lebih tinggi dari sebelumnya. Sedangkan permaianan modern (era permaianan *arcade*) nilai tidak menjadi hal yang sangat penting karena pemain hanya fokus pada menyelesaikan permainan saja dengan waktu yang terbatas. Tetapi beberapa permainan memiliki beberapa poin yang digunakan sebagai kuantitas nilai seperti poin keterampilan, banyaknya poin yang dimiliki pemain dan sumber daya yang ada dapat digunakan sebagai bagian dari permaianan. Terkadang dalam permaianan poin dapat digunakan untuk membuka satu karakter atau fasilitas tambahan yang dapat digunakan karena sebelumnya terkunci. Poin juga dapat berfungsi untuk meningkatkan *level* pemain ketika pemain berhasil mengalahkan musuh utama kemudian dapat melanjutkan permainan ke tingkat selanjutnya.

2.4 Logika *Fuzzy*

Logika *Fuzzy* adalah salah satu sistem kecerdasan buatan (*Artificial Intelegent*) yang menerapkan cara berfikir manusia dimana dapat digunakan dalam berbagai aplikasi pemrosesan data. Pertama kali diperkenalkan oleh Prof. Lotfi Zadeh seorang guru besar di University of California di Berkeley pada tahun 1965. Dalam paparannya dijelaskan mengenai ide dasar *fuzzy set* yang meliputi *inclusion, union, intersection, complement, relation dan convexity*. Logika ini dapat diumpamakan sebagai kotak hitam yang berisi metode atau cara yang digunakan untuk mengolah data yang dapat menghubungkan antara masukan dan keluaran. Sehinggadapat digunakan menjadi informasi yang dapat dimanfaatkan[3].

Gambar 2.3 Contoh pemetaan *input output* (Gelley,2000)

Menurut [3] ada beberapa kelebihan dalam menggunakan logika *fuzzy* antara lain :

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi *nonlinear* yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* didasarkan pada bahasa alami. Logika *fuzzy* menggunakan bahasa sehari – hari sehingga mudah dimengerti.

Logika Fuzzy memiliki beberapa bagian dalam pembuatannya yaitu [3]:

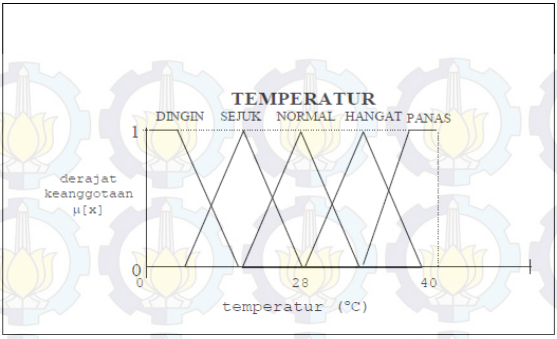
1. Himpunan *fuzzy*

   Terdapat dua atribut yang dimiliki himpunan *fuzzy* yaitu atribut linguistik dan atribut numeris. Linguistik adalah penamaan yang diberikan pada suatu domain yang merepresentasikan keadaan atau kondisi tertentu sedangkan numeris adalah suatu nilai (angka) yang menunjukkan jangkauan dari suatu variabel.salah satu contoh himpunan *fuzzy* terlihat pada Gambar 2.4.

   Nilai keanggotaan suatu anggota x dalam suatu himpunan *fuzzy* A(μA[x]) memiliki 2 kemungkinan yaitu:

   a. Satu (1), yang berarti suatu anggota menjadi bagian dalam suatu himpunan, atau
   b. Nol (0), yang berarti suatu anggota tidak menjadi bagian dalam suatu himpunan.

   Bagian lainnya dari himpunan *fuzzy* adalah semesta pembicaraan dan domain. Semesta pembicaraan adalah jangkauan nilai yang diperbolehkan untuk diolah dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan



Gambar 2.4 Himpunan *fuzzy* pada variable temperatur[3]

positif maupun negatif. Contoh dari semesta pembicaraan adalah:

a. Semesta pembicaraan untuk variabel nilai: [0 100]
b. Semesta pembicaraan untuk variabel temperatur : [0 40]

Sedangkan Domain adalah jangkauan nilai yang terdapat pada masing – masing himpunan *fuzzy* yang yang diijinkan untuk dioperasikan. Domain merupakan himpunan bilangan real yang naik (bertambah) secara monoton dari kiri ke kanan. Contoh domain himpunan fuzzy:

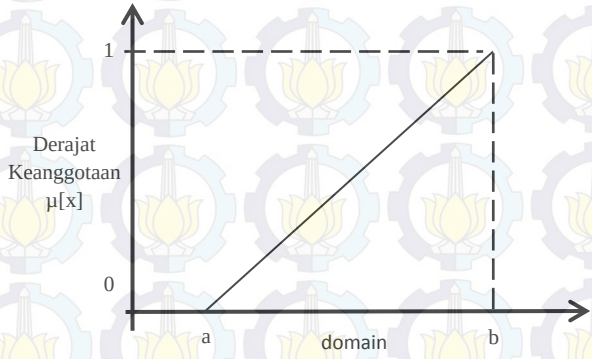
a. Sangat kurang = [0 30]
b. Kurang = [20 50]
c. Cukup = [45 75]
d. Baik = [70 85]
e. Sangat baik= [80 100]

2. Fungsi Keanggotaan

Merupakan Kurva yang digunakan untuk memetakan data masukan kedalam nilai keanggotaannya atau dapat disebut juga derajat keanggotaan. Salah satu cara yang digunakan untuk mendapatkan nilai derajat keanggotaan adalah dengan pendekatan fungsi. Terdapat beberapa pendekatan fungsi yang dapat digunakan yaitu[3] :

a. Representasi Linear

Pada representasi linear terdapat dua keadaan, pertama nilai yang dimulai dari nilai domain yang memiliki derajat keanggotaan nol(0) yang bergerak ke kanan (naik) menuju nilai derajat keanggotaan yang lebih tinggi seperti pada Gambar 2.5.



Gambar 2.5 Representasi linier naik[3]

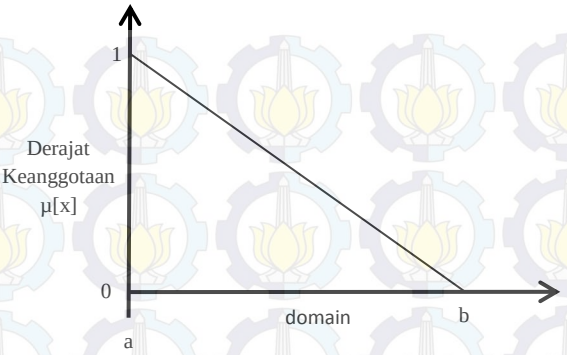
Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x-a) / (b-a); & a < x < b \\ 1; & x \geq b \end{cases} \quad (2.1)$$

Kedua, Nilai dimulai dari nilai domain yang memiliki derajat keanggotaan satu(1) yang bergerak ke kanan (turun) menuju nilai derajat keanggotaan yang lebih rendah seperti pada Gambar 2.6.

Fungsi keanggotaan:

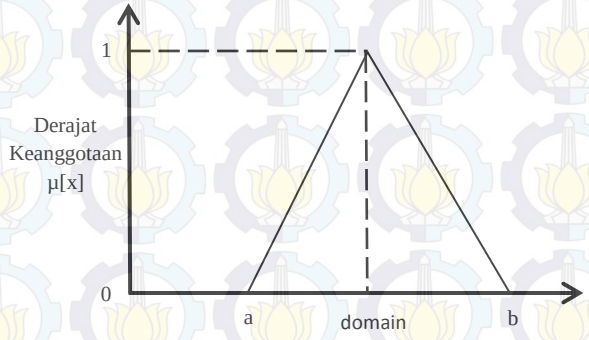
$$\mu[x] = \begin{cases} 0; & x \geq b \\ (b-x) / (b-a) & a < x < b \\ 1; & x \leq a \end{cases} \quad (2.2)$$



Gambar 2.6 Representasi linier turun[3]

b. Representasi Kurva Segitiga

Pada kurva segtiga terdapat 3 nilai yang menjadi acuannya yaitu 2 nilai yang bernilai derajat keanggotaan nol (0) dan satu nilai yang memiliki derajat keanggotaan satu (1) Seperti terlihat pada gambar 2.7.



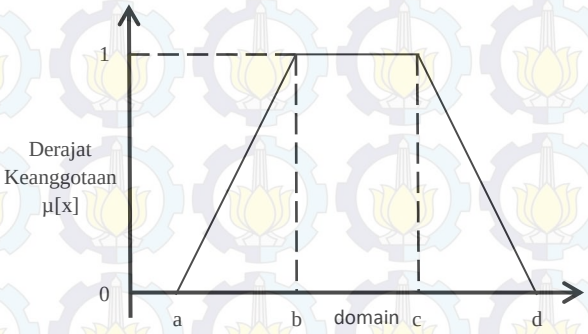
Gambar 2.7 Representasi Kurva Segitiga [3]

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \geq c \text{ atau } x \leq a \\ (x-a) / (b-a) & a < x < b \\ (c-x) / (c-b) & b < x < c \end{cases} \quad (2.3)$$

c. Representasi Kurva Trapesium

Pada kurva trapesium ada 4 nilai yang menjadi patokan. Dua nilai memiliki derajat keanggotaan nol (0) dan dua nilai lainnya memiliki derajat keanggotaan satu (1) yang terletak diantara nilai dengan derajat keanggotaan nol (0)



Gambar 2.8 Representasi Kurva Trapesium[3]

3. Fuzzyfikasi

Fuzzyfikasi adalah proses untuk mengubah variabel numerik menjadi variabel *fuzzy* (linguistik) dimana data masukan yang masih dalam bentuk numerik diubah terlebih dahulu menjadi variabel *fuzzy* melalui fungsi keanggotaan yang telah dibentuk sebelumnya sehingga nantinya data tersebut dapat digunakan menjadi suatu informasi yang akan diproses dalam pengolahan *fuzzy*.

Dapat dikatakan juga fuzzyfikasi merupakan pemetaan titik-titik numerik $x = (x^1,.....x^n)^T \in U$ ke himpunan *fuzzy* A di *U*. *U* adalah semesta pembicaraan. Paling tidak ada dua kemungkinan pemetaan,yaitu[3]:

1. Fuzzyfikasi *singleton*: A adalah *fuzzy singleton* dengan *support* x , artinya $\mu_A(x') = 1$ untuk x' = x dan $\mu_A(x') = 0$ untuk $x' \in U$ yang lain dengan x' = x .
2. Fuzzyfikasi *nonsingleton* : $\mu_A(x) = 1$ dan $\mu_A(x')$ menurun dari 1 sebagaimana x' bergerak menjauh dari x .

4. *Fuzzy Inference System* (FIS) Mamdani

Ada 2 jenis FIS yang paling kita kenal yaitu Mamdani dan Sugeno. Masukan yang digunakan dalam FIS adalah bilangan tertentu yang menghasilkan keluaran berupa bilangan tertentu. Penerapan bahasa linguistik dapat digunakan sebagai masukan yang sebelumnya dikonversi terlebih dahulu. Kemudian dilakukan penalaran berdasarkan aturan – aturan yang sudah disusun dan mengubahnya menjadi keluaran yang bersifat teliti.

Prof. Ebrahim Mamdani merupakan pelopor aplikasi *fuzzy set* dalam bidang kontrol. Bersama kawan – kawannya dari Queen Mary College London, Mamdani membangun aplikasi pertama dan utama dari *fuzzy set*. Metode Mamdani ini juga dikenal sebagai metode *Max-Min* yang diperkenalkan pada tahun 1975. Terdapat empat tahapan untuk mendapat keluaran dalam FIS Mamdani yaitu[3]:

a. Pembentukan Himpunan *Fuzzy*

Pembentukan himpunan *fuzzy* bertujuan untuk memberikan jangkauan nilai yang diperbolehkan dalam suatu kurva yang setiap nilainya memiliki derajat keanggotaan antara 0 sampai dengan 1. Pada logika *boolean* digambarkan nilai "benar" dan "salah" sedangkan pada *fuzzy* digambarkan dengan ungkapan misalnya : "sangat lambat","agak sedang", "sangat cepat"dan lain-lain untuk mengambarkan setiap domainnya.

b. Aplikasi fungsi implikasi (aturan)

Dari himpunan *fuzzy* yang telah dibentuk, proses selanjutnya adalah membuat aturan – aturan *fuzzy* yang kemudian akan diolah dalam fungsi implikasi. Dari aturan – aturan *fuzzy* tersebut, setiap aturan akan menghasilkan nilai yang diperoleh dari fungsi implikasi yang ditetapkan Untuk FIS Mamdani fungsi implikasi yang digunakan adalah *Min*, dimana nilai derajat keanggotaan yang paling kecil dari masing – masing aturan yang akan diambil nilainya.

c. Komposisi aturan

Pada tahapan ini, setelah dilakukan fungsi implikasi kemudian dari setiap nilai yang diperoleh akan di gabungkan menjadi satu kurva komposisi aturan. Berbeda dengan penalaran boolean yang memiliki nilai benar dan

salah, pada FIS Mamdani jika terdapat beberapa aturan dalam sistem maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Menurut [3] terdapat 3 metode yang digunakan dalam inferensi sistem *fuzzy*, yaitu:

1) Metode *Max* (*Maximum*)
   Metode *Max* akan menghasilkan keluaran yang berupa daerah (himpunan) yang diperoleh dari fungsi implikasi nilai derajat keanggotaan masing – masing aturan yang selanjutnya seluruh aturan digabungkan dengan menggunakan operator OR sehingga akan menjadi kurva yang memiliki daerah berdasarkan gabungan dari hasil komposisi aturan. Secara umum dapat dituliskan:

   $$\mu sf[xi] \leftarrow max(\mu sf[xi], \mu kf[xi]) \quad (2.4)$$

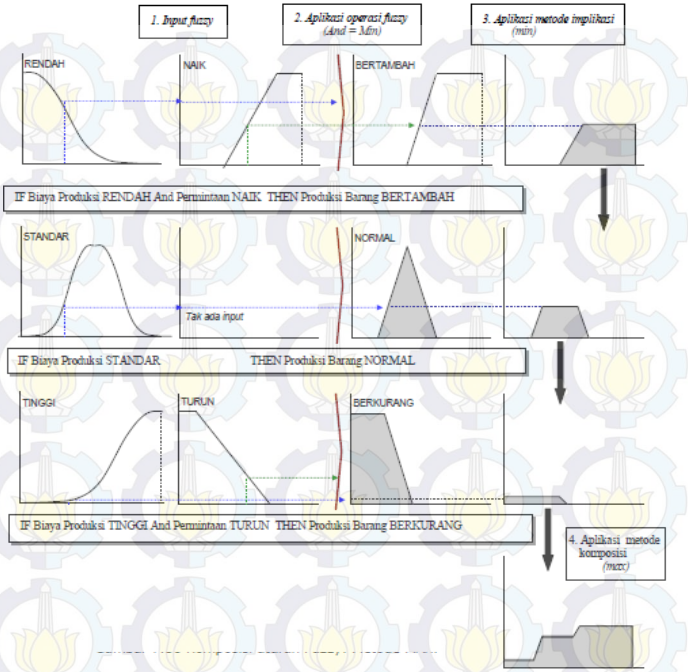
   dengan:
   a) μsf[xi] = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;
   b) μkf[xi] = nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

   Contoh inferensi dengan menerapkan metode *Max* untuk proses komposisi aturan seperti terlihat pada Gambar 2.9.

2) Metode *Addotive* (Sum)
   Setelah mendapatkan hasil komposisi aturan, pada metode Sum akan dilakukan *bounded-sum* yang akan menghasilkan solusi himpunan *fuzzy.*
   Secara umum dituliskan:

   $$\mu sf[xi] \leftarrow \mu min(1, \mu sf[xi] + \mu kf[xi]) \quad (2.2)$$

   dengan:
   a) μsf[xi] = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;
   b) μkf[xi] = nilai keanggotaan konsekuen *fuzzy* aturan ke-i;



Gambar 2.9 Komposisi Aturan *Fuzzy* : Metode *Max*[3]

3) Metode Probabilistik OR (probor)
Untuk metode Probor hasil akhir himpunan *fuzzy* diperoleh dengan cara melakukan *dot product* terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan:

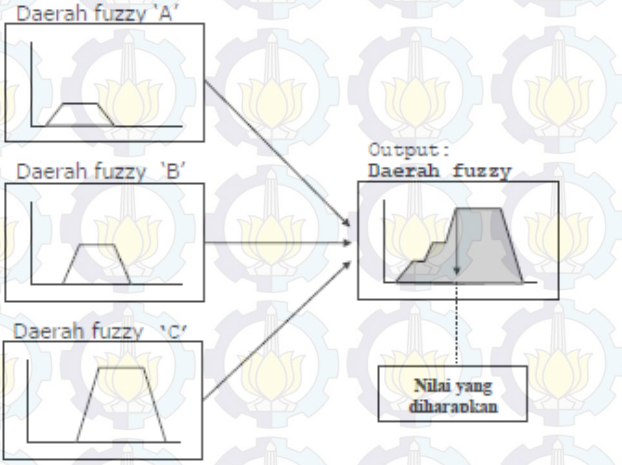
$$\mu sf[xi] \leftarrow \mu\ (\mu sf[xi] + \mu kf[xi]) - (\mu sf[xi] * \mu kf[xi]) \quad (2.3)$$

dengan:

a) μsf[xi] = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;
b) μkf[xi] = nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

d. Penegasan (defuzzy)

Setelah melalui proses komposisi aturan selanjutnya hasil tersebut akan digunakan sebagai masukan dalam proses defuzzyfikasi yang akan menghasilkan suatu nilai yang terdapat pada domain himpunan *fuzzy*. Sehingga jika terdapat suatu himpunan *fuzzy* dalam jangkauan tertentu maka hasil keluarannya diambil dari suatu nilai *crsip* yang ada pada himpunan *fuzzy* tersebut seperti terlihat pada Gambar 2.10.



Gambar 2.10 Proses Defuzzyfikasi [3]

Berdasarkan [3] ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani yaitu :

1) Metode Centroid (*Composite Moment*)

Hasil akhir logika *fuzzy* didapatkan dengan cara mengambil titik pusat (z*) daerah *fuzzy*. Secara umum dirumuskan:

$$Z^* = \frac{\int_Z Z\mu(Z)dz}{\int_Z \mu(Z)dz} \qquad (2.5)$$

$$Z^* = \frac{\sum_{j-1}^{n} Zj\mu(Zj)}{\sum_{j-1}^{n} \mu(Zj)} \quad (2.6)$$

2) Metode *Mean of Maximum* (MOM)
Pada metode ini, hasil akhir didapatkan dengan mengambil dilai rata – rata dari domain yang mempunyai nilai keanggotaan paling tinggi (maksimum).

$$v_{o=} \sum_{j=1}^{j} \frac{v_j}{J} \quad (2.7)$$

$$v_j = v\, \mu_v (v) \quad (2.8)$$

v_o : nilai keluaran
J : jumlah harga maksimum
v_j : nilai keluaran maksimum ke - j
$\mu_v(v)$: derajat keanggotaan elemen – elemen pada fuzzy set v
v : semesta pembicaraan

3) Metode Bisektor
Untuk metode bisektor hasil akhir didapat dengan cara mengambil nilai pada domain *fuzzy* yang dengan nilai keanggotaannya setengah dari keseluruhan total nilai keanggotaan pada daerah *fuzzy*. Secara umum dituliskan:

$$Z_{p\ sedemikian\ hingga} \int_{\Re 1}^{p} \mu(z)dz = \int_{p}^{\Re n} \mu(z)dz \quad (2.6)$$

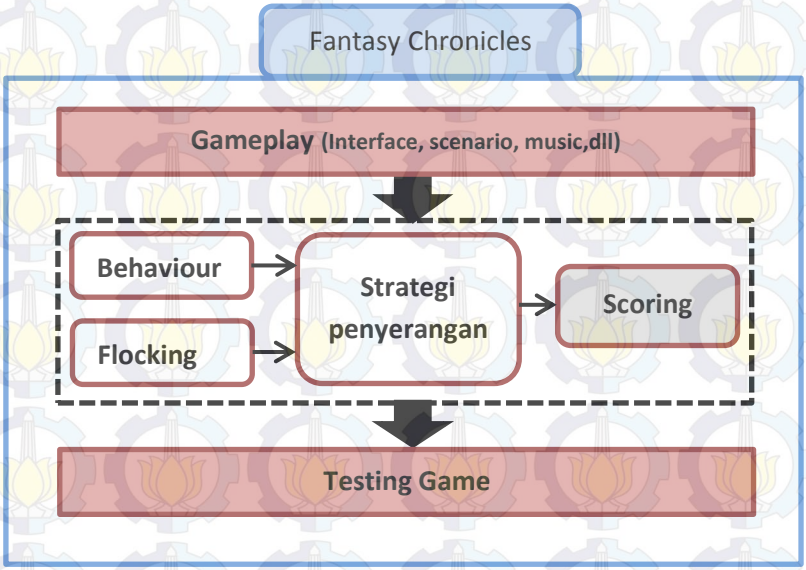
4) Metode *Largest of Maximum* (LOM)
Setelah mendapatkan kurva dari hasil komposisi aturan, pada metode LOM untuk mendapatkan nilai keluaran hasil akhir dari logika *fuzzy* didapatkan dari nilai terbesar yang terdapat pada domain yang memiliki nilai keanggotaan maksimum

5) Metode *Smallest of Maximum* (SOM)
Metode SOM berkebalikan dengan metode LOM jika pada metode LOM yang diambil adalah nilai terbesarnya sedangkan pada metode SOM yang digunakan sebagai solusi logika *fuzzy* adalah nilai terkecil yang terdapat pada nilai keanggotaan yang paling tinggi.

--Halaman ini sengaja dikosongkan--

BAB 3
PERANCANGAN SISTEM

Pada bab ini dibahas mengenai rancangan desain sistem yang akan diterapkan pada sebuah permainan(*game*) *Fantasy Chronicles* dimana permainan (*game)* ini dalam bentuk tiga dimensi yang mengambil tema atau *genre* semi RTS, g*ame* ini dikerjakan oleh lima orang mahasiswa jurusan Teknik Elektro ITS. Rancangan sistem *game Fantasy Chronicles* secara menyeluruh dari penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Desain *game Fantasy Chronicles 3D*

Dari blok diagram pada gambar 3.1, pada penelitian ini yang akan dibahas hanya pada sistem *scoring* yang akan digunakan pada permainan tersebut. Penelitian meliputi perancangan sistem, perancangan variabel, dan perancangan aturan *fuzzy*.

3.1. Desain Sistem

Rancangan desain sistem akan dibagi menjadi empat tahapan untuk mendapatkan hasil yang sesuai. Tahapan desain sistem yang akan dikerjakan ditunjukan seperti pada Gambar 3.2

Gambar 3.2 Tahapan desain sistem

Tahapan pertama yang dilakukan adalah menentukan parameter apa saja yang akan digunakan sebagai masukan yang akan diolah pada sistem *fuzzy*. Setelah menentukan parameter selanjutnya ditentukan fungsi keanggotaan untuk setiap parameter yang berupa representasi kurva yang akan menggambarkan domain dari parameter tersebut. Selanjutnya akan dibuat aturan *fuzzy* untuk sistem *scoring* yang mencakup parameter yang sudah ditentukan sebelumnya. Aturan *fuzzy* akan dibuat untuk menentukan hasil akhir permainan yang berupa predikat yang akan didapatkan pemain yang direpresentasikan dalam bentuk bintang.

3.2. Perancangan Variable masukan

Parameter yang digunakan sebagai masukan merupakan properti yang dimiliki oleh unit yang terdapat pada *game*. Untuk menghitung *scoring* dari satu permainan digunakan parameter Kayu, Poin, Markas dan Pasukan untuk mendapatkan hasil predikat pemain pada setiap permainan.

3.2.1 Parameter Kayu

Kayu merupakan *resource* utama yang dimiliki oleh masing – masing unit. *Resource* akan bertambah 15 poin setiap 5 detik dan akan berkurang jika digunakan untuk membangun pasukan sesuai harga dari masing – masing pasukan. Jumlah kayu yang dihitung untuk hasil akhir permainan adalah sisa terakhir kayu pada saat permainan berakhir. Kayu memiliki jumlah maksimal sebesar 1000 yang berarti lumbung kayu penuh dan jumlah minimal 0 yang berarti kayu kosong.

3.2.2. Parameter Poin

Parameter Poin diperoleh dari perbandingan antara jumlah poin yang dikumpulan jika pemain berhasil mengalahkan lawan dan poin yang dipakai saat membangun pasukan. Masing – masing tipe lawan memiliki poin yang berbeda- beda. Untuk menghitung nilai parameter poin digunakan rumus:

*(poin dipakai / point didapat) *100* (3.1)

Hasil nilai poin akan berkisar antara 0 – 100. Nilai 0 berarti poin yang didapat tidak ada yang digunakan. Nilai 100 berarti poin yang didapat digunakan semua.

3.2.3. Parameter Pasukan

Pasukan adalah perbandingan antara pasukan yang dibangun oleh pemain dan jumlah pasukan pemain yang mati. Untuk menghitung nilai parameter pasukan digunakan rumus:

*(pasukan yang mati / pasukan yang dibangun) *100* (3.2)

Hasil nilai pasukan akan berkisar antara 0 – 100. Nilai 0 berarti pasukan yang dibangun tidak ada yang mati. Sedangan nilai 100 berarti semua pasukan pemain mati melawan musuh.

3.2.4. Parameter Markas

Parameter markas merupakan jumlah nilai yang diperoleh dari *health point* masing – masing bangunan yang dimiliki pemain pada saat permainan berakhir. Setiap bangunan memiliki *health point* sebesar

1000 karena pemain memiliki tiga markas maka jumlah total maksimal keseluruhan markas sebesar 3000. Besar nilai markas akan berkisar antara 0 – 3000. Nilai 0 berarti bangunan pemain tidak ada yang tersisa atau kalah. Nilai 3000 berarti bangunan pemain tidak ada yang hancur atau utuh.

3.2.5. Parameter Predikat

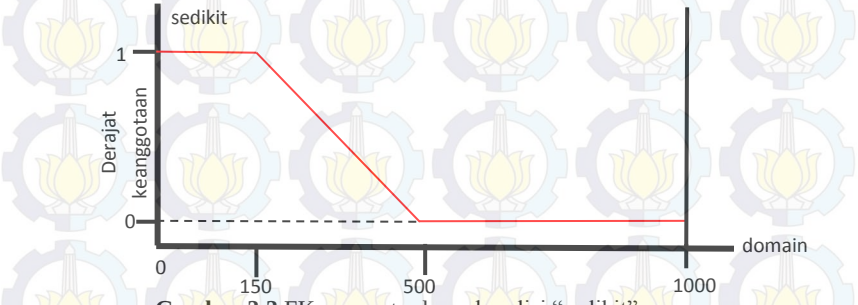
Predikat adalah *level* yang didapat pemain pada akhir permainan berdasarkan hasil permainannya yang dihitung dari kayu, poin, pasukan dan markas. Predikat memiliki range dari 0 – 10, nilai 0 berarti permainan pemain buruk (*poor*) sedangkan nilai 10 berarti permainan pemain sangat baik (*excelent*).

3.3. Perancangan Fungsi Keanggotaan Kayu

Fungsi keanggotaan digunakan untuk merepresentasikan parameter kayu pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga dan kurva trapesium .

3.3.1. Fungsi Keanggotaan Parameter Kayu Kondisi "sedikit"

Untuk kondisi "sedikit" digunakan representasi kurva trapesium dengan jangkauan antara 0 sampai dengan 500. Memiliki nilai maksimum pada range 0 – 150 dan nilai minimum pada nilai 500. Untuk fungsi keanggotaan kayu kondisi "sedikit" terlihat seperti pada Gambar 3.3.



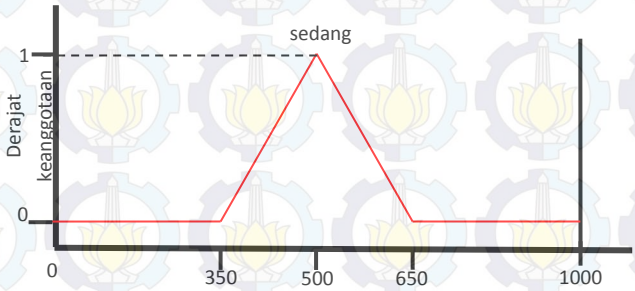
Gambar 3.3 FK parameter kayu kondisi "sedikit"

Untuk menghitung derajat keanggotaan parameter kayu kondisi "sedikit" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{kayu}Sedikit(x) = \begin{cases} 1, x \leq 150 \\ (500-x)/(500-150), 150 \leq x \leq 500 \\ 0, x \geq 500 \end{cases} \quad (3.3)$$

3.3.2. Fungsi Keanggotaan Parameter Kayu Kondisi "sedang"

Untuk kondisi "sedang" digunakan representasi kurva segitiga dengan jangkauan antara 350 sampai dengan 650. Memiliki nilai maksimum pada nilai 500 dan nilai minimum pada nilai 350 dan 650. Untuk fungsi keanggotaan kayu kondisi "sedang" terlihat seperti pada Gambar 3.4.



Gambar 3.4 FK parameter kayu kondisi "sedang"

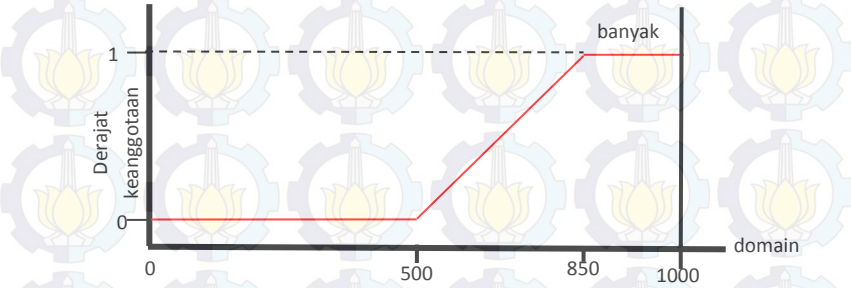
Untuk menghitung derajat keanggotaan parameter kayu kondisi "sedang" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{kayu}Sedang(x) = \begin{cases} 0, x \leq 350 \; dan \; x \geq 650 \\ (x-350)/(500-350), 350 \leq x \leq 500 \\ (650-x)/(650-500), 500 \leq x \leq 650 \end{cases} \quad (3.4)$$

3.3.3. Fungsi Keanggotaan Parameter Kayu Kondisi "banyak"

Untuk kondisi "banyak" digunakan representasi kurva trapesium dengan jangkauan antara 500 sampai dengan 1000. Memiliki nilai maksimum pada range 850 – 1000 dan nilai minimum pada nilai

500. Untuk fungsi keanggotaan kayu kondisi "banyak" terlihat seperti pada Gambar 3.5.



Gambar 3.5 FK parameter kayu kondisi "banyak"

Untuk menghitung derajat keanggotaan parameter kayu kondisi "banyak" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{kayu}Banyak(x) = \begin{cases} 0, x \leq 500 \\ (x-500)/(850-500), 500 \leq x \leq 850 \\ 1, 850 \leq x \leq 1000 \end{cases} \quad (3.5)$$

3.4. Perancangan Fungsi Keanggotaan Poin

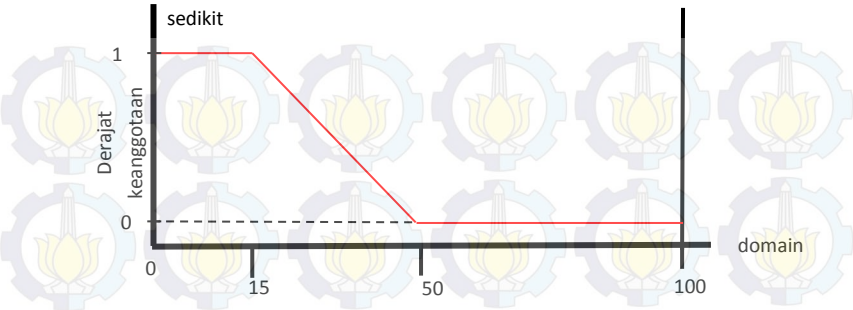
Fungsi keanggotaan digunakan untuk merepresentasikan parameter poin pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga dan kurva trapesium .

3.4.1. Fungsi Keanggotaan Parameter Poin Kondisi "sedikit"

Untuk kondisi "sedikit" digunakan representasi kurva trapesium dengan jangkauan antara 0 sampai dengan 50. Memiliki nilai maksimum pada range 0 – 15 dan nilai minimum pada nilai 50. Untuk fungsi keanggotaan poin kondisi "sedikit" terlihat seperti pada Gambar 3.6.

Untuk menghitung derajat keanggotaan parameter poin kondisi "sedikit" dari setiap nilai yang dimasukan digunakan rumus :

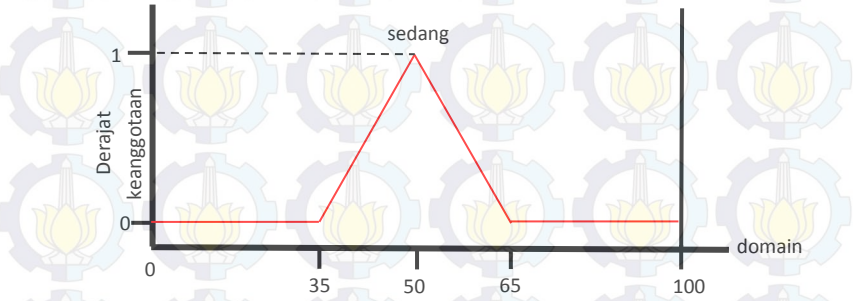
$$\mu_{poin}sedikit(x) = \begin{cases} 0, x \geq 50 \\ (50-x)/(50-15), 15 \leq x \leq 50 \\ 1, x \leq 15 \end{cases} \quad (3.6)$$



Gambar 3.6 FK parameter poin kondisi "sedikit"

3.4.2. Fungsi Keanggotaan Parameter Poin Kondisi "sedang"

Untuk kondisi "sedang" digunakan representasi kurva segitiga dengan jangkauan antara 35 sampai dengan 65. Memiliki nilai maksimum pada nilai 50 dan nilai minimum pada nilai 35 dan 65. Untuk fungsi keanggotaan poin kondisi "sedang" terlihat seperti pada Gambar 3.7.



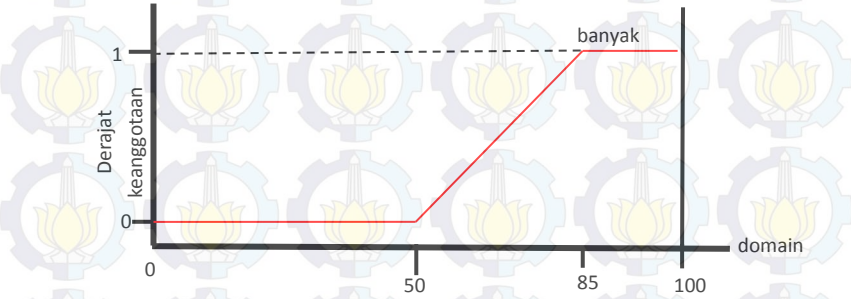
Gambar 3.7 FK parameter poin kondisi "sedang"

Untuk menghitung derajat keanggotaan parameter poin kondisi "sedang" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{poin}sedang(x) = \begin{cases} 0, x \leq 35 \; atau \; x \geq 65 \\ (x-35)/(50-35), 35 \leq x \leq 50 \\ (65-x)/(65-50), 50 \leq x \leq 65 \end{cases} \quad (3.7)$$

3.4.3. Fungsi Keanggotaan Parameter Poin Kondisi "banyak"

Untuk kondisi "banyak" digunakan representasi kurva trapesium dengan jangkauan antara 50 sampai dengan 100. Memiliki nilai maksimum pada range 85 – 100 dan nilai minimum pada nilai 50. Untuk fungsi keanggotaan poin kondisi "banyak" terlihat seperti pada Gambar 3.8.



Gambar 3.8 FK parameter poin kondisi "banyak"

Untuk menghitung derajat keanggotaan parameter poin kondisi "banyak" dari setiap nilai yang dimasukan digunakan rumus :

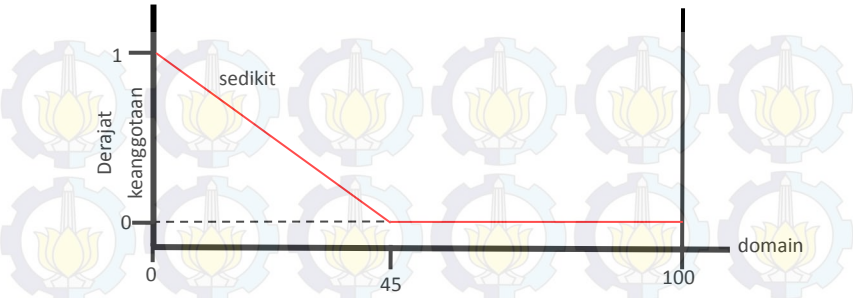
$$\mu_{poin}Banyak(x) = \begin{cases} 0\,, x \leq 50 \\ (x-50)/(85-50)\,, 50 \leq x \leq 85 \\ 1\,, 85 \leq x \leq 100 \end{cases} \quad (3.8)$$

3.5. Perancangan Fungsi Keanggotaan Pasukan

Fungsi keanggotaan digunakan untuk merepresentasikan parameter pasukan pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga.

3.5.1. Fungsi Keanggotaan Parameter Pasukan Kondisi "sedikit"

Untuk kondisi "sedikit" digunakan representasi kurva segitiga dengan jangkauan antara 0 samapi dengan 45. Memiliki nilai maksimum pada nilai 0 dan nilai minimum pada nilai 45. Untuk fungsi keanggotaan pasukan kondisi "sedikit" terlihat seperti pada Gambar 3.9



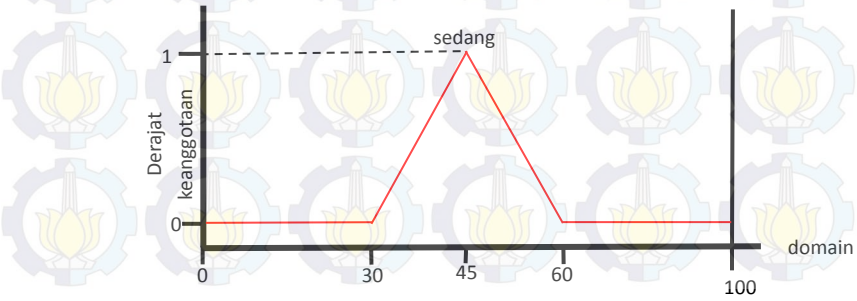
Gambar 3.9 FK parameter pasukan kondisi "sedikit"

Untuk menghitung derajat keanggotaan parameter pasukan kondisi "sedikit" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pas}Sedikit(x) = \begin{cases} 1, x \leq 0 \\ (45-x)/(45-0), 0 \leq x \leq 45 \\ 0, x \geq 45 \end{cases} \quad (3.9)$$

3.5.2. Fungsi Keanggotaan Parameter Pasukan Kondisi "sedang"

Untuk kondisi "sedang" digunakan representasi kurva segitiga dengan jangkauan antara 30 sampai dengan 60. Memiliki nilai maksimum pada range nilai 45 dan nilai minimum pada nilai 60 dan 30. Untuk fungsi keanggotaan pasukan kondisi "sedang" terlihat seperti pada Gambar 3.10.



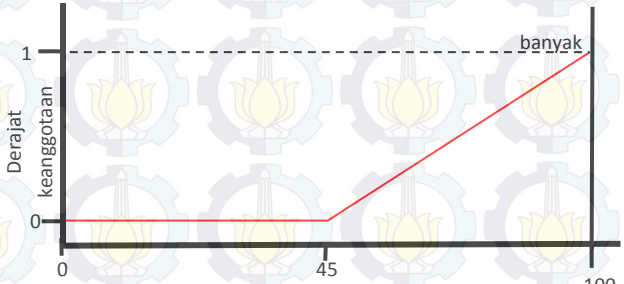
Gambar 3.10 FK parameter pasukan kondisi "sedang"

Untuk menghitung derajat keanggotaan parameter pasukan kondisi “sedang” dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pas}Sedang(x) = \begin{cases} (45-x)/(45-30), 30 \le x \le 45 \\ (x-45)/(60-45), 45 \le x \le 60 \\ 0, x \le 30 \; atau \; x \ge 60 \end{cases} \quad (3.10)$$

3.5.3. Fungsi Keanggotaan Parameter Pasukan Kondisi “banyak”

Untuk kondisi “banyak” digunakan representasi kurva segitiga dengan jangkauan antara 45 sampai dengan 100. Memiliki nilai maksimum nilai 100 dan nilai minimum pada nilai 45. Untuk fungsi keanggotaan pasukan kondisi “banyak” terlihat seperti pada Gambar 3.11.



Gambar 3.11 FK parameter pasukan kondisi “banyak”

Untuk menghitung derajat keanggotaan parameter pasukan kondisi “banyak” dari setiap nilai yang dimasukan digunakan rumus :

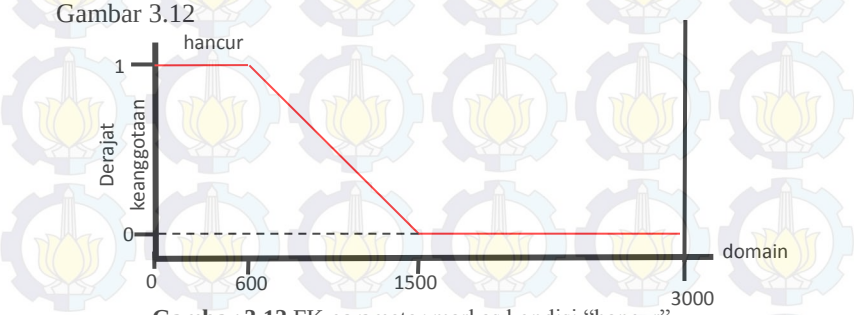
$$\mu_{pas}Banyak(x) = \begin{cases} 0, x \le 45 \\ (100-x)/(100-45), 45 \le x \le 100 \\ 1, x \ge 100 \end{cases} \quad (3.11)$$

3.6. Perancangan Fungsi Keanggotaan Markas

Fungsi keanggotaan digunakan untuk merepresentasikan parameter markas pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga dan trapesium.

3.6.1. Fungsi Keanggotaan Parameter Markas Kondisi "hancur"

Untuk kondisi "hancur" digunakan representasi kurva segitiga dengan jangkauan antara 0 sampai dengan 1500. Memiliki nilai maksimum pada range 0 - 600 dan nilai minimum pada nilai 1500. Untuk fungsi keanggotaan markas kondisi "hancur" terlihat seperti pada Gambar 3.12



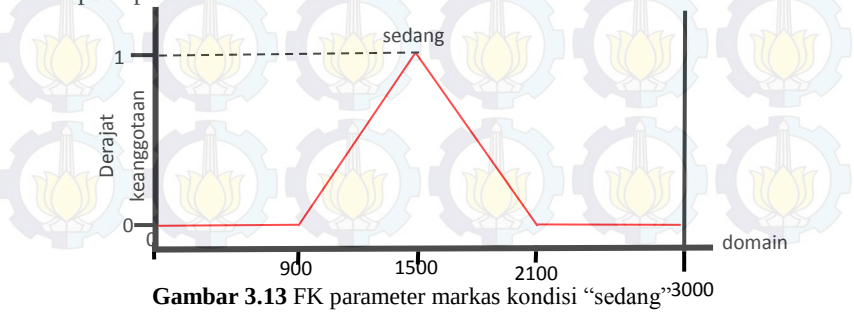
Gambar 3.12 FK parameter markas kondisi "hancur"

Untuk menghitung derajat keanggotaan parameter markas kondisi "hancur" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{mar}Hancur(x) = \begin{cases} 1, x \leq 600 \\ (x-600)/(1500-600), 600 \leq x \leq 1500 \\ 0, x \geq 1500 \end{cases} \quad (3.12)$$

3.6.2. Fungsi Keanggotaan Parameter Markas Kondisi "sedang"

Untuk kondisi "sedang" digunakan representasi kurva segitiga dengan jangkauan antara 900 sampai dengan 2100. Memiliki nilai maksimum pada range 1500 dan nilai minimum pada nilai 900 dan 1500. Untuk fungsi keanggotaan markas kondisi "sedang" terlihat seperti pada Gambar 3.13



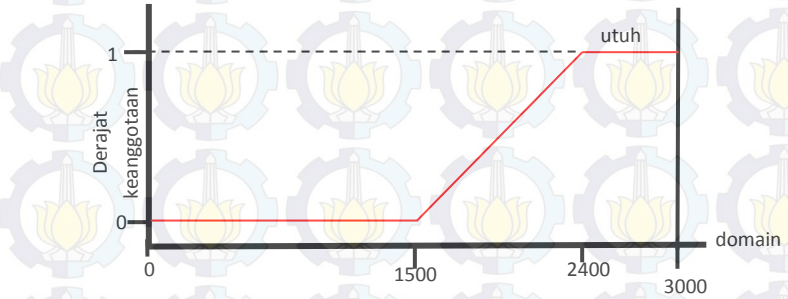
Gambar 3.13 FK parameter markas kondisi "sedang"

Untuk menghitung derajat keanggotaan parameter markas kondisi “sedang” dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{mar}sedang(x) = \begin{cases} 0\,, x \leq 900\ atau\ x \geq 2100 \\ (1500 - x)/(1500 - 900)\,, 900 \leq x \leq 1500 \\ (x - 1500)/(2100 - 1500)\,, 1500 \leq x \leq 2100 \end{cases} \quad (3.13)$$

3.6.3. Fungsi Keanggotaan Parameter Markas Kondisi “utuh”

Untuk kondisi “utuh” digunakan representasi kurva trapesium dengan jangkauan antara 1500 sampai dengan 3000. Memiliki nilai maksimum pada range 2400 - 3000 dan nilai minimum pada nilai 1500. Untuk fungsi keanggotaan markas kondisi “utuh” terlihat seperti pada Gambar 3.14



Gambar 3.14 FK parameter markas kondisi “utuh”

Untuk menghitung derajat keanggotaan parameter markas kondisi “utuh” dari setiap nilai yang dimasukan digunakan rumus :

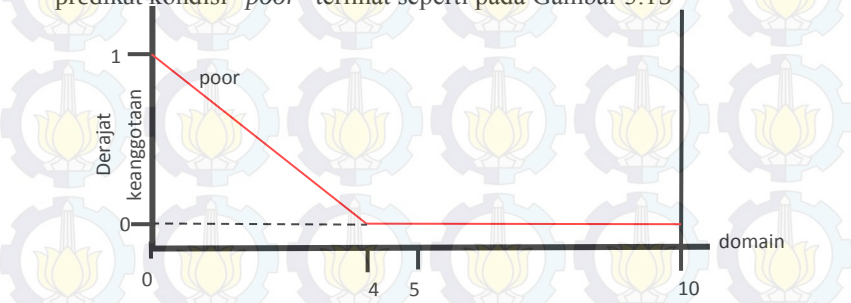
$$\mu_{mar}Utuh(x) = \begin{cases} 0\,, x \leq 1500 \\ (2400 - x)/(2400 - 1500)\,, 1500 \leq x \leq 2400 \\ 1, x \geq 2400 \end{cases} \quad (3.14)$$

3.7. Perancangan Fungsi Keanggotaan Predikat

Fungsi keanggotaan digunakan untuk merepresentasikan parameter predikat pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga.

3.7.1. Fungsi Keanggotaan Parameter Predikat Kondisi "*poor*"

Untuk kondisi "*poor*" digunakan representasi kurva segitiga dengan jangkauan antara 0 sampai dengan 4. Memiliki nilai maksimum pada nilai 0 dan nilai minimum pada nilai 4. Untuk fungsi keanggotaan predikat kondisi "*poor*" terlihat seperti pada Gambar 3.15



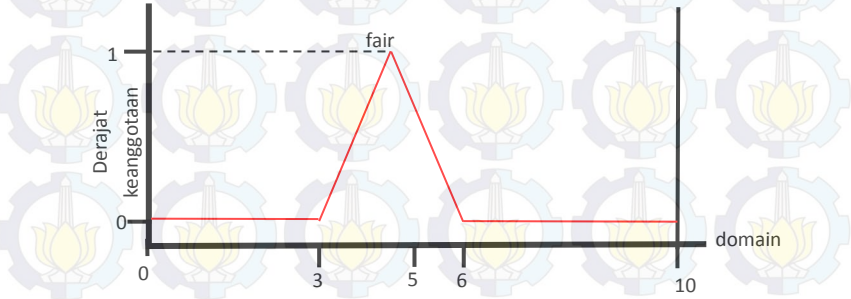
Gambar 3.15 FK parameter predikat kondisi "*poor*"

Untuk menghitung derajat keanggotaan parameter predikat kondisi "*poor*" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pre}poor(x) = \begin{cases} 1, x \leq 0 \\ (4-x)/(4-0), 0 \leq x \leq 4 \\ 0, x \geq 4 \end{cases} \quad (3.15)$$

3.7.2. Fungsi Keanggotaan Parameter Predikat Kondisi "*fair*"

Untuk kondisi "*fair*" digunakan representasi kurva segitiga dengan jangkauan antara 3 sampai dengan 6. Memiliki nilai maksimum pada nilai 4.5 dan nilai minimum pada nilai 4 dan 6. Untuk fungsi keanggotaan predikat kondisi "*fair*" terlihat seperti pada Gambar 3.16



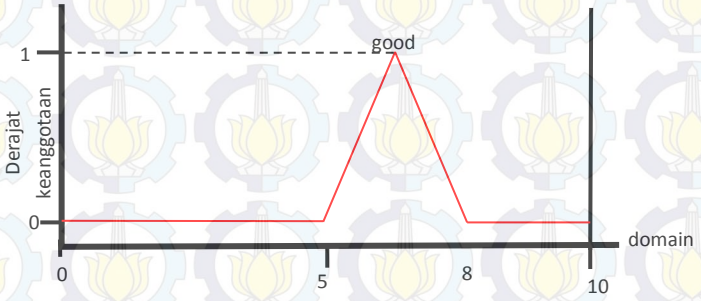
Gambar 3.16 FK parameter predikat kondisi "*fair*"

Untuk menghitung derajat keanggotaan parameter predikat kondisi "*fair*" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pre}fair(x) = \begin{cases} 0\,, x \leq 3 \; atau \; x \geq 6 \\ (4.5 - x)/(4.5 - 3)\,, 3 \leq x \leq 4.5 \\ (6 - x)/(6 - 4.5)\,, 4.5 \leq x \leq 6 \end{cases} \quad (3.16)$$

3.7.3. Fungsi Keanggotaan Parameter Predikat Kondisi "*good*"

Untuk kondisi "*good*" digunakan representasi kurva segitiga dengan jangkauan antara 5 sampai dengan 8. Memiliki nilai maksimum pada nilai 6.5 dan nilai minimum pada nilai 5 dan 8. Untuk fungsi keanggotaan predikat kondisi "*good*" terlihat seperti pada Gambar 3.17



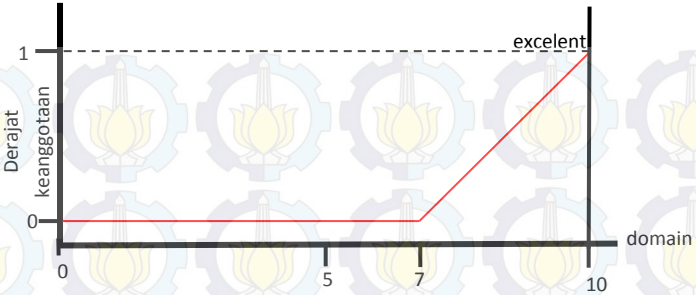
Gambar 3.17 FK parameter predikat kondisi "good"

Untuk menghitung derajat keanggotaan parameter predikat kondisi "*good*" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pre}Good(x) = \begin{cases} 0\,, x \leq 5 \; atau \; x \geq 8 \\ (6.5 - x)/(6.5 - 5)\,, 5 \leq x \leq 6.5 \\ (8 - x)/(8 - 6.5)\,, 6.5 \leq x \leq 8 \end{cases} \quad (3.17)$$

3.7.4. Fungsi Keanggotaan Parameter Predikat Kondisi "*excelent*"

Untuk kondisi "*excelent*" digunakan representasi kurva segitiga dengan jangkauan antara 7 sampai dengan 10. Memiliki nilai maksimum pada nilai 10 dan nilai minimum pada nilai 7. Untuk fungsi keanggotaan predikat kondisi "*excelent*" terlihat seperti pada Gambar 3.18



Gambar 3.18 FK parameter predikat kondisi "*excelent*"

Untuk menghitung derajat keanggotaan parameter predikat kondisi "*excelent*" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pre}Exc(x) = \begin{cases} 0\,, x \leq 7 \\ (x-7)/(10-7)\,, 7 \leq x \leq 10 \\ 1\,, x \geq 10 \end{cases} \quad (3.18)$$

3.8. Desain Aturan *Fuzzy (Rules Base)*

Aturan *fuzzy* berfungsi untuk memberikan aturan *IF -THEN* pada parameter yang sebelumnya sudah dirancang. Aturan *fuzzy* digunakan untuk menentukan predikat dengan parameter kayu, poin, markas dan pasukan. *Rules based* untuk predikat seperti pada Tabel 3.4 terdiri dari 81 aturan

Tabel 3.4: *Rule based fuzzy* Predikat

Rules	Masukan				Keluaran
	Markas	Kayu	Poin	Pasukan	Predikat
1	Hancur	Sedikit	Sedikit	Sedikit	*Fair*
2	Hancur	Sedikit	Sedikit	Sedang	*Fair*
3	Hancur	Sedikit	Sedikit	Banyak	*Poor*
4	Hancur	Sedikit	Sedang	Sedikit	*Poor*
5	Hancur	Sedikit	Sedang	Sedang	*Poor*
6	Hancur	Sedikit	Sedang	Banyak	*Poor*
7	Hancur	Sedikit	Banyak	Sedikit	*Poor*
8	Hancur	Sedikit	Banyak	Sedang	*Poor*
9	Hancur	Sedikit	Banyak	Banyak	*Poor*
10	Hancur	Sedang	Sedikit	Sedikit	*Fair*
11	Hancur	Sedang	Sedikit	Sedang	*Fair*

12	Hancur	Sedang	Sedikit	Banyak	*Poor*
13	Hancur	Sedang	Sedang	Sedikit	*Poor*
14	Hancur	Sedang	Sedang	Sedang	*Poor*
15	Hancur	Sedang	Sedang	Banyak	*Poor*
16	Hancur	Sedang	Banyak	Sedikit	*Poor*
17	Hancur	Sedang	Banyak	Sedang	*Poor*
18	Hancur	Sedang	Banyak	Banyak	*Poor*
19	Hancur	Banyak	Sedikit	Sedikit	*Fair*
20	Hancur	Banyak	Sedikit	Sedang	*Fair*
21	Hancur	Banyak	Sedikit	Banyak	*Poor*
22	Hancur	Banyak	Sedang	Sedikit	*Fair*
23	Hancur	Banyak	Sedang	Sedang	*Fair*
24	Hancur	Banyak	Sedang	Banyak	*Poor*
25	Hancur	Banyak	Banyak	Sedikit	*Fair*
26	Hancur	Banyak	Banyak	Sedang	*Poor*
27	Hancur	Banyak	Banyak	Banyak	*Poor*
28	Sedang	Sedikit	Sedikit	Sedikit	*Good*
29	Sedang	Sedikit	Sedikit	Sedang	*Good*
30	Sedang	Sedikit	Sedikit	Banyak	*Fair*
31	Sedang	Sedikit	Sedang	Sedikit	*Fair*
32	Sedang	Sedikit	Sedang	Sedang	*Fair*
33	Sedang	Sedikit	Sedang	Banyak	*Fair*
34	Sedang	Sedikit	Banyak	Sedikit	*Good*
35	Sedang	Sedikit	Banyak	Sedang	*Fair*
36	Sedang	Sedikit	Banyak	Banyak	*Fair*
37	Sedang	Sedang	Sedikit	Sedikit	*Good*
38	Sedang	Sedang	Sedikit	Sedang	*Good*
39	Sedang	Sedang	Sedikit	Banyak	*Fair*
40	Sedang	Sedang	Sedang	Sedikit	*Good*
41	Sedang	Sedang	Sedang	Sedang	*Good*
42	Sedang	Sedang	Sedang	Banyak	*Fair*
43	Sedang	Sedang	Banyak	Sedikit	*Fair*
44	Sedang	Sedang	Banyak	Sedang	*Fair*
45	Sedang	Sedang	Banyak	Banyak	*Fair*
46	Sedang	Banyak	Sedikit	Sedikit	*Good*
47	Sedang	Banyak	Sedikit	Sedang	*Good*
48	Sedang	Banyak	Sedikit	Banyak	*Good*

49	Sedang	Banyak	Sedang	Sedikit	*Good*
50	Sedang	Banyak	Sedang	Sedang	*Good*
51	Sedang	Banyak	Sedang	Banyak	*Fair*
52	Sedang	Banyak	Banyak	Sedikit	*Fair*
53	Sedang	Banyak	Banyak	Sedang	*Fair*
54	Sedang	Banyak	Banyak	Banyak	*fair*
55	Utuh	Sedikit	Sedikit	Sedikit	*Good*
56	Utuh	Sedikit	Sedikit	Sedang	*Good*
57	Utuh	Sedikit	Sedikit	Banyak	*Good*
58	Utuh	Sedikit	Sedang	Sedikit	*Excellent*
59	Utuh	Sedikit	Sedang	Sedang	*Good*
60	Utuh	Sedikit	Sedang	Banyak	*Good*
61	Utuh	Sedikit	Banyak	Sedikit	*Good*
62	Utuh	Sedikit	Banyak	Sedang	*Good*
63	Utuh	Sedikit	Banyak	Banyak	*Good*
64	Utuh	Sedang	Sedikit	Sedikit	*Excellent*
65	Utuh	Sedang	Sedikit	Sedang	*Good*
66	Utuh	Sedang	Sedikit	Banyak	*Good*
67	Utuh	Sedang	Sedang	Sedikit	*Excellent*
68	Utuh	Sedang	Sedang	Sedang	*Excellent*
69	Utuh	Sedang	Sedang	Banyak	*Good*
70	Utuh	Sedang	Banyak	Sedikit	*Good*
71	Utuh	Sedang	Banyak	Sedang	*Good*
72	Utuh	Sedang	Banyak	Banyak	*Good*
73	Utuh	Banyak	Sedikit	Sedikit	*Excellent*
74	Utuh	Banyak	Sedikit	Sedang	*Excellent*
75	Utuh	Banyak	Sedikit	Banyak	*Good*
76	Utuh	Banyak	Sedang	Sedikit	*Excellent*
77	Utuh	Banyak	Sedang	Sedang	*Excellent*
78	Utuh	Banyak	Sedang	Banyak	*Good*
79	Utuh	Banyak	Banyak	Sedikit	*Good*
80	Utuh	Banyak	Banyak	Sedang	*Good*
81	Utuh	Banyak	Banyak	Banyak	*Good*

--Halaman ini sengaja dikosongkan--

BAB 4
PENGUJIAN DAN ANALISA

Pada bab ini dilakukan pengujian dan analisa terhadap sistem *scoring* yang telah dibangun untuk mengetahui apakah sistem dapat diimplementasikan dan sesuai yang diharapan.

4.1 Skenario Pengujian

Pengujian dilakukan terhadap model *fuzzy* yang sebelumnya sudah ditentukan. Pengujian dilakukan dengan mengambil beberapa *sample* data yang diimplementasikan pada *game*. Untuk pengujian, sistem *fuzzy* diimplementasikan pada permainan yang telah dibangun. Pengujian dilakukan pada 2 tahapan. Pertama pengujian implementasi parameter yang ada pada sistem dan kedua pengujian penerapan metode dengan menerapkan dua kondisi permainan yaitu kondisi menang dan kondisi kalah dengan variasi hasil akhir yang berbeda – beda yang hasil predikatnya akan direpresentasikan dalam bentuk bintang..

4.2 Pengujian Implementasi Parameter

Pengujian parameter ini bertujuan untuk mengetahui apakah parameter yang akan digunakan sudah berjalan sesuai dengan fungsinya agar tidak terjadi kesalahan ketika parameter tersebut akan digunakan dalam sistem *scoring*. Pengujian parameter dilakukan pada semua parameter yang terlibat. Implementasi tiap - tiap parameter yang akan digunakan terlihat seperti pada Gambar 4.1.

Gambar 4.1 Implementasi variabel masukan

Pada gambar 4.1 terlihat implementasi pada masing – masing parameter berjalan sesuai dengan fungsinya. Pada parameter kayu jumlah kayu pada saat gambar diambil terlihat berjumlah 574 kayu hasil tersebut didapatkan dari total kayu awal sebesar 500 ditambahkan dengan kayu yang didapat selama 1 menit 18 detik sebesar 234 dan dikurangkan dengan kayu yang dipakai sebesar 160 sehingga sisa kayu sebesar 574.). Perhitungan untuk parameter kayu sebagai berikut :

kayu awal + Kayu selema 1 menit 18 detik – kayu dipakai

= 500 + 234 – 160 = 574

Untuk parameter poin jumlah poin sebesar 16 diperoleh dari point yang dipakai sebesar 48 dibagi dengan poin yang didapat sebesar 300 dikalikan dengan 100. Perhitungan untuk parameter kayu sebagai berikut :

(poin dipakai / poin didapai) 100 = (48/300) * 100 = 16*

Pada parameter pasukan dihitung dari jumlah pasukan yang mati sebesar 3 dibagi dengan pasukan yang dibangun sebesar 12 dikalikan dengan 100 sehingga menghasilkan nilai 25. Perhitungan untuk parameter kayu sebagai berikut :

(pasukan yang mati / pasukan yang dingaun)) 100 = (3/12) * 100 = 25*

Sedangkan pada markas bernilai 3000 didapat dari jumlah *health point* markas yang dimiliki pemain.

4.3 Pengujian Sistem *Scoring*

Pengujian sistem *scoring* dilakukan untuk mengetahui apakah *scoring* yang dibuat dengan *fuzzy* sudah berjalan dengan baik. Pengujian dilakukan dengan membandingkan hasil akhir permainan dengan beberapa kondisi predikat yang sudah ditentukan .

4.3.1. Pengujian Hasil Akhir Kondisi "*Poor*"

Pada pengujian ini akan diperoleh hasil akhir permainan dengan predikat "*poor*" yang mana diperoleh dari hasil pertempuran yang menghasilkan kondisi markas "hancur", kondisi kayu "sedikit", kondisi poin "sedikit", dan kondisi pasukan "banyak" pada akhir permainan. Gambar pertempuran hingga hasil akhir terlihat pada gambar 4.2.

Gambar 4.2 Hasil Akhir Kondisi "*poor*"
(a) Markas musuh utuh
(b) Markas musuh sisa 2
(c) Markas musuh sisa 1
(d) Markas musuh hancur
(e) Hasil akhir pertempuran

4.3.2. Pengujian Hasil Akhir Kondisi "*Fair*"

Pada pengujian ini akan diperoleh hasil akhir permainan dengan predikat "*fair*" yang mana diperoleh dari hasil pertempuran yang menghasilkan kondisi markas "sedang", kondisi kayu "sedikit", kondisi poin "sedikit", dan kondisi pasukan "banyak" pada akhir permainan. Gambar pertempuran hingga hasil akhir terlihat pada gambar 4.3.

(a)

(b)

(c)

(d)

(e)

Gambar 4.3 Hasil Akhir Kondisi "*fair*"
(a) Markas musuh utuh
(b) Markas musuh sisa 2
(c) Markas musuh sisa 1
(d) Markas pemain sisa 2
(e) Hasil akhir pertempuran

4.3.3. Pengujian Hasil Akhir Kondisi "*Good*"

Pada pengujian ini akan diperoleh hasil akhir permainan dengan predikat "*good*" yang mana diperoleh dari hasil pertempuran yang menghasilkan kondisi markas "utuh", kondisi kayu "sedikit", kondisi poin "sedikit", dan kondisi pasukan "banyak" pada akhir permainan. Gambar pertempuran hingga hasil akhir terlihat pada gambar 4.4.

(a)

(b)

(c)

(d)

(e)

Gambar 4.4 Hasil Akhir Kondisi "*Good*"
(a) Markas musuh utuh
(b) Markas musuh sisa 2
(c) Markas musuh sisa 1
(d) Markas pemain sisa 2
(e) Hasil akhir pertempuran

4.3.4. Pengujian Hasil Akhir Kondisi "*Excelent*"

Pada pengujian ini akan diperoleh hasil akhir permainan dengan predikat "*Excelent*" yang mana diperoleh dari hasil pertempuran yang menghasilkan kondisi markas "utuh", kondisi kayu "banyak", kondisi poin "sedikit", dan kondisi pasukan "sedikit" pada akhir permainan. Gambar pertempuran hingga hasil akhir terlihat pada gambar 4.5.

(a)

(b)

(c)

(d)

(e)

Gambar 4.5 Hasil Akhir Kondisi "*excelent*"
(a) Markas musuh utuh
(b) Markas musuh sisa 2
(c) Markas musuh sisa 1
(d) Markas musuh hancur
(e) Hasil akhir pertempuran

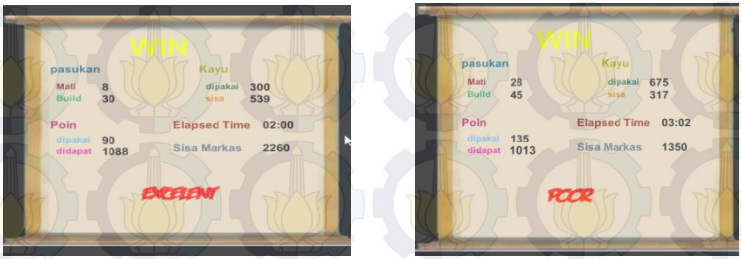
4.4 Pengujian Metode

Pengujian metode bertujuan untuk mengetahui apakah metode yang digunakan dapat diterapkan dalam permaianan. Dan juga untuk mengetahui hasil akhir permainan yang dilakukan dengan menggunakan skenario pertempuran. Pengujian dilakukan dengan kondisi permainan menang dan kalah dengan menggunakan metode *fuzzy* dan dibandingkan dengan tanpa menggunakan metode *fuzzy*.

4.4.1 Pengujian Hasil Akhir kondisi Menang

Pada pengujian ini akan dilakukan sepuluh kali pengujian untuk mendapatkan hasil akhir permainan yang berbeda - beda. Kemudian hasilnya akan dibandingkan dengan penghitungan hasil akhir tanpa menggunakan *fuzzy* (berdasarkan aturan tetap / statis). Penghitungan hasil akhir permainan akan melibatkan 4 parameter yaitu sisa *health point* markas pemain yang berhasil dipertahankan, perbandingan pasukan yang dibangun oleh pemain dan pasukan yang dibunuh musuh, perbandingan poin yang dipakai dan yang didapat oleh pemain, serta sisa kayu yang dimiliki pemain saat permainan berakhir.

Perhitungan hasil akhir tanpa menggunakan *fuzzy* memiliki aturan yaitu jika pada akhir permaianan markas pemain utuh maka mendapatkan bintang 3. Jika markas pemain sisa dua mendapatkan bintang 2. Dan jika markas pemain sisa satu maka mendapatkan bintang 1.Sedangkan untuk penghitungan hasil akhir dengan *fuzzy* pemain akan mendapatkan bintang antara 0 – 3 yang nilainya sesuai dengan nilai keluaran yang dihasilkan dari penghitungan logika *fuzzy*. Pada Gambar 4.5 terlihat salah satu hasil akhir permainan yang mendapat predikat "*excelent*" dan "*poor*" dengan hasil menang. Hasil dari sepuluh kali pengujian dengan hasil menang terdapat pada Tabel 4.1



Gambar 4.5 pengujian metode hasil permainan menang, predikat "*excelent*" dan "*poor*"

Tabel 4.1: Hasil Pengujian kondisi menang

Uji ke-	Markas	Kayu	Poin	Pasukan	Hasil penghitungan *fuzzy*	Representasi bintang	
						Fuzzy	Statis
1	1000	101	22	83	2.1667		
2	1850	4	31	60	5.499		
3	2000	28	17	60	5.736		
4	3000	578	8	20	8.76		
5	2260	539	8	26	8.74		
6	2560	107	17	51	6.584		
7	3000	647	32	27	8.681		
8	2140	119	18	65	6.584		
9	2410	56	15	45	6.650		
10	1350	317	13	62	3.291		

Pada hasil pengujian Table 4.1 terdapat beberapa hasil dengan bintang yang sama tetapi memiliki kondisi yang berbeda. Pada pengujian ke -6 , ke- 8, ke -9 mendapatkan bintang 2 pada pengujian *fuzzy* tetapi pada pengujian statis ketiganya mendapat bintang tiga. Untuk hasil lainnya, pada pengujian ke -4 sampai dengan ke -9 pada pengujian statis menujukan hasil bintang yang sama. Besar persentase hasil bintang yang sama antara statis dan *fuzzy* adalah sebesar 60% . Persentase tersebut didapat dari hasil perolehan bintang yang terdapat pada pengujian ke 1,2,3,4,5,dan 7. Sedangkan persentase hasil bintang yang tidak sama antara statis dan *fuzzy* sebesar 40% yang diperoleh dari pengujian ke 6,8,9, dan10.

Dari hasil pengujian Table 4.1 jika dihitung nilai standar deviasinya (variasi data) maka nilai untuk pengujian *fuzzy* sebesar 0.7378. sedangkan pada pengujian statis nilainya sebesar 0.7017. dari kedua niai tersebut menunjukan bawah pengujian *fuzzy* memberikan nilai lebih bervariasi / lebih dinamik dariada pengujian statis karena memiliki nilai standar deviasi yang lebih besar.

4.4.2 Pengujian Hasil Akhir kondisi Kalah

Pada pengujian kondisi kalah ini hanya akan dilakukan tiga kali pengujian untuk mendapatkan hasil akhir permainan dengan kondisi kalah. Kemudian hasilnya akan dibandingkan dengan penghitungan hasil akhir tanpa menggunakan *fuzzy* (berdasarkan aturan tetap / statis).

Perhitungan hasil akhir tanpa menggunakan *fuzzy* untuk kondisi kalah memiliki aturan yaitu pemain tidak akan mendapatkan bintang jika kondisinya kalah. Sedangkan untuk penghitungan hasil akhir dengan *fuzzy* bintang yang didapat antara 0 – 3 yang nilainya sesuai dengan nilai keluaran yang dihasilkan dari penghitungan logika *fuzzy*. Pada Gambar 4.6 terlihat salah satu hasil akhir permainan dengan hasil kalah. Hasil dari tiga kali pengujian dengan hasil kalah terdapat pada Tabel 4.2

Gambar 4.6 pengujian metode hasil permainan kalah predikat “poor”

Tabel 4.2: Hasil Pengujian kondisi kalah

Uji ke-	Markas	Kayu	Poin	Pasukan	Hasil penghitungan *fuzzy*	Representasi Bintang	
						Fuzzy	Statis
1	0	150	35	98	1.607		-
2	0	13	21	87.5	1.390		-
3	0	502	47	85	1.491		-

Dari hasil pengujian pada table 4.2 terdapat hasil bintang yang sama tetapi memiliki kondisi yang berbeda. Karena dalam aturan *fuzzy* untuk kondisi kalah / markas hancur hanya terdapat dua kemungkinan predikat yaitu "*poor*" dan "*fair*". Sedangkan pada hasil akhir tanpa *fuzzy* pemain tidak akan mendapatkan hasil apapun jika kalah. Hal ini menjelaskan bahwa dengan *fuzzy* kita tetap dapat memberi nilai pada hasil kerja pemain walaupun pemain kalah dalam permainan.

--Halaman ini sengaja dikosongkan--

DAFTAR PUSTAKA

[1] Graf Alan.“ Fuzzy Logic Approach for Modelling Multiplayer Game Scoring System”, Siemens d.d, Croatia, 2005.

[2] Johanna M. M. Goertz and Fran¸cois Maniquet, “On the Informational Efficiency of Simple Scoring Rules”, University of Guelph and Universit´e catholique de Louvain, 2008.

[3] Kusumadewi,S & Purnomo. 2004. “Aplikasi Logika Fuzzy Untuk Pendukung Keputusan”. Yogyakarta.Graha ilmu

[4] Li, Yifan, Musilek Petr, & Wyard-Scott. “Fuzzy Logic in Agent-Based Game Design”, Department of Electrical and Computer Engineering,University of Alberta Canada, 2004. IEEE 0-7803-8376-1/04

[5] Stene Sindre Berg. “Artificial Intelligence Techniques in Real-Time Strategy Games - Architecture and Combat Behavior”, Department of Computer and Information Science , Norwegian University of Science and Technology, 2006

[6] Rollings. Andrew, Dave Morris. “*Game Architecture and Design”*. Indianapolis, Indiana. New Riders Publisher, 2004.

--Halaman ini sengaja dikosongkan--

BAB 5
PENUTUP

5.1. Kesimpulan

Dari hasil pengujian Tugas Akhir ini pada bab sebelumnya dapat disimpulkan bahwa :

1. Nilai standar deviasi (variasi data) untuk pengujian *fuzzy* sebesar 0.7378 sedangkan pada pengujian statis nilainya sebesar 0.7017.
2. Penghitungan scoring menggunakan *fuzzy* memberikan nilai lebih bervariasi / lebih dinamis daripada penghitungan statis karena memiliki nilai standar deviasi yang lebih besar
3. Besar persentase hasil bintang yang sama antara penghitungan statis dan *fuzzy* sebesar 60%. Sisanya sebesar 40% adalah hasil perolehan bintang yang berbeda.

5.2. Saran

1. Dapat ditambahkan parameter – parameter lainnya yang terlibat dalam permainan yang dapat dimanfaatkan untuk menentukan hasil akhir permainan.
2. Permainan dapat dikembangkan menjadi permainan *multiplayer* dan dapat memanfaatkan *fuzzy* hasil akhir sebagai referensi aturan untuk peringkat (*ranking*) pemain karena hasilnya yang dinamis .

--Halaman ini sengaja dikosongkan--

LAMPIRAN

Result.CS

```
using UnityEngine;
using System.Collections;
using AForge.Fuzzy;
using System;

public class result : MonoBehaviour
{
	// Use this for initialization
	public GUIText playerkill;
	public GUIText currentgold;
	public GUIText currentpoin;
	public GUIText elapsedtime;
	public GUIText hasilstar;
	public GUIText
playerbuild,status,buildtx,deadtx,goldlosttx,pointlosttx,predikat;
	public GUITexture bintang;
	float kill,poin,build,dead,pointlost,goldlost;
	int bangunan;
	float gold;
	string time,sts;
	private InferenceSystem IS;
	float hasil,NewAngle;
	double newBarWidth,z;

	void fuzzyfuzzya()
	{
		// Linguistic labels (fuzzy sets) that compose the
distances
		//menentukan variable fuzzy kayu
		FuzzySet fsgsedikit = new FuzzySet("sedikit", new
TrapezoidalFunction(150, 500, TrapezoidalFunction.EdgeType.Right));
		FuzzySet fsgsedang = new FuzzySet("sedang", new
TrapezoidalFunction(350, 500, 650));
```

```
                FuzzySet fsgbanyak = new FuzzySet("banyak", new
TrapezoidalFunction(500, 1000, TrapezoidalFunction.EdgeType.Left));

                LinguisticVariable lvgold = new
LinguisticVariable("gold", 0, 1000);
                lvgold.AddLabel(fsgsedikit);
                lvgold.AddLabel(fsgsedang);
                lvgold.AddLabel(fsgbanyak);
                //----------------------------//

                //menentukan variable fuzzy POINT
                FuzzySet fspsedikit = new FuzzySet("sedikit", new
TrapezoidalFunction(15, 50, TrapezoidalFunction.EdgeType.Right));
                FuzzySet fspsedang = new FuzzySet("sedang", new
TrapezoidalFunction(35,50,65));
                FuzzySet fspbanyak = new FuzzySet("banyak", new
TrapezoidalFunction(50,100, TrapezoidalFunction.EdgeType.Left));

                LinguisticVariable lvpoint = new
LinguisticVariable("point", 0, 100);
                lvpoint.AddLabel(fspsedikit);
                lvpoint.AddLabel(fspsedang);
                lvpoint.AddLabel(fspbanyak );

                //----------------------------//

                //menentukan variable fuzzy MARKAS
                FuzzySet fsbhancur = new FuzzySet("hancur", new
TrapezoidalFunction(600, 1500, TrapezoidalFunction.EdgeType.Right));
                FuzzySet fsbsedang = new FuzzySet("sedang", new
TrapezoidalFunction(900, 1500, 2100));
                FuzzySet fsbutuh = new FuzzySet("utuh", new
TrapezoidalFunction(1500, 2400, TrapezoidalFunction.EdgeType.Left));

                LinguisticVariable lvbangunan = new
LinguisticVariable("markas", 0, 3000);
                lvbangunan.AddLabel(fsbhancur);
```

```
                lvbangunan.AddLabel(fsbsedang);
                lvbangunan.AddLabel(fsbutuh );
                //--------------------------//

                //menentukan variable fuzzy pasukan
                FuzzySet fsksedikit = new FuzzySet("sedikit", new
TrapezoidalFunction(0, 45, TrapezoidalFunction.EdgeType.Right));
                FuzzySet fsksedang = new FuzzySet("sedang", new
TrapezoidalFunction(30, 40, 60));
                FuzzySet fskbanyak = new FuzzySet("banyak", new
TrapezoidalFunction(45, 100, TrapezoidalFunction.EdgeType.Left));

                LinguisticVariable lvek = new LinguisticVariable("ek",
0, 100);
                lvek.AddLabel(fsksedikit);
                lvek.AddLabel(fsksedang);
                lvek.AddLabel(fskbanyak);
                //--------------------------//


                // Linguistic labels (fuzzy sets) that compose the angle
                // menentukan variable fuzzy predikat
                FuzzySet fspemula = new FuzzySet("poor", new
TrapezoidalFunction(0, 4, TrapezoidalFunction.EdgeType.Right));
                FuzzySet fsinter = new FuzzySet("fair", new
TrapezoidalFunction(3, 4, 6));
                FuzzySet fsadvan = new FuzzySet("good", new
TrapezoidalFunction(5, 7, 8));
                FuzzySet fsexpert = new FuzzySet("exc", new
TrapezoidalFunction(7, 10, TrapezoidalFunction.EdgeType.Left));

                // predikat
                LinguisticVariable lvpredikat = new
LinguisticVariable("predikat", 0, 10);
                lvpredikat.AddLabel(fspemula);
                lvpredikat.AddLabel(fsinter);
                lvpredikat.AddLabel(fsadvan);
```

```
lvpredikat.AddLabel(fsexpert);

// The database
Database fuzzyDB = new Database();
fuzzyDB.AddVariable(lvgold);
fuzzyDB.AddVariable(lvpoint);
fuzzyDB.AddVariable(lvbangunan);
fuzzyDB.AddVariable(lvek);
fuzzyDB.AddVariable(lvpredikat);

// Creating the inference system
IS = new InferenceSystem(fuzzyDB, new
CentroidDefuzzifier(1000));
IS.NewRule("Rule 1", "IF markas IS hancur AND gold IS
sedikit AND point IS sedikit AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 2", "IF markas IS hancur AND gold IS
sedikit AND point IS sedikit AND ek IS sedang THEN predikat IS fair");
IS.NewRule("Rule 3", "IF markas IS hancur AND gold IS
sedikit AND point IS sedikit AND ek IS banyak THEN predikat IS poor");
IS.NewRule("Rule 4", "IF markas IS hancur AND gold IS
sedikit AND point IS sedang AND ek IS sedikit THEN predikat IS poor");
IS.NewRule("Rule 5", "IF markas IS hancur AND gold IS
sedikit AND point IS sedang AND ek IS sedang THEN predikat IS poor");
IS.NewRule("Rule 6", "IF markas IS hancur AND gold IS
sedikit AND point IS sedang AND ek IS banyak THEN predikat IS poor");
IS.NewRule("Rule 7", "IF markas IS hancur AND gold IS
sedikit AND point IS banyak AND ek IS sedikit THEN predikat IS poor");
IS.NewRule("Rule 8", "IF markas IS hancur AND gold IS
sedikit AND point IS banyak AND ek IS sedang THEN predikat IS poor");
IS.NewRule("Rule 9", "IF markas IS hancur AND gold IS
sedikit AND point IS banyak AND ek IS banyak THEN predikat IS poor");
IS.NewRule("Rule 10", "IF markas IS hancur AND gold
IS sedang AND point IS sedikit AND ek IS sedikit THEN predikat IS fair");

IS.NewRule("Rule 11", "IF markas IS hancur AND gold
IS sedang AND point IS sedikit AND ek IS sedang THEN predikat IS fair");
```

```
IS.NewRule("Rule 12", "IF markas IS hancur AND gold
IS sedang AND point IS sedikit AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 13", "IF markas IS hancur AND gold
IS sedang AND point IS sedang AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 14", "IF markas IS hancur AND gold
IS sedang AND point IS sedang AND ek IS sedang THEN predikat IS
fair");
IS.NewRule("Rule 15", "IF markas IS hancur AND gold
IS sedang AND point IS sedang AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 16", "IF markas IS hancur AND gold
IS sedang AND point IS banyak AND ek IS sedikit THEN predikat IS
poor");
IS.NewRule("Rule 17", "IF markas IS hancur AND gold
IS sedang AND point IS banyak AND ek IS sedang THEN predikat IS
poor");
IS.NewRule("Rule 18", "IF markas IS hancur AND gold
IS sedang AND point IS banyak AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 19", "IF markas IS hancur AND gold
IS banyak AND point IS sedikit AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 20", "IF markas IS hancur AND gold
IS banyak AND point IS sedikit AND ek IS sedang THEN predikat IS fair");

IS.NewRule("Rule 21", "IF markas IS hancur AND gold
IS banyak AND point IS sedikit AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 22", "IF markas IS hancur AND gold
IS banyak AND point IS sedang AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 23", "IF markas IS hancur AND gold
IS banyak AND point IS sedang AND ek IS sedang THEN predikat IS
fair");
IS.NewRule("Rule 24", "IF markas IS hancur AND gold
IS banyak AND point IS sedang AND ek IS banyak THEN predikat IS
poor");
```

```
IS.NewRule("Rule 25", "IF markas IS hancur AND gold
IS banyak AND point IS banyak AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 26", "IF markas IS hancur AND gold
IS banyak AND point IS banyak AND ek IS sedang THEN predikat IS
poor");
IS.NewRule("Rule 27", "IF markas IS hancur AND gold
IS banyak AND point IS banyak AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 28", "IF markas IS sedang AND gold
IS sedikit AND point IS sedikit AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 29", "IF markas IS sedang AND gold
IS sedikit AND point IS sedikit AND ek IS sedang THEN predikat IS
good");
IS.NewRule("Rule 30", "IF markas IS sedang AND gold
IS sedikit AND point IS sedikit AND ek IS banyak THEN predikat IS fair");

IS.NewRule("Rule 31", "IF markas IS sedang AND gold
IS sedikit AND point IS sedang AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 32", "IF markas IS sedang AND gold
IS sedikit AND point IS sedang AND ek IS sedang THEN predikat IS fair");
IS.NewRule("Rule 33", "IF markas IS sedang AND gold
IS sedikit AND point IS sedang AND ek IS banyak THEN predikat IS fair");
IS.NewRule("Rule 34", "IF markas IS sedang AND gold
IS sedikit AND point IS banyak AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 35", "IF markas IS sedang AND gold
IS sedikit AND point IS banyak AND ek IS sedang THEN predikat IS fair");
IS.NewRule("Rule 36", "IF markas IS sedang AND gold
IS sedikit AND point IS banyak AND ek IS banyak THEN predikat IS fair");
IS.NewRule("Rule 37", "IF markas IS sedang AND gold
IS sedang AND point IS sedikit AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 38", "IF markas IS sedang AND gold
IS sedang AND point IS sedikit AND ek IS sedang THEN predikat IS
good");
```

```
IS.NewRule("Rule 39", "IF markas IS sedang AND gold
IS sedang AND point IS sedikit AND ek IS banyak THEN predikat IS fair");
IS.NewRule("Rule 40", "IF markas IS sedang AND gold
IS sedang AND point IS sedang AND ek IS sedikit THEN predikat IS
good");

IS.NewRule("Rule 41", "IF markas IS sedang AND gold
IS sedang AND point IS sedang AND ek IS sedang THEN predikat IS
good");
IS.NewRule("Rule 42", "IF markas IS sedang AND gold
IS sedang AND point IS sedang AND ek IS banyak THEN predikat IS
fair");
IS.NewRule("Rule 43", "IF markas IS sedang AND gold
IS sedang AND point IS banyak AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 44", "IF markas IS sedang AND gold
IS sedang AND point IS banyak AND ek IS sedang THEN predikat IS
fair");
IS.NewRule("Rule 45", "IF markas IS sedang AND gold
IS sedang AND point IS banyak AND ek IS banyak THEN predikat IS
fair");
IS.NewRule("Rule 46", "IF markas IS sedang AND gold
IS banyak AND point IS sedikit AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 47", "IF markas IS sedang AND gold
IS banyak AND point IS sedikit AND ek IS sedang THEN predikat IS
good");
IS.NewRule("Rule 48", "IF markas IS sedang AND gold
IS banyak AND point IS sedikit AND ek IS banyak THEN predikat IS
good");
IS.NewRule("Rule 49", "IF markas IS sedang AND gold
IS banyak AND point IS sedang AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 50", "IF markas IS sedang AND gold
IS banyak AND point IS sedang AND ek IS sedang THEN predikat IS
good");
```

```
IS.NewRule("Rule 51", "IF markas IS sedang AND gold
IS banyak AND point IS sedang AND ek IS banyak THEN predikat IS
fair");
IS.NewRule("Rule 52", "IF markas IS sedang AND gold
IS banyak AND point IS banyak AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 53", "IF markas IS sedang AND gold
IS banyak AND point IS banyak AND ek IS sedang THEN predikat IS
fair");
IS.NewRule("Rule 54", "IF markas IS sedang AND gold
IS banyak AND point IS banyak AND ek IS banyak THEN predikat IS
fair");
IS.NewRule("Rule 55", "IF markas IS utuh AND gold IS
sedikit AND point IS sedikit AND ek IS sedikit THEN predikat IS good");
IS.NewRule("Rule 56", "IF markas IS utuh AND gold IS
sedikit AND point IS sedikit AND ek IS sedang THEN predikat IS good");
IS.NewRule("Rule 57", "IF markas IS utuh AND gold IS
sedikit AND point IS sedikit AND ek IS banyak THEN predikat IS good");
IS.NewRule("Rule 58", "IF markas IS utuh AND gold IS
sedikit AND point IS sedang AND ek IS sedikit THEN predikat IS exc");
IS.NewRule("Rule 59", "IF markas IS utuh AND gold IS
sedikit AND point IS sedang AND ek IS sedang THEN predikat IS good");
IS.NewRule("Rule 60", "IF markas IS utuh AND gold IS
sedikit AND point IS sedang AND ek IS banyak THEN predikat IS good");

IS.NewRule("Rule 61", "IF markas IS utuh AND gold IS
sedikit AND point IS banyak AND ek IS sedikit THEN predikat IS good");
IS.NewRule("Rule 62", "IF markas IS utuh AND gold IS
sedikit AND point IS banyak AND ek IS sedang THEN predikat IS good");
IS.NewRule("Rule 63", "IF markas IS utuh AND gold IS
sedikit AND point IS banyak AND ek IS banyak THEN predikat IS good");
IS.NewRule("Rule 64", "IF markas IS utuh AND gold IS
sedang AND point IS sedikit AND ek IS sedikit THEN predikat IS exc");
IS.NewRule("Rule 65", "IF markas IS utuh AND gold IS
sedang AND point IS sedikit AND ek IS sedang THEN predikat IS good");
IS.NewRule("Rule 66", "IF markas IS utuh AND gold IS
sedang AND point IS sedikit AND ek IS banyak THEN predikat IS good");
```

```
                IS.NewRule("Rule 67", "IF markas IS utuh AND gold IS
sedang AND point IS sedang AND ek IS sedikit THEN predikat IS exc");
                IS.NewRule("Rule 68", "IF markas IS utuh AND gold IS
sedang AND point IS sedang AND ek IS sedang THEN predikat IS exc");
                IS.NewRule("Rule 69", "IF markas IS utuh AND gold IS
sedang AND point IS sedang AND ek IS banyak THEN predikat IS good");
                IS.NewRule("Rule 70", "IF markas IS utuh AND gold IS
sedang AND point IS banyak AND ek IS sedikit THEN predikat IS good");

                IS.NewRule("Rule 71", "IF markas IS utuh AND gold IS
sedang AND point IS banyak AND ek IS sedang THEN predikat IS good");
                IS.NewRule("Rule 72", "IF markas IS utuh AND gold IS
sedang AND point IS banyak AND ek IS banyak THEN predikat IS good");
                IS.NewRule("Rule 73", "IF markas IS utuh AND gold IS
banyak AND point IS sedikit AND ek IS sedikit THEN predikat IS exc");
                IS.NewRule("Rule 74", "IF markas IS utuh AND gold IS
banyak AND point IS sedikit AND ek IS sedang THEN predikat IS exc");
                IS.NewRule("Rule 75", "IF markas IS utuh AND gold IS
banyak AND point IS sedikit AND ek IS banyak THEN predikat IS good");
                IS.NewRule("Rule 76", "IF markas IS utuh AND gold IS
banyak AND point IS sedang AND ek IS sedikit THEN predikat IS exc");
                IS.NewRule("Rule 77", "IF markas IS utuh AND gold IS
banyak AND point IS sedang AND ek IS sedang THEN predikat IS exc");
                IS.NewRule("Rule 78", "IF markas IS utuh AND gold IS
banyak AND point IS sedang AND ek IS banyak THEN predikat IS good");
                IS.NewRule("Rule 79", "IF markas IS utuh AND gold IS
banyak AND point IS banyak AND ek IS sedikit THEN predikat IS good");
                IS.NewRule("Rule 80", "IF markas IS utuh AND gold IS
banyak AND point IS banyak AND ek IS sedang THEN predikat IS good");
                IS.NewRule("Rule 81", "IF markas IS utuh AND gold IS
banyak AND point IS banyak AND ek IS banyak THEN predikat IS good");

        }

        void Start ()
                {
                fuzzyfuzzya();
```

```
kill = PlayerPrefs.GetFloat("enemymati");
poin = PlayerPrefs.GetFloat("poinplayer");
bangunan = PlayerPrefs.GetInt("markas");
time = PlayerPrefs.GetString("waktu");
gold = PlayerPrefs.GetFloat("jmlgold");
sts = PlayerPrefs.GetString("status");

goldlost = PlayerPrefs.GetFloat ("goldlost");
pointlost = PlayerPrefs.GetFloat("pointlost");
dead = PlayerPrefs.GetFloat("dead");
build = PlayerPrefs.GetFloat("build");

elapsedtime.text = "" + time;
playerbuild.text ="" + bangunan;
status.text = status.text + " " + sts;
deadtx.text = "" + dead;
buildtx.text = "" + build;
goldlosttx.text ="" + goldlost;
pointlosttx.text = "" + pointlost;
Debug.Log (kill+","+poin+","+time+""+gold);

IS.SetInput("gold",gold);
IS.SetInput("point",poin);
IS.SetInput("ek",kill);
IS.SetInput("markas",bangunan);

try
{
        NewAngle = IS.Evaluate("predikat");
        z = Math.Round(NewAngle);
        hasilstar.text = "" + z.ToString ();
}
catch ( Exception ){
}

if (z <= 3) {
        predikat.text = "POOR";
```

```
        }
        else if (z > 3 || z <= 5) {
                predikat.text = "FAIR";
        }
        else if (z > 5 || z <= 8) {
                predikat.text = "GOOD";
        }
        else {
                predikat.text = "EXCELENT";
        }

}

        // Update is called once per frame
        void Update ()
        {}
```

--Halaman ini sengaja dikosongkan--

--Halaman ini sengaja dikosongkan--

SCORING BASED INTELLIGENT AGENT USING FUZZY LOGIC IN REAL TIME STRATEGY(RTS) GAME

Name **: Kartika Dwi Handini**
Supervisor **: 1. Mochamad Hariadi, ST.,M.Sc.,Ph.D.**
2. Dr. Supeno Mardi S N, ST., MT.

ABSTRACT

One part of the game that can attract players is assessment (scoring) is applied. Exploiting the property of each unit in RTS games can be used for modeling scoring appropriate to the circumstances at the time of the game diberakhir. By using fuzzy logic, some properties can be combined to get the value (score) that can be used for the purpose in the game. Changes in the condition of the property each unit contained in RTS games are used as inputs are then processed with fuzzy logic. Standard deviation (variation data) for scoring using fuzzy logic at 0.7378. whereas the static scoring value by 0.7017. Fuzzy logic is able to generate value (score) is more varied so that the results are more dynamic. 3. Large percentage of the same star between the static and fuzzy calculation by 60%. The rest of 40% is the result of the acquisition of different stars.

Keyword*: Fuzzy, Real Time Strategy, Scoring, game*

--Halaman ini sengaja dikosongkan--

KATA PENGANTAR

Puji dan Syukur kehadirat Allah SWT atas segala limpahan berkah, rahmat serta hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul : ***Scoring* Berbasis Agen Cerdas Menggunakan Logika *Fuzzy* pada Permainan (G*ame*) *Real Time Strategy* (RTS).**

Penelitian ini disusun dalam rangka pemenuhan bidang riset di Jurusan Teknik Elektro ITS, Bidang Studi Teknik Komputer dan Telematika, serta digunakan sebagai persyaratan menyelesaikan pendidikan S-1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, danBapak yang telah memberikan dorongan spiritual dan material serta seluruh kerabat dan kolega penulis yang banyak membantu proses dalam menyelesaikan buku penelitian ini.
2. Bapak Dr. Tri Arief Sardjono, S.T., M.T. selaku Ketua Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember
3. Secara khusus penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Mochamad Hariadi, ST.,M.Sc.,Ph.D. dan Bapak Dr. Supeno Mardi S N, ST., MT. atas bimbingan selama mengerjakan penelitian.
4. Bapak-ibu dosen pengajar Bidang Studi Teknik Komputer dan Telematika, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Seluruh teman-teman Lintas Jalur Elektro Genap 2012 serta teman-teman Bidang Studi Teknik Komputer dan Telematika.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun serta menghatur maaf atas segala kekurangan yang ada dalam penulisan buku ini. Semoga penelitian ini dapat memberikan manfaat bagi kita semua.Amin.

Surabaya, Juni 2015

Penulis

--Halaman ini sengaja dikosongkan--

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR ..... **..i**
LEMBAR PENGESAHAN ..... **iii**
ABSTRAK ..... **…v**
ABSTRACT ..... **.vii**
KATA PENGANTAR ..... **...ix**
DAFTAR ISI ..... **...xi**
DAFTAR GAMBAR ..... **..xv**
DAFTAR TABEL ..... **.xvii**
BAB I PENDAHULUAN **1**
1.1. Latar Belakang 1
1.2. Permasalahan 2
1.3. Tujuan 2
1.4. Batasan Masalah 2
1.5. Metodologi 2
1.6. Sistematika Penulisan 4
BAB 2 TEORI PENUNJANG **5**
2.1. *Fantasy Chronicles* 5
2.2. *Game Real Time Strategy* 6
2.3. *Scoring* pada Permainan 7
2.4. Logika *Fuzzy* 8
BAB 3 PERANCANGAN SISTEM **21**
3.1. Desain Sistem 22
3.2. Perancangan Variable Masukan 22
3.2.1. Parameter Kayu 23

3.2.2. Parameter Poin 23
3.2.3. Paramener Pasukan 23
3.2.4. Parameter Markas 23
3.2.5. Parameter Predikat 24
3.3. Perancangan Fungsi Keanggotaan Kayu 24
3.3.1. Fungsi Keanggotaan Parameter Kayu Kondisi “sedikit” 24
3.3.2. Fungsi Keanggotaan Parameter Kayu Kondisi “sedang” 25
3.3.3. Fungsi Keanggotaan Parameter Kayu Kondisi “banyak” 25
3.4. Perancangan Fungsi Keanggotaan Poin 26
3.4.1. Fungsi Keanggotaan Parameter Poin Kondisi “sedikit” 26
3.4.2. Fungsi Keanggotaan Parameter Poin Kondisi “sedang” 27
3.4.3. Fungsi Keanggotaan Parameter Poin Kondisi “banyak” 28
3.5. Perancangan Fungsi Keanggotaan Pasukan 28
3.5.1. Fungsi Keanggotaan Parameter Pasukan Kondisi “sedikit” . 28
3.5.2. Fungsi Keanggotaan Parameter Pasukan Kondisi “sedang” 29
3.5.3. Fungsi Keanggotaan Parameter Pasukan Kondisi “banyak”30
3.6. Perancangan Fungsi Keanggotaan Markas 30
3.6.1. Fungsi Keanggotaan Parameter Markas Kondisi “hancur” .. 31
3.6.2. Fungsi Keanggotaan Parameter Markas Kondisi “sedang” . 31
3.6.3. Fungsi Keanggotaan Parameter Markas Kondisi “utuh” 32
3.7. Perancangan Fungsi Keanggotaan Predikat 32
3.7.1. Fungsi Keanggotaan Parameter Predikat Kondisi “poor” 33
3.7.2. Fungsi Keanggotaan Parameter Predikat Kondisi “fair” 33
3.7.3. Fungsi Keanggotaan Parameter Predikat Kondisi “good” ... 34
3.7.4. Fungsi Keanggotaan Parameter Predikat Kondisi “excelent”34

3.8. Desain Aturan *Fuzzy* (*Rules Base*) 35
BAB 4 PENGUJIAN DAN ANALISA 39
4.1. Skenario Pengujian 39
4.2. Pengujian Implemantasi Parameter 39
4.3. Pengujian Sistem *Scoring* 40
4.3.1. Pengujian Hasil Akhir Kondisi "poor" 41
4.3.2. Pengujian Hasil Akhir Kondisi "fair" 42
4.3.3. Pengujian Hasil Akhir Kondisi "good" 43
4.3.4. Pengujian Hasil Akhir Kondisi "excelent" 44
4.4. Pengujian Metode 45
4.4.1. Pengujian Hasil Akhir Kondisi Menang 45
4.4.2. Pengujian Hasil Akhir Kondisi Kalah 47
BAB 5 PENUTUP 51
5.1. Kesimpulan 51
5.2. Saran 51
DAFTAR PUSTAKA 53
LAMPIRAN 55
RIWAYAT HIDUP 67

--Halaman ini sengaja dikosongkan--

DAFTAR TABEL

Tabel 3.4: Rule based fuzzy Predikat 35
Tabel 4.1: Hasil Pengujian kondisi menang 46
Tabel 4.2: Hasil Pengujian kondisi kalah 48

--Halaman ini sengaja dikosongkan--

DAFTAR GAMBAR

Gambar 1.1 Metodologi penelitian 3
Gambar 2.1 Tampilan Awal Permainan *Fantasy Chronicles* 3D 6
Gambar 2.2 Game RTS Command & Conquer (techxav.com) 7
Gambar 2.3 Contoh pemetaan *input output* (Gelley,2000) 8
Gambar 2.4 Himpunan *fuzzy* pada variable temperatur[3] 10
Gambar 2.5 Representasi linier naik 11
Gambar 2.6 Representasi linier turun 12
Gambar 2.7 Representasi Kurva Segitiga 12
Gambar 2.8 Representasi Kurva Trapesium 13
Gambar 2.9 Komposisi Aturan *Fuzzy* : Metode *Max*[3] 16
Gambar 2.10 Proses Defuzzyfikasi 17
Gambar 3.1 Desain *game Fantasy Chronicles 3D* 21
Gambar 3.2 Tahapan desain sistem 22
Gambar 3.3 FK parameter kayu kondisi “sedikit” 24
Gambar 3.4 FK parameter kayu kondisi “sedang” 25
Gambar 3.5 FK parameter kayu kondisi “banyak” 26
Gambar 3.6 FK parameter poin kondisi “sedikit” 27
Gambar 3.7 FK parameter poin kondisi “sedang” 27
Gambar 3.8 FK parameter poin kondisi “banyak” 28
Gambar 3.9 FK parameter pasukan kondisi “sedikit” 29
Gambar 3.10 FK parameter pasukan kondisi “sedang” 29
Gambar 3.11 FK parameter pasukan kondisi “banyak” 30
Gambar 3.12 FK parameter markas kondisi “hancur” 31
Gambar 3.13 FK parameter markas kondisi “sedang” 31
Gambar 3.14 FK parameter markas kondisi “utuh” 32
Gambar 3.15 FK parameter predikat kondisi “poor” 33

Gambar 3.16 FK parameter predikat kondisi “fair” 33
Gambar 3.17 FK parameter predikat kondisi “good”..... 34
Gambar 3.18 FK parameter predikat kondisi “excelent” 35
Gambar 4.1 Implementasi variabel masukan 39
Gambar 4.2 Hasil Akhir Kondisi “poor”..... 41
Gambar 4.3 Hasil Akhir Kondisi “fair” 42
Gambar 4.4 Hasil Akhir Kondisi “Good” 43
Gambar 4.5 Hasil Akhir Kondisi “excelent” 44
Gambar 4.5 pengujian metode hasil permainan menang, predikat “excelent” dan “poor” 45
Gambar 4.6 pengujian metode hasil permainan kalah predikat “poor”. 47

RIWAYAT HIDUP

Kartika Dwi Handini. lahir di Tegal pada 11 April 1991. Anak kedua dari dua bersaudara ini menyelesaikan pendidikan dasar di SDN Randugunting 3 Tegal. Kemudian menempuh jalur pendidikan di SMP Negeri 10 Tegal. Setelah itu melanjutkan pendidikan di SMK Telkom Sandhy Putra Purwokerto dengan mengambil jurusan Rekayasa Perangkat Lunak (RPL) pada tahun 2006 hingga tahun 2009. Melanjutkan pendidikan ke jenjang perkuliahan di Jurusan D3 Teknik Elektro Politeknik Negeri Semarang (POLINES) prodi Teknik Informatika dari tahun 2009 sampai 2012. Dan melanjutkan kuliah S1 di Lintas Jalur ITS jurusan Teknik Elektro prodi Teknik Komputer dan Telematika. (kartikadwihandini@gmail.com).

BAB 1
PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi yang semakin pesat khususnya dalam bidang komputer membuat komputer saat ini bukan hanya sebagai perangkat yang digunakan untuk mengolah data saja, tetapi juga dapat digunakan sebagai sarana hiburan. Pemanfaatan komputer sebagai sarana hiburan dapat di implementasikan dalam berbagai macam bentuk seperti permainan komputer (*Game*). *Game* dengan alur cerita yang menarik serta gambar yang berkualitas baik akan lebih diminati oleh pengguna sehingga membuat pemain tidak merasa cepat bosan dan ingin melanjutkan *game* hingga selesai. Sebuah *game* memiliki 5 komponen penting yaitu fitur, *gameplay*, *interface*, aturan, dan desain [6]. Salah satu bagian yang termasuk dalam aturan sebuah *game* adalah aturan *scoring* yang akan digunakan. *Scoring* bertujuan untuk memberikan nilai akhir pada sebuah permainan dan merupakan komponen penting yang dapat menjaga minat pemain untuk tetap bermain [1].

Dalam *game* RTS memiliki dua aspek penilaian yaitu aspek ekonomi yang berupa sumber daya yang dimiliki dan aspek militer yang dapat dilihat dari pasukan dan *basecamp* yang ada [5]. Pada beberapa *game* yang ada saat ini, sumber daya biasanya digunakan untuk membeli / meningkatkan keterampilan pemain / senjata . Sedangkan pasukan dan *basecamp* tidak begitu diperhitungkan dalam *scoring* hasil akhir permainan. Seperti dalam permainan *Clash Of Clans,* sumber daya digunakan untuk membeli perlengkapan *basecamp* dan pasukan tetapi untuk menentukan hasil akhir permainan hanya melihat dari seberapa besar bagian dari bangunan yang berhasil dihancurkan tanpa melihat kondisi pasukan dan sumber daya yang didapat dan digunakan. Sehingga diperlukan sebuah sistem *scoring* yang dapat mencakup semua aspek penilaian dalam *game* RTS.

Dalam Tugas Akhir ini akan dibuat *scoring* yang dapat menentukan hasil akhir permainan dengan melibatkan semua aspek penilaian yang ada pada permainan. Karena parameter *scoring* yang digunakan beragam maka *scoring* akan diolah menggunakan logika *fuzzy*. Logika *fuzzy* adalah cabang dari sistem kecerdasan buatan (*Artificial Intelegent*) yang mengemulasi kemampuan manusia dalam berfikir ke dalam bentuk algoritma[3]. Logika *fuzzy* juga dapat digunakan dalam berbagai aplikasi

pemrosesan data yang tidak dapat direpresentasikan dalam bentuk biner. Selain itu logika *fuzzy* memiliki beberapa kelebihan diantaranya mampu beradaptasi dengan perubahan, mudah diimplementasikan, dan didasarkan pada bahasa alami. Dalam permainan komputer (*game*) logika *fuzzy* adalah teknik kecerdasan lainnya yang dapat digunakan untuk meningkatkan kinerja *game*. *Fuzzy* mampu menangani permasalahan yang kompleks dengan. komputasi yang rendah[4].

1.2. Permasalahan

Masalah yang diharapkan untuk ditemukan solusinya melalui tugas akhir ini adalah penghitungan *scoring* hasil akhir pada suatu permainan peperangan memiliki kecenderungan tidak memperhitungkan nilai – nilai dari seluruh aspek yang terdapat pada permainan.

1.3. Tujuan

Tujuan dan capaian yang diharapkan tercapai setelah selesainya tugas akhir ini adalah sebagai berikut:

1. Menghasilkan pemodelan *scoring* menggunakan logika *fuzzy* yang sesuai dengan kondisi keadaan saat permainan berakhir.
2. Menentukan hasil akhir permainan dengan logika *fuzzy* berdasarkan nilai yang diperoleh dari parameter yang terdapat pada permainan.

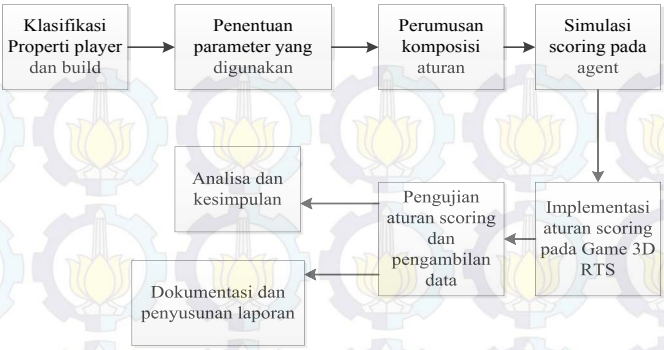
1.4. Batasan Masalah

Batasan masalah dari Tugas Akhir ini adalah:

1. Metode *fuzzy* yang digunakan adalah *fuzzy* Mamdani dengan fungsi keanggotaan segitiga dan trapesium.
2. Permainan yang dibuat adalah permainan *Real Time Strategy* yang bersifat *Single player.*
3. Permainan yang dibuat berbasis *Desktop Game.*

1.5. Metodologi

Metodologi yang digunakan pada pengerjaan tugas akhir ini dijabarkan dengan blok diagram pada Gambar 1.1.



Gambar 1.1 Metodologi penelitian

Penjelasan dari blok diagram Gambar 1.1 sebagai berikut:

1. Klasifikasi Properti *Player* dan *building*
   Tahap awal adalah pengklasifikasian properti *game* yang terdapat pada *player* dan *building* (bangunan).
2. Penentuan parameter yang digunakan
   Pada bagian ini dilakukan pengumpulan properti - properti yang efisien yang berhubungan dengan *player* dan *building* yang dapat dimanfaatkan dalam membuat pemodelan *scoring*. Parameter masukan ini digunakan sebagai variabel acuan yang akan diolah nantinya
3. Perumusan komposisi aturan
   Sebelum melakukan simulasi yang akan digunakan perlu ditentukan komposisi aturan *fuzzy* yang akan digunakan dalam *scoring*. Tahapan ini dilakukan untuk menentukan aturan yang diterapkan dalam *scoring* agar pemodelan *scoring* yang dibuat sesuai yang diharapkan.
4. Simulasi *scoring* pada agen.
   Simulasi pada *scoring* dilakukan pada *player* dan *building*. Pada *player* akan ditentukan beberapa kondisi sesuai komposisi aturan yang sudah dibuat. Jika terdapat komposisi aturan yang kurang sesuai dapat terlihat hasilnya pada tahapan ini.
5. Implementasi aturan *scoring* pada *Game* 3D RTS
   Setelah melakukan simulasi, selanjutnya hasilnya akan diterapkan pada *game* 3D RTS. *Tools* yang digunakan adalah Unity3D. Implementasi diterapkan pada *game* pertempuran RTS.

6. Pengujian aturan *scoring* dan pengambilan data
   Setelah implementasi dilakukan , maka proses pengukuran dapat dilakukan. Pada tahapan ini, dilakukan pengumpulan data untuk berbagai kondisi parameter yang telah ditentukan untuk pengujian. Tujuan akhirnya adalah menentukan komposisi aturan mana yang tepat untuk diterapkan.
7. Analisa dan Kesimpulan
   Analisa data dan pengambilan kesimpulan mengacu pada data hasil pengukuran.
8. Dokumentasi dan Penyusunan Laporan
   Tahapan ini meliputi pembuatan laporan tugas akhir dan penulisan jurnal ilmiah. Pembuatan laporan tugas akhir untuk beberapa bagian dilakukan bersesuaian dengan pengerjaan tahapan-tahapan diatas. Sedangkan jurnal ilmiah dilakukan setelah laporan tugas akhir selesai.

1.6. Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini terbagi dalam lima bab, masing-masing bab diuraikan sebagai berikut:

1. Bab 1 merupakan bab pendahuluan yang berisi latar belakang, permasalahan, tujuan tugas akhir, batasan masalah, metodologi, sistematika penulisan dan relevansi
2. Bab 2 merupakan bab yang membahas tentang teori penunjang dan literatur yang digunakan dalam penyusunan tugas akhir
3. Bab 3 membahas tentang perancangan sistem dan implementasi sistem
4. Bab 4 berisi tentang pengujian dari sistem yang telah dibangun
5. Bab 5 merupakan bab penutup laporan yang berisi tentang kesimpulan yang diperoleh dari pembuatan tugas akhir, serta saran-saran untuk pengembangan sistem lebih lanjut.

BAB 2
TEORI PENUNJANG

Untuk mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi dengan demikian penelitian ini lebih terarah

Pada penelitian [2] menunjukkan bahwa parameter yang efisien dapat digunakan sebagai acuan dalam aturan pembuatan *score* sederhana. Dengan menggunakan parameter yang efisien maka dihasilkan probabilitas aturan yang relevan dengan hasil yang diinginkan. Kemudian penelitian [1] membahas mengenai sistem *scoring* dalam permainan menggunakan pendekatan statistik dan pendekatan logika *fuzzy*. Hasil yang didapat menjelaskan bahwa pendekatan logika *fuzzy* jauh lebih fleksibel dibandingkan dengan statistik. Dengan mengacu pada hasil pertandingan dan perbedaan nilai antara pemain, hasil akhir peringkat pemain dapat berubah secara fleksibel sesuai parameter acuannya.

2.1 *Fantasy Chronicles*

Fantasy Chronicles 3D merupakan permainan ber*genre Real Time Strategy* yang dikembangkan oleh tim beranggotakan 5 orang dari lintas jalur teknik komputer dan telematika. Permainan ini memiliki latar belakang terjadi pertempuran antara 2 ras utama yaitu *White Warrior* (NPC *player*) dan *Black Warrior* (NPC musuh). Kedua ras tersebut memiliki 4 kelompok yang sama yang terdiri dari penyerangan, pertahanan, bangunan, dan pekerja. Dalam permainan ini kedua ras saling berhadapan untuk memperebutkan kekuasaan dan saling menghancurkan. Di kelompok penyerangan ras *White Warrior* memiliki 6 *macam* (NPC *player*) yang berperang menggunakan pedang dan sihir api. Sedangkan di ras *Black Warrior* memiliki 6 *macam* (NPC musuh) yang berperang menggunakan senjata (pedang, kapak,tongkat,gada) dan sihir api hitam. Permainan *Fantasy Chronicles* 3D ini meliputi 5 bagian penting yaitu perilaku NPC, *flocking,* strategi penyerangan, *scoring* dan *Gameplay*. Kelima unsur tersebut nantinya akan dijadikan satu dan diujicobakan ke *game* yang sebenarnya. Gambar 2.1 merupakan tampilan awal dari permainan *Fantasy Chronicles* 3D.

Gambar 2.1 Tampilan Awal Permainan *Fantasy Chronicles* 3D

2.2 *Game Real Time Strategy*

Real time strategy (RTS) Merupakan salah satu jenis permainan komputer yang mencakup *game* peperangan yang terjadi secara "*real time*". "*Real time"* berarti bahwa adanya interaksi langsung pada permainan antara komputer dan pengguna. Pada permainan RTS terdapat 2 aspek yaitu aspek ekonomi dan aspek militer sehingga permainan jenis RTS termasuk jenis permainan yang sulit untuk dikuasai. Dalam permainan RTS biasanya pemain diminta untuk mengumpulkan sumber daya, membangun tentara, dan berperang untuk mengalahkan musuh. Saat ini sudah banyak permainan RTS yang telah dirilis, antara lain yang terkenal adalah "StarCraft" , "Warcraft"," Command & Conquer " , " The Age of Empires, dan masih banyak lagi. Contoh lain dari permainan RTS adalah SimCity yang dirilis oleh EA Games dimana permainan ini tidak memiliki bagian perperangan tetapi fokus pada perencanaan kota dan ekonomi. *Platform* yang paling populer untuk memainkan permainan ini adalah komputer pribadi, tetapi beberapa permainan tersebut dirilis dalam bentuk lain seperti *Sony Playstation.* Sebuah survei yang dibuat oleh *The Entertainment Software Association* di Amerika Serikat menyatakan bahwa hampir

30.8% dari penjualan permainan untuk komputer pribadi adalah permainan RTS[5].

Gambar 2.2 Game RTS Command & Conquer (techxav.com)

Pada awal permainan yang sedang dimainkan , pemain biasanya memiliki sedikit sumber daya dan unit , tetapi memiliki kemungkinan memperoleh lebih banyak sumber daya . Tugasnya membangun atau memperluas markas, dengan tujuan mengumpulkan sumber daya dan menghabiskan sumber daya untuk memproduksi tentara . Pemain juga harus memastikan apakah akan mampu bertahan pada tahap awal dengan unit-unit produksi atau struktur defensif. Beberapa *game* memiliki cerita dengan tujuan tertentu yang harus dipenuhi , tetapi secara umum adalah tentang kontrol sumber daya, menempatkan unit untuk penggunaan yang baik dan mencapai kekuatan militer yang unggul . Permainan ini biasanya berakhir ketika seorang pemain telah menghancurkan semua musuhnya , atau ketika seorang pemain mencapai beberapa tujuan yang sudah ditentukan. Dalam permainan tujuan ini biasanya ditentukan dalam misi yang diletakan pada *storyline* dan biasanya dapat diakses saat bermain *game*.

2.3 *Scoring* pada Permainan

Dalam sebuah permainan, nilai biasanya mengacu pada kuantitas dari seorang pemain atau sebuah tim. Biasanya diukur berdasarkan poin yang terkumpul dan setiap kejadian yang ada dalam permainan dapat menambah atau mengurai poin dalam nilai. Pada kebanyakan *game* indikator yang digunakan adalah poin dari tingkat keberhasilan dalam permainan atau dalam permainan kompetisi *goal* untuk mendapatkan nilai yang lebih dari lawan juga dapat digunakan sebagai indikator.

Pada masa permaianan *video game* pemain akan terus bermain untuk mendapatkan nilai yang lebih tinggi dari sebelumnya. Sedangkan permaianan modern (era permaianan *arcade*) nilai tidak menjadi hal yang sangat penting karena pemain hanya fokus pada menyelesaikan permainan saja dengan waktu yang terbatas. Tetapi beberapa permainan memiliki beberapa poin yang digunakan sebagai kuantitas nilai seperti poin keterampilan, banyaknya poin yang dimiliki pemain dan sumber daya yang ada dapat digunakan sebagai bagian dari permaianan. Terkadang dalam permaianan poin dapat digunakan untuk membuka satu karakter atau fasilitas tambahan yang dapat digunakan karena sebelumnya terkunci. Poin juga dapat berfungsi untuk meningkatkan *level* pemain ketika pemain berhasil mengalahkan musuh utama kemudian dapat melanjutkan permainan ke tingkat selanjutnya.

2.4 Logika *Fuzzy*

Logika *Fuzzy* adalah salah satu sistem kecerdasan buatan (*Artificial Intelegent*) yang menerapkan cara berfikir manusia dimana dapat digunakan dalam berbagai aplikasi pemrosesan data. Pertama kali diperkenalkan oleh Prof. Lotfi Zadeh seorang guru besar di University of California di Berkeley pada tahun 1965. Dalam paparannya dijelaskan mengenai ide dasar *fuzzy set* yang meliputi *inclusion, union, intersection, complement, relation dan convexity*. Logika ini dapat diumpamakan sebagai kotak hitam yang berisi metode atau cara yang digunakan untuk mengolah data yang dapat menghubungkan antara masukan dan keluaran. Sehinggadapat digunakan menjadi informasi yang dapat dimanfaatkan[3].

Gambar 2.3 Contoh pemetaan *input output* (Gelley,2000)

Menurut [3] ada beberapa kelebihan dalam menggunakan logika *fuzzy* antara lain :

1. Konsep logika *fuzzy* mudah dimengerti. Konsep matematis yang mendasari penalaran *fuzzy* sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* sangat fleksibel.
3. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.
4. Logika *fuzzy* mampu memodelkan fungsi-fungsi *nonlinear* yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan.
6. Logika *fuzzy* didasarkan pada bahasa alami. Logika *fuzzy* menggunakan bahasa sehari – hari sehingga mudah dimengerti.

Logika Fuzzy memiliki beberapa bagian dalam pembuatannya yaitu [3]:

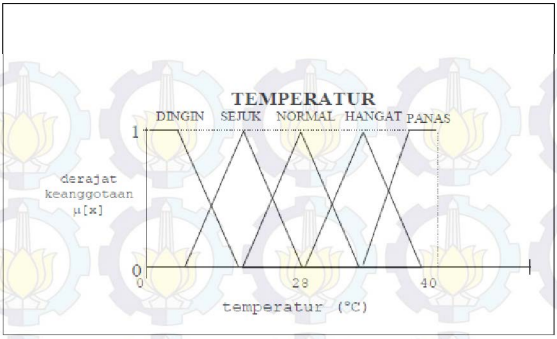
1. Himpunan *fuzzy*

   Terdapat dua atribut yang dimiliki himpunan *fuzzy* yaitu atribut linguistik dan atribut numeris. Linguistik adalah penamaan yang diberikan pada suatu domain yang merepresentasikan keadaan atau kondisi tertentu sedangkan numeris adalah suatu nilai (angka) yang menunjukkan jangkauan dari suatu variabel.salah satu contoh himpunan *fuzzy* terlihat pada Gambar 2.4.

   Nilai keanggotaan suatu anggota x dalam suatu himpunan *fuzzy* A(μA[x]) memiliki 2 kemungkinan yaitu:

   a. Satu (1), yang berarti suatu anggota menjadi bagian dalam suatu himpunan, atau
   b. Nol (0), yang berarti suatu anggota tidak menjadi bagian dalam suatu himpunan.

   Bagian lainnya dari himpunan *fuzzy* adalah semesta pembicaraan dan domain. Semesta pembicaraan adalah jangkauan nilai yang diperbolehkan untuk diolah dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan



Gambar 2.4 Himpunan *fuzzy* pada variable temperatur[3]

positif maupun negatif. Contoh dari semesta pembicaraan adalah:

a. Semesta pembicaraan untuk variabel nilai: [0 100]
b. Semesta pembicaraan untuk variabel temperatur : [0 40]

Sedangkan Domain adalah jangkauan nilai yang terdapat pada masing – masing himpunan *fuzzy* yang yang diijinkan untuk dioperasikan. Domain merupakan himpunan bilangan real yang naik (bertambah) secara monoton dari kiri ke kanan. Contoh domain himpunan fuzzy:

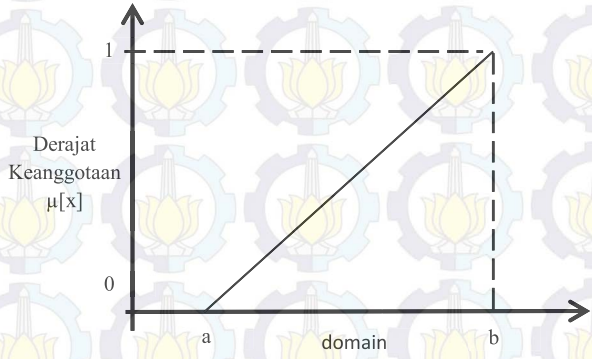
a. Sangat kurang = [0 30]
b. Kurang = [20 50]
c. Cukup = [45 75]
d. Baik = [70 85]
e. Sangat baik= [80 100]

2. Fungsi Keanggotaan

Merupakan Kurva yang digunakan untuk memetakan data masukan kedalam nilai keanggotaannya atau dapat disebut juga derajat keanggotaan. Salah satu cara yang digunakan untuk mendapatkan nilai derajat keanggotaan adalah dengan pendekatan fungsi. Terdapat beberapa pendekatan fungsi yang dapat digunakan yaitu[3] :

a. Representasi Linear

Pada representasi linear terdapat dua keadaan, pertama nilai yang dimulai dari nilai domain yang memiliki derajat keanggotaan nol(0) yang bergerak ke kanan (naik) menuju nilai derajat keanggotaan yang lebih tinggi seperti pada Gambar 2.5.



Gambar 2.5 Representasi linier naik[3]

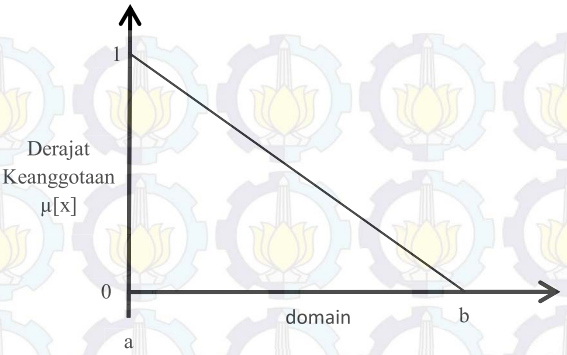
Fungsi keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x-a) / (b-a); & a < x < b \\ 1; & x \geq b \end{cases} \quad (2.1)$$

Kedua, Nilai dimulai dari nilai domain yang memiliki derajat keanggotaan satu(1) yang bergerak ke kanan (turun) menuju nilai derajat keanggotaan yang lebih rendah seperti pada Gambar 2.6.

Fungsi keanggotaan:

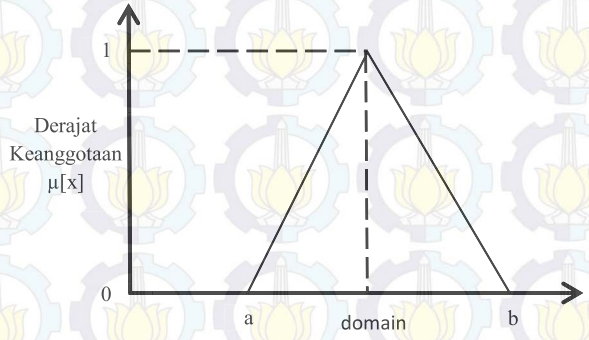
$$\mu[x] = \begin{cases} 0; & x \geq b \\ (b-x) / (b-a) & a < x < b \\ 1; & x \leq a \end{cases} \quad (2.2)$$



Gambar 2.6 Representasi linier turun[3]

b. Representasi Kurva Segitiga

Pada kurva segtiga terdapat 3 nilai yang menjadi acuannya yaitu 2 nilai yang bernilai derajat keanggotaan nol (0) dan satu nilai yang memiliki derajat keanggotaan satu (1) Seperti terlihat pada gambar 2.7.



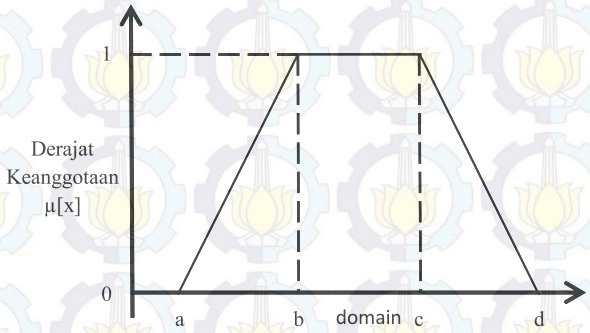
Gambar 2.7 Representasi Kurva Segitiga [3]

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & x \geq c \text{ atau } x \leq a \\ (x-a) / (b-a) & a < x < b \\ (c-x) / (c-b) & b < x < c \end{cases} \quad (2.3)$$

c. Representasi Kurva Trapesium

Pada kurva trapesium ada 4 nilai yang menjadi patokan. Dua nilai memiliki derajat keanggotaan nol (0) dan dua nilai lainnya memiliki derajat keanggotaan satu (1) yang terletak diantara nilai dengan derajat keanggotaan nol (0)



Gambar 2.8 Representasi Kurva Trapesium[3]

3. Fuzzyfikasi

Fuzzyfikasi adalah proses untuk mengubah variabel numerik menjadi variabel *fuzzy* (linguistik) dimana data masukan yang masih dalam bentuk numerik diubah terlebih dahulu menjadi variabel *fuzzy* melalui fungsi keanggotaan yang telah dibentuk sebelumnya sehingga nantinya data tersebut dapat digunakan menjadi suatu informasi yang akan diproses dalam pengolahan *fuzzy*.

Dapat dikatakan juga fuzzyfikasi merupakan pemetaan titik-titik numerik $x = (x^1,.....x^n)^T \in$ ke himpunan *fuzzy* A di *U*. *U* adalah semesta pembicaraan. Paling tidak ada dua kemungkinan pemetaan,yaitu[3]:

1. Fuzzyfikasi *singleton*: A adalah *fuzzy singleton* dengan *support* x , artinya $\mu_A (x') = 1$ untuk x' = x dan $\mu_A(x') = 0$ untuk $x' \in U$ yang lain dengan x' = x .
2. Fuzzyfikasi *nonsingleton* : $\mu_A (x) = 1$ dan $\mu_A (x')$ menurun dari 1 sebagaimana x' bergerak menjauh dari x .

4. *Fuzzy Inference System* (FIS) Mamdani

Ada 2 jenis FIS yang paling kita kenal yaitu Mamdani dan Sugeno. Masukan yang digunakan dalam FIS adalah bilangan tertentu yang menghasilkan keluaran berupa bilangan tertentu. Penerapan bahasa linguistik dapat digunakan sebagai masukan yang sebelumnya dikonversi terlebih dahulu. Kemudian dilakukan penalaran berdasarkan aturan – aturan yang sudah disusun dan mengubahnya menjadi keluaran yang bersifat teliti.

Prof. Ebrahim Mamdani merupakan pelopor aplikasi *fuzzy set* dalam bidang kontrol. Bersama kawan – kawannya dari Queen Mary College London, Mamdani membangun aplikasi pertama dan utama dari *fuzzy set*. Metode Mamdani ini juga dikenal sebagai metode *Max-Min* yang diperkenalkan pada tahun 1975. Terdapat empat tahapan untuk mendapat keluaran dalam FIS Mamdani yaitu[3]:

a. Pembentukan Himpunan *Fuzzy*

Pembentukan himpunan *fuzzy* bertujuan untuk memberikan jangkauan nilai yang diperbolehkan dalam suatu kurva yang setiap nilainya memiliki derajat keanggotaan antara 0 sampai dengan 1. Pada logika *boolean* digambarkan nilai "benar" dan "salah" sedangkan pada *fuzzy* digambarkan dengan ungkapan misalnya : "sangat lambat","agak sedang", "sangat cepat"dan lain-lain untuk mengambarkan setiap domainnya.

b. Aplikasi fungsi implikasi (aturan)

Dari himpunan *fuzzy* yang telah dibentuk, proses selanjutnya adalah membuat aturan – aturan *fuzzy* yang kemudian akan diolah dalam fungsi implikasi. Dari aturan – aturan *fuzzy* tersebut, setiap aturan akan menghasilkan nilai yang diperoleh dari fungsi implikasi yang ditetapkan Untuk FIS Mamdani fungsi implikasi yang digunakan adalah *Min*, dimana nilai derajat keanggotaan yang paling kecil dari masing – masing aturan yang akan diambil nilainya.

c. Komposisi aturan

Pada tahapan ini, setelah dilakukan fungsi implikasi kemudian dari setiap nilai yang diperoleh akan di gabungkan menjadi satu kurva komposisi aturan. Berbeda dengan penalaran boolean yang memiliki nilai benar dan

salah, pada FIS Mamdani jika terdapat beberapa aturan dalam sistem maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Menurut [3] terdapat 3 metode yang digunakan dalam inferensi sistem *fuzzy*, yaitu:

1) Metode *Max* (*Maximum*)
   Metode *Max* akan menghasilkan keluaran yang berupa daerah (himpunan) yang diperoleh dari fungsi implikasi nilai derajat keanggotaan masing – masing aturan yang selanjutnya seluruh aturan digabungkan dengan menggunakan operator OR sehingga akan menjadi kurva yang memiliki daerah berdasarkan gabungan dari hasil komposisi aturan. Secara umum dapat dituliskan:

   $$\mu sf[xi] \leftarrow max(\mu sf[xi], \mu kf[xi]) \quad (2.4)$$

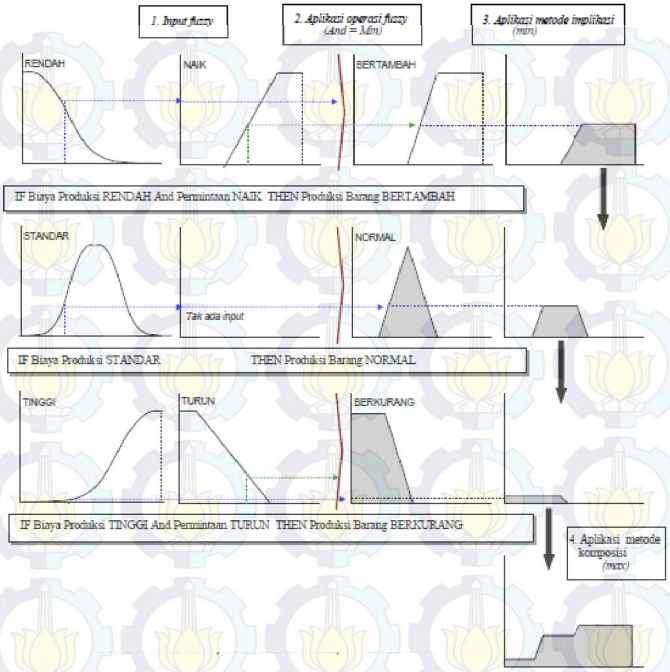
   dengan:
   a) μsf[xi] = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;
   b) μkf[xi] = nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

   Contoh inferensi dengan menerapkan metode *Max* untuk proses komposisi aturan seperti terlihat pada Gambar 2.9.

2) Metode *Addotive* (Sum)
   Setelah mendapatkan hasil komposisi aturan, pada metode Sum akan dilakukan *bounded-sum* yang akan menghasilkan solusi himpunan *fuzzy*.
   Secara umum dituliskan:

   $$\mu sf[xi] \leftarrow \mu min(1, \mu sf[xi] + \mu kf[xi]) \quad (2.2)$$

   dengan:
   a) μsf[xi] = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;
   b) μkf[xi] = nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

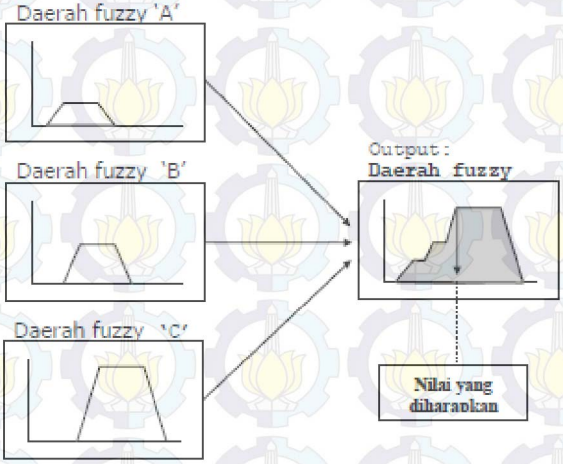


Gambar 2.9 Komposisi Aturan *Fuzzy* : Metode *Max*[3]

3) Metode Probabilistik OR (probor)
Untuk metode Probor hasil akhir himpunan *fuzzy* diperoleh dengan cara melakukan *dot product* terhadap semua *output* daerah *fuzzy*. Secara umum dituliskan:
$\mu sf[xi] \leftarrow \mu\ (\mu sf[xi] + \mu kf[xi]) - (\mu sf[xi] * \mu kf[xi])$ (2.3)
dengan:
   a) µsf[xi] = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i;
   b) µkf[xi] = nilai keanggotaan konsekuen *fuzzy* aturan ke-i;

d. Penegasan (defuzzy)

Setelah melalui proses komposisi aturan selanjutnya hasil tersebut akan digunakan sebagai masukan dalam proses defuzzyfikasi yang akan menghasilkan suatu nilai yang terdapat pada domain himpunan *fuzzy*. Sehingga jika terdapat suatu himpunan *fuzzy* dalam jangkauan tertentu maka hasil keluarannya diambil dari suatu nilai *crsip* yang ada pada himpunan *fuzzy* tersebut seperti terlihat pada Gambar 2.10.



Gambar 2.10 Proses Defuzzyfikasi [3]

Berdasarkan [3] ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani yaitu :

1) Metode Centroid (*Composite Moment*)

Hasil akhir logika *fuzzy* didapatkan dengan cara mengambil titik pusat (z*) daerah *fuzzy*. Secara umum dirumuskan:

$$Z^* = \frac{\int_Z Z\mu(Z)dz}{\int_Z \mu(Z)dz} \qquad (2.5)$$

$$Z^* = \frac{\sum_{j-1}^{n} Zj\mu(Zj)}{\sum_{j-1}^{n} \mu(Zj)} \quad (2.6)$$

2) Metode *Mean of Maximum* (MOM)
Pada metode ini, hasil akhir didapatkan dengan mengambil dilai rata – rata dari domain yang mempunyai nilai keanggotaan paling tinggi (maksimum).

$$v_{o=} \sum_{j=1}^{j} \frac{v_j}{J} \quad (2.7)$$

$$v_j = v\, \mu_v (v) \quad (2.8)$$

v_o : nilai keluaran
J : jumlah harga maksimum
v_j : nilai keluaran maksimum ke - j
$\mu_v (v)$: derajat keanggotaan elemen – elemen pada fuzzy set v
v : semesta pembicaraan

3) Metode Bisektor
Untuk metode bisektor hasil akhir didapat dengan cara mengambil nilai pada domain *fuzzy* yang dengan nilai keanggotaannya setengah dari keseluruhan total nilai keanggotaan pada daerah *fuzzy*. Secara umum dituliskan:

$$Z_{p\ sedemikian\ hingga}\ \int_{\Re 1}^{p} \mu(z)dz = \int_{p}^{\Re n} \mu(z)dz \quad (2.6)$$

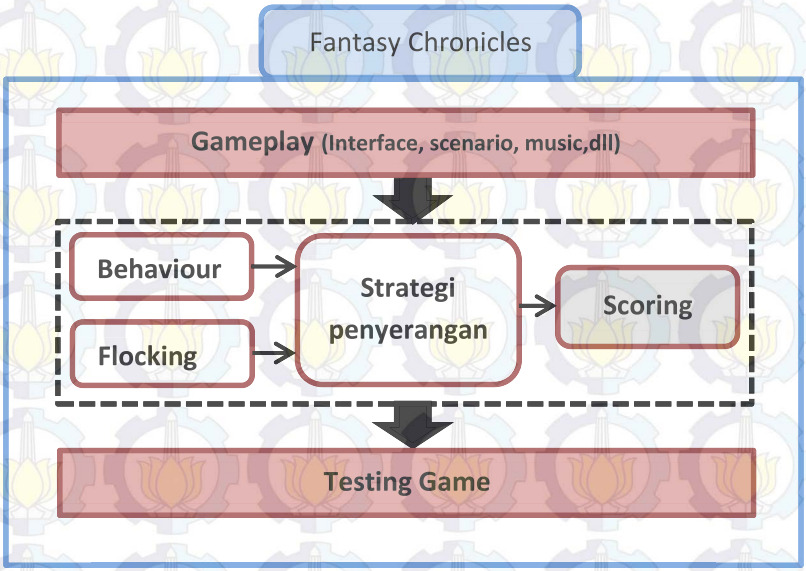
4) Metode *Largest of Maximum* (LOM)
Setelah mendapatkan kurva dari hasil komposisi aturan, pada metode LOM untuk mendapatkan nilai keluaran hasil akhir dari logika *fuzzy* didapatkan dari nilai terbesar yang terdapat pada domain yang memiliki nilai keanggotaan maksimum

5) Metode *Smallest of Maximum* (SOM)
Metode SOM berkebalikan dengan metode LOM jika pada metode LOM yang diambil adalah nilai terbesarnya sedangkan pada metode SOM yang digunakan sebagai solusi logika *fuzzy* adalah nilai terkecil yang terdapat pada nilai keanggotaan yang paling tinggi.

--Halaman ini sengaja dikosongkan--

BAB 3
PERANCANGAN SISTEM

Pada bab ini dibahas mengenai rancangan desain sistem yang akan diterapkan pada sebuah permainan(*game*) *Fantasy Chronicles* dimana permainan (*game)* ini dalam bentuk tiga dimensi yang mengambil tema atau *genre* semi RTS, *game* ini dikerjakan oleh lima orang mahasiswa jurusan Teknik Elektro ITS. Rancangan sistem *game Fantasy Chronicles* secara menyeluruh dari penelitian ini dapat dilihat pada gambar 3.1.



Gambar 3.1 Desain *game Fantasy Chronicles 3D*

Dari blok diagram pada gambar 3.1, pada penelitian ini yang akan dibahas hanya pada sistem *scoring* yang akan digunakan pada permainan tersebut. Penelitian meliputi perancangan sistem, perancangan variabel, dan perancangan aturan *fuzzy*.

3.1. Desain Sistem

Rancangan desain sistem akan dibagi menjadi empat tahapan untuk mendapatkan hasil yang sesuai. Tahapan desain sistem yang akan dikerjakan ditunjukan seperti pada Gambar 3.2

Gambar 3.2 Tahapan desain sistem

Tahapan pertama yang dilakukan adalah menentukan parameter apa saja yang akan digunakan sebagai masukan yang akan diolah pada sistem *fuzzy*. Setelah menentukan parameter selanjutnya ditentukan fungsi keanggotaan untuk setiap parameter yang berupa representasi kurva yang akan menggambarkan domain dari parameter tersebut. Selanjutnya akan dibuat aturan *fuzzy* untuk sistem *scoring* yang mencakup parameter yang sudah ditentukan sebelumnya. Aturan *fuzzy* akan dibuat untuk menentukan hasil akhir permainan yang berupa predikat yang akan didapatkan pemain yang direpresentasikan dalam bentuk bintang.

3.2. Perancangan Variable masukan

Parameter yang digunakan sebagai masukan merupakan properti yang dimiliki oleh unit yang terdapat pada *game*. Untuk menghitung *scoring* dari satu permainan digunakan parameter Kayu, Poin, Markas dan Pasukan untuk mendapatkan hasil predikat pemain pada setiap permainan.

3.2.1 Parameter Kayu

Kayu merupakan *resource* utama yang dimiliki oleh masing – masing unit. *Resource* akan bertambah 15 poin setiap 5 detik dan akan berkurang jika digunakan untuk membangun pasukan sesuai harga dari masing – masing pasukan. Jumlah kayu yang dihitung untuk hasil akhir permainan adalah sisa terakhir kayu pada saat permainan berakhir. Kayu memiliki jumlah maksimal sebesar 1000 yang berarti lumbung kayu penuh dan jumlah minimal 0 yang berarti kayu kosong.

3.2.2. Parameter Poin

Parameter Poin diperoleh dari perbandingan antara jumlah poin yang dikumpulan jika pemain berhasil mengalahkan lawan dan poin yang dipakai saat membangun pasukan. Masing – masing tipe lawan memiliki poin yang berbeda- beda. Untuk menghitung nilai parameter poin digunakan rumus:

*(poin dipakai / point didapat) *100* (3.1)

Hasil nilai poin akan berkisar antara 0 – 100. Nilai 0 berarti poin yang didapat tidak ada yang digunakan. Nilai 100 berarti poin yang didapat digunakan semua.

3.2.3. Parameter Pasukan

Pasukan adalah perbandingan antara pasukan yang dibangun oleh pemain dan jumlah pasukan pemain yang mati. Untuk menghitung nilai parameter pasukan digunakan rumus:

*(pasukan yang mati / pasukan yang dibangun) *100* (3.2)

Hasil nilai pasukan akan berkisar antara 0 – 100. Nilai 0 berarti pasukan yang dibangun tidak ada yang mati. Sedangan nilai 100 berarti semua pasukan pemain mati melawan musuh.

3.2.4. Parameter Markas

Parameter markas merupakan jumlah nilai yang diperoleh dari *health point* masing – masing bangunan yang dimiliki pemain pada saat permainan berakhir. Setiap bangunan memiliki *health point* sebesar

1000 karena pemain memiliki tiga markas maka jumlah total maksimal keseluruhan markas sebesar 3000. Besar nilai markas akan berkisar antara 0 – 3000. Nilai 0 berarti bangunan pemain tidak ada yang tersisa atau kalah. Nilai 3000 berarti bangunan pemain tidak ada yang hancur atau utuh.

3.2.5. Parameter Predikat

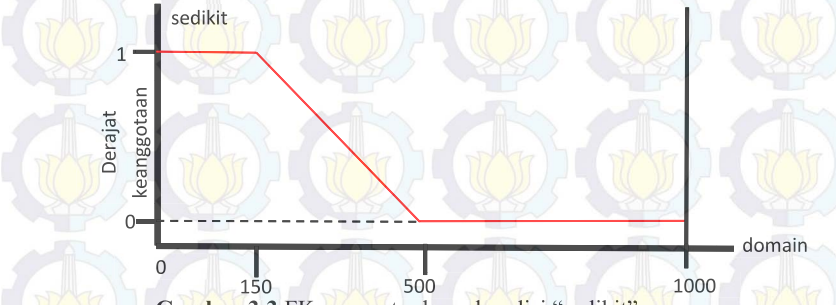
Predikat adalah *level* yang didapat pemain pada akhir permainan berdasarkan hasil permainannya yang dihitung dari kayu, poin, pasukan dan markas. Predikat memiliki range dari 0 – 10, nilai 0 berarti permainan pemain buruk (*poor*) sedangkan nilai 10 berarti permainan pemain sangat baik (*excelent*).

3.3. Perancangan Fungsi Keanggotaan Kayu

Fungsi keanggotaan digunakan untuk merepresentasikan parameter kayu pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga dan kurva trapesium .

3.3.1. Fungsi Keanggotaan Parameter Kayu Kondisi "sedikit"

Untuk kondisi "sedikit" digunakan representasi kurva trapesium dengan jangkauan antara 0 sampai dengan 500. Memiliki nilai maksimum pada range 0 – 150 dan nilai minimum pada nilai 500. Untuk fungsi keanggotaan kayu kondisi "sedikit" terlihat seperti pada Gambar 3.3.



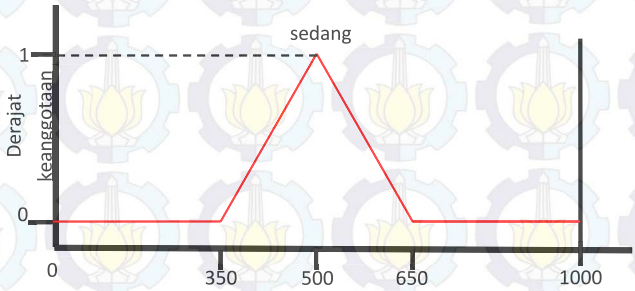
Gambar 3.3 FK parameter kayu kondisi "sedikit"

Untuk menghitung derajat keanggotaan parameter kayu kondisi "sedikit" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{kayu}Sedikit(x) = \begin{cases} 1, x \leq 150 \\ (500 - x)/(500 - 150), 150 \leq x \leq 500 \\ 0, x \geq 500 \end{cases} \quad (3.3)$$

3.3.2. Fungsi Keanggotaan Parameter Kayu Kondisi "sedang"

Untuk kondisi "sedang" digunakan representasi kurva segitiga dengan jangkauan antara 350 sampai dengan 650. Memiliki nilai maksimum pada nilai 500 dan nilai minimum pada nilai 350 dan 650. Untuk fungsi keanggotaan kayu kondisi "sedang" terlihat seperti pada Gambar 3.4.



Gambar 3.4 FK parameter kayu kondisi "sedang"

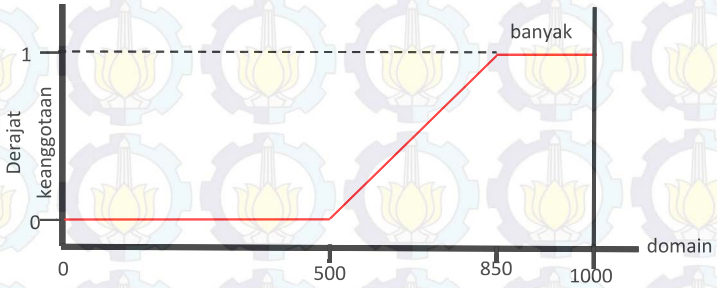
Untuk menghitung derajat keanggotaan parameter kayu kondisi "sedang" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{kayu}Sedang(x) = \begin{cases} 0, x \leq 350 \; dan \; x \geq 650 \\ (x - 350)/(500 - 350), 350 \leq x \leq 500 \\ (650 - x)/(650 - 500), 500 \leq x \leq 650 \end{cases} \quad (3.4)$$

3.3.3. Fungsi Keanggotaan Parameter Kayu Kondisi "banyak"

Untuk kondisi "banyak" digunakan representasi kurva trapesium dengan jangkauan antara 500 sampai dengan 1000. Memiliki nilai maksimum pada range 850 – 1000 dan nilai minimum pada nilai

500. Untuk fungsi keanggotaan kayu kondisi "banyak" terlihat seperti pada Gambar 3.5.



Gambar 3.5 FK parameter kayu kondisi "banyak"

Untuk menghitung derajat keanggotaan parameter kayu kondisi "banyak" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{kayu}Banyak(x) = \begin{cases} 0, x \leq 500 \\ (x-500)/(850-500), 500 \leq x \leq 850 \\ 1, 850 \leq x \leq 1000 \end{cases} \quad (3.5)$$

3.4. Perancangan Fungsi Keanggotaan Poin

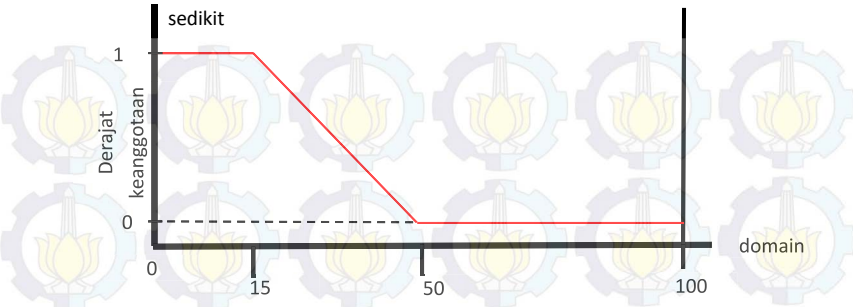
Fungsi keanggotaan digunakan untuk merepresentasikan parameter poin pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga dan kurva trapesium .

3.4.1. Fungsi Keanggotaan Parameter Poin Kondisi "sedikit"

Untuk kondisi "sedikit" digunakan representasi kurva trapesium dengan jangkauan antara 0 sampai dengan 50. Memiliki nilai maksimum pada range 0 – 15 dan nilai minimum pada nilai 50. Untuk fungsi keanggotaan poin kondisi "sedikit" terlihat seperti pada Gambar 3.6.

Untuk menghitung derajat keanggotaan parameter poin kondisi "sedikit" dari setiap nilai yang dimasukan digunakan rumus :

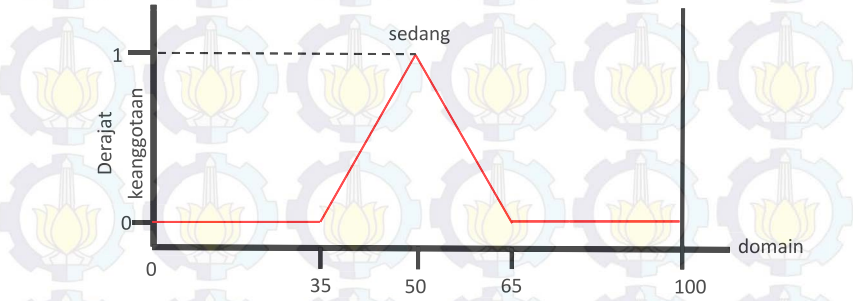
$$\mu_{poin}sedikit(x) = \begin{cases} 0, x \geq 50 \\ (50-x)/(50-15), 15 \leq x \leq 50 \\ 1, x \leq 15 \end{cases} \quad (3.6)$$



Gambar 3.6 FK parameter poin kondisi "sedikit"

3.4.2. Fungsi Keanggotaan Parameter Poin Kondisi "sedang"

Untuk kondisi "sedang" digunakan representasi kurva segitiga dengan jangkauan antara 35 sampai dengan 65. Memiliki nilai maksimum pada nilai 50 dan nilai minimum pada nilai 35 dan 65. Untuk fungsi keanggotaan poin kondisi "sedang" terlihat seperti pada Gambar 3.7.



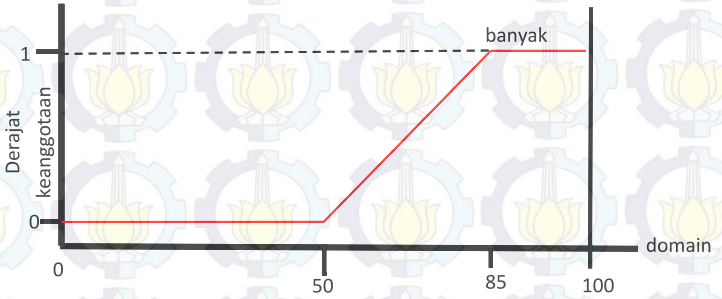
Gambar 3.7 FK parameter poin kondisi "sedang"

Untuk menghitung derajat keanggotaan parameter poin kondisi "sedang" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{poin}sedang(x) = \begin{cases} 0, x \leq 35 \; atau \; x \geq 65 \\ (x-35)/(50-35), 35 \leq x \leq 50 \\ (65-x)/(65-50), 50 \leq x \leq 65 \end{cases} \quad (3.7)$$

3.4.3. Fungsi Keanggotaan Parameter Poin Kondisi "banyak"

Untuk kondisi "banyak" digunakan representasi kurva trapesium dengan jangkauan antara 50 sampai dengan 100. Memiliki nilai maksimum pada range 85 – 100 dan nilai minimum pada nilai 50. Untuk fungsi keanggotaan poin kondisi "banyak" terlihat seperti pada Gambar 3.8.



Gambar 3.8 FK parameter poin kondisi "banyak"

Untuk menghitung derajat keanggotaan parameter poin kondisi "banyak" dari setiap nilai yang dimasukan digunakan rumus :

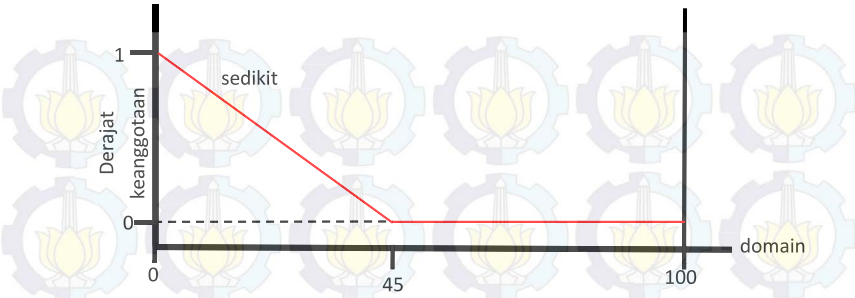
$$\mu_{poin}Banyak(x) = \begin{cases} 0\,, x \leq 50 \\ (x-50)/(85-50)\,, 50 \leq x \leq 85 \\ 1\,, 85 \leq x \leq 100 \end{cases} \quad (3.8)$$

3.5. Perancangan Fungsi Keanggotaan Pasukan

Fungsi keanggotaan digunakan untuk merepresentasikan parameter pasukan pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga.

3.5.1. Fungsi Keanggotaan Parameter Pasukan Kondisi "sedikit"

Untuk kondisi "sedikit" digunakan representasi kurva segitiga dengan jangkauan antara 0 samapi dengan 45. Memiliki nilai maksimum pada nilai 0 dan nilai minimum pada nilai 45. Untuk fungsi keanggotaan pasukan kondisi "sedikit" terlihat seperti pada Gambar 3.9



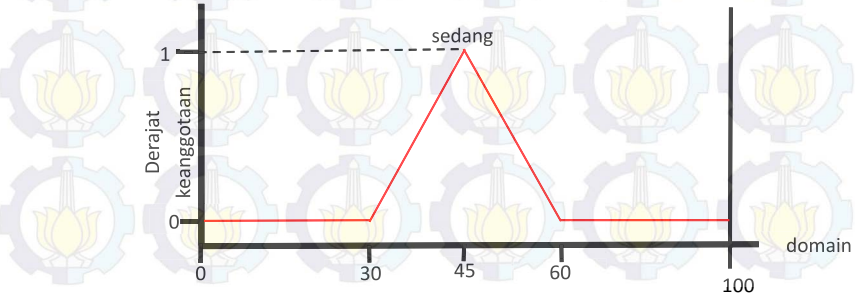
Gambar 3.9 FK parameter pasukan kondisi "sedikit"

Untuk menghitung derajat keanggotaan parameter pasukan kondisi "sedikit" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pas}Sedikit(x) = \begin{cases} 1, x \leq 0 \\ (45-x)/(45-0), 0 \leq x \leq 45 \\ 0, x \geq 45 \end{cases} \quad (3.9)$$

3.5.2. Fungsi Keanggotaan Parameter Pasukan Kondisi "sedang"

Untuk kondisi "sedang" digunakan representasi kurva segitiga dengan jangkauan antara 30 sampai dengan 60. Memiliki nilai maksimum pada range nilai 45 dan nilai minimum pada nilai 60 dan 30. Untuk fungsi keanggotaan pasukan kondisi "sedang" terlihat seperti pada Gambar 3.10.



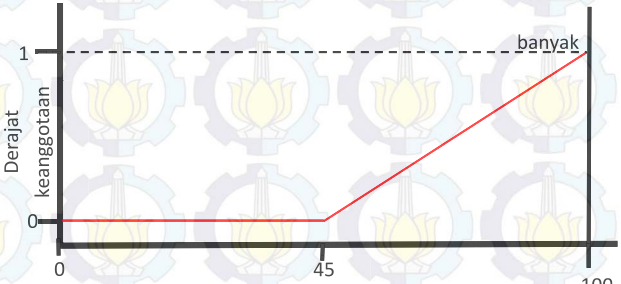
Gambar 3.10 FK parameter pasukan kondisi "sedang"

Untuk menghitung derajat keanggotaan parameter pasukan kondisi "sedang" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pas}Sedang(x) = \begin{cases} (45-x)/(45-30), 30 \le x \le 45 \\ (x-45)/(60-45), 45 \le x \le 60 \\ 0, x \le 30 \; atau \; x \ge 60 \end{cases} \quad (3.10)$$

3.5.3. Fungsi Keanggotaan Parameter Pasukan Kondisi "banyak"

Untuk kondisi "banyak" digunakan representasi kurva segitiga dengan jangkauan antara 45 sampai dengan 100. Memiliki nilai maksimum nilai 100 dan nilai minimum pada nilai 45. Untuk fungsi keanggotaan pasukan kondisi "banyak" terlihat seperti pada Gambar 3.11.



Gambar 3.11 FK parameter pasukan kondisi "banyak"

Untuk menghitung derajat keanggotaan parameter pasukan kondisi "banyak" dari setiap nilai yang dimasukan digunakan rumus :

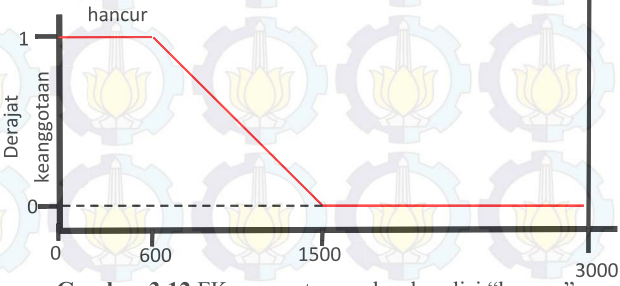
$$\mu_{pas}Banyak(x) = \begin{cases} 0, x \le 45 \\ (100-x)/(100-45), 45 \le x \le 100 \\ 1, x \ge 100 \end{cases} \quad (3.11)$$

3.6. Perancangan Fungsi Keanggotaan Markas

Fungsi keanggotaan digunakan untuk merepresentasikan parameter markas pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga dan trapesium.

3.6.1. Fungsi Keanggotaan Parameter Markas Kondisi "hancur"

Untuk kondisi "hancur" digunakan representasi kurva segitiga dengan jangkauan antara 0 sampai dengan 1500. Memiliki nilai maksimum pada range 0 - 600 dan nilai minimum pada nilai 1500. Untuk fungsi keanggotaan markas kondisi "hancur" terlihat seperti pada Gambar 3.12



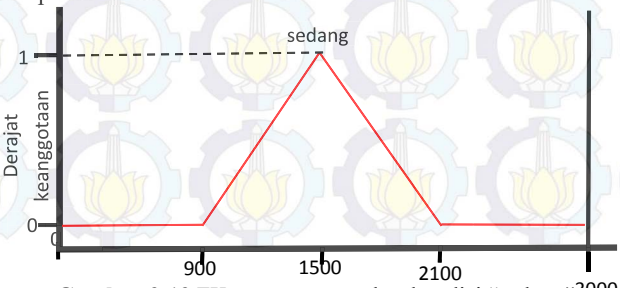
Gambar 3.12 FK parameter markas kondisi "hancur"

Untuk menghitung derajat keanggotaan parameter markas kondisi "hancur" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{mar}Hancur(x) = \begin{cases} 1, x \leq 600 \\ (x-600)/(1500-600), 600 \leq x \leq 1500 \\ 0, x \geq 1500 \end{cases} \quad (3.12)$$

3.6.2. Fungsi Keanggotaan Parameter Markas Kondisi "sedang"

Untuk kondisi "sedang" digunakan representasi kurva segitiga dengan jangkauan antara 900 sampai dengan 2100. Memiliki nilai maksimum pada range 1500 dan nilai minimum pada nilai 900 dan 1500. Untuk fungsi keanggotaan markas kondisi "sedang" terlihat seperti pada Gambar 3.13



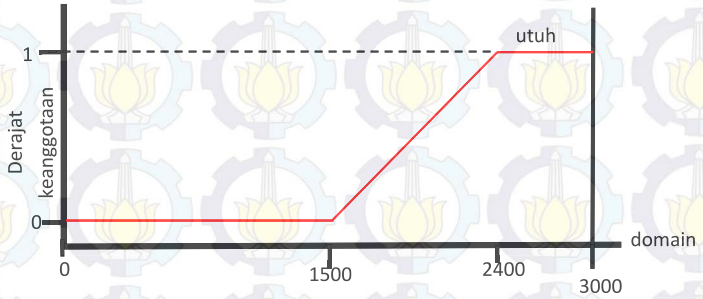
Gambar 3.13 FK parameter markas kondisi "sedang"

Untuk menghitung derajat keanggotaan parameter markas kondisi "sedang" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{mar}sedang(x) = \begin{cases} 0\,, x \leq 900\ atau\ x \geq 2100 \\ (1500 - x)/(1500 - 900)\,, 900 \leq x \leq 1500 \\ (x - 1500)/(2100 - 1500)\,, 1500 \leq x \leq 2100 \end{cases} \quad (3.13)$$

3.6.3. Fungsi Keanggotaan Parameter Markas Kondisi "utuh"

Untuk kondisi "utuh" digunakan representasi kurva trapesium dengan jangkauan antara 1500 sampai dengan 3000. Memiliki nilai maksimum pada range 2400 - 3000 dan nilai minimum pada nilai 1500. Untuk fungsi keanggotaan markas kondisi "utuh" terlihat seperti pada Gambar 3.14



Gambar 3.14 FK parameter markas kondisi "utuh"

Untuk menghitung derajat keanggotaan parameter markas kondisi "utuh" dari setiap nilai yang dimasukan digunakan rumus :

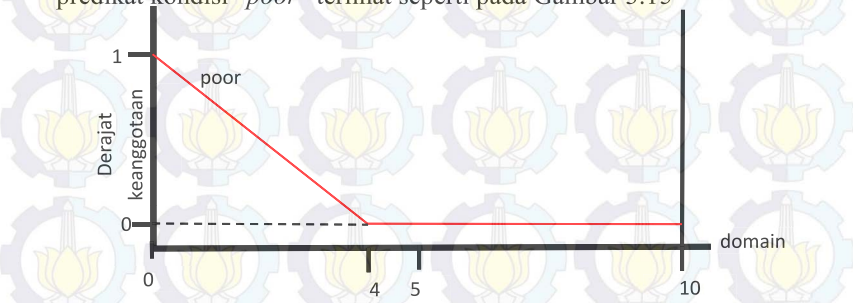
$$\mu_{mar}Utuh(x) = \begin{cases} 0\,, x \leq 1500 \\ (2400 - x)/(2400 - 1500)\,, 1500 \leq x \leq 2400 \\ 1, x \geq 2400 \end{cases} \quad (3.14)$$

3.7. Perancangan Fungsi Keanggotaan Predikat

Fungsi keanggotaan digunakan untuk merepresentasikan parameter predikat pada suatu kurva. Fungsi keanggotaan yang akan digunakan adalah kurva segitiga.

3.7.1. Fungsi Keanggotaan Parameter Predikat Kondisi "*poor*"

Untuk kondisi "*poor*" digunakan representasi kurva segitiga dengan jangkauan antara 0 sampai dengan 4. Memiliki nilai maksimum pada nilai 0 dan nilai minimum pada nilai 4. Untuk fungsi keanggotaan predikat kondisi "*poor*" terlihat seperti pada Gambar 3.15



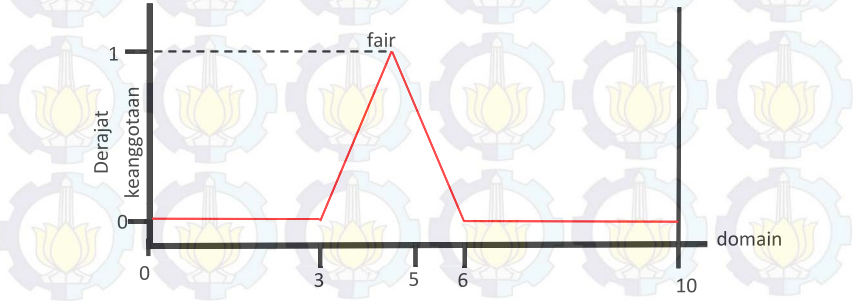
Gambar 3.15 FK parameter predikat kondisi "*poor*"

Untuk menghitung derajat keanggotaan parameter predikat kondisi "*poor*" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pre}poor(x) = \begin{cases} 1, x \leq 0 \\ (4-x)/(4-0), 0 \leq x \leq 4 \\ 0, x \geq 4 \end{cases} \quad (3.15)$$

3.7.2. Fungsi Keanggotaan Parameter Predikat Kondisi "*fair*"

Untuk kondisi "*fair*" digunakan representasi kurva segitiga dengan jangkauan antara 3 sampai dengan 6. Memiliki nilai maksimum pada nilai 4.5 dan nilai minimum pada nilai 4 dan 6. Untuk fungsi keanggotaan predikat kondisi "*fair*" terlihat seperti pada Gambar 3.16



Gambar 3.16 FK parameter predikat kondisi "*fair*"

Untuk menghitung derajat keanggotaan parameter predikat kondisi "*fair*" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pre} fair(x) = \begin{Bmatrix} 0\,, x \leq 3 \; atau \; x \geq 6 \\ (4.5 - x)/(4.5 - 3)\,, 3 \leq x \leq 4.5 \\ (6 - x)/(6 - 4.5)\,, 4.5 \leq x \leq 6 \end{Bmatrix} \quad (3.16)$$

3.7.3. Fungsi Keanggotaan Parameter Predikat Kondisi "*good*"

Untuk kondisi "*good*" digunakan representasi kurva segitiga dengan jangkauan antara 5 sampai dengan 8. Memiliki nilai maksimum pada nilai 6.5 dan nilai minimum pada nilai 5 dan 8. Untuk fungsi keanggotaan predikat kondisi "*good*" terlihat seperti pada Gambar 3.17

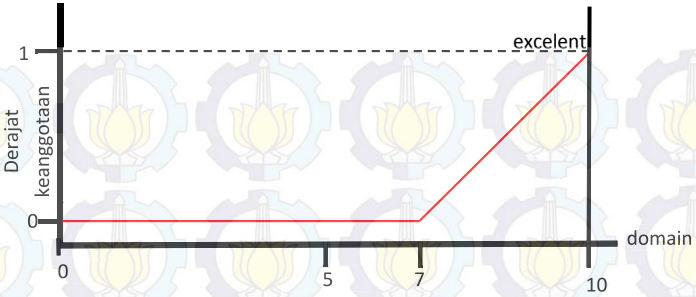
Gambar 3.17 FK parameter predikat kondisi "good"

Untuk menghitung derajat keanggotaan parameter predikat kondisi "*good*" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pre} Good(x) = \begin{Bmatrix} 0\,, x \leq 5 \; atau \; x \geq 8 \\ (6.5 - x)/(6.5 - 5)\,, 5 \leq x \leq 6.5 \\ (8 - x)/(8 - 6.5)\,, 6.5 \leq x \leq 8 \end{Bmatrix} \quad (3.17)$$

3.7.4. Fungsi Keanggotaan Parameter Predikat Kondisi "*excelent*"

Untuk kondisi "*excelent*" digunakan representasi kurva segitiga dengan jangkauan antara 7 sampai dengan 10. Memiliki nilai maksimum pada nilai 10 dan nilai minimum pada nilai 7. Untuk fungsi keanggotaan predikat kondisi "*excelent*" terlihat seperti pada Gambar 3.18



Gambar 3.18 FK parameter predikat kondisi "*excelent*"

Untuk menghitung derajat keanggotaan parameter predikat kondisi "*excelent*" dari setiap nilai yang dimasukan digunakan rumus :

$$\mu_{pre}Exc(x) = \begin{cases} 0\,, x \leq 7 \\ (x-7)/(10-7)\,, 7 \leq x \leq 10 \\ 1\,, x \geq 10 \end{cases} \quad (3.18)$$

3.8. Desain Aturan *Fuzzy (Rules Base)*

Aturan *fuzzy* berfungsi untuk memberikan aturan *IF -THEN* pada parameter yang sebelumnya sudah dirancang. Aturan *fuzzy* digunakan untuk menentukan predikat dengan parameter kayu, poin, markas dan pasukan. *Rules based* untuk predikat seperti pada Tabel 3.4 terdiri dari 81 aturan

Tabel 3.4: *Rule based fuzzy* Predikat

Rules	Masukan				Keluaran
	Markas	Kayu	Poin	Pasukan	Predikat
1	Hancur	Sedikit	Sedikit	Sedikit	*Fair*
2	Hancur	Sedikit	Sedikit	Sedang	*Fair*
3	Hancur	Sedikit	Sedikit	Banyak	*Poor*
4	Hancur	Sedikit	Sedang	Sedikit	*Poor*
5	Hancur	Sedikit	Sedang	Sedang	*Poor*
6	Hancur	Sedikit	Sedang	Banyak	*Poor*
7	Hancur	Sedikit	Banyak	Sedikit	*Poor*
8	Hancur	Sedikit	Banyak	Sedang	*Poor*
9	Hancur	Sedikit	Banyak	Banyak	*Poor*
10	Hancur	Sedang	Sedikit	Sedikit	*Fair*
11	Hancur	Sedang	Sedikit	Sedang	*Fair*

12	Hancur	Sedang	Sedikit	Banyak	*Poor*
13	Hancur	Sedang	Sedang	Sedikit	*Poor*
14	Hancur	Sedang	Sedang	Sedang	*Poor*
15	Hancur	Sedang	Sedang	Banyak	*Poor*
16	Hancur	Sedang	Banyak	Sedikit	*Poor*
17	Hancur	Sedang	Banyak	Sedang	*Poor*
18	Hancur	Sedang	Banyak	Banyak	*Poor*
19	Hancur	Banyak	Sedikit	Sedikit	*Fair*
20	Hancur	Banyak	Sedikit	Sedang	*Fair*
21	Hancur	Banyak	Sedikit	Banyak	*Poor*
22	Hancur	Banyak	Sedang	Sedikit	*Fair*
23	Hancur	Banyak	Sedang	Sedang	*Fair*
24	Hancur	Banyak	Sedang	Banyak	*Poor*
25	Hancur	Banyak	Banyak	Sedikit	*Fair*
26	Hancur	Banyak	Banyak	Sedang	*Poor*
27	Hancur	Banyak	Banyak	Banyak	*Poor*
28	Sedang	Sedikit	Sedikit	Sedikit	*Good*
29	Sedang	Sedikit	Sedikit	Sedang	*Good*
30	Sedang	Sedikit	Sedikit	Banyak	*Fair*
31	Sedang	Sedikit	Sedang	Sedikit	*Fair*
32	Sedang	Sedikit	Sedang	Sedang	*Fair*
33	Sedang	Sedikit	Sedang	Banyak	*Fair*
34	Sedang	Sedikit	Banyak	Sedikit	*Good*
35	Sedang	Sedikit	Banyak	Sedang	*Fair*
36	Sedang	Sedikit	Banyak	Banyak	*Fair*
37	Sedang	Sedang	Sedikit	Sedikit	*Good*
38	Sedang	Sedang	Sedikit	Sedang	*Good*
39	Sedang	Sedang	Sedikit	Banyak	*Fair*
40	Sedang	Sedang	Sedang	Sedikit	*Good*
41	Sedang	Sedang	Sedang	Sedang	*Good*
42	Sedang	Sedang	Sedang	Banyak	*Fair*
43	Sedang	Sedang	Banyak	Sedikit	*Fair*
44	Sedang	Sedang	Banyak	Sedang	*Fair*
45	Sedang	Sedang	Banyak	Banyak	*Fair*
46	Sedang	Banyak	Sedikit	Sedikit	*Good*
47	Sedang	Banyak	Sedikit	Sedang	*Good*
48	Sedang	Banyak	Sedikit	Banyak	*Good*

49	Sedang	Banyak	Sedang	Sedikit	*Good*
50	Sedang	Banyak	Sedang	Sedang	*Good*
51	Sedang	Banyak	Sedang	Banyak	*Fair*
52	Sedang	Banyak	Banyak	Sedikit	*Fair*
53	Sedang	Banyak	Banyak	Sedang	*Fair*
54	Sedang	Banyak	Banyak	Banyak	*fair*
55	Utuh	Sedikit	Sedikit	Sedikit	*Good*
56	Utuh	Sedikit	Sedikit	Sedang	*Good*
57	Utuh	Sedikit	Sedikit	Banyak	*Good*
58	Utuh	Sedikit	Sedang	Sedikit	*Excellent*
59	Utuh	Sedikit	Sedang	Sedang	*Good*
60	Utuh	Sedikit	Sedang	Banyak	*Good*
61	Utuh	Sedikit	Banyak	Sedikit	*Good*
62	Utuh	Sedikit	Banyak	Sedang	*Good*
63	Utuh	Sedikit	Banyak	Banyak	*Good*
64	Utuh	Sedang	Sedikit	Sedikit	*Excellent*
65	Utuh	Sedang	Sedikit	Sedang	*Good*
66	Utuh	Sedang	Sedikit	Banyak	*Good*
67	Utuh	Sedang	Sedang	Sedikit	*Excellent*
68	Utuh	Sedang	Sedang	Sedang	*Excellent*
69	Utuh	Sedang	Sedang	Banyak	*Good*
70	Utuh	Sedang	Banyak	Sedikit	*Good*
71	Utuh	Sedang	Banyak	Sedang	*Good*
72	Utuh	Sedang	Banyak	Banyak	*Good*
73	Utuh	Banyak	Sedikit	Sedikit	*Excellent*
74	Utuh	Banyak	Sedikit	Sedang	*Excellent*
75	Utuh	Banyak	Sedikit	Banyak	*Good*
76	Utuh	Banyak	Sedang	Sedikit	*Excellent*
77	Utuh	Banyak	Sedang	Sedang	*Excellent*
78	Utuh	Banyak	Sedang	Banyak	*Good*
79	Utuh	Banyak	Banyak	Sedikit	*Good*
80	Utuh	Banyak	Banyak	Sedang	*Good*
81	Utuh	Banyak	Banyak	Banyak	*Good*

--Halaman ini sengaja dikosongkan--

BAB 4
PENGUJIAN DAN ANALISA

Pada bab ini dilakukan pengujian dan analisa terhadap sistem *scoring* yang telah dibangun untuk mengetahui apakah sistem dapat diimplementasikan dan sesuai yang diharapan.

4.1 Skenario Pengujian

Pengujian dilakukan terhadap model *fuzzy* yang sebelumnya sudah ditentukan. Pengujian dilakukan dengan mengambil beberapa *sample* data yang diimplementasikan pada *game*. Untuk pengujian, sistem *fuzzy* diimplementasikan pada permainan yang telah dibangun. Pengujian dilakukan pada 2 tahapan. Pertama pengujian implementasi parameter yang ada pada sistem dan kedua pengujian penerapan metode dengan menerapkan dua kondisi permainan yaitu kondisi menang dan kondisi kalah dengan variasi hasil akhir yang berbeda – beda yang hasil predikatnya akan direpresentasikan dalam bentuk bintang..

4.2 Pengujian Implementasi Parameter

Pengujian parameter ini bertujuan untuk mengetahui apakah parameter yang akan digunakan sudah berjalan sesuai dengan fungsinya agar tidak terjadi kesalahan ketika parameter tersebut akan digunakan dalam sistem *scoring*. Pengujian parameter dilakukan pada semua parameter yang terlibat. Implementasi tiap - tiap parameter yang akan digunakan terlihat seperti pada Gambar 4.1.

Gambar 4.1 Implementasi variabel masukan

Pada gambar 4.1 terlihat implementasi pada masing – masing parameter berjalan sesuai dengan fungsinya. Pada parameter kayu jumlah kayu pada saat gambar diambil terlihat berjumlah 574 kayu hasil tersebut didapatkan dari total kayu awal sebesar 500 ditambahkan dengan kayu yang didapat selama 1 menit 18 detik sebesar 234 dan dikurangkan dengan kayu yang dipakai sebesar 160 sehingga sisa kayu sebesar 574.). Perhitungan untuk parameter kayu sebagai berikut :

kayu awal + Kayu selema 1 menit 18 detik – kayu dipakai

= 500 + 234 – 160 = 574

Untuk parameter poin jumlah poin sebesar 16 diperoleh dari point yang dipakai sebesar 48 dibagi dengan poin yang didapat sebesar 300 dikalikan dengan 100. Perhitungan untuk parameter kayu sebagai berikut :

(poin dipakai / poin didapai) 100 = (48/300) * 100 = 16*

Pada parameter pasukan dihitung dari jumlah pasukan yang mati sebesar 3 dibagi dengan pasukan yang dibangun sebesar 12 dikalikan dengan 100 sehingga menghasilkan nilai 25. Perhitungan untuk parameter kayu sebagai berikut :

(pasukan yang mati / pasukan yang dingaun)) 100 = (3/12) * 100 = 25*

Sedangkan pada markas bernilai 3000 didapat dari jumlah *health point* markas yang dimiliki pemain.

4.3 Pengujian Sistem *Scoring*

Pengujian sistem *scoring* dilakukan untuk mengetahui apakah *scoring* yang dibuat dengan *fuzzy* sudah berjalan dengan baik. Pengujian dilakukan dengan membandingkan hasil akhir permainan dengan beberapa kondisi predikat yang sudah ditentukan .

4.3.1. Pengujian Hasil Akhir Kondisi "*Poor*"

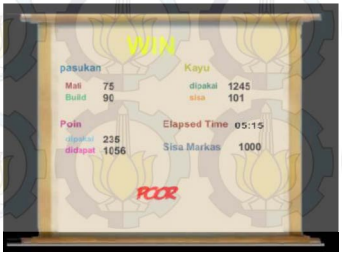
Pada pengujian ini akan diperoleh hasil akhir permainan dengan predikat "*poor*" yang mana diperoleh dari hasil pertempuran yang menghasilkan kondisi markas "hancur", kondisi kayu "sedikit", kondisi poin "sedikit", dan kondisi pasukan "banyak" pada akhir permainan. Gambar pertempuran hingga hasil akhir terlihat pada gambar 4.2.

(a)

(b)

(c)

(d)



(e)

Gambar 4.2 Hasil Akhir Kondisi "*poor*"
(a) Markas musuh utuh
(b) Markas musuh sisa 2
(c) Markas musuh sisa 1
(d) Markas musuh hancur
(e) Hasil akhir pertempuran

4.3.2. Pengujian Hasil Akhir Kondisi "*Fair*"

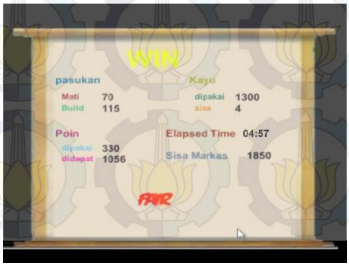
Pada pengujian ini akan diperoleh hasil akhir permainan dengan predikat "*fair*" yang mana diperoleh dari hasil pertempuran yang menghasilkan kondisi markas "sedang", kondisi kayu "sedikit", kondisi poin "sedikit", dan kondisi pasukan "banyak" pada akhir permainan. Gambar pertempuran hingga hasil akhir terlihat pada gambar 4.3.

(a)

(b)

(c)

(d)



(e)

Gambar 4.3 Hasil Akhir Kondisi "*fair*"
(a) Markas musuh utuh
(b) Markas musuh sisa 2
(c) Markas musuh sisa 1
(d) Markas pemain sisa 2
(e) Hasil akhir pertempuran

4.3.3. Pengujian Hasil Akhir Kondisi "*Good*"

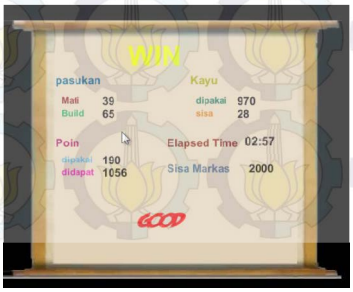
Pada pengujian ini akan diperoleh hasil akhir permainan dengan predikat "*good*" yang mana diperoleh dari hasil pertempuran yang menghasilkan kondisi markas "utuh", kondisi kayu "sedikit", kondisi poin "sedikit", dan kondisi pasukan "banyak" pada akhir permainan. Gambar pertempuran hingga hasil akhir terlihat pada gambar 4.4.

(a)

(b)

(c)

(d)



(e)

Gambar 4.4 Hasil Akhir Kondisi "*Good*"
(a) Markas musuh utuh
(b) Markas musuh sisa 2
(c) Markas musuh sisa 1
(d) Markas pemain sisa 2
(e) Hasil akhir pertempuran

4.3.4. Pengujian Hasil Akhir Kondisi "*Excelent*"

Pada pengujian ini akan diperoleh hasil akhir permainan dengan predikat "*Excelent*" yang mana diperoleh dari hasil pertempuran yang menghasilkan kondisi markas "utuh", kondisi kayu "banyak", kondisi poin "sedikit", dan kondisi pasukan "sedikit" pada akhir permainan. Gambar pertempuran hingga hasil akhir terlihat pada gambar 4.5.

(a)

(b)

(c)

(d)

(e)

Gambar 4.5 Hasil Akhir Kondisi "*excelent*"
(a) Markas musuh utuh
(b) Markas musuh sisa 2
(c) Markas musuh sisa 1
(d) Markas musuh hancur
(e) Hasil akhir pertempuran

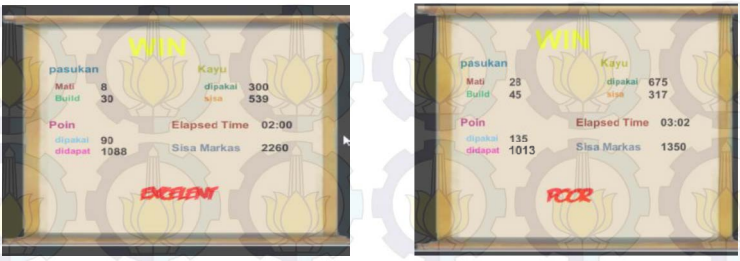
4.4 Pengujian Metode

Pengujian metode bertujuan untuk mengetahui apakah metode yang digunakan dapat diterapkan dalam permaianan. Dan juga untuk mengetahui hasil akhir permainan yang dilakukan dengan menggunakan skenario pertempuran. Pengujian dilakukan dengan kondisi permainan menang dan kalah dengan menggunakan metode *fuzzy* dan dibandingkan dengan tanpa menggunakan metode *fuzzy*.

4.4.1 Pengujian Hasil Akhir kondisi Menang

Pada pengujian ini akan dilakukan sepuluh kali pengujian untuk mendapatkan hasil akhir permainan yang berbeda - beda. Kemudian hasilnya akan dibandingkan dengan penghitungan hasil akhir tanpa menggunakan *fuzzy* (berdasarkan aturan tetap / statis). Penghitungan hasil akhir permainan akan melibatkan 4 parameter yaitu sisa *health point* markas pemain yang berhasil dipertahankan, perbandingan pasukan yang dibangun oleh pemain dan pasukan yang dibunuh musuh, perbandingan poin yang dipakai dan yang didapat oleh pemain, serta sisa kayu yang dimiliki pemain saat permainan berakhir.

Perhitungan hasil akhir tanpa menggunakan *fuzzy* memiliki aturan yaitu jika pada akhir permaianan markas pemain utuh maka mendapatkan bintang 3. Jika markas pemain sisa dua mendapatkan bintang 2. Dan jika markas pemain sisa satu maka mendapatkan bintang 1.Sedangkan untuk penghitungan hasil akhir dengan *fuzzy* pemain akan mendapatkan bintang antara 0 – 3 yang nilainya sesuai dengan nilai keluaran yang dihasilkan dari penghitungan logika *fuzzy*. Pada Gambar 4.5 terlihat salah satu hasil akhir permainan yang mendapat predikat "*excelent*" dan "*poor*" dengan hasil menang. Hasil dari sepuluh kali pengujian dengan hasil menang terdapat pada Tabel 4.1



Gambar 4.5 pengujian metode hasil permainan menang, predikat "*excelent*" dan "*poor*"

Tabel 4.1: Hasil Pengujian kondisi menang

Uji ke-	Markas	Kayu	Poin	Pasukan	Hasil penghitungan *fuzzy*	Representasi bintang	
						Fuzzy	Statis
1	1000	101	22	83	2.1667		
2	1850	4	31	60	5.499		
3	2000	28	17	60	5.736		
4	3000	578	8	20	8.76		
5	2260	539	8	26	8.74		
6	2560	107	17	51	6.584		
7	3000	647	32	27	8.681		
8	2140	119	18	65	6.584		
9	2410	56	15	45	6.650		
10	1350	317	13	62	3.291		

Pada hasil pengujian Table 4.1 terdapat beberapa hasil dengan bintang yang sama tetapi memiliki kondisi yang berbeda. Pada pengujian ke -6 , ke- 8, ke -9 mendapatkan bintang 2 pada pengujian *fuzzy* tetapi pada pengujian statis ketiganya mendapat bintang tiga. Untuk hasil lainnya, pada pengujian ke -4 sampai dengan ke -9 pada pengujian statis menujukan hasil bintang yang sama. Besar persentase hasil bintang yang sama antara statis dan *fuzzy* adalah sebesar 60% . Persentase tersebut didapat dari hasil perolehan bintang yang terdapat pada pengujian ke 1,2,3,4,5,dan 7. Sedangkan persentase hasil bintang yang tidak sama antara statis dan *fuzzy* sebesar 40% yang diperoleh dari pengujian ke 6,8,9, dan10.

Dari hasil pengujian Table 4.1 jika dihitung nilai standar deviasinya (variasi data) maka nilai untuk pengujian *fuzzy* sebesar 0.7378. sedangkan pada pengujian statis nilainya sebesar 0.7017. dari kedua niai tersebut menunjukan bawah pengujian *fuzzy* memberikan nilai lebih bervariasi / lebih dinamik dariada pengujian statis karena memiliki nilai standar deviasi yang lebih besar.

4.4.2 Pengujian Hasil Akhir kondisi Kalah

Pada pengujian kondisi kalah ini hanya akan dilakukan tiga kali pengujian untuk mendapatkan hasil akhir permainan dengan kondisi kalah. Kemudian hasilnya akan dibandingkan dengan penghitungan hasil akhir tanpa menggunakan *fuzzy* (berdasarkan aturan tetap / statis).

Perhitungan hasil akhir tanpa menggunakan *fuzzy* untuk kondisi kalah memiliki aturan yaitu pemain tidak akan mendapatkan bintang jika kondisinya kalah. Sedangkan untuk penghitungan hasil akhir dengan *fuzzy* bintang yang didapat antara 0 – 3 yang nilainya sesuai dengan nilai keluaran yang dihasilkan dari penghitungan logika *fuzzy*. Pada Gambar 4.6 terlihat salah satu hasil akhir permainan dengan hasil kalah. Hasil dari tiga kali pengujian dengan hasil kalah terdapat pada Tabel 4.2

Gambar 4.6 pengujian metode hasil permainan kalah predikat “poor”

Tabel 4.2: Hasil Pengujian kondisi kalah

Uji ke-	Markas	Kayu	Poin	Pasukan	Hasil penghitungan *fuzzy*	Representasi Bintang	
						Fuzzy	Statis
1	0	150	35	98	1.607		-
2	0	13	21	87.5	1.390		-
3	0	502	47	85	1.491		-

Dari hasil pengujian pada table 4.2 terdapat hasil bintang yang sama tetapi memiliki kondisi yang berbeda. Karena dalam aturan *fuzzy* untuk kondisi kalah / markas hancur hanya terdapat dua kemungkinan predikat yaitu "*poor*" dan "*fair*". Sedangkan pada hasil akhir tanpa *fuzzy* pemain tidak akan mendapatkan hasil apapun jika kalah. Hal ini menjelaskan bahwa dengan *fuzzy* kita tetap dapat memberi nilai pada hasil kerja pemain walaupun pemain kalah dalam permainan.

--Halaman ini sengaja dikosongkan--

DAFTAR PUSTAKA

[1] Graf Alan.“ Fuzzy Logic Approach for Modelling Multiplayer Game Scoring System”, Siemens d.d, Croatia, 2005.

[2] Johanna M. M. Goertz and Fran¸cois Maniquet, “On the Informational Efficiency of Simple Scoring Rules”, University of Guelph and Universit´e catholique de Louvain, 2008.

[3] Kusumadewi,S & Purnomo. 2004. “Aplikasi Logika Fuzzy Untuk Pendukung Keputusan”. Yogyakarta.Graha ilmu

[4] Li, Yifan, Musilek Petr, & Wyard-Scott. “Fuzzy Logic in Agent-Based Game Design”, Department of Electrical and Computer Engineering,University of Alberta Canada, 2004. IEEE 0-7803-8376-1/04

[5] Stene Sindre Berg. “Artificial Intelligence Techniques in Real-Time Strategy Games - Architecture and Combat Behavior”, Department of Computer and Information Science , Norwegian University of Science and Technology, 2006

[6] Rollings. Andrew, Dave Morris. “*Game Architecture and Design”*. Indianapolis, Indiana. New Riders Publisher, 2004.

--Halaman ini sengaja dikosongkan--

BAB 5
PENUTUP

5.1. Kesimpulan

Dari hasil pengujian Tugas Akhir ini pada bab sebelumnya dapat disimpulkan bahwa :

1. Nilai standar deviasi (variasi data) untuk pengujian *fuzzy* sebesar 0.7378 sedangkan pada pengujian statis nilainya sebesar 0.7017.
2. Penghitungan scoring menggunakan *fuzzy* memberikan nilai lebih bervariasi / lebih dinamis daripada penghitungan statis karena memiliki nilai standar deviasi yang lebih besar
3. Besar persentase hasil bintang yang sama antara penghitungan statis dan *fuzzy* sebesar 60%. Sisanya sebesar 40% adalah hasil perolehan bintang yang berbeda.

5.2. Saran

1. Dapat ditambahkan parameter – parameter lainnya yang terlibat dalam permainan yang dapat dimanfaatkan untuk menentukan hasil akhir permainan.
2. Permainan dapat dikembangkan menjadi permainan *multiplayer* dan dapat memanfaatkan *fuzzy* hasil akhir sebagai referensi aturan untuk peringkat (*ranking*) pemain karena hasilnya yang dinamis .

--Halaman ini sengaja dikosongkan--

LAMPIRAN

Result.CS

```
using UnityEngine;
using System.Collections;
using AForge.Fuzzy;
using System;

public class result : MonoBehaviour
{
	// Use this for initialization
	public GUIText playerkill;
	public GUIText currentgold;
	public GUIText currentpoin;
	public GUIText elapsedtime;
	public GUIText hasilstar;
	public GUIText
playerbuild,status,buildtx,deadtx,goldlosttx,pointlosttx,predikat;
	public GUITexture bintang;
	float kill,poin,build,dead,pointlost,goldlost;
	int bangunan;
	float gold;
	string time,sts;
	private InferenceSystem IS;
	float hasil,NewAngle;
	double newBarWidth,z;

	void fuzzyfuzzya()
	{
		// Linguistic labels (fuzzy sets) that compose the
distances
		//menentukan variable fuzzy kayu
		FuzzySet fsgsedikit = new FuzzySet("sedikit", new
TrapezoidalFunction(150, 500, TrapezoidalFunction.EdgeType.Right));
		FuzzySet fsgsedang = new FuzzySet("sedang", new
TrapezoidalFunction(350, 500, 650));
```

```
                FuzzySet fsgbanyak = new FuzzySet("banyak", new
TrapezoidalFunction(500, 1000, TrapezoidalFunction.EdgeType.Left));

                LinguisticVariable lvgold = new
LinguisticVariable("gold", 0, 1000);
                lvgold.AddLabel(fsgsedikit);
                lvgold.AddLabel(fsgsedang);
                lvgold.AddLabel(fsgbanyak);
                //----------------------------//

                //menentukan variable fuzzy POINT
                FuzzySet fspsedikit = new FuzzySet("sedikit", new
TrapezoidalFunction(15, 50, TrapezoidalFunction.EdgeType.Right));
                FuzzySet fspsedang = new FuzzySet("sedang", new
TrapezoidalFunction(35,50,65));
                FuzzySet fspbanyak = new FuzzySet("banyak", new
TrapezoidalFunction(50,100, TrapezoidalFunction.EdgeType.Left));

                LinguisticVariable lvpoint = new
LinguisticVariable("point", 0, 100);
                lvpoint.AddLabel(fspsedikit);
                lvpoint.AddLabel(fspsedang);
                lvpoint.AddLabel(fspbanyak );

                //----------------------------//

                //menentukan variable fuzzy MARKAS
                FuzzySet fsbhancur = new FuzzySet("hancur", new
TrapezoidalFunction(600, 1500, TrapezoidalFunction.EdgeType.Right));
                FuzzySet fsbsedang = new FuzzySet("sedang", new
TrapezoidalFunction(900, 1500, 2100));
                FuzzySet fsbutuh = new FuzzySet("utuh", new
TrapezoidalFunction(1500, 2400, TrapezoidalFunction.EdgeType.Left));

                LinguisticVariable lvbangunan = new
LinguisticVariable("markas", 0, 3000);
                lvbangunan.AddLabel(fsbhancur);
```

```
                lvbangunan.AddLabel(fsbsedang);
                lvbangunan.AddLabel(fsbutuh );
                //---------------------------//

                //menentukan variable fuzzy pasukan
                FuzzySet fsksedikit = new FuzzySet("sedikit", new
TrapezoidalFunction(0, 45, TrapezoidalFunction.EdgeType.Right));
                FuzzySet fsksedang = new FuzzySet("sedang", new
TrapezoidalFunction(30, 40, 60));
                FuzzySet fskbanyak = new FuzzySet("banyak", new
TrapezoidalFunction(45, 100, TrapezoidalFunction.EdgeType.Left));

                LinguisticVariable lvek = new LinguisticVariable("ek",
0, 100);
                lvek.AddLabel(fsksedikit);
                lvek.AddLabel(fsksedang);
                lvek.AddLabel(fskbanyak);
                //---------------------------//


                // Linguistic labels (fuzzy sets) that compose the angle
                // menentukan variable fuzzy predikat
                FuzzySet fspemula = new FuzzySet("poor", new
TrapezoidalFunction(0, 4, TrapezoidalFunction.EdgeType.Right));
                FuzzySet fsinter = new FuzzySet("fair", new
TrapezoidalFunction(3, 4, 6));
                FuzzySet fsadvan = new FuzzySet("good", new
TrapezoidalFunction(5, 7, 8));
                FuzzySet fsexpert = new FuzzySet("exc", new
TrapezoidalFunction(7, 10, TrapezoidalFunction.EdgeType.Left));

                // predikat
                LinguisticVariable lvpredikat = new
LinguisticVariable("predikat", 0, 10);
                lvpredikat.AddLabel(fspemula);
                lvpredikat.AddLabel(fsinter);
                lvpredikat.AddLabel(fsadvan);
```

```
                lvpredikat.AddLabel(fsexpert);

                // The database
                Database fuzzyDB = new Database();
                fuzzyDB.AddVariable(lvgold);
                fuzzyDB.AddVariable(lvpoint);
                fuzzyDB.AddVariable(lvbangunan);
                fuzzyDB.AddVariable(lvek);
                fuzzyDB.AddVariable(lvpredikat);

                // Creating the inference system
                IS = new InferenceSystem(fuzzyDB, new
CentroidDefuzzifier(1000));
                IS.NewRule("Rule 1", "IF markas IS hancur AND gold IS
sedikit AND point IS sedikit AND ek IS sedikit THEN predikat IS fair");
                IS.NewRule("Rule 2", "IF markas IS hancur AND gold IS
sedikit AND point IS sedikit AND ek IS sedang THEN predikat IS fair");
                IS.NewRule("Rule 3", "IF markas IS hancur AND gold IS
sedikit AND point IS sedikit AND ek IS banyak THEN predikat IS poor");
                IS.NewRule("Rule 4", "IF markas IS hancur AND gold IS
sedikit AND point IS sedang AND ek IS sedikit THEN predikat IS poor");
                IS.NewRule("Rule 5", "IF markas IS hancur AND gold IS
sedikit AND point IS sedang AND ek IS sedang THEN predikat IS poor");
                IS.NewRule("Rule 6", "IF markas IS hancur AND gold IS
sedikit AND point IS sedang AND ek IS banyak THEN predikat IS poor");
                IS.NewRule("Rule 7", "IF markas IS hancur AND gold IS
sedikit AND point IS banyak AND ek IS sedikit THEN predikat IS poor");
                IS.NewRule("Rule 8", "IF markas IS hancur AND gold IS
sedikit AND point IS banyak AND ek IS sedang THEN predikat IS poor");
                IS.NewRule("Rule 9", "IF markas IS hancur AND gold IS
sedikit AND point IS banyak AND ek IS banyak THEN predikat IS poor");
                IS.NewRule("Rule 10", "IF markas IS hancur AND gold
IS sedang AND point IS sedikit AND ek IS sedikit THEN predikat IS fair");

                IS.NewRule("Rule 11", "IF markas IS hancur AND gold
IS sedang AND point IS sedikit AND ek IS sedang THEN predikat IS fair");
```

```
IS.NewRule("Rule 12", "IF markas IS hancur AND gold
IS sedang AND point IS sedikit AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 13", "IF markas IS hancur AND gold
IS sedang AND point IS sedang AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 14", "IF markas IS hancur AND gold
IS sedang AND point IS sedang AND ek IS sedang THEN predikat IS
fair");
IS.NewRule("Rule 15", "IF markas IS hancur AND gold
IS sedang AND point IS sedang AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 16", "IF markas IS hancur AND gold
IS sedang AND point IS banyak AND ek IS sedikit THEN predikat IS
poor");
IS.NewRule("Rule 17", "IF markas IS hancur AND gold
IS sedang AND point IS banyak AND ek IS sedang THEN predikat IS
poor");
IS.NewRule("Rule 18", "IF markas IS hancur AND gold
IS sedang AND point IS banyak AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 19", "IF markas IS hancur AND gold
IS banyak AND point IS sedikit AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 20", "IF markas IS hancur AND gold
IS banyak AND point IS sedikit AND ek IS sedang THEN predikat IS fair");

IS.NewRule("Rule 21", "IF markas IS hancur AND gold
IS banyak AND point IS sedikit AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 22", "IF markas IS hancur AND gold
IS banyak AND point IS sedang AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 23", "IF markas IS hancur AND gold
IS banyak AND point IS sedang AND ek IS sedang THEN predikat IS
fair");
IS.NewRule("Rule 24", "IF markas IS hancur AND gold
IS banyak AND point IS sedang AND ek IS banyak THEN predikat IS
poor");
```

```
IS.NewRule("Rule 25", "IF markas IS hancur AND gold
IS banyak AND point IS banyak AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 26", "IF markas IS hancur AND gold
IS banyak AND point IS banyak AND ek IS sedang THEN predikat IS
poor");
IS.NewRule("Rule 27", "IF markas IS hancur AND gold
IS banyak AND point IS banyak AND ek IS banyak THEN predikat IS
poor");
IS.NewRule("Rule 28", "IF markas IS sedang AND gold
IS sedikit AND point IS sedikit AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 29", "IF markas IS sedang AND gold
IS sedikit AND point IS sedikit AND ek IS sedang THEN predikat IS
good");
IS.NewRule("Rule 30", "IF markas IS sedang AND gold
IS sedikit AND point IS sedikit AND ek IS banyak THEN predikat IS fair");

IS.NewRule("Rule 31", "IF markas IS sedang AND gold
IS sedikit AND point IS sedang AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 32", "IF markas IS sedang AND gold
IS sedikit AND point IS sedang AND ek IS sedang THEN predikat IS fair");
IS.NewRule("Rule 33", "IF markas IS sedang AND gold
IS sedikit AND point IS sedang AND ek IS banyak THEN predikat IS fair");
IS.NewRule("Rule 34", "IF markas IS sedang AND gold
IS sedikit AND point IS banyak AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 35", "IF markas IS sedang AND gold
IS sedikit AND point IS banyak AND ek IS sedang THEN predikat IS fair");
IS.NewRule("Rule 36", "IF markas IS sedang AND gold
IS sedikit AND point IS banyak AND ek IS banyak THEN predikat IS fair");
IS.NewRule("Rule 37", "IF markas IS sedang AND gold
IS sedang AND point IS sedikit AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 38", "IF markas IS sedang AND gold
IS sedang AND point IS sedikit AND ek IS sedang THEN predikat IS
good");
```

```
IS.NewRule("Rule 39", "IF markas IS sedang AND gold
IS sedang AND point IS sedikit AND ek IS banyak THEN predikat IS fair");
IS.NewRule("Rule 40", "IF markas IS sedang AND gold
IS sedang AND point IS sedang AND ek IS sedikit THEN predikat IS
good");

IS.NewRule("Rule 41", "IF markas IS sedang AND gold
IS sedang AND point IS sedang AND ek IS sedang THEN predikat IS
good");
IS.NewRule("Rule 42", "IF markas IS sedang AND gold
IS sedang AND point IS sedang AND ek IS banyak THEN predikat IS
fair");
IS.NewRule("Rule 43", "IF markas IS sedang AND gold
IS sedang AND point IS banyak AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 44", "IF markas IS sedang AND gold
IS sedang AND point IS banyak AND ek IS sedang THEN predikat IS
fair");
IS.NewRule("Rule 45", "IF markas IS sedang AND gold
IS sedang AND point IS banyak AND ek IS banyak THEN predikat IS
fair");
IS.NewRule("Rule 46", "IF markas IS sedang AND gold
IS banyak AND point IS sedikit AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 47", "IF markas IS sedang AND gold
IS banyak AND point IS sedikit AND ek IS sedang THEN predikat IS
good");
IS.NewRule("Rule 48", "IF markas IS sedang AND gold
IS banyak AND point IS sedikit AND ek IS banyak THEN predikat IS
good");
IS.NewRule("Rule 49", "IF markas IS sedang AND gold
IS banyak AND point IS sedang AND ek IS sedikit THEN predikat IS
good");
IS.NewRule("Rule 50", "IF markas IS sedang AND gold
IS banyak AND point IS sedang AND ek IS sedang THEN predikat IS
good");
```

```
IS.NewRule("Rule 51", "IF markas IS sedang AND gold
IS banyak AND point IS sedang AND ek IS banyak THEN predikat IS
fair");
IS.NewRule("Rule 52", "IF markas IS sedang AND gold
IS banyak AND point IS banyak AND ek IS sedikit THEN predikat IS fair");
IS.NewRule("Rule 53", "IF markas IS sedang AND gold
IS banyak AND point IS banyak AND ek IS sedang THEN predikat IS
fair");
IS.NewRule("Rule 54", "IF markas IS sedang AND gold
IS banyak AND point IS banyak AND ek IS banyak THEN predikat IS
fair");
IS.NewRule("Rule 55", "IF markas IS utuh AND gold IS
sedikit AND point IS sedikit AND ek IS sedikit THEN predikat IS good");
IS.NewRule("Rule 56", "IF markas IS utuh AND gold IS
sedikit AND point IS sedikit AND ek IS sedang THEN predikat IS good");
IS.NewRule("Rule 57", "IF markas IS utuh AND gold IS
sedikit AND point IS sedikit AND ek IS banyak THEN predikat IS good");
IS.NewRule("Rule 58", "IF markas IS utuh AND gold IS
sedikit AND point IS sedang AND ek IS sedikit THEN predikat IS exc");
IS.NewRule("Rule 59", "IF markas IS utuh AND gold IS
sedikit AND point IS sedang AND ek IS sedang THEN predikat IS good");
IS.NewRule("Rule 60", "IF markas IS utuh AND gold IS
sedikit AND point IS sedang AND ek IS banyak THEN predikat IS good");

IS.NewRule("Rule 61", "IF markas IS utuh AND gold IS
sedikit AND point IS banyak AND ek IS sedikit THEN predikat IS good");
IS.NewRule("Rule 62", "IF markas IS utuh AND gold IS
sedikit AND point IS banyak AND ek IS sedang THEN predikat IS good");
IS.NewRule("Rule 63", "IF markas IS utuh AND gold IS
sedikit AND point IS banyak AND ek IS banyak THEN predikat IS good");
IS.NewRule("Rule 64", "IF markas IS utuh AND gold IS
sedang AND point IS sedikit AND ek IS sedikit THEN predikat IS exc");
IS.NewRule("Rule 65", "IF markas IS utuh AND gold IS
sedang AND point IS sedikit AND ek IS sedang THEN predikat IS good");
IS.NewRule("Rule 66", "IF markas IS utuh AND gold IS
sedang AND point IS sedikit AND ek IS banyak THEN predikat IS good");
```

```
                IS.NewRule("Rule 67", "IF markas IS utuh AND gold IS
sedang AND point IS sedang AND ek IS sedikit THEN predikat IS exc");
                IS.NewRule("Rule 68", "IF markas IS utuh AND gold IS
sedang AND point IS sedang AND ek IS sedang THEN predikat IS exc");
                IS.NewRule("Rule 69", "IF markas IS utuh AND gold IS
sedang AND point IS sedang AND ek IS banyak THEN predikat IS good");
                IS.NewRule("Rule 70", "IF markas IS utuh AND gold IS
sedang AND point IS banyak AND ek IS sedikit THEN predikat IS good");

                IS.NewRule("Rule 71", "IF markas IS utuh AND gold IS
sedang AND point IS banyak AND ek IS sedang THEN predikat IS good");
                IS.NewRule("Rule 72", "IF markas IS utuh AND gold IS
sedang AND point IS banyak AND ek IS banyak THEN predikat IS good");
                IS.NewRule("Rule 73", "IF markas IS utuh AND gold IS
banyak AND point IS sedikit AND ek IS sedikit THEN predikat IS exc");
                IS.NewRule("Rule 74", "IF markas IS utuh AND gold IS
banyak AND point IS sedikit AND ek IS sedang THEN predikat IS exc");
                IS.NewRule("Rule 75", "IF markas IS utuh AND gold IS
banyak AND point IS sedikit AND ek IS banyak THEN predikat IS good");
                IS.NewRule("Rule 76", "IF markas IS utuh AND gold IS
banyak AND point IS sedang AND ek IS sedikit THEN predikat IS exc");
                IS.NewRule("Rule 77", "IF markas IS utuh AND gold IS
banyak AND point IS sedang AND ek IS sedang THEN predikat IS exc");
                IS.NewRule("Rule 78", "IF markas IS utuh AND gold IS
banyak AND point IS sedang AND ek IS banyak THEN predikat IS good");
                IS.NewRule("Rule 79", "IF markas IS utuh AND gold IS
banyak AND point IS banyak AND ek IS sedikit THEN predikat IS good");
                IS.NewRule("Rule 80", "IF markas IS utuh AND gold IS
banyak AND point IS banyak AND ek IS sedang THEN predikat IS good");
                IS.NewRule("Rule 81", "IF markas IS utuh AND gold IS
banyak AND point IS banyak AND ek IS banyak THEN predikat IS good");

        }

        void Start ()
                {
                fuzzyfuzzya();
```

```
kill = PlayerPrefs.GetFloat("enemymati");
poin = PlayerPrefs.GetFloat("poinplayer");
bangunan = PlayerPrefs.GetInt("markas");
time = PlayerPrefs.GetString("waktu");
gold = PlayerPrefs.GetFloat("jmlgold");
sts = PlayerPrefs.GetString("status");

goldlost = PlayerPrefs.GetFloat ("goldlost");
pointlost = PlayerPrefs.GetFloat("pointlost");
dead = PlayerPrefs.GetFloat("dead");
build = PlayerPrefs.GetFloat("build");

elapsedtime.text = "" + time;
playerbuild.text ="" + bangunan;
status.text = status.text + " " + sts;
deadtx.text = "" + dead;
buildtx.text = "" + build;
goldlosttx.text ="" + goldlost;
pointlosttx.text = "" + pointlost;
Debug.Log (kill+","+poin+","+time+""+gold);

IS.SetInput("gold",gold);
IS.SetInput("point",poin);
IS.SetInput("ek",kill);
IS.SetInput("markas",bangunan);

try
{
        NewAngle = IS.Evaluate("predikat");
        z = Math.Round(NewAngle);
        hasilstar.text = "" + z.ToString ();
}
catch ( Exception ){
}

if (z <= 3) {
        predikat.text = "POOR";
```

```
        }
        else if (z > 3 || z <= 5) {
                predikat.text = "FAIR";
        }
        else if (z > 5 || z <= 8) {
                predikat.text = "GOOD";
        }
        else {
                predikat.text = "EXCELENT";
        }

}

        // Update is called once per frame
        void Update ()
        {}
```

--Halaman ini sengaja dikosongkan--