

**PENERAPAN LOGIKA *FUZZY SUGENO* DALAM GAME  
ARJUNA MENCARI CINTA**

**SKRIPSI**



**Disusun Oleh:**

**PRASETIO DIMAS YULIANTO**

**12.18.021**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

**2016**

---

LEMBAR PERSETUJUAN

PENERAPAN LOGIKA FUZZY SUGENO DALAM GAME  
ARJUNA MENCARI CINTA

SKRIPSI

*Disusun dan Diajukan Untuk Melengkapi Dan Memenuhi Persyaratan Guna  
Mencapai Gelar Sarjana Komputer Strata Satu (S-1)*

Disusun Oleh :

Prasetio Dimas Yulianto

12.18.021

Diperiksa dan disetujui oleh

Dosen Pembimbing I

Dosen Pembimbing II

Yosep Agus Pranoto, ST, MT  
NIP.1031000432

Febriana Santi Wahyuni, S.Kom, M.Kom  
NIP. P. 1031600425

Mengetahui,  
Ketua Jurusan Teknik Informatika S-1

Joseph Dedy Irawan, ST, MT  
NIP. 197404162005011002

PROGRAM STUDI TEKNIK INFORMATIKA S-1  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2016

MILIK  
PERPUSTAKAAN  
STN MALANG

**LEMBAR KEASLIAN  
PERNYATAAN KEASLIAN SKRIPSI**

Saya yang bertanda tangan dibawah ini :

Nama : Prasetio Dimas Yulianto

NIM : 12.18.021

Program Studi : Teknik Informatika S-1

Fakultas : Fakultas Teknologi Industri

Menyatakan dengan sesungguhnya bahwa Skripsi saya yang berjudul :

**“Penerapan Logika Fuzzy Sugeno Dalam Game Arjuna Mencari Cinta”**

Adalah skripsi sendiri bukan duplikasi serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, 10 Januari 2016

Yang mem:

  
  
Prasetio Dimas Yulianto  
12.18.021

# PENERAPAN LOGIKA FUZZY PADA GAME MENCARI CINTA

Prasctio Dimas Yulianto

Program Studi Teknik Informatika S-1  
Fakultas Teknologi Industri  
Institut Teknologi Nasional Malang  
Jl. Raya Karanglo Km. 2 Tasikmadu-Malang  
Email : Sebastian.hitsugaya@gmail.com

Dosen Pembimbing : 1. Yosep Agus Pranoto, ST, MT  
2. Febriana Santi W S, S.Kom, M.Kom

## Abstrak

*Game sekarang ini tidak hanya menjadi bagian dari kehidupan anak-anak saja, namun juga telah merambah pada kehidupan orang-orang dewasa. Jalan ceritanya yang seru, tampilan grafisnya yang menarik, dan yang paling utama tentu karena permainannya yang menghibur membuat masyarakat dari berbagai lapisan usia senang bermain game. Jenis game yang banyak diminati saat ini adalah Role Playing Game (RPG).*

*Metode yang digunakan untuk membuat dan merancang game arjuna mencari cintya ini adalah dengan cara mengimplementasikan Rpg Maker VX Ace dan menggabungkan logika fuzzy sugeno untuk kecerdasan buatan yang diterapkan untuk musuh dalam karakter bos pada game, logika fuzzy sugeno di letakkan pada skill dalam karakter boss sehingga lebih realistic dan menghibur. Desain layout, desain karakter, desain map, semua menggunakan Rpg Maker Vx Ace dan untuk pengimplementasian Fuzzy sugeno menggunakan matlab.*

*Dari 10 hasil pengujian fungsional, terdapat 1 hasil yang tidak sesuai hasil pengujian, dapat disimpulkan bahwa game berfungsi dengan baik 90%. Dalam pengujian AI dilakukan perbandingan pengujian dalam matlab dan pengujian dalam game. Output yang dihasilkan pengujian pada matlab dan game terlihat sama dan 100% sesuai. Pengujian control pemain terdapat 6 pengujian, semua pengujian berjalan dengan baik tanpa ada error dan control berjalan dengan baik 100%.*

**Kata Kunci :** *Permainan, Logika Fuzzy Sugeno, Rpg, Matlab*

## KATA PENGANTAR

Puji syukur alhamdulillah penulis panjatkan kehadirat Allah SWT atas berkat, rahmat, taufik dan hidayah-Nya, penyusunan skripsi yang berjudul "Penerapan Logika Fuzzy Sugeno Dalam Game Rpg Arjuna Mencari Cinta" dapat diselesaikan dengan baik. Shalawat serta salam senantiasa tercurah kepada junjungan Nabi besar Muhammad SAW beserta keluarga, sahabat, kerabat, dan pengikut beliau hingga akhir zaman.

Penulis menyadari bahwa dalam proses penulisan skripsi ini banyak mengalami kendala, namun berkat bantuan, bimbingan, kerjasama dari berbagai pihak dan berkah dari Allah SWT sehingga kendala – kendala yang dihadapi tersebut dapat diatasi. Untuk itu penulis menyampaikan ucapan terima kasih dan penghargaan kepada Bapak dan Ibu yang senantiasa mendoakan, memberikan bantuan moril, materi, dan nasehat selama penulis menjalani pendidikan. Selanjutnya ucapan terima kasih penulis sampaikan pula kepada:

1. Bapak Dr. Ir. Lalu Mulyadi, MTA, selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. Anang Subardi, MT, selaku Dekan Fakultas Teknologi Industri, Institut Teknologi Nasional Malang.
3. Bapak Joseph Dedy Irawan, ST, MT, selaku Ketua Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.
4. Bapak Yosep Agus Pranoto, ST.MT. selaku Dosen Pembimbing I, yang selalu memberikan bimbingan dan masukan.
5. Ibu Vebriana Santi W S, S.Kom, M.Kom selaku Dosen Pembimbing II, yang selalu memberikan bimbingan dan masukan.
6. Semua dosen Program Studi Teknik Informatika yang telah membantu dalam penulisan dan masukan.
7. Saudara/i, keluarga, sahabat, yang senantiasa mendoakan & memberikan dukungan kepada penulis dalam proses pembuatan program dan laporan skripsi.

8. Semua teman-teman berbagai angkatan yang telah memberikan doa dan dukungannya dalam menyelesaikan skripsi ini.

Dengan segala kerendahan hati, penulis menyadari masih banyak terdapat kekurangan-kekurangan, sehingga penulis mengharapkan adanya saran dan kritik yang bersifat membangun demi kesempurnaan skripsi ini.

Malang, Januari 2016

Penulis

3.2.3 Item.....	14
3.2.4 Map.....	14
3.2.5 Perancangan Logika Fuzzy Pada Karakter Boss.....	16
<b>BAB IV IMPLEMENTASI DAN PENGUJIAN</b> .....	<b>20</b>
4.1 Implementasi hasil.....	20
4.1.1 Tampilan Menu Utama.....	20
4.1.2 Tampilan Menu Karakter .....	21
4.1.3 Tampilan Stage 1 .....	21
4.1.4 Tampilan Stage 2.....	22
4.1.5 Tampilan Stage 3.....	23
4.1.6 Tampilan Stage 4.....	24
4.1.7 Tampilan Battle .....	25
4.2 Pengujian Fungsional .....	25
4.2.1 Pengujian AI .....	27
4.2.2 Pengujian Kontrol Player .....	27
<b>BAB V PENUTUP</b> .....	<b>29</b>
5.1 Kesimpulan .....	29
5.2 Saran .....	29
<b>DAFTAR PUSTAKA</b> .....	<b>30</b>
<b>LAMPIRAN</b> .....	<b>31</b>

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Saat ini terdapat beberapa media untuk bermain game yang paling populer yaitu konsol game, komputer, dan smartphone. Komputer adalah salah satu target media yang paling banyak digunakan para pengembang di industri game untuk meluncurkan produknya. Hal ini dikarenakan komputer merupakan media yang paling cepat perkembangannya dan memungkinkan *developergame* untuk memaksimalkan kinerja game mereka, hal ini dikarenakan dari sisi *hardware* yang cepat sekali berkembang, dan makin banyaknya *game engine* yang didesain untuk komputer dan dapat digunakan secara gratis.

*Fuzzy Logic* merupakan salah satu metode untuk membangun kecerdasan buatan pada sebuah sistem. *Fuzzy Logic* memberikan toleransi pada nilai abu – abu, tidak seperti *Crisp Logic* yang hanya menerima masukan 1 dan 0. Sehingga *Fuzzy Logic* lebih fleksibel mulai dari menerima masukan hingga menentukan aturan – aturan yang merupakan representasi dari pakar(Septian, 2014).

Pada game biasanya *fuzzy logic* digunakan pada karakter *NPC (Non Playable Character)* yaitu karakter yang digerakan oleh kecerdasan buatan yang digunakan untuk mendukung game tersebut. Seperti karakter musuh *fuzzy logic* dapat digunakan untuk menentukan gerakan dan aksi dari musuh tanpa melibatkan pengguna game.

Di balik semua itu, tersirat keinginan dari putra-putri Indonesia untuk membuat *game* mereka sendiri, tetapi masih saja terbentur dengan masalah ilmu, biaya dan tingkat kesulitan pembuatan *game* yang memang cukup tinggi, padahal kreativitas, inovasi dan imajinasi mereka tidak kalah dengan pengembang *game* luar negeri.

Banyak tokoh-tokoh pewayangan jawa yang dapat diangkat ceritanya menjadi sebuah cerita game yang menarik, diantaranya adalah kisah arjuna dan Srikandi.



Arjuna adalah pangeran dari alengaka dan istrinya, tetapi suatu saat istrinya tersebut di culik oleh buto cakel yang ternyata juga menyukai dewi srikandi, kemudian oleh arjuna di selamatkan dan sangat menarik untuk diangkat dalam game dan membuat penulis sebagai salah satu dari penikmat game tersebut ikut mencoba membuat game sendiri dengan judul "Arjuna Mencari Cinta Menggunakan Program RPG Maker VX".

## 1.2 Rumusan Masalah

Berdasarkan pada latar belakang masalah diatas, maka didapatkan rumusan masalah yang akan dibahas yaitu :

1. Bagaimana merancang *game rpg* ?
2. Bagaimana implementasi metode *Fuzzy sugeno* pada karakter Bos dalam *game rpg* ?

## 1.3 Batasan Masalah

Agar pembahasan dalam skripsi ini tidak meluas, maka ditentukan beberapa batasan masalah diantaranya, yaitu :

1. *Game* dibuat menggunakan *Rpg Maker Vx Ace*
2. *Game* dibuat hanya untuk *single player*, *game* ini dimainkan pada PC, dan hanya menceritakan petualangan arjuna menyelamatkan dewi srikandi.
3. *Game* hanya terdiri dari 4 *stage* yang bergenre *role play game(RPG)*.

## 1.4 Tujuan

Adapun tujuan dari pembuatan game ini yaitu :

1. Membangun sebuah *game rpg* menggunakan perangkat lunak *Rpg Maker Vx Ace*.
  2. Mengimplementasikan Logika Fuzzy Sugeno pada Boss dalam game.
  3. Mengimplementasikan dan menceritakan tokoh pewayangan arjuna dan Buto cakil dalam sebuah *game rpg*.
-

## BAB II

### LANDASAN TEORI

#### 2.1 Game

*Game* adalah suatu sistem yang di buat oleh seseorang yang bisa di tujukan sebagai sarana hiburan atau sebagai sarana pembelajaran, karena sifat *game* tersebut yang bisa membuat seseorang yang memainkannya bisa senang dan nyaman. *Game* sendiri memiliki aturan-aturan tertentu dimana pemain akan terlibat di dalam suatu permasalahan sehingga dapat menghasilkan suatu hasil yang dapat diukur yaitu menang atau kalah. *Game* merupakan sebuah media hiburan untuk berbagai kalangan. Seiring perkembangan zaman, musuh musuh pada *game* sudah semakin cerdas. Hal ini dimaksudkan dengan tujuan untuk membuat *game* menjadi lebih menarik dan memberikan tantangan kepada pemain. *Game Adventure* dengan grafis 2D merupakan salah satu *genre game* yang menggunakan metode *Finite State Machine* sebagai musuh pemain. (Rostianingsih. 2013)

*Game Adventure* merupakan *genre game* yang lebih menonjolkan jalan cerita disbanding aksi dalam *game*. *Adventure Game* adalah salah satu *genre game* yang bisa dikatakan jenis paling tua. Beberapa jenis *game* seperti *First Person Shooter (FPS)*, *Role Playing Game (RPG)*, dan *Real Team Strategy (RTS)* merupakan hasil pengembangan dari *game adventure* ini. (Rostianingsih. 2013)

##### 1.1.1 Syarat Perancangan Game

Berikut adalah syarat utama dalam merancang sebuah *game*, yaitu: (Rostianingsih. 2013)

##### 1. *Game are not linear*

Pada umumnya di dalam sebuah *game* terdapat tingkat kesulitan yang berbeda. Tetapi dalam menyelesaikan setiap kasus di dalam *game*, pemain tidak harus menyelesaikannya secara *linier* ataupun berurutan. Melainkan pemain dapat menyelesaikannya dengan mendahulukan cara yang dianggap mudah.

Berdasarkan alur tersebut, maka *Game* dikatakan tidak *linear* (lurus). (Rostianingsih, 2013)

## 2. *Game have a goal*

Di dalam merancang sebuah *Game*, *Game* harus memiliki tujuan dan hasil akhir permainan. Seperti halnya pada permainan catur, pemain harus mengalahkan lawan dengan cara "*checkmate*". Atau seperti pada *Game RPG*, pemain harus bertemu dengan salah satu tokoh di *game* tersebut untuk memperoleh *item* yang berguna pada perjalanan selanjutnya. (Rostianingsih, 2013)

## 3. *Game Must be winnable*

Di dalam merancang sebuah *Game*, *Game* tersebut juga harus memiliki cara untuk dapat dimenangkan oleh pemain. Jadi, di dalam permainan tersebut antara pemain dengan musuh atau antara pemain dengan pemain memiliki kesempatan menang yang sama besar. (Rostianingsih, 2013)

## 4. *Start, Middle and Ending Of The Game*

Di dalam sebuah *game*, *game* wajib memiliki posisi awal (*starting point*) yang mana pada posisi ini pemain akan mengawali permainannya. Sedangkan untuk *Middle*, merupakan alur perjalanan atau proses yang terjadi selama permainan sehingga sampai pada tujuan permainan atau akhir dari permainan tersebut (*Ending*). (Rostianingsih, 2013)

## 2.2 Genre Game

*Genre* atau ragam permainan *video* digunakan untuk menggolongkan permainan *video* berdasarkan interaksi bidang permainannya, bukan hanya perbedaan visual maupun naratif. Permainan *video* diklasifikasikan independen berdasarkan pengaturan atau isi dari dunia permainan tersebut, tidak seperti film ataupun buku. (Dewi, 2012)

Berikut ini adalah daftar umum yang digunakan untuk menentukan *genre* dari sebuah permainan *video*. Daftar ini tidak berarti lengkap karena pada dasarnya permainan *video* berubah dengan cepat. Sebuah permainan biasanya juga memiliki lebih dari satu *genre* (*Hybrid-genre*). *Genre* sebenarnya dari sebuah permainan terbuka bagi para setiap individual untuk menentukan. (Dewi, 2012)

Contoh *genre-genre* dari game adalah :

a. Aksi

Genre ini merupakan macam game yang paling populer. Game jenis ini membutuhkan kemampuan reflex pemain. Salah satu subgenre action yang populer adalah First Person Shooter (FPS). Pada game FPS diperlukan kecepatan berfikir. Game ini dibuat seolah-olah pemain yang berada dalam suasana tersebut.

b. Aksi Petualangan

Genre ini memadukan game play aksi dan petualangan. Contohnya pemain diajak untuk menelusuri gua bawah tanah sambil mengalahkan musuh, dan mencari artefak kuno, atau menyeberangi sungai . (Dewi, 2012)

c. Simulasi, Konstruksi dan Manajemen

Pemain dalam game ini diberi keleluasaan untuk membangun dan suatu proyek tertentu dengan bahan baku yang terbatas. (Dewi, 2012)

d. Role Playing Games (RPG)

Dalam RPG pemain dapat memilih satu karakter untuk dimainkan. Seiring dengan naiknya level game, karakter tersebut dapat berubah, bertambah kemampuannya, bertambah senjatanya, atau bertambah hewan peliharaannya. (Dewi, 2012)

e. Strategi

Genre strategi menitikberatkan pada kemampuan pada kemampuan berpikir dan organisasi. Game strategi dibedakan menjadi dua, yaitu Turn Based Strategy dan Real Time Strategy. Jika real time strategi mengharuskan pemain membuat keputusan dan secara bersamaan pihak lawan juga beraksi hingga menimbulkan serangkaian kejadian dalam waktu yang sebenarnya, sedangkan turn based strategi pemain bergantian menjalankan taktiknya. Saat pemain mengambil langkah, pihak lawan menunggu. Demikian juga sebaliknya. (Dewi, 2012)

f. Balapan

Pemain dapat memilih kendaraan, lalu melaju di arena balap. Tujuannya yaitu mencapai garis finish tercepat. (Dewi, 2012)

g. Olahraga

---

Genre ini membawa olahraga ke dalam sebuah komputer atau konsol. Biasanya gameplay dibuat semirip mungkin dengan kondisi olahraga yang sebenarnya. (Dewi, 2012)

#### h. Puzzle

Genre puzzle menyajikan teka-teki, menyamakan warna bola, perhitungan matematika, menyusun balok, atau mengenal huruf dan gambar. i. Permainan Kata Word game sering dirancang untuk menguji kemampuan dengan bahasa atau untuk mengeksplorasi sifat-sifatnya. Word Game umumnya digunakan sebagai sumber hiburan, tetapi telah dibuktikan untuk melayani suatu tujuan pendidikan juga. Berdasarkan uraian di atas dapat disimpulkan game adalah suatu hasil dari proses multimedia berupa alat untuk bersenang-senang dan dapat digunakan sebagai media untuk pembelajaran. (Dewi, 2012)

### 2.3 Kecerdasan Buatan

Kecerdasan buatan (bahasa Inggris: *Artificial Intelligence* atau *AI*) didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu *entitas* buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. (Purba, Kristo Radion. 2013)

Logika *fuzzy* yang pertama kali diperkenalkan oleh Lotfi A. Zadeh, memiliki derajat keanggotaan dalam rentang 0(nol) hingga 1(satu), berbeda dengan logika digital yang hanya memiliki dua nilai yaitu 1(satu) atau 0(nol). Proses *fuzzy inference* dapat dibagi dalam lima bagian, yaitu :

Fuzzyfikasi Input : FIS mengambil masukan-masukan dan menentukan derajat keanggotaannya dalam semua *fuzzy set*.

Operasi logika fuzzy :Hasil akhir dari operasi ini adalah derajat kebenaran *antecedent* yang berupa bilangan tunggal.

Implikasi :Merupakan proses mendapatkan *consequent* atau keluaran sebuah IF-THEN *rule* berdasarkan derajat kebenaran *antecedent*. Proses ini menggunakan mengambil nilai MIN/terkecil dari dua bilangan : Hasil operasi *fuzzy logic OR* dan *fuzzy set* banyak.

## BAB III

### ANALISA DAN PERANCANGAN SISTEM

#### 3.1 Analisa Game

Kebutuhan di bagi menjadi dua, yaitu kebutuhan perangkat keras dan kebutuhan perangkat lunak. Diuraikan sebagai berikut :

##### 3.1.1 Analisa Kebutuhan Game

Dalam pembuatan game ini membutuhkan perancangan konsep yang berguna untuk membantu dalam pembuatan game. Tahapan ini diawali dengan mengenal dan menganalisa spesifikasi dari game yang akan dibuat. Hasil analisis spesifikasi game tersebut kemudian dilanjutkan dengan menentukan struktur alur dari game yang akan di buat seperti level dari game, sifat-sifat dari musuh, aplikasi yang di gunakan untuk membuat game dan spesifikasi dari game tersebut.

Dalam tahap ini konsep rancangan Game Arjuna Mencari Cinta adalah:

- a. Menggunakan model *AI Fuzzy Sugeno* untuk penerapan sifat musuhnya.
- b. Dalam game ini player mempunyai fitur untuk mengambil item ketika mengalahkan musuh atau menyelesaikan perintah dari npc yang bisa membantu menyelesaikan game tersebut.
- c. Player di game ini juga dapat bergerak ke kanan, kiri, atas dan bawah dengan menekan tombol arah.
- d. Palyer akan battle proses ketika pemain mendekati musuh.
- e. Fitur level dalam game tersebut diletakkan dalam karakter pemain dan musuh.

##### 3.1.2 Kebutuhan Perangkat Keras

Spesifikasi perangkat keras untuk membuat game Arjuna Mencari Cinta :

1. AMD A6-3400M APU with Radeon™ HD Graphics 1.49 GHz
2. Memori 2GB
3. Harddisk 320GB
4. VGA ATI Radeon HD 4800 Series 512MB

Kebutuhan minimum untuk menjalankan game yang dibuat ini adalah :

1. Intel Pentium IV atau keatas
2. Memori 512MB
3. Free space IID 2GB
4. VGA 256MB

### 3.1.3 Analisa Kebutuhan Perangkat Lunak

Perangkat keras tidak berarti tanpa perangkat lunak begitu juga sebaliknya. Jadi perangkat lunak dan perangkat keras saling mendukung satu sama lain. Perangkat keras hanya berfungsi jika diberikan instruksi-intruksi kepadanya. Instruksi-instruksi inilah disebut dengan perangkat lunak. Untuk menjalankan aplikasi game ini digunakan beberapa perangkat lunak, antara lain :

1. Sistem Operasi Windows
2. Machintosh,
3. Atau Linux

## 3.2 Perancangan Game

Perancangan game di buat agar game dapat berjalan optimal dan sesuai dengan keinginan. Game dirancang berbasis Role Play Game (RPG) dengan tema cerita pewayangan Jawa, yang menceritakan petualangan pangeran Arjuna untuk menyelamatkan Dewi Srikandi sang istri dari sekapan Buto Cakel.

### 3.2.1 Game Play

1. Pada stage 1 terdapat 10 monster dan 1 bos stage. Boss stage tidak bisa di lawan jika semua monster yang ada di stage tidak dikalahkan. Pada stage 1 juga terdapat intro tentang percakapan ketika akan menyelamatkan Srikandi.
  2. Setelah mengalahkan boss di level 1 akan di lanjutkan di stage 2, di stage 2 Arjuna akan bergabung dengan Bima untuk membantu Arjuna, di dalam stage 2 terdapat save point jika ingin menyimpan data pada stage 1. Tidak ada boss di stage 2 dikarenakan akan dilanjutkan ke stage 3.
  3. Pada stage 3 hampir sama dengan stage 1 dan 2, tetapi di stage 3 player akan di suruh mencari item oleh npc dan mendapatkan item bonus untuk mengalahkan boss stage tersebut.
-

4. Pada stage 4 pemain akan langsung melawan boss utama yaitu buto cakil dan menyelamatkan dewi srikandi.
5. Permainan dinyatakan selesai atau game over ketika mengalahkan boss. Jika pemain kalah dalam pertarungan, pemain tidak akan game over, tetapi dapat mengulang kembali. Pada setiap stage terdapat item untuk menambah kekuatan dan mengalahkan monster. Level pada permainan di letakkan pada pemain dan musuh, setiap stage berbeda level dan tingkat kekuatannya.







### 3.2.2 Karakter



Berikut adalah karakter atau objek yang ada dalam game beserta nama, gambar dan keterangan dari karakter atau objek tersebut. Agar mempermudah dalam pemahaman secara mendetail, Table karakter atau objek dapat dilihat pada table 3.1.

Tabel : 3.1 Karakter dan objek pada game

No	Karakter/Objek	Gambar	Keterangan
1	arjuna		Karakter utama yang dijalankan
2	Bima		karakter tambahan jika ingin party bersama
3	Buto Rambut geni		Boss pada stege 1





No	Karakter/Objek	Gambar	Keterangan
4	Setan		Monster yang ada pada stage 1
5	Dedemit		monster yang ada pada stage 2.
6	Kakek-kakek		Karkater npc stage 3
7	Boss Buto Cakel		Boss pada stage terakhir
8	Npc Kakek Pengelana		Npc pada stage 3 untuk memberi petunjuk agar mencari item
9	Macan Purwo		Bos pada stage 4

No	Karakter/Objek	Gambar	Keterangan
10	portal		Tempat untuk pindah lokasi
11	Save Point		Tempat untuk pemain menyimpan permainan yang sudah dimainkan

### 3.2.3 Item

Pada rancangan ini menjelaskan tentang bentuk tampilan *item* yang terdapat dalam *game* Arjuna Mencari Cinta. Seperti yang terlihat pada table 3.2.

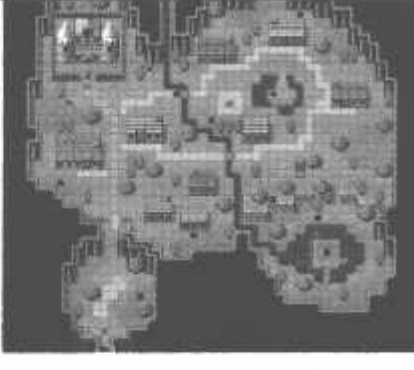
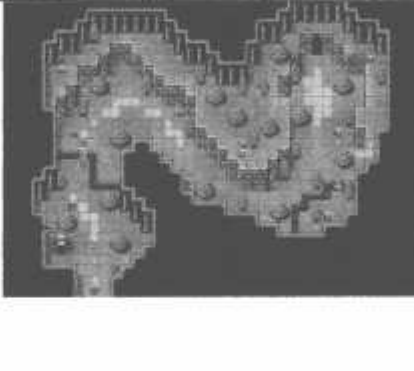
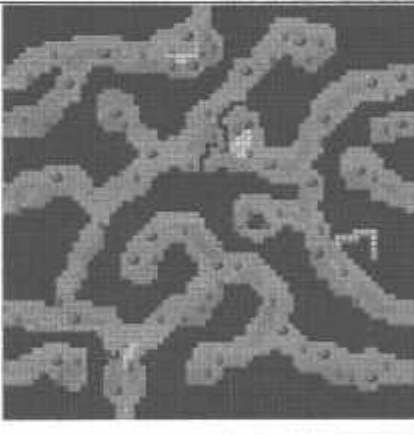
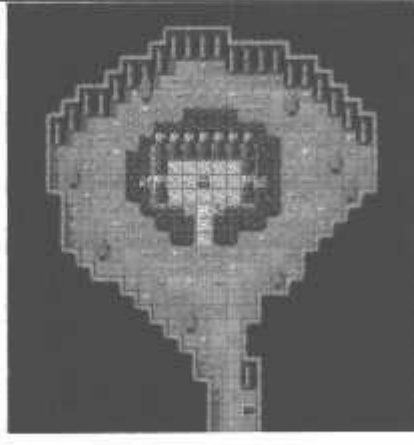
Tabel : 3.2 Item

Item	Deskripsi Item
	Drop item ketika mnyelesaikan perintah
	Item dari npc

### 3.2.4 Map

Pada rancangan ini menjelaskan tentang bentuk tampilan peta yang terdapat dalam *game* Arjuna Mencari Cinta seperti pada table 3.3.

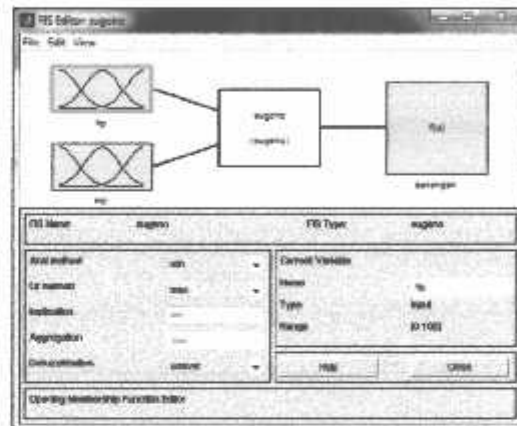
Tabel : 3.3 Stage

Map	Deskripsi map
	<p>stage 1 berlatarbelakang sebuah desa dan terdapat sungai dan pohon dan pertokoan.</p>
	<p><i>Stage level 2</i> berlatar belalang hutan, dan hutan tersebut bernama alas gondo mayet</p>
	<p><i>Stage level 3</i> menuju ke hutan yang dapat membingungkan</p>
	<p><i>Stage</i> terakhir berada pada hutan kerajaan buto caket seperti pulau kecil yang indah</p>

Tabel 3.3 menunjukkan tampilan map-map yang ada dalam game. Map yang ada dalam game adalah area yang dapat dimainkan pemain, di dalam map terdapat monster-monster, danau pohon batu dan lain-lain.

### 3.2.5 Perancangan Logika Fuzzy Pada Karakter Boss

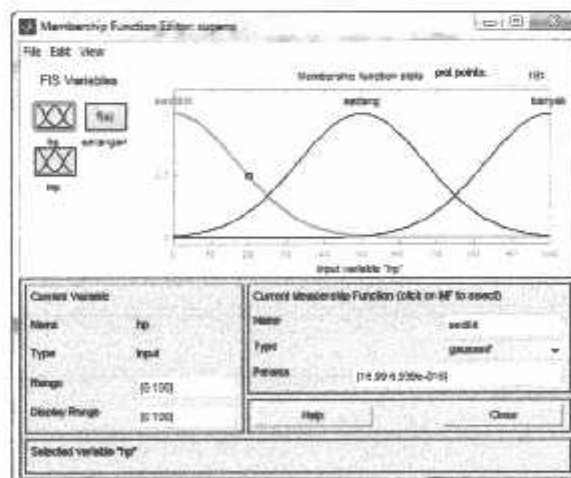
#### a) Input dan Output



Gambar 3.1 Input dan Output

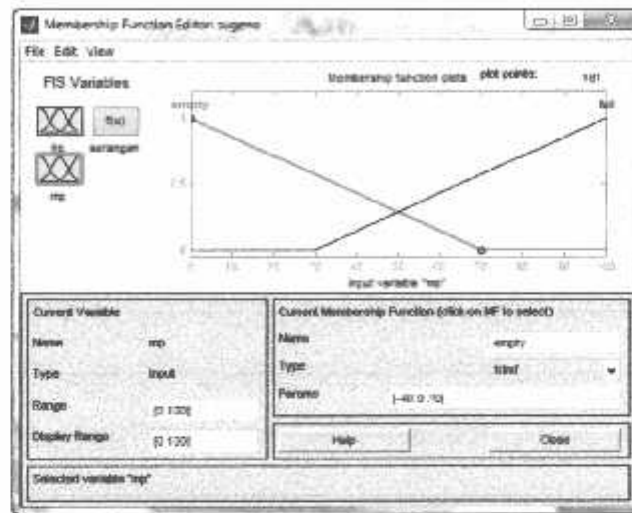
Gambar 3.1 menunjukkan input dan output hp dan mp dalam logika fuzzy sugeno. Terdapat 2 input dan 1 output dalam fuzzy pada boss. Untuk input yaitu hp(hit point/nyawa) dan mp (mana point/tenaga), untuk output yaitu serangan yang dikeluarkan oleh boss.

#### b) Fungsi Keanggotaan



Gambar 3.2 fungsi keanggotaan hp

Gambar 3.2 menunjukkan fungsi keanggotaan hp dari logika fuzzy sugeno. Di dalam fungsi keanggotaan hp terdapat range 1-100, 0-45 yaitu sedikit, 45-75 yaitu sedang, 75-100 yaitu banyak



Gambar 3.3 fungsi keanggotaan mp

Gambar 3.3 menunjukkan fungsi keanggotaan dari mp fuzzy sugeno. Di dalam fungsi keanggotaan mp terdapat range 0-100, untuk range 0-50 mp dinyatakan empty dan 50-100mp dinyatakan full.



Gambar 3.4 fungsi keanggotaan serangan

3. Jika hp banyak atau mp musuh penuh, maka serangan musuh akan kuat
  4. Jika mp sedang atau mp musuh penuh, maka musuh akan melakukan serangan sedang
  5. Jika hp sedang atau mp musuh habis maka musuh akan melakukan serangan biasa
-

## BAB IV IMPLEMENTASI DAN PENGUJIAN

### 4.1 Implementasi Hasil

Berikut tahapan hasil implementasi game “Arjuna Mencari Cinta” menggunakan engine Rpg Maker VX Ace. Tampilan yang terdapat pada game yaitu tampilan menu utama, stage 1, stage 2, stage 3, stage 4 dan tampilan battle. Tampilan-tampilan tersebut adalah sebagai berikut :

#### 4.1.1 Tampilan Menu Utama

Tampilan menu utama adalah tampilan yang muncul pada saat membuka game “Arjuna Mencari Cinta”. Pada saat tampilan terdapat pilihan new game, continue dan shutdown. Seperti pada gambar 4.1.



*Gambar 4.1 Tampilan menu utama*

### 4.1.2 Tampilan Menu Karakter

Tampilan menu karakter terdapat ketika pemain menekan tombol esc pada keyboard dan akan muncul keterangan karakter. Dan dalam menu karakter terdapat pilihan menu item, skill, equipment, status, dan end game. Menu-menu tersebut dapat dilihat agar pemain dapat melihat kekuatan si karakter. Tampilan menu karakter dapat dilihat pada gambar 4.2.



*Gambar 4.2 Tampilan menu karakter*

### 4.1.3 Tampilan stage 1

Pada stage 1 pemain akan mempunyai scenario untuk mengalahkan semua monster-monster yang ada dalam stage tersebut agar bisa mengalahkan bos pada stage tersebut. Dan pemain harus menemui bos terlebih dahulu sebelum melawan monster-monster. Setelah mengalahkan semua monster, pemain baru bisa melawan boss stage tersebut dan jika menang dapat dilanjutkan ke stage selanjutnya. Tampilan stage 1 dapat dilihat pada gambar 4.3.





*Gambar 4.3 Tampilan stage 1*

#### 4.1.4 Tampilan stage 2

Pada stage 2 hampir sama alur game pada stage 1, bedanya dalam stage 2 tidak terdapat bos dan hanya terdapat gerbang menuju alas gondo mayet. Dan pada stage 2 arjuna akan dipertemukan dengan saudaranya yang bernama bima, dan di stage tersebut arjuna dapat memilih untuk mengajak bergabung bima untuk berpetualang. Jika iya, bima akan bergabung dengan arjuna untuk berpetualang. Di dalam stage 2 tersebut disediakan save point agar pemain dapat menyimpan permainan yang sudah di mainkan. Tampilan stage 2 dapat dilihat pada gambar 4.4.



*Gambar 4.4 Tampilan stage 2*

#### **4.1.5 Tampilan Stage 3**

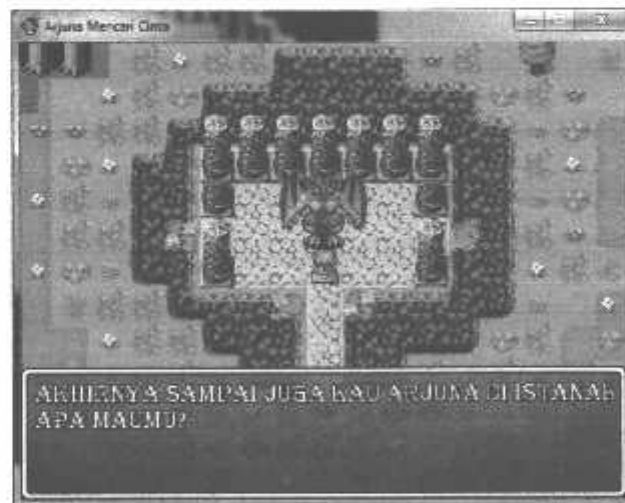
Dalam stage 3 ini merupakan kelanjutan dari stage 2 yang mempunyai scenario bahwa arjuna dan bima bertemu dengan kakek-kakek dan harus menolongnya agar bisa menyelesaikan game pada stage tersebut. Dan tugasnya adalah untuk mencari keris pusaka sebanyak 10 buah. Dan setelah itu pemain akan mendapatkan drop item untuk melawan bos pada stage tersebut. Jika belum menyelesaikan tugasnya pemain tidak dapat melawan bos stage tersebut. Bos pada stage tersebut dirancang untuk tidak dapat di serang atau bertarung sebelum misi yang diberikan si kakek belem terpenuhi. Dan di dalam stage 3 juga di isi monster-monster yang dapat di lawan agar pemain tidak bosan dalam stage tersebut dan untuk menambah level pemain. Tampilan level 3 dapat dilihat pada gambar 4.5.



*Gambar 4.5 Tampilan stage 3*

#### 4.1.6 Tampilan Stage 4

Di dalam stage 4 ini arjuna langsung berhadapan dengan bos utama buto cakel dan menyelamatkan dewi srikandi. Permainan dinyatakan selesai atau tamat ketika arjuna mengalahkan buto cakel. Tampilan stage 4 dapat dilihat pada gambar 4.6



*Gambar 4.6 Tampilan lvl 4*

#### 4.1.7 Tampilan Battle

Tampilan battle terlihat berhadapan antara pemain dan musuh. Pemain dinyatakan menang apabila sudah mengalahkan musuh dan tidak mati dalam pertarungan. Game ini dirancang agar pemain tidak mati, ketika pemain tidak berhasil mengalahkan musuh, pemain akan kembali ke map dan dapat bertarung lagi dengan musuh ketika sudah naik level atau pemain sudah dirasa cukup kuat untuk bertarung dengan musuh. Tampilan battle proses dapat dilihat pada gambar 4.7.



Gambar 4.7 Tampilan Battle

#### 4.2 Pengujian Fungsional

Pengujian fungsional adalah pengujian mengenai proses fungsional yang ada dalam game. Hasil dari pengujian dapat dilihat pada tabel 4.1.

#### 4.1 Tabel pengujian fungsional

No.	Fungsi	Output	Hasil
1	Healt bar player	Berkurang jika terserang monster	Sesuai
2	Healt bar monster	Bisa berkurang jika terserang oleh player	Sesuai
3	Portal	Pemain dapat berpindah tempat sesuai dengan tempat yang di tuju	Sesuai
4	Save point	Pemain dapat menyimpan hasil permainannya dan dapat meneruskan kembali permainannya	Sesuai
5	battle	Tampilan battle sudah bisa berhadapan antara pemain dan musuh.	Sesuai
6	attack	Musuh dapat berkurang hpnya jika di serang	Sesuai
7	Music dan suara	Music dan suara berbeda-beda dalam setiap stage	sesuai
8	video	Tidak dapat menampilkan video ketika ending atau sudah mengalahkan bos terakhir	Tidak sesuai
9	Dialog box	Pemain dapat memilih jawaban dan hasil jawaban sesuai yang dipilih	sesuai
10	Party	Pemain dapat mengajak npc agar ikut bermain	sesuai

Kesimpulan: dari sepuluh pengujian, terdapat 1 hasil yang tidak dapat berjalan dengan baik atau gagal, jadi dapat diartikan bahwa prosentase pengujian game 90%.

#### 4.2.1 Pengujian AI(Artificial Intelligence)

Pengujian AI(Artificial Intelligence) adalah pengujian mengenai fungsi yang berkaitan dengan AI(Artificial Intelligence) yang ada dalam game. Pengujian AI dilakukan di matlab, karena aplikasi ini dapat mendukung pengujian fuzzy sugeno. Dan kemudian dibandingkan dengan pengujian yang dilakukan dalam game. Dengan membandingkan 3 contoh data masukan pada game yang dibuat dan pada *Fuzzy Logic Toolbox* pada Matlab didapat hasil seperti pada Tabel 4.2.

4.2 Tabel pengujian AI

No	Input		Output		Hasil
	Health Point	Mana Point	Game	Matlab	
1	100%	100%	K. Kuat	83	Sesuai
2	30%	25%	S. Biasa	23	sesuai
3	20%	10%	S. Biasa	25	sesuai

Pengujian dilakukan ketika pemmain melawan boss buto cakel. Dari hasil pengujian AI terdapat 3 contoh serangan yang dihasilkan oleh bos masing-masing serangan berbeda nilai hp dan mpnya, dan berdasarkan hasil dari ke tiga pengujian AI yang dilakukan, dapat disimpulkan bahwa AI 100% berjalan sesuai dengan game dan pengujian matlab.

#### 4.2.2 Pengujian Control Player

Pengujian control player adalah pengujian fungsi dari setiap tombol yang diterapkan untuk menggerakkan karakter tabel 4.3.

Tabel 4.3 pengujian control player

No.	Tombol	Fungsi	Output
1	←	Bergerak kekiri	Pemain bergerak ke kiri(Sesuai)
2	→	Bergerak kekanan	Pemain bergerak ke kanan(Sesuai)
3	↑	Bergerak atas	Pemain bergerak ke atas(Sesuai)
4	↓	Bergerak ke atas	Pemain bergerak keatas (Sesuai)
5	esc	Menuju menu	Muncul tampilan menu (Sesuai)
6	enter	Mengeksekusi pilihan	pemain dapat mengeksekusi pilihan (Sesuai)

Hasil pengujian control player, 6 dari 6 pengujian yang dilakukan semuanya berjalan dengan baik atau 100% berjalan dengan normal.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Kesimpulan yang dapat diambil dari penerapan Logika Fuzzy sugeno pada game “Arjuna Mencari cinta” adalah sebagai berikut :

1. Dari 10 pengujian fungsional yang dilakukan, terdapat 1 kesalahan dalam game, sehingga fungsionalitas game berfungsi 90%.
2. Dalam pengujian AI, dilakukan 2 pengujian, yaitu pengujian yang dilakukan dalam matlab dan dilakukan dalam game, dari hasil pengujian didapatkan perbandingan yang dihasilkan dalam pengujian, dari 3 perbandingan pengujian dihasilkan bahwa perbandingan hasil dari matlab dan game sama atau 100% memenuhi rule yang telah ditetapkan
3. Dari hasil pengujian control gerak menunjukkan bahwa 6 dari 6 tombol pada control player berjalan sesuai dengan fungsi dan berjalan 100%.

#### **5.2. Saran**

Untuk meningkatkan kelayakan pada game Arjuna Mencari Cinta ini, maka ada beberapa saran diantaranya :

1. Dalam setiap stage sebaiknya di tambahkan musuh yang lebih fariatif dan random untuk menambah keseruan game.
2. Dalam game Arjuna Mencari Cinta akan lebih menarik jika stagenya dapat di tambahkan agar pemain mendapatkan tantangan dalam bermain.
3. Ketika battle, skill yang dimiliki pemain dan musuh ditambahkan agar tampilan ketika battle lebih menarik dan seru.



## Daftar Pustaka

- Arif, Y. Miftachul, F. Kurniawan, F. Nugroho. 2011. "*Desain Perubahan Perilaku pada NPC Game Menggunakan Logika Fuzzy*". Seminar on Electrical, Informatics and Education.
- Arif, Yunifa Miftachul and Ririen Kusumawati. *Attack Strategy For NPC In FPS Game Using Fuzzy Sugeno*. Basic Science International Conference, 2012.
- Dewi, Ghea Putri Fatma. 2012. *Pengembangan game edukas pengenalan nama hewan dalam bahasa inggris sebagai media pembelajaran siswa sd berbasis macromedia flash*. Jurnal skripsi.
- Fathurochman, Dery, Wina Witanti, Rezki Yuniarti. 2014. "perancangan game turn based strategy menggunakan logika fuzzy dan naive bayes classifier". *Seminar Nasional Informatika 2014 (semnasIF 2014)*.
- Purba, Kristo Radion, Rini Nur Hasanah dan M. Azis Muslim. 2013. Implementasi Logika *Fuzzy* Untuk Mengatur Perilaku Musuh dalam *Game* Bertipe *Action-RPG*. *Jurnal EECCIS Vol. 7, No. 1*
- Rostianingsih, Silvia. 2013. *Game Simulasi Finite State Machine Untuk Pertahnian Dan Peternakan*, jakarta : Universitas Kristen Retra.
- Septian, Ridwan Fadjar. 2014. *Implementasi Fuzzy Logic Metode Mamdani Untuk Pengembangan Intelegent Non-Player Character pada Game Strategi*. Bandung: UPI

# LAMPIRAN

---

## Lampiran 1. Source code battle side view

```
module REEDO_SBS
  ACTOR_START_TOP = 6.2      # Top position (in tiles) for first
actor in party
  ACTOR_START_LEFT = 14.5   # Left position (in tiles) for first
actor in party
  ACTOR_STAGGER = true      # Enables staggered layout. Set false
and chance start left to 12 for angled line.
  ACTOR_STAGGER_HORZ = 1.0  # The amount to recess each actor to the
right (in tiles)
  ACTOR_STAGGER_VERT = 0.5  # The vertical spacing between
characters (in tiles)
  ACTOR_STEPS_IN_OUT = 10   # The number of steps to take when
moving to attack
  ACTOR_STEP_DISTANCE = 0.1 # The distance to move with each step
(in tiles)

  ACTOR_DEATH_OPACITY = 125 # Transparency of dead actors
  ACTOR_DEATH_TONE = Tone.new(128, 128, 128, 0) # Overlay color of
dead actors
  ACTOR_DEATH_ANGLE = 270   # Rotation angle of dead actor sprite
  ACTOR_DEATH_OFFSET_X = -0.6 # Offset in tiles for dead actors
  ACTOR_DEATH_OFFSET_Y = -0.2 # Offset in tiles for dead actors

  VICTORY_DANCE_ENABLED = true # Actors spin and jump on
victory if alive and movable
  VICTORY_DANCE_JUMP_HEIGHT = 1.3 # The number of tiles an actor
moves upward
  VICTORY_DANCE_JUMP_HOVER = 4   # The number of frames to wait
in the air
  VICTORY_DANCE_JUMP_WAIT = 70   # The number of frames before
jumping again

  ACTOR_SHADOW_IMAGE = "Shadow" # The name of the system image
to use
  ACTOR_SHADOW_CUSTOM_PICTURE = nil # The name of a resource
picture to use instead; nil to use system image

  PREEMPTIVE_TIME = 120 # The number of frames to flash white on
preemptive strike
  SURPRISE_TIME = 90 # The number of frames to jump facing
backward on surprise
  DAMAGE_RED_TIME = 15 # Number of frames actor turns red when
damaged

  ATTACK_SKILL_ID = 1 # The id in the database for the skill
used as the default attack
  ENEMY_DEFAULT_ATTACK = 7 # AnimationId for enemy actions with
no animation
  ACTION_SPRITE_INCREMENT = 5 # The amount of movement for swinging
or pulsing action icons

  UNARMED_ICON = 175 # The icon to use if the actor has no weapon
```

```

equipped.
  UNARMED_STYLE - 2      # The style to use when there is no weapon
equipped.
  PUNCH_SPEED = 0.4
  PUNCH_DISTANCE = 32
  PULSE_WEAPON_TYPES = []      # Weapons with these weapon type
ids will pulse like items or skills
  NO_MOVE_WEAPON_TYPES = []    # Weapons with these weapon type
ids will show without moving
  PUNCH_WEAPON_TYPES = [2, 6, 10] # Weapons with these weapon type
ids will move toward the enemy and back
                                # All other weapon type ids will
swing

  # Specify the ID of enemies who should have their image flipped in
battle
  FLIP_ENEMY = [
    2, 3, 4, 5, 8, 10, 12, 13, 14, 16, 17,
    18, 19, 21, 22, 23, 25, 28, 29,

    1

  ]

  FORCE_AUTOTILE_BACKGROUND = true # attempt to use tile-based
background on all maps

end
#####
#####
## MAIN SCRIPT
#####
#####
## EDITS BEYOND THIS POINT ARE AT YOUR OWN RISK!!!
#####
#####
module BattleManager
  class << self
    alias reedo_sbs_bm_process_victory process_victory
    alias reedo_sbs_bm_process_escape process_escape
    alias reedo_sbs_bm_gain_exp gain_exp
  end

  def self.gain_exp
    if $imported && $imported["YEA-VictoryAftermath"]
      tmp_gc = []
      $game_party.all_members.each do |actor|
        tmp_gc.push(actor.reedo_sbs_gc)
        actor.reedo_sbs_gc = nil if actor.reedo_sbs_gc
      end
      reedo_sbs_bm_gain_exp
      $game_party.all_members.each_with_index do |actor, i|
        actor.reedo_sbs_gc = tmp_gc[i]
      end
    else
      reedo_sbs_bm_gain_exp
    end
  end
end

```

```

    end
  end

  def self.process_victory
    SceneManager.scene.reedo_victory_dance = true if
REEDO_SBS::VICTORY_DANCE_ENABLED
    reedo_sbs_bm_process_victory
  end

  def self.process_escape
    SceneManager.scene.reedo_turn_and_run
    SceneManager.scene.reedo_cancel_run if
!reedo_sbs_bm_process_escape
  end

  def self.reedo_preemptive?
    @preemptive
  end

  def self.reedo_surprise?
    @surprise
  end

  def self.reedo_get_actor_index
    return @actor_index
  end
end
end

class Game_Action
  def reedo_guard?
    return item == $data_skills[subject.guard_skill_id]
  end
end

class Game_Actor
  attr_accessor :reedo_sbs_gc
  attr_accessor :reedo_damage_time
  attr_accessor :reedo_is_acting

  alias reedo_sbs_ga_initialize initialize
  def initialize(actor_id)
    reedo_sbs_ga_initialize(actor_id)
    @reedo_damage_time = 0
    @reedo_is_acting = false
  end

  alias reedo_sbs_ga_perform_damage_effect perform_damage_effect
  def perform_damage_effect
    @reedo_damage_time = REEDO_SBS::DAMAGE_RED_TIME
    reedo_sbs_ga_perform_damage_effect
  end
end

class Game_Character

```

```

attr_accessor :opacity
attr_accessor :priority_type
attr_accessor :reedo_start_top
attr_accessor :reedo_jump_delta
attr_accessor :reedo_shadow_sprite
attr_accessor :reedo_is_dead
attr_accessor :reedo_action_sprite
attr_accessor :reedo_action_sprite_increment
attr_accessor :reedo_action_style

def reedo_step_on
  cmd = RPG::MoveCommand.new(Game_Character::ROUTE_STEP_ANIME_ON)
  process_move_command(cmd)
end

def reedo_step_off
  cmd = RPG::MoveCommand.new(Game_Character::RCUTE_STEP_ANIME_OFF)
  process_move_command(cmd)
end

def reedo_create_action_sprite(view)
  @reedo_action_style = 1
  @reedo_action_sprite_increment =
REEDO_SBS::ACTION_SPRITE_INCREMENT
  @reedo_action_sprite = Sprite_Base.new(view)
  @reedo_action_sprite.bitmap = Bitmap.new(24, 24)
  @reedo_action_sprite.src_rect = Rect.new(0, 0, 24, 24)
  @reedo_action_sprite.ox = 24
  @reedo_action_sprite.oy = 24
  @reedo_action_sprite.x = (self.x * 32) - 28
  @reedo_action_sprite.y = (self.y * 32) + 18
  @reedo_action_sprite.z = (self.priority_type * 100) + 1
  @reedo_action_sprite.visible = false
end

def reedo_create_shadow_sprite(view)
  @reedo_shadow_sprite = Sprite_Base.new(view)
  bmp = Cache.picture(REEDO_SBS::ACTOR_SHADOW_CUSTOM_PICTURE) if
REEDO_SBS::ACTOR_SHADOW_CUSTOM_PICTURE != nil
  bmp = Cache.system(REEDO_SBS::ACTOR_SHADOW_IMAGE) if bmp == nil
  @reedo_shadow_sprite.bitmap = bmp
  @reedo_shadow_sprite.src_rect = Rect.new(0, 0, bmp.width,
bmp.height)
  @reedo_shadow_sprite.x = (self.x * 32)
  @reedo_shadow_sprite.y = (self.y * 32)
  @reedo_shadow_sprite.z = (self.priority_type * 100) - 1
end

alias reedo_sbs_gc_update update
def update
  reedo_sbs_gc_update
  @shadow_sprite.update if @shadow_sprite != nil
end

```

```

def reedo_draw_icon(icon_index)
  @reedo_action_sprite.bitmap.clear
  @reedo_action_sprite.visible = (icon_index > 0)
  return if icon_index < 0
  bitmap = Cache.system("Iconset")
  rect = Rect.new(icon_index % 16 * 24, icon_index / 16 * 24, 24,
24)
  @reedo_action_sprite.bitmap.blit(0, 0, bitmap, rect, 255)
end
end

class Spriteset_Battle
  alias reedo_sbs_ssb_initialize initialize
  def initialize
    @reedo_dance_wait = 0
    reedo_sbs_ssb_initialize
  end

  alias reedo_sbs_ssb_dispose dispose
  def dispose
    for actor in $game_party.battle_members
      actor.reedo_sbs_gc.reedo_action_sprite.dispose
      actor.reedo_sbs_gc.reedo_shadow_sprite.dispose
      actor.reedo_sbs_gc = nil
    end
    @reedo_sprite_chars.each {|sprite| sprite.dispose }
    reedo_sbs_ssb_dispose
  end

  alias reedo_sbs_ssb_create_actors create_actors
  def create_actors
    dispose_actors if @actor_sprites != nil
    reedo_sbs_ssb_create_actors

    @reedo_game_chars = []
    @reedo_sprite_chars = []
    @actor_sprites = []
    @reedo_surprise_time = 0
    @reedo_preemptive_time = 0

    i = 1
    for actor in $game_party.battle_members
      gc = Game_Character.new
      gc.set_graphic(actor.character_name, actor.character_index)
      if BattleManager.reedo_surprise?
        gc.set_direction(6)
        @reedo_surprise_time = REEDO_SBS::SURPRISE_TIME
      elsif BattleManager.reedo_preemptive?
        @reedo_preemptive_time = REEDO_SBS::PREEMPTIVE_TIME
        @reedo_preemptive_tone = 0
        @reedo_preemptive_delta = 10
        gc.set_direction(2)
      else
        gc.set_direction(4)
      end
    end
  end
end

```

```

end
gc.priority_type = i
bc = Sprite_Character.new(@viewport1, gc)
sb = Sprite_Battler.new(@viewport1, actor)
top = REEDO_SBS::ACTOR_START_TOP + (i *
REEDO_SBS::ACTOR_STAGGER_VERT)
left = REEDO_SBS::ACTOR_START_LEFT
if REEDO_SBS::ACTOR_STAGGER
  left += REEDO_SBS::ACTOR_STAGGER_HORZ if i & 2 == 0
else
  left += REEDO_SBS::ACTOR_STAGGER_HORZ * i
end
gc.reedo_is_dead = false
gc.reedo_start_top = top
gc.reedo_jump_delta = -0.1
gc.moveto(left, top)
gc.reedo_create_shadow_sprite(@viewport1)
gc.reedo_create_action_sprite(@viewport1)
bc.update
@actor_sprites.push(sb)
@reedo_sprite_chars.push(bc)
@reedo_game_chars.push(gc)
actor.reedo_sbs_gc = gc
i += 1
end
end

alias reedo_sbs_ssb_create_enemies create_enemies
def create_enemies
  reedo_sbs_ssb_create_enemies
  for sprite in @enemy_sprites
    sprite.mirror = true if
REEDO_SBS::FLIP_ENEMY.include?(sprite.battler.enemy_id)
  end
end

alias reedo_sbs_ssb_create_blurry_background_bitmap
create_blurry_background_bitmap
def create_blurry_background_bitmap
  if REEDO_SBS::FORCE_AUTOTILE_BACKGROUND && !$BTEST
    tryName1 = battleback1_name
    return reedo_sbs_ssb_create_blurry_background_bitmap if
tryName1 == nil
    wall = Cache.battleback1(tryName1)
    tryName2 = battleback2_name
    bitmap = Bitmap.new(640, 480)
    bitmap.blit(0, 0, wall, wall.rect)
    if tryName2 != nil
      floor = Cache.battleback2(tryName2)
      bitmap.blit(0, 320, floor, floor.rect)
    end
    bitmap
  else
    reedo_sbs_ssb_create_blurry_background_bitmap

```



```

end
end

def reedo_update_victory_dance
  if @reedo_dance_wait == 0
    @actor_sprites.each_with_index do |sprite, i|
      if sprite.battler.alive? && sprite.battler.movable?
        gc = @reedo_game_chars[i]
        gc.moveto(gc.x, gc.y + gc.reedo_jump_delta)
        if gc.y < gc.reedo_start_top -
REEDO_SBS::VICTORY_DANCE_JUMP_HEIGHT
          gc.reedo_jump_delta = 0.1
          @reedo_dance_wait = REEDO_SBS::VICTORY_DANCE_JUMP_HOVER
        end
        if gc.y > gc.reedo_start_top
          gc.reedo_jump_delta = -0.1
          gc.set_direction(4)
          @reedo_dance_wait = REEDO_SBS::VICTORY_DANCE_JUMP_WAIT
        end
        if ((gc.y - REEDO_SBS::VICTORY_DANCE_JUMP_HEIGHT) *
10).to_int % (REEDO_SBS::VICTORY_DANCE_JUMP_HEIGHT / 4 * 10).to_int
== 0
          if @reedo_dance_wait == 0
            case gc.direction
              when 4; gc.set_direction(2)
              when 2; gc.set_direction(6)
              when 6; gc.set_direction(8)
              when 8; gc.set_direction(4)
            end
          end
        end
      end
    end
  else
    @reedo_dance_wait -= 1
  end
end

def reedo_update_surprise
  @actor_sprites.each_with_index do |sprite, i|
    if sprite.battler.alive? && sprite.battler.movable?
      gc = @reedo_game_chars[i]
      if @reedo_surprise_time > 1
        gc.moveto(gc.x, gc.y + gc.reedo_jump_delta)
        if gc.y < gc.reedo_start_top -
REEDO_SBS::VICTORY_DANCE_JUMP_HEIGHT
          gc.reedo_jump_delta = 0.1
        end
        if gc.y > gc.reedo_start_top
          gc.reedo_jump_delta = -0.1
        end
      else
        gc.moveto(gc.x, gc.reedo_start_top)
        gc.set_direction(4)
      end
    end
  end
end

```

```

        end
    end
end
@reedo_surprise_time -= 1
end

def reedo_update_preemptive
    @actor_sprites.each_with_index do |sprite, i|
        if sprite.battler.alive? && sprite.battler.movable?
            sc = @reedo_sprite_chars[i]
            if @reedo_preemptive_time > 1
                sc.tone.set(@reedo_preemptive_tone,@reedo_preemptive_tone,@reedo_preemptive_tone)
                @reedo_preemptive_tone += @reedo_preemptive_delta
                @reedo_preemptive_delta *= -1 if @reedo_preemptive_tone >=
255
                @reedo_preemptive_delta *= -1 if @reedo_preemptive_tone <=
0
            else
                sc.tone.set(0,0,0)
                @reedo_game_chars[i].set_direction(4)
            end
        end
    end
    @reedo_preemptive_time -- 1
end

def update_actors
    dance = false
    run = false
    if SceneManager.scene.is_a?(Scene_Battle)
        run = SceneManager.scene.reedo_run_away
        dance = SceneManager.scene.reedo_victory_dance
        reedo_update_victory_dance if dance
    end
    reedo_update_surprise if @reedo_surprise_time > 0
    reedo_update_preemptive if @reedo_preemptive_time > 0

    @actor_sprites.each_with_index do |sprite, i|
        gc = @reedo_game_chars[i]
        sc = @reedo_sprite_chars[i]
        if sprite.battler.movable? &&
(BattleManager.reedo_get_actor_index == i ||
sprite.battler.reedo_is_acting || dance || run)
            gc.reedo_step_on
        else
            gc.reedo_step_off
        end

        if gc.reedo_action_sprite.visible then
            if gc.reedo_action_style == 0 # swing
                gc.reedo_action_sprite.angle +=
gc.reedo_action_sprite.increment
            end
        end
    end
end

```

```

    gc.reedo_action_sprite_increment *= -1 if
(gc.reedo_action_sprite.angle > 90) || (gc.reedo_action_sprite.angle
< 0)
    elsif gc.reedo_action_style == 1 # pulse
        gc.reedo_action_sprite.zoom_x +=
(gc.reedo_action_sprite_increment * 0.01)
        gc.reedo_action_sprite.zoom_y +=
(gc.reedo_action_sprite_increment * 0.01)
        gc.reedo_action_sprite_increment *= -1 if
(gc.reedo_action_sprite.zoom_x > 2.0) ||
(gc.reedo_action_sprite.zoom_x < 1.0)
    elsif gc.reedo_action_style == 2 # punch
        gc.reedo_action_sprite.x +=
(gc.reedo_action_sprite_increment * REEDO_SBS::PUNCH_SPEED)
        gc.reedo_action_sprite_increment *= -1 if
(gc.reedo_action_sprite.x < gc.reedo_shadow_sprite.x -
REEDO_SBS::PUNCH_DISTANCE) || (gc.reedo_action_sprite.x >
gc.reedo_shadow_sprite.x)
    end
    gc.reedo_action_sprite.update
else
    gc.reedo_action_sprite.angle = 0
    gc.reedo_action_sprite.zoom_x = 1.0
    gc.reedo_action_sprite.zoom_y = 1.0
    gc.reedo_action_sprite.x = gc.reedo_shadow_sprite.x
end

if sprite.battler.dead?
    if gc.reedo_is_dead == false
        gc.moveto(gc.x + REEDO_SBS::ACTOR_DEATH_OFFSET_X, gc.y +
REEDO_SBS::ACTOR_DEATH_OFFSET_Y)
        gc.reedo_shadow_sprite.visible = false
        gc.opacity = REEDO_SBS::ACTOR_DEATH_OPACITY
        sc.angle = REEDO_SBS::ACTOR_DEATH_ANGLE
        sc.tone.set(REEDO_SBS::ACTOR_DEATH_TONE)
        gc.reedo_is_dead = true
    end
else
    if gc.reedo_is_dead == true
        gc.moveto(gc.x - REEDO_SBS::ACTOR_DEATH_OFFSET_X, gc.y -
REEDO_SBS::ACTOR_DEATH_OFFSET_Y)
        gc.reedo_shadow_sprite.visible = true
        gc.opacity = 255
        sc.angle = 0
        sc.tone.set(0,0,0)
        gc.reedo_is_dead = false
    end
    if sprite.battler.reedo_damage_time > 0
        sc.tone.set(255,0,0) if sc.tone.red == 0
        sprite.battler.reedo_damage_time -- 1
    else
        sc.tone.set(0,0,0) if sc.tone.red == 255
    end
end
end

```

```

    sc.update
    gc.update
    sprite.update
  end
end

def reedo_game_char(index)
  @reedo_game_chars[index]
end

def reedo_sprite_char(index)
  @reedo_sprite_chars[index]
end

def reedo_actor_sprites
  @actor_sprites
end
end

class Scene_Battle
  attr_accessor :reedo_victory_dance
  attr_accessor :reedo_run_away
  attr_reader :subject

  alias reedo_sbs_sb_start start
  def start
    @reedo_victory_dance = false
    @reedo_run_away = false
    if !$BTEST
      @reedo_map_x = $game_map.display_x
      @reedo_map_y = $game_map.display_y
      $game_map.set_display_pos(0, 0)
    else
      $game_map = Game_Map.new
      $game_map.setup(1)
    end
    reedo_sbs_sb_start
  end

  alias reedo_sbs_sb_terminate terminate
  def terminate
    if !$BTEST
      $game_map.set_display_pos(@reedo_map_x, @reedo_map_y)
    end
    reedo_sbs_sb_terminate
  end

  alias reedo_sbs_sb_execute_action execute_action
  def execute_action
    if @subject.is_a?(Game_Actor)
      @subject.reedo_is_acting = true
      @spriteset.update_actors
      wait(15)
      if !@subject.current_action.reedo_guard?

```

```

    reedo_move_inout(-REEDO_SBS::ACTOR_STEP_DISTANCE)
    gc = @subject.reedo_sbs_gc
    iconidx = -1
    if @subject.current_action.item.is_a?(RPG::Skill)
        if @subject.current_action.item.id ==
REEDO_SBS::ATTACK_SKILL_ID
            iconidx = @subject.weapons[0].icon_index if
@subject.weapons[0]
            iconidx = REEDO_SBS::UNARMED_ICON if iconidx < 0
            if iconidx == REEDO_SBS::UNARMED_ICON
                gc.reedo_action_style = REEDO_SBS::UNARMED_STYLE
            elsif
REEDO_SBS::NO_MOVE_WEAPON_TYPES.include?(@subject.weapons[0].wtype_id)
                gc.reedo_action_style = 3 #none
            elsif
REEDO_SBS::PUNCH_WEAPON_TYPES.include?(@subject.weapons[0].wtype_id)
                gc.reedo_action_style = 2 #punch
            elsif
REEDO_SBS::PULSE_WEAPON_TYPES.include?(@subject.weapons[0].wtype_id)
                gc.reedo_action_style = 1 #pulse
            else
                gc.reedo_action_style = 0 #swing
            end
        else
            iconidx = @subject.current_action.item.icon_index
            gc.reedo_action_style = 1 #pulse
        end
    elsif @subject.current_action.item.is_a?(RPG::Item)
        iconidx = @subject.current_action.item.icon_index
        gc.reedo_action_style = 1
    end
    gc.reedo_draw_icon(iconidx)
else
    reedo_move_inout(REEDO_SBS::ACTOR_STEP_DISTANCE / 2)
end
end
reedo_sbs_sb_execute_action
if @subject.is_a?(Game Actor)
    if !@subject.current_action.reedo_guard?
        gc = @subject.reedo_sbs_gc
        gc.reedo_draw_icon(-1)
        reedo_move_inout(REEDO_SBS::ACTOR_STEP_DISTANCE)
    else
        reedo_move_inout(-REEDO_SBS::ACTOR_STEP_DISTANCE / 2)
    end
    @subject.reedo_is_acting = false
end
end

alias reedo_sbs_sb_show_animation show_animation
def show_animation(targets, animation_id)
    oktargets = []
    targets.each do |sprite|

```

```

    if sprite.is_a?(Game_Actor)
      if animation_id == -1
        sprite.reedo_sbs_gc.animation_id =
REEDO_SBS::ENEMY_DEFAULT_ATTACK
      else
        sprite.reedo_sbs_gc.animation_id = animation_id
      end
    else
      oktargets.push(sprite)
    end
  end
  reedo_sbs_sb_show_animation(oktargets, animation_id)
end

def reedo_move_inout(dist)
  i = 0
  while i < REEDO_SBS::ACTOR_STEPS_IN_OUT
    gc = @subject.reedo_sbs_gc
    gc.moveto(gc.x + dist, gc.y)
    gc.reedo_shadow_sprite.x = (gc.x * 32)
    update_for_wait
    i += 1
  end
end

def reedo_turn_and_run
  @reedo_run_away = true
  @spriteset.update_actors
  wait(15)
  @spriteset.reedo_actor_sprites.each_with_index do |sprite, i|
    gc = @spriteset.reedo_game_char(i)
    gc.set_direction(6)
    gc.moveto(gc.x + 0.5, gc.y)
    gc.reedo_shadow_sprite.x = (gc.x * 32)
  end
end

def reedo_cancel_run
  @reedo_run_away = false
  @spriteset.reedo_actor_sprites.each_with_index do |sprite, i|
    gc = @spriteset.reedo_game_char(i)
    gc.set_direction(4)
    gc.moveto(gc.x - 0.5, gc.y)
    gc.reedo_shadow_sprite.x = (gc.x * 32)
  end
  return false
end
end
end

```

## Lampiran 2. Souch Code HP dan MP Bar

```
module DTP_HP
  #Whether to place the hp bar above or below the enemy
  ABOVE_MONSTER = true
  #Whether to use a custome image or not:
  #Image would be placed in Graphics/System and named Custom_HP.png
  CUSTOM_BAR = false
  #Whether to include the hp bar or not
  USE_HP_BAR = true
  #Whether to include an mp bar or not
  USE_MP_BAR = true

  #The width of the hp bar
  BAR_WIDTH = 66
  #The height of the hp bar
  BAR_HEIGHT = 5
  #The width of the border around the hp bar
  BORDER_WIDTH = 1
  #The height of the border around the hp bar
  BORDER_HEIGHT = 1
  #Offset the hp bar along the x-axis(left,right)
  BAR_OFFSET_X = 0
  #Offset the hp bar along the y-axis(up,down)
  BAR_OFFSET_Y = 0

  #Color for the back of the hp bar
  COLOR_BAR_BACK = Color.new(0,0,0,200)
  #First color for the hp bar gradient
  COLOR_BAR_1 = Color.new(255,0,0)
  #Second color for the hp bar gradient
  COLOR_BAR_2 = Color.new(200,100,100)
  #Outside border color
  COLOR_BORDER_1 = Color.new(0,0,0,185)
  #Inside border color
  COLOR_BORDER_2 = Color.new(255,255,255,185)
  #First color for the mp bar gradient
  MP_COLOR_BAR_1 = Color.new(0,175,255)
  #Second color fot he mp bar gradient
  MP_COLOR_BAR_2 = Color.new(0,0,255)

  #Whether to display text or not
  USE_TEXT = true
  #Text to be displayed, chp = current hp, mhp = max hp, php =
percentage hp
  #Examples: "php%" or "chp/mhp" or "chp - php%"
  TEXT_DISPLAY = "chp"
  #Offset for the text along the x-axis(left,right)
  TEXT_OFFSET_X = 5
  #Offset for the text along the y-axis(up,down)
  TEXT_OFFSET_Y = -24
  #Size of the displayed text
  TEXT_SIZE = Font.default_size
  #Font of the displayed text
```

```

TEXT_FONT = Font.default_name

#Show bars only when specific actor in party. Array format.
Example: [8,7]
#Set to [] to not use actor only
SPECIFIC_ACTOR = []
#Show enemy hp bar only if certain state is applied (like a scan
state)
#Set to 0 to not use state only
SCAN_STATE = 0
#Enemies will show hp bar as long as they have been affected but
scan state
#at least once before
SCAN_ONCE = false
#Hp bars will only show when you are targetting a monster
ONLY_ON_TARGET = false

#Text to display if it's a boss monster, accepts same arguments
BOSS_TEXT = "???"
#The width of the boss hp bar
BOSS_BAR_WIDTH = 66
#The height of the boss hp bar
BOSS_BAR_HEIGHT = 5
#The width of the border around the boss hp bar
BOSS_BORDER_WIDTH = 1
#The height of the border around the boss hp bar
BOSS_BORDER_HEIGHT = 1
#ID's of boss monsters in array format.
BOSS_MONSTERS = []

#Switch to disable bars
DISABLE_SWITCH = :

end
#Customization ends here

class Sprite_Battler
  alias hpbar_update update
  alias hpbar_dispose dispose
  def update
    hpbar_update
    return unless @battler.is_a?(Game_Enemy)
    if @battler
      update_hp_bar
    end
  end
end
def update_hp_bar
  boss = DTP_HP::BOSS_MONSTERS.include?(@battler.enemy_id)
  setup_bar if @hp_bar.nil?
  if @text_display.nil?
    @text_display = Sprite_Base.new(self.viewport)
    @text_display.bitmap = Bitmap.new(100,DTP_HP::TEXT_SIZE)
    @text_display.bitmap.font.size = DTP_HP::TEXT_SIZE
    @text_display.bitmap.font.name = DTP_HP::TEXT_FONT
  end
end

```



```

    @text_display.x = @hp_bar.x + DTP_HP::TEXT_OFFSET_X
    @text_display.y = @hp_bar.y + DTP_HP::TEXT_OFFSET_Y
    @text_display.z = 105
end
determine_visible
return unless @hp_bar.visible
if @hp_bar.opacity != self.opacity
    @hp_bar.opacity = self.opacity
    @mp_bar.opacity = @hp_bar.opacity if DTP_HP::USE_MP_BAR
end
@hp_bar.bitmap.clear
if !boss
    width = DTP_HP::BAR_WIDTH
    height = DTP_HP::BAR_HEIGHT
    bwidth = DTP_HP::BORDER_WIDTH
    bheight = DTP_HP::BORDER_HEIGHT
else
    width = DTP_HP::BOSS_BAR_WIDTH
    height = DTP_HP::BOSS_BAR_HEIGHT
    bwidth = DTP_HP::BOSS_BORDER_WIDTH
    bheight = DTP_HP::BOSS_BORDER_HEIGHT
end
btotal = (bwidth + bheight) * 2
rwidth = @hp_bar.bitmap.width
rheight = @hp_bar.bitmap.height
if !DTP_HP::CUSTOM_BAR && DTP_HP::USE_HP_BAR

@hp_bar.bitmap.fill_rect(0,0,rwidth,rheight,DTP_HP::COLOR_BAR_BACK)
    @hp_bar.bitmap.fill_rect(bwidth,bheight,rwidth-
bwidth*2,rheight-bheight*2,DTP_HP::COLOR_BORDER_2)

@hp_bar.bitmap.fill_rect(bwidth*2,bheight*2,width,height,DTP_HP::COL
OR_BORDER_1)
    end
    hp_width = @battler.hp_rate * width
    if DTP_HP::USE_HP_BAR

@hp_bar.bitmap.gradient_fill_rect(bwidth*2,bheight*2,hp_width,height
,DTP_HP::COLOR_BAR_1,DTP_HP::COLOR_BAR_2)
    end
    if DTP_HP::CUSTOM_BAR && DTP_HP::USE_HP_BAR
        border_bitmap = Bitmap.new("Graphics/System/Custom_HP.png")
        rect = Rect.new(0,0,border_bitmap.width,border_bitmap.height)
        @hp_bar.bitmap.blit(0,0,border_bitmap,rect)
    end
    if DTP_HP::USE_MP_BAR
        @mp_bar.bitmap.clear
        if !DTP_HP::CUSTOM_BAR

@mp_bar.bitmap.fill_rect(0,0,rwidth,rheight,DTP_HP::COLOR_BAR_BACK)
            @mp_bar.bitmap.fill_rect(bwidth,bheight,rwidth-
bwidth*2,rheight-bheight*2,DTP_HP::COLOR_BORDER_2)

@mp_bar.bitmap.fill_rect(bwidth*2,bheight*2,width,height,DTP_HP::COL

```

```

OR_BORDER_1)
  end
  mp_width = @battler.mp_rate * width

@mp_bar.bitmap.gradient_fill_rect(bwidth*2,bheight*2,mp_width,height
,DTP_HP::MP_COLOR_BAR_1,DTP_HP::MP_COLOR_BAR_2)
  if DTP_HP::CUSTOM_BAR
    border_bitmap = Bitmap.new("Graphics/System/Custom_HP.png")
    rect =
Rect.new(0,0,border_bitmap.width,border_bitmap.height)
    @mp_bar.bitmap.blit(0,0,border_bitmap,rect)
  end
  end
  return unless DTP_HP::USE_TEXT
  @text_display.opacity = @hp_bar.opacity if @text_display.opacity
!= @hp_bar.opacity
  @text_display.bitmap.clear
  text = DTP_HP::TEXT_DISPLAY.clone
  text = DTP_HP::BOSS_TEXT.clone if
DTP_HP::BOSS_MONSTERS.include?(@battler.enemy_id)
  text.gsub!(/chp/) {@battler.hp}
  text.gsub!(/mhp/) {@battler.mhp}
  text.gsub!(/php/) {(@battler.hp_rate * 100).to_i}

@text_display.bitmap.draw_text(0,0,100,@text_display.height,text)
  end
  def setup_bar
    boss = DTP_HP::BOSS_MONSTERS.include?(@battler.enemy_id)
    @hp_bar = Sprite_Base.new(self.viewport)
    if !boss
      width = DTP_HP::BAR_WIDTH + DTP_HP::BORDER_WIDTH * 4
      height = DTP_HP::BAR_HEIGHT + DTP_HP::BORDER_HEIGHT * 4
    else
      width = DTP_HP::BOSS_BAR_WIDTH + DTP_HP::BOSS_BORDER_WIDTH * 4
      height = DTP_HP::BOSS_BAR_HEIGHT + DTP_HP::BOSS_BORDER_HEIGHT
* 4
    end
    @hp_bar.bitmap = Bitmap.new(width,height)
    @hp_bar.x = self.x - @hp_bar.width / 2 + DTP_HP::BAR_OFFSET_X
    @hp_bar.y = self.y + DTP_HP::BAR_OFFSET_Y - self.bitmap.height -
@hp_bar.height
    @hp_bar.y = self.y + DTP_HP::BAR_OFFSET_Y unless
DTP_HP::ABOVE_MONSTER
    @hp_bar.x = 0 if @hp_bar.x < 0
    @hp_bar.y = 0 if @hp_bar.y < 0
    @hp_bar.z = 104
    if DTP_HP::USE_MP_BAR
      @mp_bar = Sprite_Base.new(self.viewport)
      @mp_bar.bitmap = Bitmap.new(@hp_bar.width,@hp_bar.height)
      @mp_bar.x = @hp_bar.x + 6
      @mp_bar.y = @hp_bar.y + @mp_bar.height - 3
      @mp_bar.z = 103
    end
  end
end

```

```

def determine_visible
  if !@battler.alive?
    @hp_bar.visible = false
    @mp_bar.visible = false if @mp_bar
    @text_display.visible = false
    if DTP_HP::SCAN_ONCE and DTP_HP::SCAN_STATE == 1
      $game_party.monster_scans[@battler.enemy_id] = true
    end
    return if !@battler.alive?
  end
  @hp_bar.visible = true
  if DTP_HP::SCAN_STATE != 0
    @hp_bar.visible = false
    @hp_bar.visible = true if @battler.state?(DTP_HP::SCAN_STATE)
    if DTP_HP::SCAN_ONCE
      @hp_bar.visible = true if
$game_party.monster_scans[@battler.enemy_id] == true
      $game_party.monster_scans[@battler.enemy_id] = true if
@hp_bar.visible
    end
  end
  if !DTP_HP::SPECIFIC_ACTOR.empty?
    @hp_bar.visible = false unless DTP_HP::SCAN_STATE != 0
    DTP_HP::SPECIFIC_ACTOR.each do |i|
      next unless
$game_party.battle_members.include?($game_actors[i])
      @hp_bar.visible = true
    end
  end
  if DTP_HP::ONLY_ON_TARGET
    return unless SceneManager.scene.is_a?(Scene_Battle)
    return unless SceneManager.scene.enemy_window
    @hp_bar.visible = SceneManager.scene.target_window_index ==
@battler.index
    @hp_bar.visible = false if
!SceneManager.scene.enemy_window.active
  end
  @hp_bar.visible = false if
$game_switches[DTP_HP::DISABLE_SWITCH]
  @text_display.visible = false if !@hp_bar.visible
  @text_display.visible = true if @hp_bar.visible
  @mp_bar.visible = @hp_bar.visible if DTP_HP::USE_MP_BAR
end
def dispose
  @hp_bar.dispose if @hp_bar
  @mp_bar.dispose if @mp_bar
  @text_display.dispose if @text_display
  hpbar_dispose
end
end

class Scene_Battle
  attr_reader :enemy_window
  def target_window_index

```

```
begin
  @enemy_window.enemy.index
rescue
  return -1
end
end
end

class Game_Party
  alias hp_bar_init initialize
  attr_accessor :monster_scans
  def initialize
    hp_bar_init
    @monster_scans = []
  end
end
end
```

---



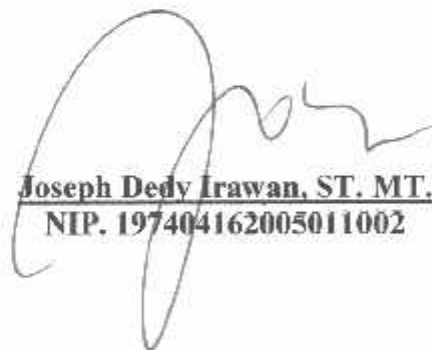
**BERITA ACARA UJIAN KOMPRESI**  
**FAKULTAS TEKNOLOGI INDUSTRI**

Nama : Prasetio Dimas Yulianto  
NIM : 12.18.021  
Program Studi : Teknik Informatika S-1  
Judul Skripsi : Penerapan Logika *Fuzzy Sugeno* Dalam Game Arjuna Mencari Cinta

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S1) pada :

Hari : Jumat  
Tanggal : 15 Januari 2016  
Nilai : 79.03(B+)

**Panitia Ujian Skripsi**  
**Ketua Majelis Penguji**



**Joseph Dedy Irawan, ST. MT.**  
**NIP. 197404162005011002**

**Dosen Penguji I**



**Sonny Prasetio, ST. MT**  
**NIP.P. 1013000433**

**Dosen Penguji II**



**Karina Auliasari, ST. M.Eng**  
**NIP.P. 1031000426**

---



**FORMULIR PERBAIKAN SKRIPSI**

Nama : Prasetio Dimas Yulianto  
Nim : 12.18.021  
Program Studi : Teknik Informatika S-1  
JudulSkripsi : Penerapan Logika *Fuzzy Sugeno* Dalam Game Arjuna Mencari Cinta

Tanggal	Penguji	Uraian	Paraf
15 Januari 2016	I	<ul style="list-style-type: none"><li>- Kesimpulan Diperbaiki</li><li>- Perbandingan Pengujian Logika Fuzzy</li><li>- Abstrak Diperbaiki</li><li>- Hasil Pengujian Dihitung Prosentasenya ( Setiap pengujian)</li></ul>	
15 Januari 2016	II	<ul style="list-style-type: none"><li>- Perbaiki latar belakang</li><li>- Perbaiki batasan masalah</li><li>- Landasan teori harus diberi sitasi sesuai daftar pustaka</li><li>- Perbaiki kesimpulan</li><li>- Perbaiki saran</li></ul>	

**AnggotaPenguji**

DosenPenguji I

**Sonny Prasetio, ST. MT**  
NIP.P. 1013000433

DosenPenguji II

**Karina Auliasari, ST. M.Eng**  
NIP.P. 1031000426

**Mengetahui,**

DosenPembimbing I

**Yosep Agus Pranoto, ST. MT**  
NIP.1031000432

DosenPembimbing II

**Febriana Santi Wahyuni, S.Kom.M.Kom**  
NIP. P. 1031000425



PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karangic, Kiri 2 Telp. (0341) 417030 Fax. (0341) 417634 Malang

Malang, 23 Oktober 2015

Nomor : ITN-593/LINF/TA/2015

Lampiran : ---

Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Yosep Agus Praoto, ST, MT  
Dosen Pembina Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,

Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : PRASETIO DIMAS YULIANTO  
Nim : 1218021  
Prodi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

**23 Oktober 2015 S/D 23 Maret 2016**

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.

Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui  
Program Studi Teknik Informatika S-1  
Ketua,

**Joseph Dedy Irawan, ST., MT.**  
NIP : 197404162005021002

Form S-4a



Malang, 23 Oktober 2015

Nomor : ITN-593/LIN.P/TA/2015  
Lampiran : ---  
Perihal : Bimbingan Skripsi

Kepada : Yth. Bpk/Ibu Febriana Santi W, S.Kom, M.Kom  
Dosen Pembina Program Studi Teknik Informatika S-1  
Institut Teknologi Nasional  
Malang

Dengan Hormat,  
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : PRASETIO DIMAS YULIANTO  
Nim : 1218021  
Prodi : Teknik Informatika S-1  
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

**23 Oktober 2015 S/D 2 Maret 2016**

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.  
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui  
Program Studi Teknik Informatika S-1  
Ketua,

**Joseph Dedy Irawan, ST., MT.**  
NIP : 197404162005021002

Form S-4a





### FORMULIR BIMBINGAN SKRIPSI

Nama : Prasctio Dimas Yulianto  
NIM : 12.18.021  
MasaBimbingan : 23 Oktober 2015 s/d 23 Maret 2016  
JudulSkripsi : Penerapan Logika Fuzzy Sugeno Dalam Game Arjuna  
Mencari Cinta

No	Tanggal	Uraian	Paraf pembimbing
1	17-11-2015	Pemberian input output dalam fuzzy dan fungsi keanggotaan	
2	19-11-2015	Memasukkan metode dalam game	
3	21-11-2015	Acc Seminar Progress	
4	06-01-2016	Acc Program	
5	07-01-2016	Revisi BAB IV	
6	08-01-2016	Revisi BAB V	
7	11-01-2016	Revisi daftar pustaka, daftar isi, daftar gambar, daftar table, lembar pernyataan	
8	12-01-2016	Acc Seminar Konferehensip	

Malang, 21 Januari 2016

Dosen Pembimbing I

Yosep Agus Pranoto, ST. MT  
NIP.1031000432

---



### FORMULIR BIMBINGAN SKRIPSI

Nama : Prasetio Dimas Yulianto  
NIM : 12.18.021  
Masa Bimbingan : 23 Oktober 2015 s/d 23 Maret 2016  
Judul Skripsi : Penerapan Logika Fuzzy Sugeno Dalam Game Arjuna Mencari Cinta

No	Tanggal	Uraian	Paraf pembimbing
1	03-11-2015	Revisi BAB I dan II	<i>km</i>
2	17-11-2015	Revisi BAB II Kurang Landasan Teori	<i>km</i>
3	18-11-2015	Acc Seminar Progress	<i>km</i>
4	08-12-2015	Revisi Laporan BAB IV	<i>km</i>
5	10-12-2015	Revisi Laporan BAB V	<i>km</i>
6	16-12-2015	Acc Seminar Hasil	<i>km</i>
7	01-01-2016	Revisi Tampilan Program	<i>km</i>
8	04-01-2016	Acc Makalah Seminar Konferehensip	<i>km</i>

Malang, 21 Januari 2016

Dosen Pembimbing II

**Febriana Santi Wahyuni, S.Kom.M.Kom**  
NIP. P. 1031500425