

**IMPLEMENTASI ALGORITMA FUZZY SUGENO  
PADA GAME COLOURATION**

**SKRIPSI**

Oleh:  
**PUTRI NUR AFRIDA**  
**NIM. 13650126**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2019**

**IMPLEMENTASI ALGORITMA FUZZY SUGENO  
PADA GAME COLOURATION**

**SKRIPSI**

**Diajukan kepada:  
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :  
PUTRI NUR AFRIDA  
NIM. 13650126**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2019**

**LEMBAR PERSETUJUAN**

**IMPLEMENTASI ALGORITMA FUZZY SUGENO  
PADA GAME COLOURATION**

**SKRIPSI**

Oleh :  
**PUTRI NUR AFRIDA**  
NIM. 13650126

Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal : 13 Desember 2019

Dosen Pembimbing I

  
Yunifa Miftachul Arif, M.T  
NIP. 19830616 201101 1 004

Dosen Pembimbing II

  
Fressy Nugroho, M.T  
NIP. 19710722 201101 1 001

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



  
D.P. Cahyo Crysdian  
NIP. 19740424 200901 1 008

**HALAMAN PENGESAHAN**

**IMPLEMENTASI ALGORITMA FUZZY SUGENO  
PADA GAME COLOURATION**

**SKRIPSI**

Oleh :  
**PUTRI NUR AFRIDA**  
NIM. 13650126

Telah Dipertahankan di Depan Dewan Penguji Skripsi  
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer ( S.Kom )  
Tanggal : 23 Desember 2019

**Susunan Dewan Penguji**

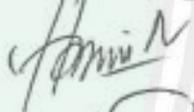
Penguji Utama : Hani Nurhayati, M.T  
NIP. 19780625 200801 2 006

Ketua Penguji : Fachrul Kurniawan, M.MT  
NIP. 19771020 200912 1 001

Sekretaris Penguji : Yunifa Miftachul Arif, M.T  
NIP. 19830616 201101 1 004

Anggota Penguji : Fressy Nugroho, M.T  
NIP. 19710722 201101 1 001

**Tanda Tangan**

()

()

()

()

Mengetahui dan Mengesahkan,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Cahyo Crysdiyan  
NIP. 19740424 200901 1 008

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Putri Nur Afrida

NIM : 13650126

Fakultas/Jurusan : Sains dan Teknologi/Teknik Informatika

Judul Skripsi : Implementasi Algoritma *Fuzzy Sugeno* Pada *Game Colouration*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila di kemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 13 Desember 2019

Yang membuat pernyataan,



Putri Nur Afrida  
NIM. 13650126

**MOTTO**

**“Let’s live while doing the things we like.”**

— *OSH*, 2012.



## HALAMAN PERSEMBAHAN

Alhamdulillah segala puji syukur kehadirat Allah SWT yang selalu memberikan kekuatan, kelapangan hati serta pikiran dan tekad sehingga saya dapat menyelesaikan skripsi ini. Sholawat serta salam kepada junjungan Nabi Muhammad SAW yang membawa kebenaran sebagai petunjuk hidup kepada seluruh umat manusia.

Ucapan terima kasih yang tak akan pernah cukup untuk membalas semua doa, kasih sayang, pengorbanan, dukungan dan segala hal-hal baik yang diberikan kedua orang tua saya, Bapak Roghib dan Ibu Evi. Semoga Allah senantiasa melindungi, dan melimpahkan kasih sayang-Nya.

Terima kasih kepada seluruh dosen-dosen Teknik Informatika UIN Malang yang selama ini mendidik saya sampai lulus. Terutama Pak Yunifa, terima kasih sudah sabar membimbing saya menyusun skripsi ini dari awal sampai selesai. Dan seluruh staff jurusan, Mbak Citra, terima kasih sudah melancarkan semua administrasi selama menjalankan ujian skripsi sampai wisuda. Semoga Allah memberikan kekuatan, kesehatan dan semoga ilmu yang telah diamalkan bermanfaat bagi seluruh mahasiswa.

Samsodi, fahmi, mbak hima, dan adek aqmar. Terima kasih banyak selalu memberi dukungan, meluangkan waktu, dan doa-doa yang selalu terpanjatkan untukku. Semoga Allah selalu melimpahkan rahmat-Nya.

Mas syafi, bebet, dody, yayan dan para sobat-sobat golongan kami yang selalu tak repoti sepanjang pengerjaan skripsi ini, terima kasih banyak. Semoga Allah membalas semua kebaikan kalian. Yok main yok!

Kiki, iphat, cikra dan alep selaku pengingat skripsiku, terima kasih banyak atas doa, nasihat, dukungan, hiburan dan canda tawanya selama ini. Semoga Allah selalu melimpahkan kebahagiaan dan kebersamaan ini sampai akhir hayat, yaa.

Terima kasih kepada teman-teman seperjuangan yang sudah berjuang bersama saling *men-support* dan saling membantu sampai terselesaikannya skripsi ini, dhofir, ayom, fikri, siti, alvi, farisa dan semua teman-teman fortinity. Semoga Allah memberikan ilmu yang manfaat dan barokah.

Dan kepada seluruh makhluk-makhluk Allah yang selalu mempertanyakan hal yang sama berulang kali. "Kapan lulus, put?" "Kapan sidang, put?". Terima kasih. Akhirnya pertanyaan kalian terjawab sudah, aku lulus!

Tak lupa kepada teman-teman Teknik Informatika angkatan 2013 yang masih berjuang menyelesaikan skripsi, semoga selalu diberi kelancaran agar lekas lulus dan wisuda.



## KATA PENGANTAR

*Assalamu'alaikum Wr. Wb.*

Segala puji syukur bagi Allah SWT. tuhan semesta alam, karena dengan segala rahmat dan karunia-Nya yang telah diberikan penulis sehingga dapat menyelesaikan skripsi dengan judul “Implementasi Algoritma *Fuzzy Sugeno* Pada *Game Colouration*” dengan baik. Selawat serta salam selalu tercurahkan kepada junjungan kita baginda Nabi Muhammad SAW. Yang membimbing umat manusia dari zaman kebodohan menuju Islam yang *rahmatan lil alamiin*.

Dalam penyelesaian skripsi ini, terdapat banyak pihak yang telah berjasa memberikan bantuan baik dalam bentuk nasihat, moril maupun materil. Atas segala jasa yang telah diberikan, penulis ingin menyampaikan doa dan ucapan terima kasih kepada:

1. Bapak Dr. Cahyo Crysdiyan, selaku ketua jurusan Teknik Informatika yang selalu memberikan semangat dan motivasi untuk terus berjuang.
2. Bapak Yunifa Miftachul Arif, M.T, selaku dosen pembimbing I yang telah meluangkan waktu untuk membimbing, mengarahkan, dan memberi masukan kepada penulis pengerjaan skripsi ini hingga akhir.
3. Bapak Fresy Nugroho, M.T, selaku dosen pembimbing II yang senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
4. Segenap jajaran dosen Teknik Informatika yang telah memberikan bimbingan ilmunan kepada penulis selama masa studi.
5. Teman-teman seperjuangan Teknik Informatika angkatan 2013 (Fortinity).
6. Semua pihak yang tidak mungkin penulis sebutkan satu persatu, atas segala yang telah diberikan kepada penulis dan dapat menjadi pelajaran.

Berbagai kekurangan dan kesalahan penulisan mungkin akan dijumpai dalam skripsi ini. Untuk itu penulis menerima segala kritik dan saran yang berguna untuk membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya dan semoga karya ini senantiasa dapat memberikan manfaat, Amiin.

*Wassalam 'alaikum Wr. Wb.*

Malang, Desember 2019

Penulis



## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>LEMBAR PERSETUJUAN</b> .....	<b>ii</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>iii</b>
<b>PERNYATAAN KEASLIAN TULISAN</b> .....	<b>iv</b>
<b>MOTTO</b> .....	<b>v</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>vi</b>
<b>KATA PENGANTAR</b> .....	<b>viii</b>
<b>DAFTAR ISI</b> .....	<b>x</b>
<b>DAFTAR GAMBAR</b> .....	<b>xii</b>
<b>DAFTAR TABEL</b> .....	<b>xiv</b>
<b>ABSTRAK</b> .....	<b>xv</b>
<b>ABSTRACT</b> .....	<b>xvi</b>
<b>المخلص</b> .....	<b>xvii</b>
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan Masalah .....	4
1.5 Manfaat Penelitian .....	4
1.6 Metode Penelitian .....	5
1.7 Sistematika Penulisan .....	6
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 <i>Game</i> .....	7
2.1.1 Pengertian <i>Game</i> .....	7
2.1.2 Jenis <i>Game</i> .....	7
2.2 Warna .....	11
2.2.1 Pembagian Warna .....	13
2.3 Logika <i>Fuzzy</i> .....	17
2.3.1 Himpunan <i>Fuzzy</i> .....	18
2.3.2 Fungsi Keanggotaan <i>Fuzzy</i> .....	20
2.3.3 Operator Dasar Pada Himpunan <i>Fuzzy</i> .....	23
2.3.4 Sistem Inferensi <i>Fuzzy</i> .....	24
2.4 <i>Game Two Dots</i> .....	28
2.5 <i>Game Candy Crush Saga</i> .....	29
2.6 Penelitian Terkait .....	31
<b>BAB III ANALISIS DAN PERANCANGAN</b>	
3.1 Perancangan Permainan .....	32
3.1.1 Deskripsi Permainan .....	32
3.1.2 Perancangan <i>Item</i> .....	32

3.1.3 Perancangan <i>Obstacle</i> .....	33
3.1.4 Perancangan <i>Level</i> .....	33
3.1.5 Perancangan Penilaian .....	35
3.1.6 Skenario Permainan .....	35
3.2 Perancangan Antarmuka Permainan .....	36
3.2.1 Perancangan Tampilan Menu Utama .....	37
3.2.2 Perancangan Tampilan <i>Level</i> Permainan .....	37
3.2.3 Perancangan Tampilan Papan Permainan .....	38
3.2.4 Perancangan Tampilan Loading Screen .....	38
3.3 <i>Finite State Machine</i> (FSM) .....	39
3.4 Perancangan <i>Fuzzy Logic</i> .....	39
3.4.1 Variabel <i>Fuzzy</i> .....	40
3.4.2 <i>Fuzzyfikasi</i> .....	40
3.4.3 <i>Fuzzy Rules</i> .....	45
3.4.4 Implikasi .....	47
3.4.5 <i>Defuzzyfikasi</i> .....	48
3.6 Contoh Perhitungan <i>Fuzzy Sugeno</i> .....	48
3.6.5 Perhitungan Manual .....	48
3.6.6 Perhitungan Matlab .....	55
<b>BAB IV HASIL DAN PEMBAHASAN</b>	
4.1 Implementasi Desain Sistem .....	60
4.1.1 Kebutuhan Perangkat keras .....	60
4.1.2 Kebutuhan Perangkat Lunak .....	60
4.2 Implementasi Permainan .....	61
4.2.1 Tampilan Main Menu .....	61
4.2.2 Tampilan <i>Level</i> Permainan .....	62
4.2.3 Tampilan Permainan .....	63
4.2.4 Tampilan <i>Game Success</i> .....	66
4.2.5 Tampilan <i>Game Failed</i> .....	66
4.3 Implementasi Algoritma <i>Fuzzy Sugeno</i> .....	67
4.3.1 Implementasi Algoritma <i>Fuzzy Sugeno</i> pada Kenaikan Tingkat Permainan .....	73
4.4 Uji Coba .....	74
4.4.1 Uji Coba Algoritma <i>Fuzzy Sugeno</i> pada Kenaikan Tingkat Permainan .....	74
4.5 Integrasi <i>Game Colouration</i> dengan Islam .....	79
<b>BAB V PENUTUP</b>	
5.1 Kesimpulan .....	82
5.2 Saran .....	82
<b>DAFTAR PUSTAKA</b> .....	<b>83</b>

## DAFTAR GAMBAR

Gambar 2.1	Warna Primer .....	14
Gambar 2.2	Warna Sekunder .....	15
Gambar 2.3	Warna Tersier .....	15
Gambar 2.4	Representasi Linear Naik .....	21
Gambar 2.5	Representasi Linear Turun .....	21
Gambar 2.6	Representasi Kurva Segitiga .....	22
Gambar 2.7	Representasi Kurva Trapesium .....	23
Gambar 2.8	Proses <i>Defuzzy</i> .....	26
Gambar 2.8	Tampilan <i>Game Two Dots</i> .....	29
Gambar 2.9	Tampilan <i>Game Candy Crush Saga</i> .....	30
Gambar 3.1	Rancangan <i>Item</i> .....	31
Gambar 3.2	Rancangan <i>Obstacle, Concrete, Breakable, Dan Blank</i> .....	32
Gambar 3.3	Rancangan Tampilan Menu Utama .....	37
Gambar 3.4	Rancangan Tampilan <i>Level</i> .....	37
Gambar 3.5	Rancangan Tampilan Papan Permainan .....	38
Gambar 3.6	Rancangan Tampilan <i>Loading Scene</i> .....	39
Gambar 3.7	<i>Finite State Machine Game Colouration</i> .....	39
Gambar 3.8	Proses <i>Fuzzy</i> .....	41
Gambar 3.9	Himpunan <i>Fuzzy</i> untuk Variabel <i>Moves</i> .....	41
Gambar 3.10	Himpunan <i>Fuzzy</i> untuk Variabel <i>Goals</i> .....	43
Gambar 3.11	Himpunan <i>Fuzzy</i> untuk Variabel <i>Score</i> .....	44
Gambar 3.12	Matlab: Himpunan <i>Fuzzy</i> untuk Variabel <i>Moves</i> .....	56
Gambar 3.13	Matlab: Himpunan <i>Fuzzy</i> untuk Variabel <i>Goals</i> .....	56
Gambar 3.14	Matlab: Himpunan <i>Fuzzy</i> untuk Variabel <i>Score</i> .....	57
Gambar 3.15	Matlab: Himpunan <i>Fuzzy</i> untuk Variabel <i>Level</i> .....	57
Gambar 3.16	Matlab: <i>Fuzzy Rules</i> .....	58
Gambar 3.17	Matlab: Hasil .....	58
Gambar 3.18	Matlab: Grafik Permukaan <i>Level</i> .....	59
Gambar 4.1	Tampilan Main Menu .....	61
Gambar 4.2	Tampilan <i>Level</i> Permainan .....	62
Gambar 4.3	Tampilan Tampilan <i>Scene</i> Nilai .....	63
Gambar 4.4	Tampilan <i>Scene Mission</i> .....	64
Gambar 4.5	Tampilan Tampilan Permainan .....	65
Gambar 4.6	Tampilan Tombol Menu Musik Aktif .....	65
Gambar 4.7	Tampilan Tombol Menu Musik Nonaktif .....	65
Gambar 4.8	Tampilan Tampilan <i>Game Success</i> .....	66
Gambar 4.9	Tampilan Tampilan <i>Game Failed</i> .....	67
Gambar 4.10	<i>Source Code</i> Derajat Keanggotaan Variabel <i>Moves</i> .....	67
Gambar 4.11	<i>Source Code</i> Derajat Keanggotaan Variabel <i>Goals</i> .....	69
Gambar 4.12	<i>Source Code</i> Derajat Keanggotaan Variabel <i>Score</i> .....	70
Gambar 4.13	<i>Source Code</i> Implementasi <i>Fuzzy Rules</i> .....	71
Gambar 4.14	<i>Source Code</i> Penerapan Fungsi Implikasi MIN dan Proses <i>Defuzzifikasi</i> .....	72
Gambar 4.15	<i>Source Code</i> Kenaikan <i>Level</i> .....	73
Gambar 4.16	Unity: <i>Console</i> Nilai Variabel <i>Input</i> dan <i>Output</i> Pengujian Pertama .....	74

Gambar 4.17 Unity: *Console* Nilai Variabel *Input* dan *Output* Pengujian Kedua .. 75  
Gambar 4.18 Unity: *Console* Nilai Variabel *Input* dan *Output* Pengujian Ketiga .. 75  
Gambar 4.19 Unity: *Console* Nilai Variabel *Input* dan *Output* Pengujian Keempat .. 76  
..... 76



## DAFTAR TABEL

Tabel 2.1	Pembagian Warna .....	16
Tabel 3.1	Pembagian Tingkat Kesulitan pada <i>Level</i> .....	34
Tabel 3.2	Pembagian <i>Obstacle</i> Pada Setiap <i>Level</i> Permainan .....	35
Tabel 3.3	Nilai linguistik Variabel <i>Fuzzy: Input</i> dan <i>Output</i> .....	40
Tabel 3.4	Domain Himpunan Keanggotaan <i>Moves</i> .....	41
Tabel 3.5	Domain Himpunan Keanggotaan <i>Goal</i> .....	42
Tabel 3.6	Domain Himpunan Keanggotaan <i>Score</i> .....	44
Tabel 4.1	Kebutuhan Perangkat Keras .....	60
Tabel 4.2	Kebutuhan Perangkat Lunak .....	61
Tabel 4.3	Keterangan Menu pada <i>Level</i> Permainan .....	62
Tabel 4.4	Hasil Pengujian Implementasi Algoritma <i>Fuzzy Sugeno</i> .....	76



## ABSTRAK

Afrida, Putri Nur. 2019. **Implementasi Algoritma Fuzzy Sugeno Pada Game Colouration**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (I) Yunifa Miftachul Arif, M.T, (II) Fressy Nugroho, M.T.

---

**Kata Kunci:** Permainan, Edukatif, Tingkat Permainan, Warna, *Fuzzy Sugeno*.

Setiap orang pasti memiliki beban dalam hidupnya masing-masing, baik dalam hal pekerjaan, studi maupun dalam kehidupan sehari-harinya. Tidak jarang beban dan perasaan tertekan yang ditanggungnya justru berpotensi membuat seseorang merasa stres. Stres merupakan suatu respons yang membuat kita lebih baik. Ketika dihadapkan dengan suatu masalah, stres akan membuat kita lebih berpikir dan berusaha keluar dari masalah tersebut. Itulah yang dinamakan mekanisme adaptasi diri terhadap stres. Namun pada umumnya tiap individu memberikan respon yang berbeda-beda ketika dilanda stres, perbedaan ini akan menghasilkan cara adaptasi yang berbeda tiap individu terhadap stres. Biasanya yang dilakukan orang-orang untuk mengurangi resiko stres adalah mencari hiburan, apapun bentuknya. Dan salah satu bentuk hiburan yang biasa dilakukan adalah dengan bermain *game*. Cara ini memang cukup efektif untuk menekan stres saat dilanda kepenatan disela-sela kerja maupun kejenuhan serta rasa bosan akan rutinitas yang dilakukan secara terus menerus.

Permainan kasual mencocokkan item-item berwarna senada bisa menjadi salah satu alternatif untuk memecah kejenuhan dan kebosanan. Permainan *Colouration* dirancang untuk menjadi sebuah permainan edukatif menarik yang bertema tentang pembagian warna menurut teori *Brewster*. Pada permainan *Colouration* ini, implementasi kecerdasan buatan diterapkan pada penentuan tingkat permainan dengan memanfaatkan algoritma *Fuzzy Sugeno*. Penggunaan *Fuzzy Sugeno* ini dapat menentukan tingkat permainan yang sesuai dengan nilai variabel *input* pemain dalam menyelesaikan setiap misi permainan. Dari hasil percobaan yang dilakukan menunjukkan akurasi metode *Fuzzy Sugeno* dalam menentukan tingkat permainan adalah 100% dengan prosentase *level* mudah 38%, *level* normal 34%, dan *level* sulit 28%. Hasil tersebut membuktikan bahwa metode ini baik dan tepat digunakan untuk penentuan tingkat permainan.

## ABSTRACT

Afrida, Putri Nur. 2019. **Implementation of Fuzzy Sugeno Algorithm on Game Colouration**. Undergraduate Thesis. Informatics Engineering Department, faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Advisers: (I) Yunifa Miftachul Arif, M.T, (II) Fresy Nugroho, M.T.

---

**Keywords:** Game, Education, Levelling, Colour, Fuzzy Sugeno.

Everyone must have a burden in their lives, both in terms of work, study or their daily lives. Commonly, the burden and pressure they got, has the potential to make someone feel stressed. Stress is a response that makes us better. When we faced a problem, stress will make us think and try to get out of the problem. That is called the mechanism of self adaptation towards stress. But generally, each individual responds differently when they got stressed, this differences will result in different ways of adaptation for each individual. Usually what people do to reduce the risk of their stress is finding entertainment, whatever it is form. And the kind of entertainment that is usually people do is playing games. This method is quite effective to reduce stress at work or their routine which is done continuously.

Casual games such as matching colored items can be an alternative to break fatigued and bored. The Colouration game is designed to be an interesting educational game about the division of colors according to Brewster's theory. In this Colouration game, the implementation of artificial intelligence is applied to determine of the level of play that matches the value of the player input variable in completing each game mission. From the results of experiments conducted, shows that the accuracy of Fuzzy Sugeno method in determining the level of play is 100% with the percentage of easy level 38%, normal level 34%, and difficult level 28%. These result prove that this method is good and appropriate for determining the level of play.

## الملخص

أفريدا ، بوتري نور. 2019. تنفيذ خوارزمية فوزي سوغنو في لعبة تلوين. البحث الجامعي. قسم المعلومة التقنية. كلية العلوم والتكنولوجيا. جامعة مولانا مالك إبراهيم مالانج الإسلامية الحكومية. المشرف (I) يونيفا مفتاح العارف، الماجيستر، (II) فرشي نوغراها، الماجيستر.

كلمات البحث : اللعبة، التعليمية، مستوى اللعبة، اللون، فوزي سوغنو.

لا بد أن يتحمل كل انسان عبئاً في حياته، سواء في العمل أو الدراسة أو حياته اليومية. ليس من النادر أن يتحمل عبئاً وشعوراً بالإجهاد الذي يشعرون به بالفعل شعوراً بالتوتر. الإجهاد هو استجابة تجعلنا أفضل. عندما تواجه المشكلة، فإن الإجهاد سيجعلنا ن فكر ونحاول الخروج من المشكلة. وذلك ما يسمى آلية التكيف النفسي للإجهاد. ولكن بشكل عام، يستجيب كل فرد بشكل مختلف عند الضغط عليه، سيؤدي هذا الاختلاف إلى طرق مختلفة للتكيف مع كل فرد. عادة ما يفعله الناس للنقص من خطر الإجهاد هو البحث على الترفيه، أيا كان شكله. وأحد أشكال الترفيه التي عادة هي ممارسة الألعاب. هذه الطريقة فعالة بالفعل لتقليل الضغط عند الشعور بالإجهاد على هامش العمل والملل والشعور بالملل الذي الروتين بشكل مستمر.

العبة عارضة مطابقة العناصر الملونة يمكن أن يكون بديلا لكسر الملل والملل. تصميم لعبة التلوين لتكون لعبة تعليمية مثيرة للاهتمام بالموضوع حول تقسيم الألوان وفقا لنظرية بروسستر. في لعبة التلوين هذا، تنفيذ الذكاء الاصطناعي على تعيين مستوى اللعبة باستخدام خوارزمية فوزي سوغنو. يمكن أن يؤدي استخدام فوزي سوغنو إلى تحديد مستوى اللعب الذي يطابق قيمة متغير إدخال اللاعب في إكمال كل مهمة لعبة. من نتائج التجارب التي أجريت، تبين أن دقة طريقة فوزي سوغنو في تحديد مستوى اللعب هي 100%. مع نسبة المستوى السهل 38% ، المستوى الطبيعي 34% ، والمستوى الصعب 28%. تبين ذلك النتائج أن هذه الطريقة جيدة ومناسبة لتكديد مستوى اللعب.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Setiap orang pasti memiliki beban dalam hidupnya masing-masing, baik dalam hal pekerjaan, studi maupun dalam kehidupan sehari-harinya. Tidak jarang beban dan perasaan tertekan yang ditanggungnya justru berpotensi membuat seseorang merasa stres. Stres muncul sejalan dengan peristiwa dan perjalanan kehidupan yang dilalui oleh individu dan terjadinya tidak dapat dihindari sepenuhnya. Menurut Acevedo dan Ekkekakis dalam bukunya menyatakan bahwa stres dapat ditimbulkan, pertama oleh karakteristik bawaan yang merupakan predisposisi keturunan dan keterbatasan psikologis individu. Kedua, dipengaruhi oleh faktor lingkungan seperti kondisi dan situasi tempat tinggal serta pengalaman masa lalu individu. Dengan demikian munculnya stres dapat disebabkan oleh faktor dari dalam diri individu maupun faktor dari luar diri individu (Acevedo et al, 2006).

Dikutip dari artikel Kompas tahun 2017, Accountemps telah melakukan survei pada 1000 karyawan Amerika Serikat yang hasilnya 60 persen partisipan setuju bahwa stres karena pekerjaan yang terus meningkat drastis selama lima tahun terakhir. Generasi yang paling stres di tempat kerja adalah generasi milenium yang lahir di era '80-an hingga pertengahan era '90-an. Sekitar 64 persen dari partisipan di kisaran usia ini merasa tertekan oleh pekerjaan mereka. Hal ini juga disetujui oleh 59 persen karyawan di usia 35 sampai 54 tahun. Namun, hanya 35 persen karyawan di usia 55 tahun ke atas yang berpendapat demikian. Kemudian, Accountemps juga menyelidiki penyebab stres di tempat kerja. Ternyata, batas waktu untuk menyelesaikan tugas atau target adalah penyebab utama (33 persen), diikuti oleh sulitnya memiliki waktu di luar pekerjaan (22 persen) dan atasan yang tidak masuk akal (22 persen).

Stres merupakan suatu respons yang membuat kita lebih baik. Ketika dihadapkan dengan suatu masalah, stres akan membuat kita lebih berpikir dan berusaha keluar dari masalah tersebut. Itulah yang dinamakan mekanisme adaptasi diri terhadap stres. Saat pemicu stres hilang, diharapkan respons adaptasi stres kita juga menghilang. Respons adaptasi adalah suatu cara untuk menjaga

keseimbangan diri kita (Andri, 2017). Namun pada umumnya tiap individu memberikan respon yang berbeda-beda ketika dilanda stres, perbedaan ini akan menghasilkan cara adaptasi yang berbeda tiap individu terhadap stres. Biasanya yang dilakukan orang-orang untuk mengurangi resiko stres adalah mencari hiburan, apapun bentuknya. Dan salah satu bentuk hiburan yang biasa dilakukan adalah dengan bermain *game*. Cara ini memang cukup efektif untuk menekan stres saat dilanda kepenatan disela-sela kerja.

Seperti pada laman artikel Techno Okezone yang menukil dari laman Indy100, peneliti menyarankan agar mengizinkan para pekerja bermain video *game* saat bekerja. Tujuannya adalah untuk mengurangi tingkat stres yang bisa terjadi pada pekerja tersebut. Peneliti menggunakan tugas berbasis komputer yang bisa menyebabkan '*cognitive fatigue*' atau kelelahan mental para partisipannya. Kemudian mereka pun diberikan waktu beristirahat selama lima menit. Selama waktu tersebut, mereka dibagi menjadi tiga grup yang berbeda. Grup pertama memanfaatkan rehat tersebut untuk bermain video *game* seperti Sushi Cat. Grup kedua lebih memilih untuk melakukan aktivitas relaksasi, sedangkan grup ketiga menghabiskan waktu rehat dengan duduk-duduk dan sibuk dengan gawai maupun komputer mereka. Dalam eksperimen tersebut ditemukan bahwa tingkat stres mereka secara keseluruhan bisa diuji melalui aktivitas ini. Sehingga performa mental mereka bisa diukur. Dari sebuah studi yang ditemukan menyatakan bahwa 66 partisipan yang tidak bermain video *game* dilaporkan merasa kurang bisa terlibat dengan pekerjaan mereka bahkan cenderung khawatir dengan hasil pekerjaan mereka. Sedangkan grup yang memutuskan untuk meluangkan waktu dengan melakukan relaksasi merasa lebih kesulitan setelahnya. Sementara itu, grup yang memutuskan untuk bermain video *game*, mereka cenderung merasa lebih baik setelah rehat. Doctoral Student in Human Factor and Cognitive Psychology di University OF Central Florida, Michael Rupp, mengatakan bahwa banyak pekerja yang berusaha untuk tetap semangat dalam bekerja. "Di mana dalam waktu seharian itu ada momen yang tidak efektif untuk melakukan pekerjaan tersebut. Mereka harus membuat sebuah rencana jangka pendek untuk bisa beristirahat agar bisa terlibat dan menikmati aktivitasnya, salah satunya

seperti bermain video *game*, yang bisa membantu mereka untuk mengisi tenaga kembali,” ungkapnya seperti dinukil dari laman Indy100.

Kejenuhan dan kebosanan akan rutinitas secara terus menerus dapat menimbulkan kemalasan, konsentrasi terganggu hingga berujung mengganggu aktifitas sehari-hari. Dalam agama Islam tidak memaksakan seseorang untuk menghilangkan rasa jenuh, namun Islam mengatur bagaimana agar seseorang tidak cepat dihindangi rasa jenuh sebagaimana di ceritakan dalam hadits riwayat Imam Muslim berikut:

قَالَ كَانَ عَبْدُ اللَّهِ يُدَكِّرُنَا كُلَّ يَوْمٍ حَمِيسٍ فَقَالَ لَهُ رَجُلٌ يَا أَبَا عَبْدِ الرَّحْمَنِ إِنَّا نُحِبُّ حَدِيثَكَ وَنُشْتَهِيهِ وَلَوْ دِدْنَا أَنَّكَ حَدَّثْتَنَا كُلَّ يَوْمٍ. فَقَالَ مَا يَمْتَعِنِي أَنْ أُحَدِّثَكُمْ إِلَّا كَرَاهِيَةٌ أَنْ أُمَلِّكُمْ. إِنَّ رَسُولَ اللَّهِ -صلى الله عليه وسلم- كَانَ يَتَخَوَّلُنَا بِالْمَوْعِظَةِ فِي الْأَيَّامِ كَرَاهِيَةَ السَّامَةِ عَلَيْنَا

Artinya:

*“Adalah Abdullah bin Mas’ud memberikan pelajaran kepada kami setiap hari Kamis, ada salah seorang yang berkata kepadanya: ‘Wahai Abu Abdur Rokhman, kami menyukai pelajaranmu dan kami biasa menghadirinya, kami sangat mengharapkan agar engkau berkenan menyampaikannya setiap hari’. Ibnu Mas’ud menjawab: ‘tidak ada yang membuatku keberatan untuk menyampaikan pelajaran kepada kalian, selain aku khawatir kalian akan bosan, sesungguhnya Rasulullah mengatur penyampaian nasehatnya pada hari tertentu, khawatir akan membuat kami jenuh’”.*

Berdasarkan hal ini penulis berinisiatif untuk membangun sebuah aplikasi permainan kasual yang didalamnya juga memuat unsur edukasi tentang warna yaitu permainan *Colouration*. Permainan ini mengangkat tema tentang pengetahuan pembagian warna. Pada permainan ini pemain diberi tantangan untuk mencocokkan warna. Selain itu, pemain harus menyelesaikan setiap misi untuk dapat melanjutkan ke tingkatan permainan selanjutnya.

Namun, agar permainan lebih menarik maka diperlukan sebuah metode yaitu kecerdasan buatan. Kecerdasan buatan pada permainan membuat kualitas interaksi permainan menjadi lebih baik (Li, Y., P., dan Wyard-Scott, L., 2004). Pada

permainan *Colouration* ini penulis mengimplementasikan algoritma *fuzzy sugeno* untuk menentukan tingkatan permainan. *Fuzzy sugeno* digunakan karena diantara ketiga *fuzzy* yang ada (*tsukamoto*, *mamdani* dan *sugeno*) *fuzzy sugeno* menghasilkan nilai output berupa konstan yang tegas. Sehingga konstanta ini dapat diterapkan langsung pada kasus permainan yang membutuhkan keputusan yang cepat (Purba, 2013). Untuk menuju tingkatan selanjutnya pemain tidak boleh sampai melewati semua misi yang diberikan agar dapat melanjutkan ke tingkat berikutnya. Dengan adanya permainan ini diharapkan membantu meredakan stres serta kejenuhan disela-sela pekerjaan atau kegiatan yang menjemukan.

### 1.2 Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka penelitian dapat dirumuskan yakni bagaimana menentukan tingkat permainan pada permainan *Colouration* menggunakan algoritma *Fuzzy Sugeno*?

### 1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

- a. Permainan bersifat *single player*.
- b. Permainan ini bertipe 2 Dimensi.
- c. Warna yang digunakan untuk tiap *item* secara garis besar terbagi menjadi tiga kelompok warna, yakni: primer, sekunder dan tersier.

### 1.4 Tujuan Masalah

Tujuan dari penelitian ini adalah menentukan tingkat permainan pada permainan *Colouration* menggunakan algoritma *Fuzzy Sugeno*.

### 1.5 Manfaat Penelitian

Manfaat pembuatan aplikasi permainan ini adalah terciptanya permainan yang kasual dan menyenangkan serta edukatif dengan disertakannya konten pengetahuan tentang pembagian kelompok warna yang terdiri dari warna primer, sekunder, dan tersier. Maka di harapkan akan lebih bermanfaat dan dapat menambah pengetahuan semua orang yang bermain permainan ini.

## 1.6 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian tentang permainan *Colouration* adalah sebagai berikut:

### 1. Studi Literatur

Pada tahap ini dilakukan pencaian dan pengumpulan literatur-literatur yang berhubungan dengan penelitian ini. Literatur di dapat dari jurnal, video, buku, dan skripsi yang sudah ada. Literatur berisi pembahasan tentang pembuatan permainan dan penerapan algoritma *fuzzy sugeno* pada permainan.

### 2. Perumusan Masalah

Tahap ini meliputi perumusan masalah, batasan-batasan masalah, dan penyelesaiannya yang akan diterapkan kedalam proses penyelesaian permainan.

### 3. Perancangan Permainan

Tahap ini melakukan analisis dari sumber literatur-literatur yang telah dikumpulkan terkait penelitian. Kemudian melakukan perancangan permainan, yakni: merancang skenario permainan, merancang antar muka permainan, dan perancangan proses dari permainan terkait.

### 4. Pembuatan Permainan

Tahap ini dilakukan pembuatan permainan dengan desain antar muka, desain *item*, desain *level* serta penerapan algoritma *fuzzy sugeno* pada permainan sehingga menghasilkan permainan yang sesuai dengan perancangan awal.

### 5. Pengintegrasian Algoritma

Tahap ini dilakukan pengintegrasian antara permainan *Colouration* dengan algoritma *fuzzy sugeno* yang digunakan untuk penentuan tingkatan permainan sesuai dengan misi yang ada.

### 6. Uji Coba dan Evaluasi Hasil

Tahap ini dilakukan proses uji coba untuk mengetahui hasil dari pembuatan permainan dengan integrasi algoritma *fuzzy sugeno* dan melakukan evaluasi dari setiap percobaan yang telah dilakukan.

### 7. Penyusunan Laporan Penelitian

Tahap ini merupakan tahap akhir dimana penyusun laporan mengenai hasil dari seluruh dokumentasi keseluruhan pelaksanaan penelitian.

### **1.7 Sistematika Penulisan**

Penulisan skripsi ini tersusun dalam lima bab dengan sistematika sebagai berikut:

#### **BAB I PENDAHULUAN**

Bab ini membahas tentang Latar Belakang, Rumusan Masalah, Batsan Masalah, Tujuan Penelitian, Manfaat Penelitian, dan Metode Penelitian.

#### **BAB II TINJAUAN PUSTAKA**

Bab ini menjelaskan teori dasar dan literature-literatur yang melandasi penyusunan penelitian.

#### **BAB III ANALISIS DAN PERANCANGAN**

Bab ini membahas perancangan permainan maupun implementasi algoritma *fuzzy sugeno* pada tingkatan permainan *Colouration*.

#### **BAB IV UJI COBA DAN PEMBAHASAN**

Bab ini membahas tentang hasil implementasi algoritma *fuzzy sugeno* terhadap permainan *Colouration* dan hasil uji coba pada permainan.

#### **BAB V PENUTUP**

Bab ini berisi tentang kesimpulan dari keseluruhan penelitian pembuatan permainan *Colouration* dan penerapan metode algoritma *fuzzy sugeno*.

## BAB II TINJAUAN PUSTAKA

### 2.1 *Game*

#### 2.1.1 *Pengertian Game*

*Game* berasal dari kata bahasa Inggris yang memiliki arti dasar permainan. Permainan dalam hal ini merujuk pada pengertian “kelincahan intelektual” (*intellectual playability*). Sementara kata “*game*” bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target yang ingin dicapai pemainnya. Kelincahan intelektual, pada tingkat tertentu, merupakan ukuran sejauh mana *game* itu menarik untuk dimainkan secara maksimal (Jason, 2008).

Menurut terdapat beberapa pengertian tentang *game* oleh beberapa ahli yaitu:

1. Menurut Agustinus Nilwan dalam bukunya *Pemrograman Animasi dan Game Profesional* terbitan Elex Media Komputindo, *game* merupakan permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami penggunaan animasi haruslah memahami pembuatan *game*. Atau jika ingin membuat *game*, maka haruslah memahami teknik dan metode animasi, sebab keduanya saling berkaitan.
2. Menurut Clark C. Abt, *Game* adalah kegiatan yang melibatkan keputusan pemain, berupaya mencapai tujuan dengan dibatasi oleh konteks tertentu (misalnya, dibatasi oleh peraturan).
3. Menurut Bernard Suits *Game* adalah upaya sukarela untuk mengatasi rintangan yang tidak perlu.
4. Menurut Greg Costikyan, *Game* adalah sebarang karya seni di mana peserta, yang disebut Pemain, membuat keputusan untuk mengelola sumberdaya yang dimilikinya melalui benda di dalam *game* demi mencapai tujuan.

#### 2.1.2 *Jenis Game*

*Genre game* adalah klasifikasi *game* yang didasari interaksi pemainnya. Visualisasi juga menjadi ukuran klasifikasi *genre* ini. Namun untuk beberapa kasus pengembang *game* membuat kompilasi antar berbagai *genre* ini. Tentu saja variasi format *game* lebih banyak (Ivan, 2009).

Berdasarkan *genre*-nya, *game* dapat diklarifikasikan menjadi beberapa jenis, antara lain (Suyanto, 2007):

1. *Maze Game*

Jenis *game* ini adalah jenis *game* yang paling awal muncul. Pada *game* ini pemain hanya mengitari *maze* (lorong-lorong yang berhubungan) untuk mencapai tujuan tertentu. Dalam beberapa *maze game*, terkadang terdapat musuh yang selalu mengejar pemain. Contoh *game* ini adalah *game* Pacman dan Digger.

2. *Board Game*

Jenis *game* ini tidak berbeda jauh dengan *game* tradisional. Sampai saat ini tidak ada variasi yang berbeda pada *game* ini, hanya tampilannya saja yang berbeda dengan *game* tradisional. Contohnya seperti *game* Monopoli.

3. *Card Game*

Hampir sama dengan *board game*, *genre* ini tidak memberikan perubahan berarti dari *game* versi tradisional yang sejenis. Contoh dari *card game* adalah *game* Solitaire dan Hearts.

4. *Battle Card Game*

Jenis *game* ini jarang masuk ke Indonesia. Contoh yang populer di luar negeri adalah *battle card* Pokemon. Ada beberapa stasiun televisi Indonesia yang menayangkan kartun tentang *battle card* ini.

5. *Quiz Game*

Salah satu yang umum dikenal adalah *game* kuis Who Wants to be A Millionaire, sebuah *game* dengan nama yang sama dari acara kuis televisi. Kita hanya perlu memilih jawaban yang benar dari beberapa pilihan jawaban.

6. *Puzzle Game*

Jenis *game* ini memberikan tantangan kepada pemainnya dengan cara memecahkan masalah yang terdapat pada *game* yang dimainkan. Contohnya *game* Jigsaw.

7. *Shoot Them Up*

Jenis ini banyak di minati karena mudah dimainkan. Biasanya musuh kita adalah berbentuk pesawat maupun jenis lain. Datang dari sebelah atas

dengan jumlah yang banyak dan tugas kita adalah menembaki dan menghancurkannya secepat dan sebanyak mungkin. Pada awalnya bentuk *game* ini adalah *game* 2 Dimensi.

#### 8. *Side Scroller Game*

*Game* jenis ini sangat terkenal dengan banyaknya *game* yang dikeluarkan dan berhasil di pasaran. *Game* ini didasarkan kepada 2D dimana pemain bergerak ke sepanjang alur *game* ke satu arah dan menyelesaikan tugasnya. Ada yang melompat, berlari, mengendap, dan menghindari halangan seperti jurang dan proyektil baik dari musuh maupun yang lainnya. Contohnya *game* Duke Nukem Asli, Commander Keen, Prince of Persia, Sonic the Hedgehog.

#### 9. *Fighting Game*

Sesuai dengan namanya, *game* ini mengutamakan pertarungan. *Game* ini memberikan kesempatan kepada pemain untuk mengkombinasikan berbagai gerakan dalam pertarungan. Contoh yang populer adalah *game* Street Fighter.

#### 10. *Racing Game*

*Game* ini memberikan *game* lomba kecepatan dari kendaraan yang dimainkan oleh pemain. Contoh *game* yang terkenal Need for Speed.

#### 11. *Flight Sim*

*Game* ini berfokus pada simulasi penerbangan. Simulasi yang diberikan meniru kondisi dari penerbangan sebenarnya, baik kondisi pesawat dan peralatannya maupun kondisi pemandangan. Contoh yang terkenal dari *game* jenis ini adalah Microsoft Flight Simulator, IL-2 Sturmovik, Apache Air Assault.

#### 12. *Turn Based Strategy Game*

Merupakan jenis *game* yang melibatkan strategi dan logika dari pemain untuk memenangkan *game*. Aturan dari *game* ini adalah setiap pemain melakukan gerakan secara bergantian. Contohnya *game* Civilization.

#### 13. *RTS (Real Time Strategy) Game*

*Game* yang melibatkan strategi dan logika dari pemain untuk memenangkan *game*. Hampir sama dengan *turn based strategy*, akan tetapi dalam *game* ini

seorang pemain tidak perlu menunggu pemain lain untuk melakukan gerakan. Jadi pemain tercepatlah yang besar kemungkinannya untuk menang. Contoh *game* yang populer adalah Warcraft.

#### 14. *Sim*

*Genre* ini menyetengahkan *game* simulasi yang berbeda dengan *Flight Simulator*. Di sini kita sebagai pemain membangun secara simulasi sebuah kota, negara, atau koloni. Kita berperan sebagai “Dewa” disini. Kita mengatur berbagai sumberdaya dan menentukan berbagai keputusan yang kita inginkan dalam proses pembangunan yang sedang terjadi.

#### 15. *First Person Shooter*

*Game* ini mengutamakan kecepatan gerakan kita di dalam *game*. Banyak baku tembak dan kita harus bertahan selama mungkin. Disebut *First Person Shooter* karena pandangan pemain adalah orang pertama (*first person*). Kita melihat tampilan dilayar seperti kita melihat dari mata kita sendiri. Contohnya Counterstrike dan Doom.

#### 16. *First Person 3D Vehicle Based*

Ini sama dengan *genre* FPS di atas hanya bedanya pandangan kita bukan dari mata tetapi dari sudut pandang kendaraan atau mesin yang kita naiki. Kendaraan bisa berbentuk kapal, tank, robot raksasa. Disini kecepatan bukan menjadi faktor utama. Kita harus berjuang dengan kemajuan kita sebaik mungkin untuk tidak dibunuh. Mirip dengan FPS tetapi lebih lambat karena kita berada di dalam sebuah kendaraan atau mesin.

#### 17. *Third Person 3D Game*

Jika *genre* FPS melihat dari sudut pandang orang pertama maka *genre* ini melihat dari sudut pandang orang ketiga. Beberapa *game* menyediakan fitur kedua sudut pandang, tetapi umumnya *game* yang dipasaran hanya menggunakan satu sudut pandang saja.

#### 18. *Role Playing Game (RPG)*

*Game* bertipe ini lebih mengutamakan cerita dan biasanya pemain di ajak masuk ke dalam cerita tersebut untuk menyelesaikan sebuah misi. Dalam *game* ini pemain akan berperan menjadi sebuah karakter dengan berbagai

atriut, seperti kesehatan, intelegensi, kekuatan dan keahlian. Contoh *game* yang terkenal adalah Final Fantasy.

#### 19. *Adventure Game*

*Game* jenis ini adalah *game* dengan mengambil konsep petualangan. Pemain bejalan menuju ke suatu tempat dan disepanjang perjalanan pemain menemukan hal-hal baru untuk di eksplorasi. Contohnya *game* Crash Bandicoot.

#### 20. *Educational and Edutainment*

Banyak pengamat *game* di Indonesia yang mengatakan hanya *genre* ini yang berhasil secara komersil (dalam artian tidak dibajak, seperti *game* keluaran luar). *Genre* ini sebenarnya lebih mengacu kepada isi dan tujuan *game*, bukan *genre* yang sebenarnya. Misalnya *game* Bobby Bola yang sebenarnya campuran dari *genre arcade* dan *side scroller*. Terkadang ada *genre* lain yang dicampurkan sehingga *genre edutainment*, bertujuan untuk memancing minat belajar anak sambil bermain.

#### 21. *Sport*

*Game* ini sama dengan *genre edutainment*. *Genre* ini hanya berdasarkan jenisnya saja, bukan berdasarkan teknologi atau spesifikasi teknis apapun. Selama *game* ini mengetengahkan *genre* olahraga maka disebut *genre sport*. Tidak peduli apakah *game* itu menggunakan gaya *arcade* 2D atau 3D maupun lainnya.

## 2.2 Warna

Menurut Kamus Besar Bahasa Indonesia, warna adalah kesan yang diperoleh mata dari cahaya yang dipantulkan oleh benda-benda yang dikenainya. Sedangkan dalam ilmu fisika, warna adalah sebuah panjang gelombang cahaya yang dihasilkan atau dipantulkan oleh benda sehingga dapat terlihat oleh mata manusia. Manusia dapat melihat cahaya dengan panjang gelombang kira-kira 380 nm, yang setara dengan warna ungu, sampai 780 nm, yang terlihat sebagai warna merah. James E. Cain berpendapat bahwa warna tercipta ketika cahaya mengenai sebuah permukaan sehingga menyerap beberapa panjang gelombang tertentu dan sisanya

dipantulkan, panjang gelombang yang dipantulkan inilah yang diinterpretasikan oleh mata kita sebagai sebuah warna. Sedangkan menurut Frank H. Mahnke, warna bukanlah bagian dari sebuah benda, ruang, ataupun permukaan, warna adalah sensasi yang disebabkan oleh kualitas cahaya tertentu yang terlihat mata dan diinterpretasikan oleh otak. Dan Jeanne Kopacz mengatakan bahwa istilah warna mengacu kepada semua sensasi visual, termasuk kosongnya warna putih dan gelapnya warna hitam, begitu juga intensitas dari merahnya lipstik dan kehalusan sebuah bata merah.

Warna tidak hanya sebatas hal fisik yang dapat kita lihat, namun warna juga ada dalam pikiran kita. Misalkannya, jika disuruh membayangkan sebuah tomat yang matang, mungkin yang ada dipikiran kita adalah tomat berwarna merah, namun yang menyebabkan kita menggambarkan warna merah bukanlah panjang gelombang cahaya antara 627-780 nm, tidak ada objek asli yang menghasilkan atau memantulkan cahaya, maka warna juga tergantung kepada kekuatan imajinasi seseorang (Rahayu, 2013).

Warna adalah estetika yang penting, karena melalui warna itulah kita dapat membedakan secara jelas keindahan suatu objek. Warna dapat didefinisikan secara subjektif atau psikologis yang merupakan pemahaman langsung oleh pengalaman indera penglihatan kita dan secara objektif atau fisik sebagai sifat cahaya yang dipancarkan.

Secara objektif atau fisik, warna diproyeksikan dari panjang gelombang (*wave length*) dan panjang gelombang warna yang masih bisa ditangkap mata manusia berkisar 380-780 nanometer. Cahaya yang tampak oleh mata merupakan salah satu bentuk pancaran energi sempit dari gelombang elektromagnetik.

Pemahaman tentang teori warna ini sudah berlangsung lama, yang dikaji dari sudut pandang ilmu pengetahuan, psikis, dan estetika. Beberapa teori warna yang mengkaji dari sudut pandang ilmu pengetahuan adalah dari Isaac Newton seorang fisikawan yang telah mengkaji teori optik yang kemudian menghasilkan penemuannya mengenai refraksi cahaya menggunakan prisma kaca segitiga yang menghasilkan warna. Kemudian ada teori komplementer yang dicetuskan oleh Brewster yang menghasilkan teori pemahaman pembagian warna menjadi

beberapa kelompok atau *clustering*, Teori ini pertama kali diungkapkan pada 1831 (Meilani, 2013).

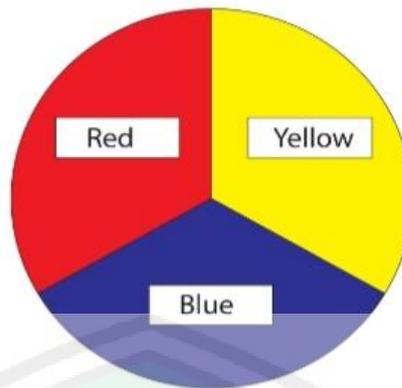
### 2.2.1 Pembagian Warna

Banyak sekali warna-warni yang ada di sekitar kita, tentu tak terhitung jumlahnya. Pada abad ke-19 hingga akhir abad ke-20, Thomas Young menemukan bahwa setiap warna yang bisa dilihat manusia dapat diperoleh dengan mencampurkan secara tepat tiga warna primer yaitu merah, kuning, dan biru. Dengan mencampurkan dua warna primer dengan perbandingan yang sama akan menghasilkan tiga warna sekunder yaitu ungu, hijau, dan oranye, serta dengan mencampur satu warna primer dan sekunder akan menghasilkan enam warna tersier yaitu. Hubungan antar warna tersebut digambarkan dalam roda warna yang mewakili seluruh spectrum warna yang terlihat (Rahayu, 2013).

Warna-warna yang ada di alam jika di sederhanakan dapat di kelompokkan menjadi 4 kategori, yaitu warna primer, warna sekunder, warna tersier, dan warna netral. Dan ini diwujudkan dalam bentuk lingkaran warna Brewster yang mampu menjelaskan teori kontras warna (komplementer), split komplementer, triad dan tetra (Ibnu Teguh, 2013).

#### a. Warna Primer

Warna primer menurut teori warna pigmen dari Brewster adalah warna-warna dasar. Warna-warna lain dibentuk dari kombinasi warna-warna primer. Pada awalnya, mengira bahwa warna primer tersusun atas warna merah, kuning, dan hijau. Namun dalam penelitian lebih lanjut, dikatakan tiga warna primer adalah: merah (Seperti darah), biru (seperti langit atau laut), dan kuning (seperti kuning telur).

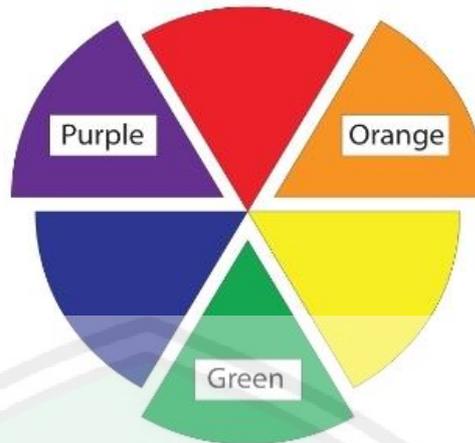


Gambar 2.1: Warna Primer

Ini kemudian dikenal sebagai warna pigmen primer yang dipakai dalam dunia seni rupa. Campuran dua warna primer menghasilkan warna sekunder. Campuran warna sekunder dengan warna primer menghasilkan warna tersier (Mardhiyah, 2014).

#### b. Warna Sekunder

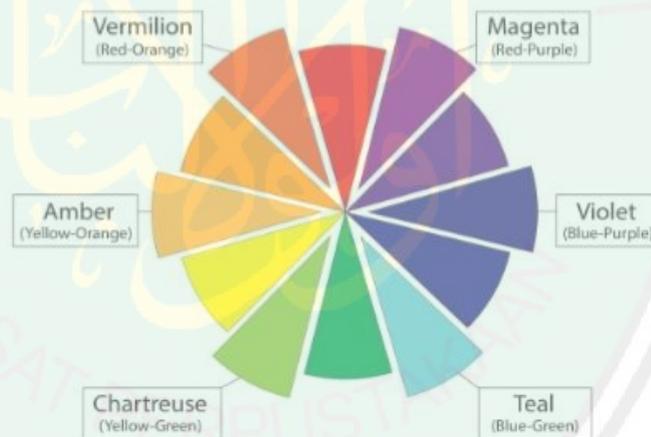
Merupakan warna yang dihasilkan dari campuran dua warna primer dalam sebuah ruang warna. Dalam peralatan grafis, terdapat tiga warna primer cahaya: (R=*Red*) merah, (G=*Green*) hijau, dan (B=*Blue*) biru atau yang lebih dikenal dengan RGB yang bila digabungkan dalam komposisi tertentu akan menghasilkan berbagai macam warna. Misalnya pencampuran 100% merah, 0% hijau, 100% biru akan menghasilkan interpretasi warna magenta. Di dalam computer kita juga mengenal berbagai warna untuk kebutuhan desain *website* maupun grafis dengan kode bilangan Hexadesimal (Mardhiyah, 2014).



Gambar 2.2: Warna Sekunder

### c. Warna Tersier

Warna tersier adalah hasil pencampuran warna primer dengan warna sekunder. Dengan pencampuran warna tersebut menghasilkan warna sebagai berikut:



Gambar 2.3: Warna Tersier

### d. Warna Netral

Warna netral, adalah warna-warna yang tidak lagi memiliki kemurnian warna atau dengan kata lain bukan merupakan warna primer maupun sekunder (Mardhiyah, 2014). Warna yang dikategorikan natural dalam *color wheel* adalah: hitam, abu-abu, dan putih. Warna-warna natural dapat

diperoleh dari warna sekunder dan tersier yang memiliki *tone* rendah atau gelap.

Tabel 2.1: Pembagian Warna

NO.	WARNA		ITEM
	KELOMPOK	NAMA	
1.	PRIMER	<i>Red</i>	
		<i>Blue</i>	
		<i>Yellow</i>	
2.	SEKUNDER	<i>Purple</i>	
		<i>Green</i>	
		<i>Orange</i>	
3.	TERSIER	<i>Vermilion</i>	
		<i>Amber</i>	
		<i>Chartreuse</i>	
		<i>Teal</i>	
		<i>Violet</i>	
		<i>Magenta</i>	
4.	Netral	<i>White</i>	
		<i>Gray</i>	
		<i>Black</i>	

### 2.3 Logika Fuzzy

Logika *Fuzzy* diperkenalkan pertama kali pada tahun 1965 oleh Prof. Lutfi A. Zadeh seorang peneliti di Universitas California di Barkley dalam bidang ilmu komputer. Profesor Zadeh beranggapan logika benar salah tidak dapat mewakili setiap pemikiran manusia, kemudian dikembangkanlah logika *fuzzy* yang dapat mempresentasikan setiap keadaan atau mewakili pemikiran manusia. Perbedaan antara logika tegas dan logika *fuzzy* terletak pada keanggotaan elemen dalam suatu himpunan. Jika dalam logika tegas suatu elemen mempunyai dua pilihan yaitu terdapat dalam himpunan atau bernilai 1 yang berarti benar dan tidak pada himpunan atau bernilai 0 yang berarti salah. Sedangkan dalam logika fuzzy, keanggotaan elemen berada di interval  $[0, 1]$  (Kusumadewi, 2003).

Logika *fuzzy* adalah peningkatan dari logika Boolean yang berhadapan dengan konsep kebenaran sebagian. Di dalam logika *fuzzy* dikenal himpunan *fuzzy*. Himpunan *fuzzy* adalah rentang nilai-nilai. Masing-masing nilai mempunyai derajat keanggotaan antara 0 sampai dengan 1. Ungkapan logika Boolean menggambarkan nilai-nilai “benar” atau “salah”. Logika *fuzzy* menggunakan ungkapan, misalnya: “sangat lambat”, “agak sedang”, “sangat cepat” dan lain-lain untuk mengungkapkan derajat intensitasnya. Di dalam himpunan *fuzzy* terdapat 2 atribut yaitu atribut numeris dan atribut linguistik. Atribut numeris merupakan suatu nilai angka yang menunjukkan ukuran dari suatu variable seperti 40, 25, 50 dsb. Sedangkan atribut linguistik merupakan penamaan suatu group yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami seperti muda, parobaya, tua.

Logika fuzzy menjadi alternatif dari berbagai sistem yang ada dalam pengambilan keputusan karena logika *fuzzy* mempunyai kelebihan sebagai berikut:

- a. Logika *fuzzy* memiliki konsep yang sangat sederhana sehingga mudah untuk dimengerti.
- b. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan dan ketidakpastian.
- c. Logika *fuzzy* memiliki toleransi terhadap data yang tidak tepat.

- d. Logika *fuzzy* mampu mensistemkan fungsi-fungsi non-linier yang sangat kompleks.
- e. Logika *fuzzy* dapat mengaplikasikan pengalaman atau pengetahuan dari para pakar.
- f. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
- g. Logika *fuzzy* didasarkan pada bahasa sehari-hari sehingga mudah dimengerti.

Logika *fuzzy* memiliki beberapa komponen yang harus dipahami seperti himpunan *fuzzy*, fungsi keanggotaan, operator pada himpunan *fuzzy*, inferensi *fuzzy* dan defuzzifikasi.

### 2.3.1 Himpunan *Fuzzy*

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu item  $x$  dalam suatu himpunan  $A$ , yang sering ditulis dengan  $\mu_A[x]$ , memiliki 2 kemungkinan (Kusumadewi S, Purnomo H, 2010) yaitu:

1. Satu (1), yang berarti bahwa suatu item menjadi anggota dalam suatu himpunan, atau
2. Nol (0), yang berarti bahwa suatu item tidak menjadi anggota dalam suatu himpunan.

Terkadang kemiripan antara keanggotaan *fuzzy* dengan probabilitas menimbulkan keracuan. Keduanya memiliki nilai pada interval  $[0,1]$ , namun interpretasi nilainya sangat berbeda antara kedua kasus tersebut. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan bernilai suatu himpunan *fuzzy* usia adalah 0,9; maka tidak perlu di permasalahakan beberapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0,9 usia berarti 10% dari himpunan tersebut diharapkan tidak muda.

Himpunan Fuzzy memiliki 2 atribut, yaitu:

1. Linguistik, yaitu penamaan suatu group yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti MUDA, PAROBAYA, TUA.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 40, 25, 50, dsb

Ada beberapa hal yang perlu diketahui dalam memahami sistem fuzzy (Sri Kusumadewi dan Purnomo, 2010), yaitu:

1. Variabel *Fuzzy*

Variable *fuzzy* merupakan variabel yang hendak dibahas dalam suatu sistem fuzzy. Contoh: umur, temperatur, permintaan, dsb.

2. Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*. Contoh:

- a. Himpunan *fuzzy* untuk variable umur: MUDA, PAROBAYA, dan TUA.
- b. Himpunan *fuzzy* untuk variable temperatur: DINGIN, SEJUK, NORMAL, HANGAT dan PANAS.

3. Semesta Pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Ada kalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya. Contoh:

- a. Semesta pembicaraan untuk variable umur:  $[0+\infty]$ .
- b. Semesta pembicaraan untuk variable temperatur:  $[0 \ 40]$ .

4. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan

*real* yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif dan bilangan negatif. Contoh:

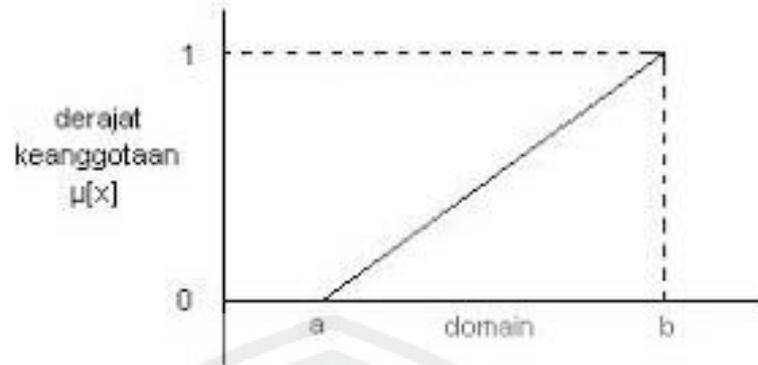
- a. Domain untuk variable umur:
  - MUDA =  $[0, 45]$
  - PAROBAYA =  $[35, 55]$
  - dan TUA =  $[45, +\infty]$
- b. Domain untuk variable temperatur:
  - DINGIN =  $[0, 20]$
  - SEJUK =  $[15, 25]$
  - NORMAL =  $[20, 30]$
  - HANGAT =  $[25, 35]$
  - dan PANAS =  $[30, 40]$

### 2.3.2 Fungsi Keanggotaan *Fuzzy*

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik input data kedalam nilai keanggotaannya (sering juga disebut dengan derajat keanggotaan) yang memiliki interval antara 0 sampai 1 (Kusumadewi, 2002). Fungsi keanggotaan yang membedakan himpunan *fuzzy* dengan himpunan tegas. Fungsi keanggotaan dapat direpresentasikan dengan berbagai cara, namun yang paling umum dan banyak dipakai dalam sistem yang dibuat berdasarkan logika *fuzzy* adalah representasi secara analitik. Pemodelan yang tepat dibutuhkan karena model *fuzzy* sensitif terhadap jenis pendeskripsian himpunan *fuzzy*. Terdapat berbagai jenis pendeskripsian himpunan *fuzzy*, fungsi yang bisa digunakan antara lain:

#### 1. Representasi Kurva *Linear*

Pada representasi *linier*, pemetaan *input* ke derajat keanggotaannya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Ada 2 keadaan himpunan *fuzzy* yang linear. Pertama, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan nol (0) bergerak ke kanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.



Gambar 2.4: Representasi Linear Naik

Fungsi keanggotaan representasi linear naik:

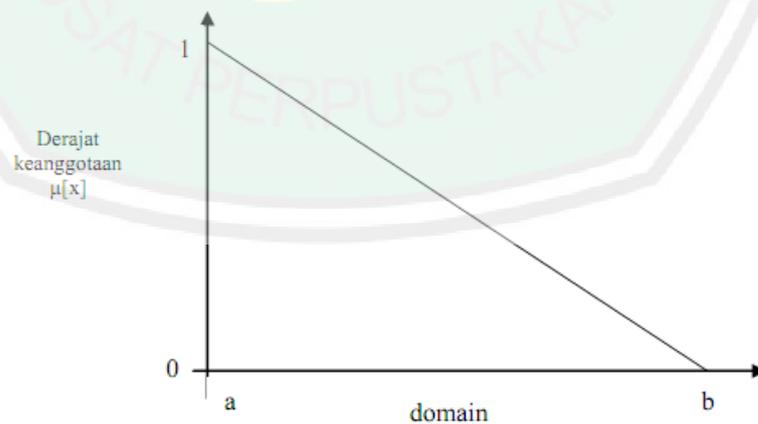
$$\mu[x] = \begin{cases} 0; & x \leq a \\ (x-a) / (b-a); & a < x < b \\ 1; & x \geq b \end{cases}$$

Keterangan:

$a$  = nilai domain terkecil saat derajat keanggotaan terkecil

$b$  = derajat keanggotaan terbesar dalam domain

Kedua, merupakan kebalikan yang pertama. Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah.



Gambar 2.5: Representasi Linear Turun

Fungsi keanggotaan representasi linear turun:

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

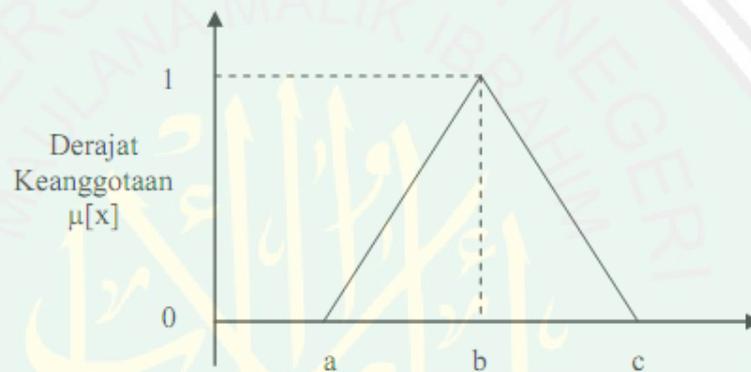
Keterangan:

$a$  = nilai domain terkecil saat derajat keanggotaan terkecil

$b$  = derajat keanggotaan terbesar dalam domain

## 2. Representasi Kurva Segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linear).



Gambar 2.6: Representasi Kurva Segitiga

Fungsi keanggotaan representasi kurva segitiga:

$$\mu[x] = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$$

Keterangan:

$a$  = nilai domain terkecil saat derajat keanggotaan terkecil

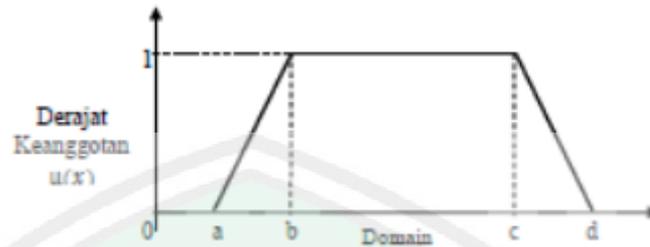
$b$  = derajat keanggotaan terbesar dalam domain

$c$  = nilai domain terbesar saat derajat keanggotaan terkecil

## 3. Representasi Kurva Trapesium

Representasi kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja ada beberapa titik yang memiliki nilai keanggotaan

1. Representasi kurva trapesium dapat dilihat pada dibawah



Gambar 2.7: Representasi Kurva Trapesium

Fungsi keanggotaan representasi kurva segitiga:

$$\mu(x) = \begin{cases} 0, & x \leq a \text{ atau } x \geq d \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \end{cases}$$

### 2.3.3 Operator Dasar Pada Himpunan *Fuzzy*

Seperti himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasikan dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau  $\alpha$ -predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh (Kusumadewi dan Purnomo, 2004: 25-27), yaitu:

#### a. Operator AND

Operator ini berhubungan dengan operasi interseksi pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B} = \min(\mu_A[x], \mu_B[y])$$

b. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B} = \max(\mu_A[x], \mu_B[y])$$

c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{A'} = 1 - \mu_A[x]$$

Karena himpunan *fuzzy* tidak dapat dibagi dengan tepat seperti halnya pada himpunan tegas, maka operasi-operasi ini diaplikasikan pada tingkat keanggotaan. Suatu elemen dikatakan menjadi anggota himpunan *fuzzy* jika (Kusumadewi, 2002: 60):

- a. Berada pada domain himpunan tersebut.
- b. Nilai kebenaran keanggotaannya  $\geq 0$ .
- c. Berada diatas ambang  $\alpha$ -predikat yang berlaku.

### 2.3.4 Sistem Inferensi *Fuzzy*

Sistem inferensi *fuzzy* merupakan tahap evaluasi pada aturan *fuzzy*. Tahap evaluasi dilakukan berdasarkan penalaran dengan menggunakan *input fuzzy* dan aturan *fuzzy* sehingga diperoleh *output* berupa himpunan *fuzzy*. Struktur dasar sistem inferensi *fuzzy* terdiri atas (Kuncahyo B, Ginardi R, Arieshanti I, 2012):

- a. Sebuah basis aturan yang berisi aturan *fuzzy if-then*.
- b. Basis data yang mendefinisikan fungsi keanggotaan himpunan *fuzzy*.
- c. Unit pengambilan keputusan yang menyatakan operasi inferensi atau aturan-aturan yang ada.

- d. Fuzzifikasi yang mentransformasikan masukan klasik (*crisp*) ke derajat tertentu sesuai dengan fungsi keanggotaan.
- e. Defuzzifikasi yang mentransformasikan hasil inferensi *fuzzy* ke dalam bentuk *crisp*.

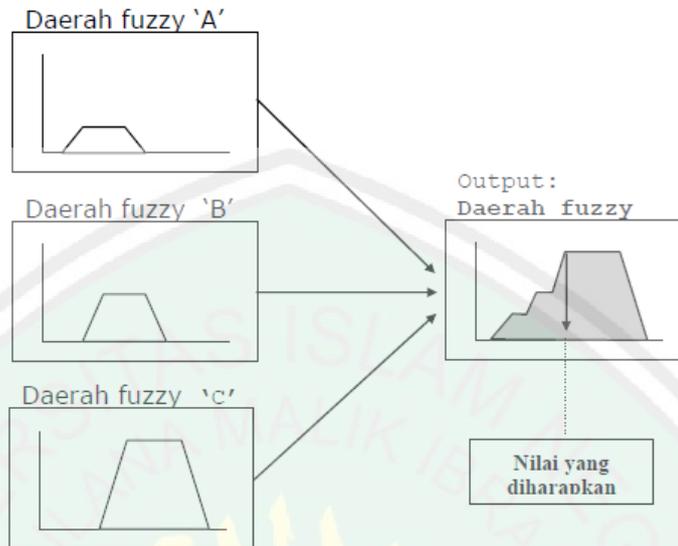
Ada 3 metode yang sering digunakan dalam sistem inferensi *fuzzy*, yaitu metode *Mamdani*, metode *Tsukamoto* dan metode *Sugeno*.

### 1. Metode Mamdani

Metode Mamdani pertama kali diperkenalkan oleh Ibrahim Mamdani pada tahun 1975. Metode ini merupakan metode yang paling sederhana dan paling sering digunakan untuk penelitian dibandingkan metode yang lain. *Input* dan *output* pada metode mamdani berupa himpunan *fuzzy* (Sri, 2002:98). Metode *Mamdani* menggunakan fungsi implikasi MIN dan agregasi MAX sehingga metode *Mamdani* juga disebut dengan metode *MIN-MAX* (*min-max inferencing*). Untuk mendapatkan *output*, diperlukan 4 tahapan (Kusumadewi dan Purnomo, 2004: 39-45), yaitu:

- 1) Pembentukan himpunan *fuzzy*  
 Pada metode *Mamdani*, baik variabel *input* maupun variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.
- 2) Aplikasi fungsi implikasi  
 Metode *Mamdani* menggunakan fungsi implikasi *MIN*
- 3) Komposisi aturan  
 Tidak seperti penalaran monoton, apabila sistem terdiri dari beberapa aturan, maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Ada 3 metode yang digunakan dalam melakukan inferensi sistem *fuzzy*, yakni: *max*, *additive*, dan *probabilistic OR* (*probor*)
- 4) Penegasan (*defuzzy*)  
*Input* dari proses *defuzzy* adalah suatu himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy*, sedangkan *output* yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut. Sehingga jika

diberikan suatu himpunan *fuzzy* dalam *range* tertentu, maka harus dapat diambil suatu nilai *crisp* tertentu sebagai *output*.



Gambar 2.8: Proses Defuzzy

Ada beberapa metode *defuzzy* yang bisa dipakai pada komposisi aturan *Mamdani*, antara lain:

a. Metode *Centroid* (*Composite Moment*)

Metode *Centroid* disebut juga metode *Center of Gravity* atau metode pusat luas (*Center of Area, CoA*). Proses defuzzifikasi pada metode *Centroid* adalah dengan mengambil nilai titik pusat ( $x^*$ ) dari daerah pada fungsi keanggotaan  $B$ . Rumus metode *centroid* (Wang, 1997:107) didefinisikan sebagai

$$x^* = \frac{\int x \mu_B(x) dx}{\int \mu_B(x) dx}$$

Untuk domain kontinu, dan

$$x^* = \frac{\sum_{i=1}^n x_i \mu_B(x_i)}{\sum_{i=1}^n \mu_B(x_i)}$$

Selain mudah dalam perhitungan, keuntungan menggunakan metode *centroid* adalah nilai *defuzzy* bergerak halus sehingga

perubahan dari suatu topologi himpunan *fuzzy* ke topologi himpunan *fuzzy* berikutnya juga bergerak secara halus.

b. Metode *Bisektor*

Metode bisektor mengambil nilai pada domain himpunan *fuzzy* yang memiliki nilai keanggotaan pada daerah *fuzzy* sebagai solusi tegas, dan didefinisikan sebagai:

$$x_p = \int_a^p \mu(x) dx = \int_p^b \mu(x) dx$$

Dengan  $a = \min(x: x \in X)$  dan  $b = \max(x: x \in X)$  sedangkan  $p = x$  yang membagi daerah inferensi menjadi dua bagian yang sama besar.

c. Metode *Mean of Maximum (MoM)*

Solusi tegas diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki nilai keanggotaan maksimum.

d. Metode *Largest of Maximum (LoM)*

Solusi tegas diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum.

e. Metode *Smallest of Maximum (SoM)*

Solusi tegas diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum.

## 2. Metode Tsukamoto

Pada metode *Tsukamoto*, setiap konsekuen pada aturan yang berbentuk *IF-THEN* harus direpresentasikan dengan suatu himpunan *fuzzy* dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, *output* hasil inferensi dari tiap-tiap aturan diberikan secara tegas (*crisp*) berdasarkan  $\alpha$ -predikat (*fire strength*). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot (Kusumadewi dan Purnomo, 2004: 33).

## 3. Metode Sugeno

Metode sugeno merupakan metode inferensi *fuzzy* untuk aturan yang direpresentasikan dalam bentuk *IF-THEN*, dimana *output* (konsekuen) system tidak berupa himpunan *fuzzy*, melainkan berupa konstanta atau persamaan *linear*.

Model *Sugeno* menggunakan fungsi keanggotaan satu (1) pada suatu nilai *crisp* tunggal dan nol (0) pada nilai *crisp* yang lain. Metode ini diperkenalkan oleh Takagi-Sugeno Kang pada tahun 1985 (Kusumadewi, 2003).

Penalaran metode *Sugeno* hampir sama dengan penalaran metode *Mamdani*, hanya saja *output* sistem tidak berupa himpunan *fuzzy* melainkan berupa konstanta atau persamaan *linear*. Ada 2 bentuk model *fuzzy sugeno*, yaitu:

- a. Model *fuzzy sugeno* orde-nol

Secara umum bentuk model *fuzzy sugeno* orde-nol adalah:

$$IF(x_1 \text{ is } A_1) \circ (x_2 \text{ is } A_2) \circ \dots \circ (x_i \text{ is } A_i) THEN z = k$$

Dengan  $A_t$  adalah himpunan ke- $i$  sebagai anteseden dan  $k$  adalah suatu konstanta (tegas) sebagai konsekuen.

- b. Model *fuzzy sugeno* orde-satu

Secara umum bentuk model *fuzzy sugeno* orde-satu adalah:

$$IF(x_1 \text{ is } A_1) \circ \dots \circ (x_i \text{ is } A_i) THEN z = p_1 * x_1 + \dots + p_i * x_i + q$$

Dengan  $A_t$  adalah himpunan ke- $i$  sebagai anteseden dan  $p_t$  adalah suatu konstanta (tegas) ke- $i$  dan  $q$  juga merupakan konstanta dalam konsekuen. Apabila komposisi aturan menggunakan metode *sugeno*, maka *defuzzy* dilakukan dengan cara mencari nilai rata-ratanya.

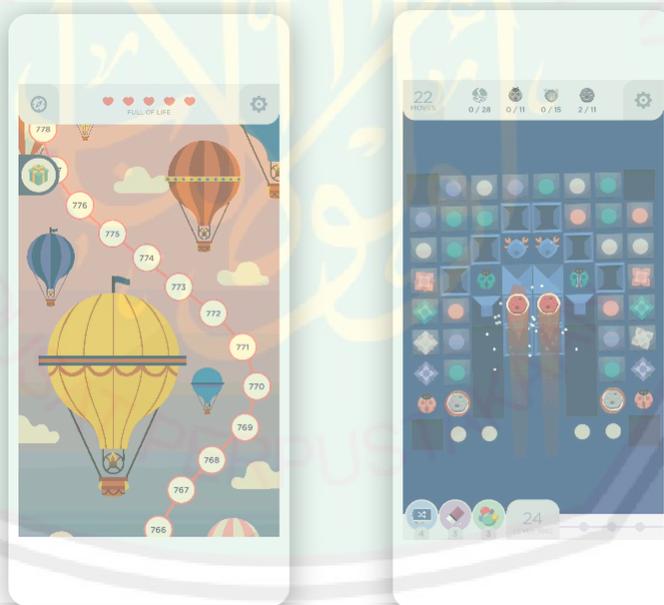
#### 2.4 *Game Two Dots*

Two Dots adalah permainan *puzzle* yang didirikan oleh Paul Murphy dan Patrick Moberg. Permainan yang berkantor pusat di New York ini tersedia untuk iOS dan Android yang dikembangkan dan dipublikasikan oleh Playdots, Inc. Namun pada platform Windows 10 Mobile dan versi Microsoft Windows tidak lagi didukung. Permainan ini dirilis untuk platform iOS pada 29 Mei 2014 dan tersedia untuk Android pada 12 November 2014. Two Dots merupakan sekuel dari Dots. Tidak seperti Dots, Two Dots memiliki *power up*, sasaran, dan *leveling*, sementara kekurangan dari permainan ini tidak ada pilihan bermain *online* secara *multi-player*. Tujuan permainan ini adalah menghubungkan *item-item* yang berwarna sama, menenggelamkan jangkar, dan memecahkan es. *Update game*

pada tanggal 1 September 2017, ada 50 tema 'dunia' yang diperbarui, yang berbeda dalam unsur permainan, hambatan serta tujuan.

Cara bermain permainan Two Dots, pemain harus menghubungkan *item-item* setidaknya dua warna yang sama untuk bergerak dan menyelesaikan misi. *Item* tersebut dapat dihubungkan secara horizontal atau vertikal, tetapi tidak dapat dihubungkan secara diagonal. Selain menghubungkan *item-item* secara horizontal dan vertikal, pemain dapat menghubungkan setidaknya empat *item* untuk membentuk persegi, kemudian semua *item-item* dalam area permainan dengan warna yang sama akan hilang. Setiap minggunya ada modus "Berburu harta karun", dimana pemain seluruh dunia dapat bersaing menjadi yang pertama untuk menyelesaikannya melalui tujuh tahap tingkatan.

Two Dots menerima cukup respon positif dari para kritikus di GameRankings dan Metascore. Sementara itu permainan Two Dots dipuji untuk desain sederhana, berbagai hambatan dan tingkat layout (Dots, 2015)



Gambar 2.9: Tampilan *Game Two Dots*

## 2.5 *Game Candy Crush Saga*

King adalah perusahaan hiburan interaktif terkemuka untuk dunia *mobile*. King telah mengembangkan lebih dari 200 judul permainan, termasuk Candy Crush,

Farm Heroes, Pet Rescue dan Bubble Witch. King memiliki 314 juta pengguna aktif bulanan pada kuartal kedua tahun 2017 di seluruh web, dan sosial platform *mobile*. Semua permainan keluaran King tersedia secara gratis, pemain juga dapat membeli *item virtual* dengan harga yang telah disesuaikan dan menanamkan fitur sosial untuk meningkatkan pengalaman pemain. King memiliki permainan studio di Stockholm, Malmo, London, Barcelona, Berlin dan Seattle, bersama dengan kantor di San Francisco, Malta, Tokyo dan Bukares. King di akuisisi oleh Activision Blizzard Inc (Nasdaq ATVI) pada 2016 Februari dan beroperasi sebagai suatu unit independen perusahaan.

Candy Crush Saga adalah permainan yang didirikan oleh Riccardo Zacconi yang sekarang menjabat sebagai Chief Executive Officer dan Sebastian Knutsson yang menjabat sebagai Chief Creative Officer. Candy Crush Saga dirilis pada tanggal 12 April 2012 untuk Facebook dan kemudian dirilis untuk *smartphone* pada tanggal 14 November 2012. Permainan ini dikembangkan oleh King. Pada Maret 2013, Candy Crush Saga mengalahkan FarmVille 2 sebagai permainan yang paling populer di Facebook dengan rata-rata pengguna bulanan 46 juta. Candy Crush Saga adalah pengembangan dari permainan Candy Crush (Discover King, 2017).



Gambar 2.10: Tampilan Game Candy Crush Saga

## 2.6 Penelitian Terkait

Berikut ini merupakan beberapa penelitian sebelumnya yang berkaitan dengan penelitian yang akan dilakukan:

Penerapan algoritma *fuzzy* mamdani diterapkan pada penelitian Latus Hermawan dan Astrid Novita Putri (2014) untuk mengatur *game scoring* pada permainan Helitap. Parameter yang digunakan terdiri dari nilai pemain, *range* antar penghalang dan kecepatan penghalang. Penelitian ini telah menguji sebanyak 100 kali, dimana 50 kali pada *game* yang diberi metode dan 50 kali yang belum diberi metode. Dari percobaan tersebut keberhasilan pemain yang menggunakan metode *fuzzy* mencapai 90% karena pada 50 kali percobaan terdapat 5 kali yang berhasil, dimana nilai yang tanpa metode lebih tinggi dengan yang memakai metode. Dari penelitian tersebut dapat disimpulkan bahwa dengan menerapkan metode *fuzzy* dapat meningkatkan nilai dan membantu pemain dalam memainkan permainan.

Penelitian lain yang berkaitan tentang *fuzzy* juga dilakukan Ardiawan Bagus Harisa, Hanny Haryanto dan Heru Agus Santoso (2016) yang menggunakan logika *fuzzy* sebagai pendekatan untuk implementasi tingkat kesulitan dinamis pada game Wayang Ramayana. Sistem pada game memiliki 4 tingkat kelas kesulitan, yakni: *Beginner*, *Medium*, *Hard*, *Very Hard*. Parameter yang digunakan pada penelitian ini terdiri dari sisa *lifepoint* (HP), sisa waktu (*TIME*), kemampuan menyelesaikan tugas (*SOLVE*), tingkat kesulitan sebelumnya (*DIFFICULTY*), tingkat keakuratan tembakan pemain (*ACCURACY*), serta banyaknya pemain melakukan percobaan (*TRY*). Parameter *input* diambil dari atribut karakter pemain saat bermain, kemudian sistem akan mengolah dan mengklarifikasi pemain yang nantinya akan berdampak pada tingkat kesulitan dalam *game* secara otomatis. Hasil dari penelitian ini logika *fuzzy* yang diimplementasikan pada sistem dapat membuat pemodelan pemain dan menentukan tingkat kesulitan permainan secara otomatis berdasarkan kemampuan pemain dalam menyelesaikan tugas.

## BAB III ANALISIS DAN PERANCANGAN

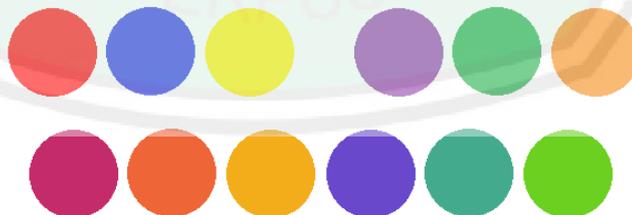
### 3.1 Perancangan Permainan

#### 3.1.1 Deskripsi Permainan

Permainan “*Coloration*” ini mengadopsi dari permainan terkenal seperti Candy Crush Saga, Two Dots, Farm Heroes Saga, Juice Cubes dan permainan dengan *genre* sejenisnya. Permainan ini ber-*genre puzzle game* dengan konten edukasi pengetahuan tentang pembagian warna yang dimainkan secara *single player*. Permainan 2D ini lebih bersifat santai, bermain sambil belajar dan dapat memberikan pembelajaran untuk pengelompokan warna-warna sesuai teori Brewster yang di bagi menjadi tiga kelompok, yakni: warna primer, warna sekunder, warna tersier dan warna netral. Pengelompokan warna Brewster ini di implementasikan pada tiap *level* permainan. Pada *level* satu sampai *level* tiga, *item* berwarna primer. *Level* empat sampai *level* enam tentang warna sekunder, dan kelompok warna terakhir yakni warna tersier dari *level* tujuh sampai *level* sembilan.

#### 3.1.2 Perancangan *Item*

Pada permainan ini kelompok warna primer, warna sekunder dan warna tersier akan ditampilkan sebagai *item* berbentuk bulat. Setiap warna dalam kelompok pembagian warna akan dijadikan *item*.



Gambar 3.1: Rancangan *Item*

### 3.1.3 Perancangan *Obstacle*

*Obstacle* atau halangan dibuat agar permainan semakin menarik dan menantang. Semakin tinggi tingkat permainan, maka *obstacle* semakin banyak.

Pada permainan ini terdapat 3 *obstacle*, yang terdiri dari:

1. *Concrete*, *item* yang dapat hancur satu kali ;
2. *Breakable*, *item* yang dapat hancur tiga kali, dan;
3. *Blank*, atau kosong jadi tidak ada *item* sama sekali.

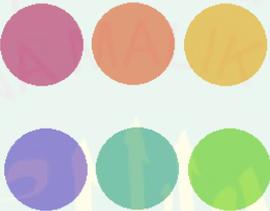


Gambar 3.2: Rancangan *Obstacle Concrete, Breakable, dan Blank*

### 3.1.4 Perancangan *Level*

Jumlah warna pada masing masing kelompok pembagian warna yang akan digunakan sebagai *item* berjumlah 12 warna. Urutan tingkatan permainan akan dimulai dari kelompok warna primer, warna sekunder sampai dengan warna tersier. Pada setiap kelompok warna akan dibagi menjadi 3 tingkat kesulitan, yaitu: mudah, normal, dan sulit. Pada *level* 1, 2, dan 3 dimulai dari kelompok warna primer, dimana untuk *level* 1 dengan tingkat kesulitan mudah, *level* 2 dengan tingkat kesulitan normal, dan *level* 3 dengan tingkat kesulitan sulit. Untuk *level* 4, 5, dan 6 menggunakan kelompok warna sekunder. *Level* 4 dengan tingkat kesulitan mudah, *level* 5 dengan tingkat kesulitan normal, dan *level* 6 dengan tingkat kesulitan sulit. Sedangkan kelompok warna tersier pada *level* 7, 8, dan 9. dimana untuk *level* 7 dengan tingkat kesulitan mudah, *level* 8 dengan tingkat kesulitan normal, dan *level* 9 dengan tingkat kesulitan sulit.

Tabel 3.1: Pembagian Tingkat Kesulitan pada *Level*

No.	Warna		<i>Level</i>	
	Kelompok	<i>Item</i>	Nomor	Tingkat Kesulitan
1.	Primer		1	Mudah
			2	Normal
			3	Sulit
2.	Sekunder		4	Mudah
			5	Normal
			6	Sulit
3.	Tersier		7	Mudah
			8	Normal
			9	Sulit

Perbedaan dari setiap tingkatan permainan ada pada jumlah *moves* (batas gerakan), jumlah *goal* (misi) dan jumlah *obstacle* (halangan). Jumlah *goals* yang dimaksud adalah jumlah banyaknya misi yang harus diselesaikan di setiap *level* agar dapat naik ke tingkatan selanjutnya. Jumlah *moves* (batas gerakan) pada *level* mudah adalah 9 *move* dan ada satu *obstacle*. Pada *level* normal, ada 2 *obstacle* dan jumlah *moves* nya adalah 12 *move*. Sedangkan pada *level* sulit, jumlah *move*-nya lebih banyak, yakni 18 *move* dengan halangan sebanyak 3 *obstacle*.

Kemudian perbedaan dari setiap *level* selanjutnya adalah pada jumlah *goals* yang harus terselesaikan. Jumlah *goals* yang dimaksud adalah jumlah banyaknya misi yang harus diselesaikan di setiap *level* agar dapat naik ke *level* selanjutnya. Pada *level* mudah, jumlah *goals* tidak lebih dari sama dengan 20 *item*. Pada *level* normal, jumlah *goals* tidak lebih dari sama dengan 30 *item*. Sedangkan pada *level* sulit, jumlah *goals* tidak lebih dari sama dengan 40 *item*. Pada tiap *level* permainan hanya ada 3 *item* saja, meskipun di *level* mudah, normal, dan sulit.

Selain itu, perbedaan pada jumlah *obstacle* pada tiap *level* adalah pada tingkat kesulitan. Semakin tinggi *level*, maka semakin banyak *obstacle*. Pada *level*

mudah, jumlah *obstacle* sebanyak satu yakni *blank*. Pada *level* normal, jumlah *obstacle* tidak lebih dari sama dengan 2 *obstacle*: *blank* dan *breakable*. Sedangkan pada *level* sulit, jumlah *obstacle* tidak lebih dari sama dengan 3 *obstacle*: *blank*, *concrete* dan *breakable*.

Tabel 3.2: Pembagian *Obstacle* pada Setiap *Level* Permainan

No.	Level		Jumlah <i>Obstacle</i>		
	Nomor	Tingkat Kesulitan	<i>Blank</i>	<i>Breakable</i>	<i>Concrete</i>
1	1	Mudah	8	-	-
2	2	Normal	8	4	-
3	3	Sulit	8	2	6
4	4	Mudah	8	-	-
5	5	Normal	8	14	-
6	6	Sulit	6	6	12
7	7	Mudah	10	-	-
8	8	Normal	14	6	-
9	9	Sulit	6	6	6

### 3.1.5 Perancangan Penilaian

Pada saat pemain menyelesaikan permainan, diakhir permainan pemain akan mendapatkan nilai. Nilai menjadi salah satu tolak ukur kemampuan pemain. Pemain tersebut akan mendapatkan nilai dalam bentuk bintang dan *score*. Untuk nilai, semakin besar nilai yang didapatkan maka semakin bagus. Sedangkan untuk catatan *moves* (batas gerak) semakin banyak sisa yang tercatat maka semakin baik. Aturan pemberian nilai pada permainan ini setidaknya menyambungkan dua *item* dengan warna yang sama dan setiap *item* warna diberi nilai 5.

### 3.1.6 Skenario Permainan

Pada permainan ini pemain akan dihadapkan dengan permainan pembagian warna menurut teori *Brewster*. Pemain akan bertugas untuk menyelesaikan *goals* yang telah ditentukan dengan ketentuan jumlah *moves* yang disediakan. Ketika

pemain berhasil menyelesaikan *goals* dengan benar dan tidak melampaui jumlah *moves*, maka pemain akan mendapatkan tampilan *loading screen* “*SUCCESS*”, jumlah *score* dan bintang. Akan tetapi jika pemain tidak menyelesaikan *goals* sesuai dengan yang ditentukan, dan kehabisan *moves* maka pemain akan mendapatkan tampilan *loading screen* “*FAILED*” dan tidak dapat naik ke *level* selanjutnya. Selanjutnya, apabila pemain dapat menyelesaikan seluruh *goals* dan tidak kehabisan *moves*, maka pemain telah berhasil menyelesaikan permainan dan dapat melanjutkan *level* permainan selanjutnya.

Pemain akan memulai permainan dari *Level 1: Primer*, yang terdiri dari *item* berwarna merah, biru dan kuning dan berakhir di *Level 9: Tersier*, yang terdiri dari *item* berwarna Merah Keungunan, Jingga Kemerahan, Jingga Kekuningan, Biru Keunguan, Biru Kehijauan, dan Hijau Kekuningan. Di tiap *level* terdapat beberapa halangan. Semakin tinggi *level* maka halangan, misi *goals* akan semakin banyak, sedangkan *moves* akan semakin sedikit.

Pada saat awal permainan, pemain dapat menyelesaikan misi yang sudah ditentukan pada *level* mudah, statistik dari permainan akan menentukan ke *level* manakah pemain akan bermain berikutnya. Artinya, pemain bisa saja melewati beberapa *level* jika memang statistik dari pemainnya memenuhi kriteria yang telah ditentukan. Misalnya, pemain dapat langsung bermain pada *level* normal pada tingkatan selanjutnya tanpa harus bermain pada *level* mudah terlebih dahulu atau langsung bermain pada *level* sulit tanpa harus bermain pada *level* mudah dan normal terlebih dahulu. Penentuan *level* ini akan diproses menggunakan *Fuzzy Sugeno*.

### 3.2 Perancangan Antarmuka Permainan

Pada saat pemain memainkan permainan ini, pemain akan berinteraksi dengan antarmuka permainan. Tampilan antarmuka permainan yang sesuai akan menambah minat pemain untuk memainkannya. Tampilan antarmuka dirancang sebagus mungkin agar menarik dan menyenangkan saat dimainkan.

### 3.2.1 Perancangan Tampilan Menu Utama

Bagian menu utama merupakan bagian yang pertama kali dijumpai oleh pemain saat akan memulai permainan. Pada bagian menu utama akan disediakan pilihan menu “PLAY” yang berfungsi untuk memulai permainan.



Gambar 3.3: Rancangan Tampilan Menu Utama

### 3.2.2 Perancangan Tampilan Level Permainan

Pada bagian *level* Permainan, akan ditampilkan tahap tingkatan yang akan dilalui pemain. Selain itu juga akan ditampilkan informasi jumlah nyawa. Pada bagian ini juga terdapat menu “Keluar”, “Home”, dan “Pengaturan”.



Gambar 3.4: Rancangan Tampilan Level

### 3.2.3 Perancangan Tampilan Papan Permainan

Pada bagian Papan Permainan, akan ditampilkan *item-item* yang harus diselesaikan sesuai dengan target. Selain itu juga akan ditampilkan informasi terkait *Goal* (misi), *Moves* (batas gerak), *Score* (nilai), keterangan *Level*. Pada bagian ini juga terdapat menu “Keluar” dan “Pengaturan”. Pada pengaturan, pemain dapat mengaktifkan atau menonaktifkan musik atau *backsound game*.



Gambar 3.5: Rancangan Tampilan Papan Permainan

### 3.2.4 Perancangan Tampilan Loading Screen

Pada bagian ini akan ditampilkan informasi terkait keterangan “*Success*”, “*Failed*” dan “*Mission*”. Pada *loading screen success* terdapat tombol “*Continue*” yang berfungsi untuk meneruskan permainan di *level* selanjutnya dan pada *loading screen failed* terdapat tombol “*Play Again*” untuk memainkan permainan lagi agar dapat naik *level*. Kemudian *loading screen mission* terdapat keterangan brapa banyak *item* yang harus diselesaikan dan tombol play untuk melanjutkan permainan.



### 3.4.1 Variabel Fuzzy

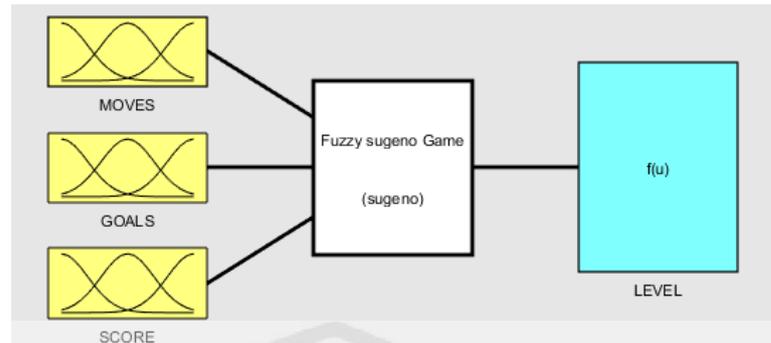
Variabel (*input*) yang digunakan sebagai penentu *level (output)* pada *puzzle* ada 3, yaitu variabel *Moves*, variabel *Goals*, variabel *Score*. Sedangkan variabel *output*-nya adalah variabel *Level*. Dari tiga variabel yang digunakan, maka nilai linguistiknya sebagai berikut:

Tabel 3.3: Nilai linguistik Variabel Fuzzy: *Input* dan *Output*

Variabel		Himpunan Fuzzy
<i>Input</i>	<i>Moves</i> (Batas Gerak)	Sedikit
		Sedang
		Banyak
	<i>Goals</i> (Jumlah Misi)	Sedikit
		Sedang
		Banyak
	<i>Score</i> (Nilai)	Sedikit
		Sedang
		Banyak
<i>Output</i>	<i>Level</i>	Mudah
		Normal
		Sulit

### 3.4.2 Fuzzyfikasi

*Fuzzyfikasi* merupakan proses memetakan nilai *crisp* (numerik) ke dalam himpunan *fuzzy* dan menentukan derajat keanggotaannya. Secara garis besar pemetaan nilai *crisp* ke dalam himpunan *fuzzy* dijelaskan pada gambar dibawah ini

Gambar 3.8: Proses *Fuzzy*

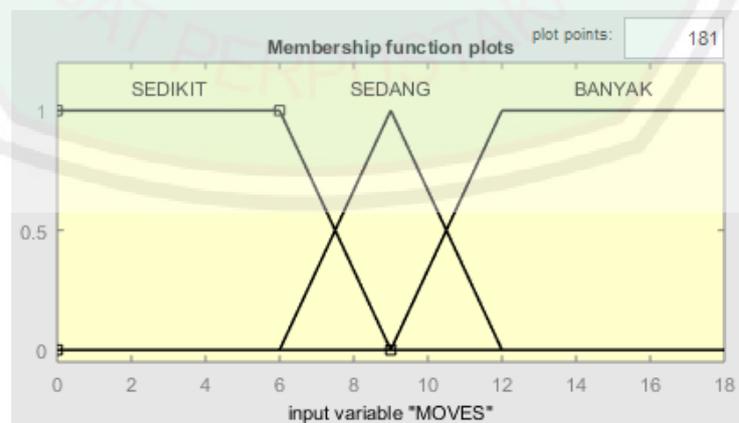
Berdasarkan *Fuzzy Inference System* (FIS) maka akan dilakukan pemetaan sebagai berikut:

### 1. Variabel *Moves* (batas gerak)

Variabel *Moves* terdapat 3 himpunan *fuzzy*, yaitu: Sedikit, Sedang, dan Banyak. Semesta pembicaraan pada variabel *Moves* adalah  $[0 \ 18]$ . Sedangkan untuk domain himpunan *fuzzy*-nya adalah sebagai berikut:

Tabel 3.4: Domain Himpunan Keanggotaan *Moves*

Himpunan <i>Fuzzy</i>	Domain
Sedikit	0 - 9
Sedang	6 - 12
Banyak	9 - 18

Gambar 3.9: Himpunan *Fuzzy* untuk Variabel *Moves*

Pada himpunan Sedikit fungsi keanggotaan yang digunakan adalah fungsi representasi linear turun, sehingga untuk mendapatkan derajat keanggotaan pada himpunan Sedikit dengan menggunakan rumus:

$$\mu_{\text{MovesSedikit}}[x] = \begin{cases} 1; & x \leq 6 \\ \frac{9-x}{9-6}; & 6 < x < 9 \\ 0; & x \geq 9 \end{cases} \quad (3.1)$$

Pada himpunan Sedang fungsi keanggotaan yang digunakan adalah fungsi representasi kurva segitiga, sehingga untuk mendapatkan derajat keanggotaan pada himpunan Sedang dengan menggunakan rumus:

$$\mu_{\text{MovesSedang}}[x] = \begin{cases} 1; & x \leq 6 \\ \frac{x-6}{9-6}; & 6 \leq x \leq 9 \\ \frac{12-x}{12-9}; & 9 \leq x \leq 12 \\ 0; & x \leq 9 \parallel x \geq 12 \end{cases} \quad (3.2)$$

Pada himpunan Banyak fungsi keanggotaan yang digunakan adalah fungsi representasi linear naik, sehingga untuk mendapatkan derajat keanggotaan pada himpunan Banyak dengan menggunakan rumus:

$$\mu_{\text{MovesBanyak}}[x] = \begin{cases} 0; & x \leq 9 \\ \frac{x-9}{12-9}; & 9 < x < 12 \\ 1; & x \geq 12 \end{cases} \quad (3.3)$$

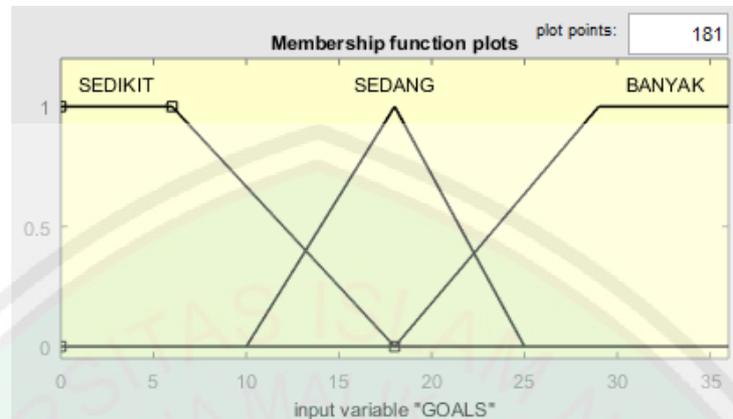
## 2. Variabel Goals (jumlah misi yang harus diselesaikan)

Variabel *goals* terdapat 3 himpunan *fuzzy*, yaitu: Sedikit, Sedang, dan Banyak. Semesta pembicaraan pada variable *goals* adalah [0 36]. Sedangkan untuk domain himpunan *fuzzy*-nya adalah sebagai berikut:

Tabel 3.5: Domain Himpunan Keanggotaan Goals

Himpunan Fuzzy	Domain
Sedikit	0 - 18

Sedang	10 - 25
Banyak	18 - 36



Gambar 3.10: Himpunan Fuzzy untuk Variabel Goals

Pada himpunan Sedikit fungsi keanggotaan yang digunakan adalah fungsi representasi linear turun, sehingga untuk mendapatkan derajat keanggotaan pada himpunan Sedikit dengan menggunakan rumus:

$$\mu_{GoalsSedikit}[x] = \begin{cases} 1; & x \leq 6 \\ \frac{18-x}{18-6}; & 6 < x < 18 \\ 0; & x \geq 18 \end{cases} \quad (3.4)$$

Pada himpunan Sedang fungsi keanggotaan yang digunakan adalah fungsi representasi kurva segitiga, sehingga untuk mendapatkan derajat keanggotaan pada himpunan Sedang dengan menggunakan rumus:

$$\mu_{GoalsSedang}[x] = \begin{cases} 1; & x \leq 10 \\ \frac{x-10}{18-10}; & 10 \leq x \leq 18 \\ \frac{25-x}{25-18}; & 18 < x < 25 \\ 0; & x \leq 18 \parallel x \geq 25 \end{cases} \quad (3.5)$$

Pada himpunan Banyak fungsi keanggotaan yang digunakan adalah fungsi representasi linear naik, sehingga untuk mendapatkan derajat keanggotaan pada himpunan Banyak dengan menggunakan rumus:

$$\mu_{GoalsBanyak}[x] = \begin{cases} 0; & x \leq 18 \\ \frac{x-18}{29-18}; & 18 < x < 29 \\ 1; & x \geq 29 \end{cases} \quad (3.6)$$

### 3. Variabel Score

Variabel *score* terdapat 3 himpunan *fuzzy*, yaitu: Sedikit, Sedang, dan Banyak. Semesta pembicaraan pada variable *Score* adalah [0 100]. Sedangkan untuk domain himpunan *fuzzy*-nya adalah sebagai berikut:

Tabel 3.6: Domain Himpunan Keanggotaan *Score*

Himpunan <i>Fuzzy</i>	Domain
Sedikit	0 - 30
Sedang	20 - 40
Banyak	30 - 100



Gambar 3.11: Himpunan *Fuzzy* untuk Variabel *Score*

Pada himpunan Sedikit fungsi keanggotaan yang digunakan adalah fungsi representasi linear turun, sehingga untuk mendapatkan derajat keanggotaan pada himpunan Banyak dengan menggunakan rumus:

$$\mu_{HintSedikit}[x] = \begin{cases} 1; & x \leq 15 \\ \frac{30-x}{30-15}; & 15 < x < 30 \\ 0; & x \geq 30 \end{cases} \quad (3.7)$$

Pada himpunan Sedang fungsi keanggotaan yang digunakan adalah fungsi representasi kurva segitiga, sehingga untuk mendapatkan derajat keanggotaan pada himpunan Sedang dengan menggunakan rumus:

$$\mu_{HintSedang}[x] = \begin{cases} 1; & x \leq 20 \\ \frac{x-20}{30-20}; & 20 \leq x \leq 30 \\ \frac{40-x}{40-30}; & 30 \leq x \leq 40 \\ 0; & x \leq 30 \parallel x \geq 40 \end{cases} \quad (3.8)$$

Pada himpunan Banyak fungsi keanggotaan yang digunakan adalah fungsi representasi linear naik, sehingga untuk mendapatkan derajat keanggotaan pada himpunan Banyak dengan menggunakan rumus:

$$\mu_{HintBanyak}[x] = \begin{cases} 0; & x \leq 30 \\ \frac{x-30}{45-30}; & 30 < x < 45 \\ 1; & x \geq 45 \end{cases} \quad (3.9)$$

#### 4. Variabel Level

Variabel *level* terdapat 3 himpunan *fuzzy*, yaitu: Mudah, Normal, dan Sulit. Variabel ini merupakan variabel *output* atau keluaran pada proses *fuzzy* ini. Semua himpunan pada variabel *level* menggunakan fungsi keanggotaan *singleton*.

##### 3.4.3 Fuzzy Rules

Model *fuzzy sugeno* yang dipakai adalah model *fuzzy sugeno* orde-nol dengan komposisi aturan sebagai berikut:

[R1] *IF Moves is SEDIKIT and Goal is SEDIKIT and Score is SEDIKIT*

*THEN Level is MUDAH* (1)

- [R2] *IF Moves is SEDIKIT and Goal is SEDIKIT and Score is SEDANG  
THEN Level is MUDAH (1)*
- [R3] *IF Moves is SEDIKIT and Goal is SEDIKIT and Score is BANYAK  
THEN Level is NORMAL (2)*
- [R4] *IF Moves is SEDIKIT and Goal is SEDANG and Score is SEDIKIT  
THEN Level is MUDAH (1)*
- [R5] *IF Moves is SEDIKIT and Goal is SEDANG and Score is SEDANG  
THEN Level is MUDAH (1)*
- [R6] *IF Moves is SEDIKIT and Goal is SEDANG and Score is BANYAK  
THEN Level is MUDAH (1)*
- [R7] *IF Moves is SEDIKIT and Goal is BANYAK and Score is SEDIKIT  
THEN Level is NORMAL (2)*
- [R8] *IF Moves is SEDIKIT and Goal is BANYAK and Score is SEDANG  
THEN Level is NORMAL (2)*
- [R9] *IF Moves is SEDIKIT and Goal is BANYAK and Score is BANYAK  
THEN Level is SULIT (3)*
- [R10] *IF Moves is SEDANG and Goal is SEDIKIT and Score is SEDIKIT  
THEN Level is MUDAH (1)*
- [R11] *IF Moves is SEDANG and Goal is SEDIKIT and Score is SEDANG  
THEN Level is MUDAH (1)*
- [R12] *IF Moves is SEDANG and Goal is SEDIKIT and Score is BANYAK  
THEN Level is SULIT (3)*
- [R13] *IF Moves is SEDANG and Goal is SEDANG and Score is SEDIKIT  
THEN Level is MUDAH (1)*
- [R14] *IF Moves is SEDANG and Goal is SEDANG and Score is SEDANG  
THEN Level is MUDAH (1)*
- [R15] *IF Moves is SEDANG and Goal is SEDANG and Score is BANYAK  
THEN Level is SULIT (3)*
- [R16] *IF Moves is SEDANG and Goal is BANYAK and Score is SEDIKIT  
THEN Level is MUDAH (1)*
- [R17] *IF Moves is SEDANG and Goal is BANYAK and Score is SEDANG  
THEN Level is NORMAL (2)*

[R18] *IF Moves is SEDANG and Goal is BANYAK and Score is BANYAK  
THEN Level is SULIT (3)*

[R19] *IF Moves is BANYAK and Goal is SEDIKIT and Score is SEDIKIT  
THEN Level is NORMAL (2)*

[R20] *IF Moves is BANYAK and Goal is SEDIKIT and Score is SEDANG  
THEN Level is NORMAL (2)*

[R21] *IF Moves is BANYAK and Goal is SEDIKIT and Score is BANYAK  
THEN Level is SULIT (3)*

[R22] *IF Moves is BANYAK and Goal is SEDANG and Score is SEDIKIT  
THEN Level is NORMAL (2)*

[R23] *IF Moves is BANYAK and Goal is SEDANG and Score is SEDANG  
THEN Level is SULIT (3)*

[R24] *IF Moves is BANYAK and Goal is SEDANG and Score is BANYAK  
THEN Level is SULIT (3)*

[R25] *IF Moves is BANYAK and Goal is BANYAK and Score is SEDIKIT  
THEN Level is SULIT (3)*

[R26] *IF Moves is BANYAK and Goals is BANYAK and Score is SEDANG  
THEN Level is SULIT (3)*

[R27] *IF Moves is BANYAK and Goal is BANYAK and Score is BANYAK  
THEN Level is SULIT (3)*

#### 3.4.4 Implikasi

Pada *Fuzzy Sugeno*, fungsi implikasi yang digunakan adalah fungsi implikasi min (*minimum*). Fungsi implikasi min ini akan di aplikasikan terhadap setiap aturan yang telah dibuat. Pada fungsi implikasi min, *output* didapat dengan mengambil derajat keanggotaan terendah dari variabel *input*-nya. Secara umum fungsi ini dirumuskan sebagaimana berikut:

$$\alpha - \text{predikat}_i = \min(\mu \text{Moves}_i[x]; \mu \text{Goals}_i[x]; \mu \text{Hint}_i[x]) \quad (3.10)$$

Hasil dari fungsi implikasi pada setiap aturan akan digunakan untuk proses *defuzzifikasi*.

### 3.5.5 Defuzzyfikasi

Proses *defuzzyfikasi* pada metode *Sugeno* dilakukan dengan cara mencari rata-rata dari hasil fungsi implikasi pada komposisi aturan yang telah ditentukan, sehingga keluaran (*output*) akhirnya adalah sebuah *weighted average*. Secara umum proses *defuzzyfikasi* ini dirumuskan sebagai berikut:

$$z = \frac{\sum_{i=1}^n \alpha\text{-predikat}_i z_i}{\sum_{i=1}^n \alpha\text{-predikat}_i} \quad (3.11)$$

Di mana:

$z$  = Keluaran (*output*) akhir.

$n$  = Jumlah aturan.

$\alpha - \text{predikat}_i$  = *Fire strength* hasil fungsi implikasi aturan ke- $i$ .

$z_i$  = Keluaran (*output*) dari aturan ke- $i$ .

Nilai  $z$  yang berupa nilai desimal akan dibulatkan ke nilai satuan. Hasil pembulatan nilai  $z$  ini akan digunakan sebagai parameter penentuan *output level* dengan aturan sebagai berikut:

- Jika hasil pembulatan nilai  $z$  sama dengan 1 maka *level*-nya adalah Mudah
- Jika hasil pembulatan nilai  $z$  sama dengan 2 maka *level*-nya adalah Normal
- Jika hasil pembulatan nilai  $z$  sama dengan 3 maka *level*-nya adalah Sulit

## 3.6 Contoh Perhitungan Fuzzy Sugeno

### 3.6.5 Perhitungan Manual

Seorang pemain telah menyelesaikan permainan pada *level* 1: Primer. Dengan catatan semua *goals* telah terselesaikan dengan jumlah total 15 *item*, *moves* yang digunakan 10, dan hasil jumlah total 50 *score*. Apakah pemain dapat melanjutkan ke *level* berikutnya? Jika iya, di tingkat kesulitan manakah pemain akan bermain selanjutnya?

**a. Fuzzifikasi**

Yaitu memetakan nilai crisp dari *moves*, *goals* dan *score* ke dalam himpunan *Fuzzy* dan menentukan derajat keanggotaannya

**Derajat keanggotaan variabel Moves:**

Perhitungan variabel *moves* dengan nilai  $x = 10$  terhadap himpunan sedikit, himpunan sedang, dan himpunan banyak:

$$\mu_{\text{MovesSedikit}}[x] = 0; x \geq 9$$

$$\mu_{\text{MovesSedang}}[x] = \frac{12 - x}{12 - 9}; 6 \leq x \leq 12$$

$$\mu_{\text{MovesBanyak}}[x] = \frac{x - 9}{12 - 9}; 9 < x < 12$$

Sehingga hasil perhitungannya berdasarkan rumus linier turun, segitiga, dan linier naik di peroleh derajat keanggotaan moves sedikit, sedang, dan banyak adalah:

$$\mu_{\text{MovesSedikit}}[10] = 0$$

$$\mu_{\text{MovesSedang}}[10] = 0,6666$$

$$\mu_{\text{MovesBanyak}}[10] = 0,3333$$

**Derajat keanggotaan variabel Goal:**

Perhitungan variabel *goal* dengan nilai  $x = 15$  terhadap himpunan sedikit, himpunan sedang, dan himpunan banyak:

$$\mu_{\text{GoalSedikit}}[x] = \frac{18 - x}{18 - 6}; 6 < x < 18$$

$$\mu_{\text{GoalSedang}}[x] = \frac{x - 10}{18 - 10}; 10 \leq x \leq 18$$

$$\mu_{\text{GoalBanyak}}[x] = 1; x \leq 18$$

Sehingga hasil perhitungannya berdasarkan rumus linier turun, segitiga, dan linier naik di peroleh derajat keanggotaan moves sedikit, sedang, dan banyak adalah:

$$\mu_{\text{GoalSedikit}}[15] = 0.25$$

$$\mu_{\text{GoalSedang}}[15] = 0.625$$

$$\mu_{\text{GoalBanyak}}[15] = 1$$

**Derajat keanggotaan variabel Score:**

Perhitungan variabel *score* dengan nilai  $x = 50$  terhadap himpunan sedikit, himpunan sedang, dan himpunan banyak:

$$\mu_{ScoreSedikit}[x] = 0; x \geq 30$$

$$\mu_{ScoreSedang}[x] = 0; x \leq 30 \parallel x \geq 40$$

$$\mu_{ScoreBanyak}[x] = 1; x \geq 40$$

Sehingga hasil perhitungan berdasarkan rumus linier turun, segitiga, dan linier naik di peroleh derajat keanggotaan moves sedikit, sedang, dan banyak adalah:

$$\mu_{ScoreSedikit}[50] = 0$$

$$\mu_{ScoreSedang}[50] = 0$$

$$\mu_{ScoreBanyak}[50] = 1$$

**b. Implikasi**

Selanjutnya dilakukan penerapan fungsi implikasi min pada setiap aturan yang ada untuk mendapatkan nilai  $\alpha$  – **predikat** dari masing-masing aturan.

$$\begin{aligned} [R1] \alpha - predikat_1 &= \min(\mu_{MovesSedikit}[10]; \mu_{GoalSedikit}[15]; \\ &\quad \mu_{ScoreSedikit}[50]) \\ &= \min(0; 0,25; 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} [R2] \alpha - predikat_2 &= \min(\mu_{MovesSedikit}[10]; \mu_{GoalSedikit}[15]; \\ &\quad \mu_{ScoreSedang}[50]) \\ &= \min(0; 0,25; 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} [R3] \alpha - predikat_3 &= \min(\mu_{MovesSedikit}[10]; \mu_{GoalSedikit}[15]; \\ &\quad \mu_{ScoreBanyak}[50]) \\ &= \min(0; 0,25; 1) \\ &= 0 \end{aligned}$$

$$\begin{aligned} [R4] \alpha - predikat_4 &= \min(\mu_{MovesSedikit}[10]; \mu_{GoalSedang}[15]; \\ &\quad \mu_{ScoreSedikit}[50]) \\ &= \min(0; 0,625; 0) \end{aligned}$$

$$= 0$$

$$\begin{aligned} [R5] \alpha - predikat_5 &= \min(\mu MovesSedikit[10]; \mu GoalSedang[15]; \\ &\quad \mu ScoreSedang[50]) \\ &= \min(0; 0,625; 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} [R6] \alpha - predikat_6 &= \min(\mu MovesSedikit[10]; \mu GoalSedang[15]; \\ &\quad \mu ScoreBanyak[50]) \\ &= \min(0; 0,625; 1) \\ &= 0 \end{aligned}$$

$$\begin{aligned} [R7] \alpha - predikat_7 &= \min(\mu MovesSedikit[10]; \mu GoalBanyak[15]; \\ &\quad \mu ScoreSedikit[50]) \\ &= \min(0; 1; 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} [R8] \alpha - predikat_8 &= \min(\mu MovesSedikit[10]; \mu GoalBanyak[15]; \\ &\quad \mu ScoreSedang[50]) \\ &= \min(0; 1; 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned} [R9] \alpha - predikat_9 &= \min(\mu MovesSedikit[10]; \mu GoalBanyak[15]; \\ &\quad \mu ScoreBanyak[50]) \\ &= \min(0; 1; 1) \\ &= 0 \end{aligned}$$

$$\begin{aligned} [R10] \alpha - predikat_{10} &= \min(\mu MovesSedang[10]; \mu GoalSedikit[15]; \\ &\quad \mu ScoreSedikit[50]) \\ &= \min(0,6666; 0,25; 0) \\ &= 0 \end{aligned}$$

$$\begin{aligned}
 [\text{R11}] \alpha - \text{predikat}_{11} &= \min(\mu\text{MovesSedang}[10]; \mu\text{GoalSedikit}[15]; \\
 &\quad \mu\text{ScoreSedang}[50]) \\
 &= \min(0,6666; 0,25; 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R12}] \alpha - \text{predikat}_{12} &= \min(\mu\text{MovesSedang}[10]; \mu\text{GoalSedikit}[15]; \\
 &\quad \mu\text{ScoreBanyak}[50]) \\
 &= \min(0,6666; 0,25; 1) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R13}] \alpha - \text{predikat}_{13} &= \min(\mu\text{MovesSedang}[10]; \mu\text{GoalSedang}[15]; \\
 &\quad \mu\text{ScoreSedikit}[50]) \\
 &= \min(0,6666; 0,625; 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R14}] \alpha - \text{predikat}_{14} &= \min(\mu\text{MovesSedang}[10]; \mu\text{GoalSedang}[15]; \\
 &\quad \mu\text{ScoreSedang}[50]) \\
 &= \min(0,6666; 0,625; 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R15}] \alpha - \text{predikat}_{15} &= \min(\mu\text{MovesSedang}[10]; \mu\text{GoalSedang}[15]; \\
 &\quad \mu\text{ScoreBanyak}[50]) \\
 &= \min(0,6666; 0,625; 1) \\
 &= 0,625
 \end{aligned}$$

$$\begin{aligned}
 [\text{R16}] \alpha - \text{predikat}_{16} &= \min(\mu\text{MovesSedang}[10]; \mu\text{GoalBanyak}[15]; \\
 &\quad \mu\text{ScoreSedikit}[50]) \\
 &= \min(0,6666; 1; 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R17}] \alpha - \text{predikat}_{17} &= \min(\mu\text{MovesSedang}[10]; \mu\text{GoalBanyak}[15]; \\
 &\quad \mu\text{ScoreSedang}[50])
 \end{aligned}$$

$$= \min(0,6666; 1; 0)$$

$$= 0$$

$$[R18] \alpha - \text{predikat}_{18} = \min(\mu\text{MovesSedang}[10]; \mu\text{GoalBanyak}[15];$$

$$\mu\text{ScoreBanyak}[50])$$

$$= \min(0,6666; 1; 1)$$

$$= 0,6666$$

$$[R19] \alpha - \text{predikat}_{19} = \min(\mu\text{MovesBanyak}[10]; \mu\text{GoalSedikit}[15];$$

$$\mu\text{ScoreSedikit}[50])$$

$$= \min(0,3333; 0,25; 0)$$

$$= 0$$

$$[R20] \alpha - \text{predikat}_{20} = \min(\mu\text{MovesBanyak}[10]; \mu\text{GoalSedikit}[15];$$

$$\mu\text{ScoreSedang}[50])$$

$$= \min(0,3333; 0,25; 0)$$

$$= 0$$

$$[R21] \alpha - \text{predikat}_{21} = \min(\mu\text{MovesBanyak}[10]; \mu\text{GoalSedikit}[15];$$

$$\mu\text{ScoreBanyak}[50])$$

$$= \min(0,3333; 0,25; 1)$$

$$= 0,25$$

$$[R22] \alpha - \text{predikat}_{22} = \min(\mu\text{MovesBanyak}[10]; \mu\text{GoalSedang}[15];$$

$$\mu\text{ScoreSedikit}[50])$$

$$= \min(0,3333; 0,625; 0)$$

$$= 0$$

$$[R23] \alpha - \text{predikat}_{23} = \min(\mu\text{MovesBanyak}[10]; \mu\text{GoalSedang}[15];$$

$$\mu\text{ScoreSedang}[50])$$

$$= \min(0,3333; 0,625; 0)$$

$$= 0$$

$$\begin{aligned}
 [\text{R24}] \alpha - \text{predikat}_{24} &= \min(\mu\text{MovesBanyak}[10]; \mu\text{GoalSedang}[15]; \\
 &\quad \mu\text{ScoreBanyak}[50]) \\
 &= \min(0,3333; 0,625; 1) \\
 &= 0,3333
 \end{aligned}$$

$$\begin{aligned}
 [\text{R25}] \alpha - \text{predikat}_{25} &= \min(\mu\text{MovesBanyak}[10]; \mu\text{GoalBanyak}[15]; \\
 &\quad \mu\text{ScoreSedikit}[50]) \\
 &= \min(0,3333; 1; 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R26}] \alpha - \text{predikat}_{26} &= \min(\mu\text{MovesBanyak}[10]; \mu\text{GoalsBanyak}[15]; \\
 &\quad \mu\text{ScoreSedang}[50]) \\
 &= \min(0,3333; 1; 0) \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 [\text{R27}] \alpha - \text{predikat}_{27} &= \min(\mu\text{MovesBanyak}[10]; \mu\text{GoalBanyak}[15]; \\
 &\quad \mu\text{ScoreBanyak}[50]) \\
 &= \min(0,3333; 1; 1) \\
 &= 0,3333
 \end{aligned}$$

### c. Defuzzifikasi

Pada tahap ini menentukan variabel linguisitik keputusan dari setiap *rules* yang telah dibuat. Setelah nilai  $\alpha - \text{predikat}$  dari masing-masing aturan sudah didapat, maka selanjutnya dilakukan proses *defuzzy* untuk mendapatkan nilai  $z$ .

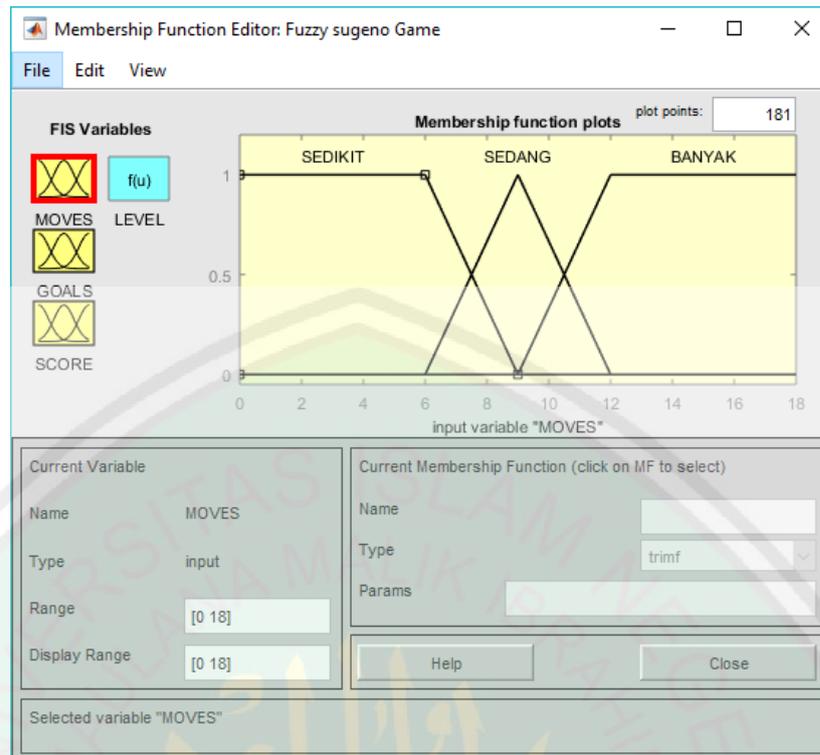
$$z = \frac{\sum_{i=1}^n \alpha - \text{predikat}_i z_i}{\sum_{i=1}^n \alpha - \text{predikat}_i} = \frac{\sum_{i=1}^{27} \alpha - \text{predikat}_i z_i}{\sum_{i=1}^{27} \alpha - \text{predikat}_i}$$

$$\begin{aligned}
& \alpha - \text{predikat}_1 * z_1 + \alpha - \text{predikat}_2 * z_2 + \alpha - \text{predikat}_3 * z_3 + \\
& \alpha - \text{predikat}_4 * z_4 + \alpha - \text{predikat}_5 * z_5 + \alpha - \text{predikat}_6 * z_6 + \\
& \alpha - \text{predikat}_7 * z_7 + \alpha - \text{predikat}_8 * z_8 + \alpha - \text{predikat}_9 * z_9 + \\
& \alpha - \text{predikat}_{10} * z_{10} + \alpha - \text{predikat}_{11} * z_{11} + \alpha - \text{predikat}_{12} * z_{12} + \\
& \alpha - \text{predikat}_{13} * z_{13} + \alpha - \text{predikat}_{14} * z_{14} + \alpha - \text{predikat}_{15} * z_{15} + \\
& \alpha - \text{predikat}_{16} * z_{16} + \alpha - \text{predikat}_{17} * z_{17} + \alpha - \text{predikat}_{18} * z_{18} + \\
& \alpha - \text{predikat}_{19} * z_{19} + \alpha - \text{predikat}_{20} * z_{20} + \alpha - \text{predikat}_{21} * z_{21} + \\
& \alpha - \text{predikat}_{22} * z_{22} + \alpha - \text{predikat}_{23} * z_{23} + \alpha - \text{predikat}_{24} * z_{24} + \\
& \alpha - \text{predikat}_{25} * z_{25} + \alpha - \text{predikat}_{26} * z_{26} + \alpha - \text{predikat}_{27} * z_{27} \\
= & \frac{\alpha - \text{predikat}_1 + \alpha - \text{predikat}_2 + \alpha - \text{predikat}_3 + \alpha - \text{predikat}_4 + \\
& \alpha - \text{predikat}_5 + \alpha - \text{predikat}_6 + \alpha - \text{predikat}_7 + \alpha - \text{predikat}_8 + \\
& \alpha - \text{predikat}_9 + \alpha - \text{predikat}_{10} + \alpha - \text{predikat}_{11} + \alpha - \text{predikat}_{12} + \\
& \alpha - \text{predikat}_{13} + \alpha - \text{predikat}_{14} + \alpha - \text{predikat}_{15} + \alpha - \text{predikat}_{16} + \\
& \alpha - \text{predikat}_{17} + \alpha - \text{predikat}_{18} + \alpha - \text{predikat}_{19} + \alpha - \text{predikat}_{20} + \\
& \alpha - \text{predikat}_{21} + \alpha - \text{predikat}_{22} + \alpha - \text{predikat}_{23} + \alpha - \text{predikat}_{24} + \\
& \alpha - \text{predikat}_{25} + \alpha - \text{predikat}_{26} + \alpha - \text{predikat}_{27}}{0 * 1 + 0 * 1 + 0 * 2 + 0 * 1 + 0 * 1 + 0 * 1 + \\
& 0 * 2 + 0 * 2 + 0 * 3 + 0 * 1 + 0 * 1 + 0 * 3 + \\
& 0 * 1 + 0 * 1 + 0,625 * 3 + 0 * 1 + 0 * 2 + 0,66666 * 3 + 0 * 2 + \\
& 0 * 2 + 0,25 * 3 + 0 * 2 + 0 * 3 + 0,3333 * 3 + 0 * 3 + 0 * 3 + 0,3333 * 3} \\
= & \frac{0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + \\
& 0 + 0 + 0,625 + 0 + 0 + 0,6666 + 0 + 0 + 0,25 + 0 + \\
& 0 + 0,3333 + 0 + 0 + 0,3333}{0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + \\
& 0 + 0 + 0,625 + 0 + 0 + 0,6666 + 0 + 0 + 0,25 + 0 + \\
& 0 + 0,3333 + 0 + 0 + 0,3333} \\
= & \frac{6,6246}{2,2082} \\
= & 3
\end{aligned}$$

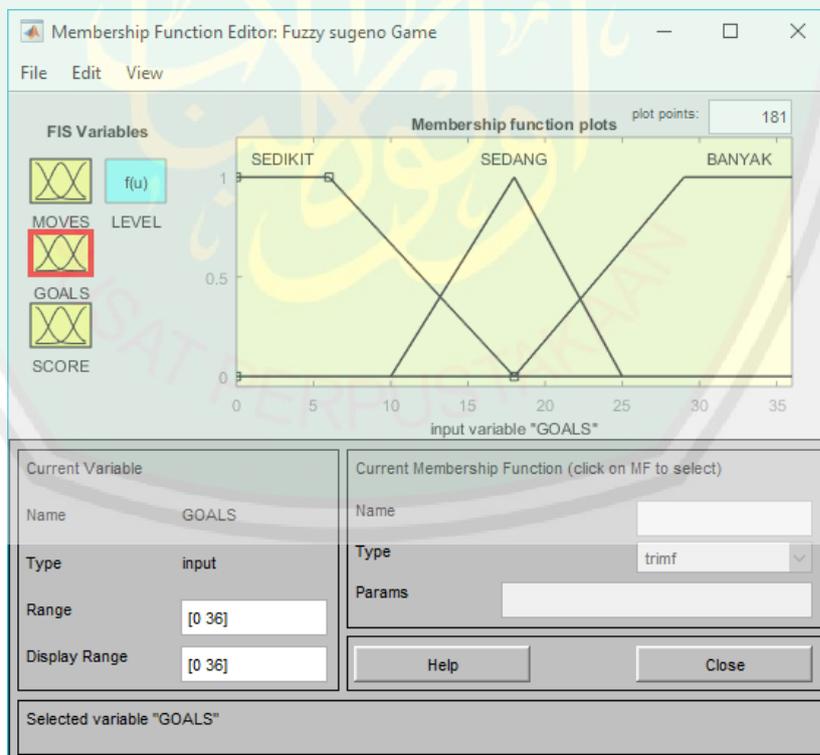
Sesuai dengan aturan penentuan *level* yang telah dibuat, hasil nilai  $z$  adalah 3 maka pemain dapat bermain ke *level* selanjutnya dengan tingkat kesulitan adalah sulit.

### 3.6.6 Perhitungan Matlab

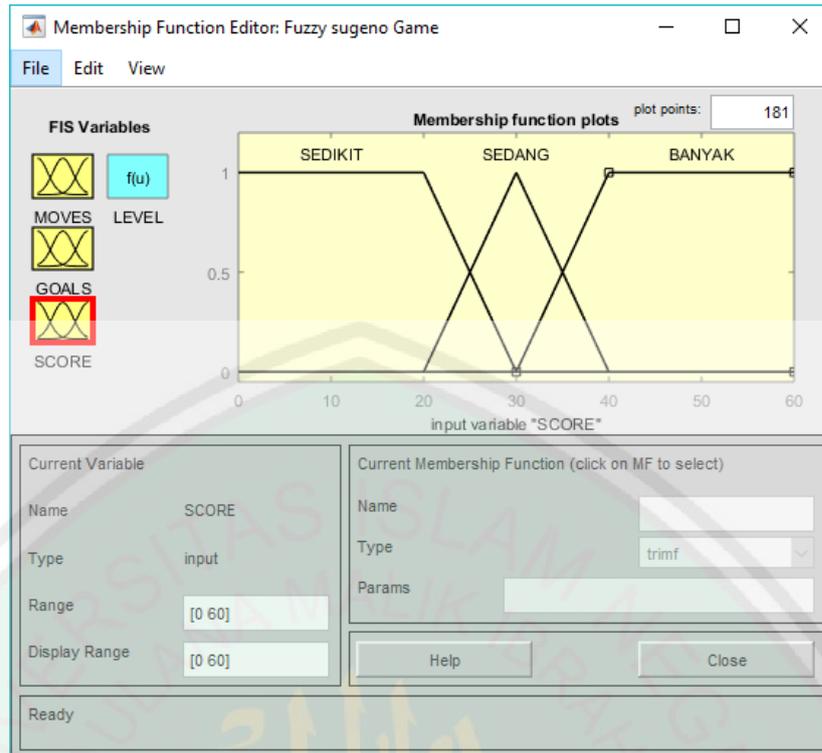
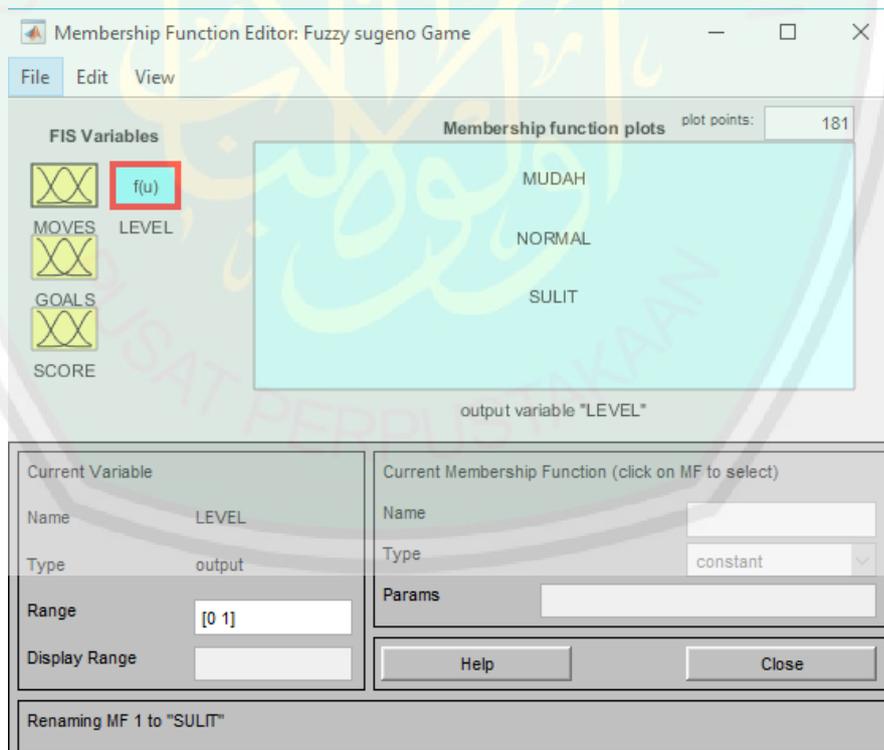
Selain mencoba dengan perhitungan manual, juga dilakukan dengan menggunakan aplikasi Matlab. Pada aplikasi Matlab terdapat *Fuzzy Logic Tool* yang bisa digunakan untuk simulasi penerapan *Fuzzy Sugeno* ini.

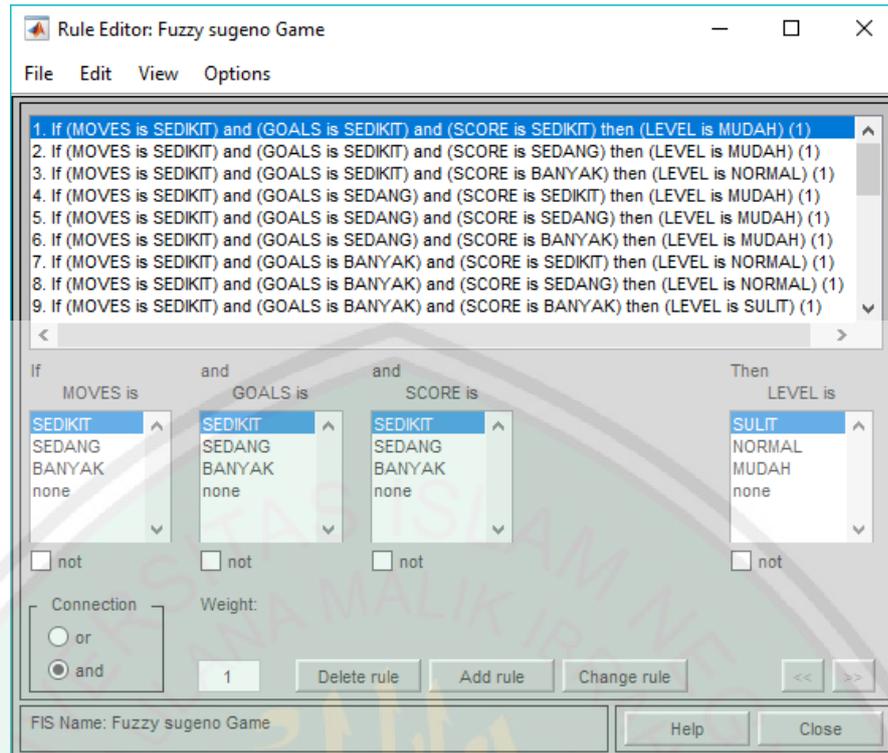


Gambar 3.12: Matlab: Himpunan *Fuzzy* untuk Variabel *Moves*

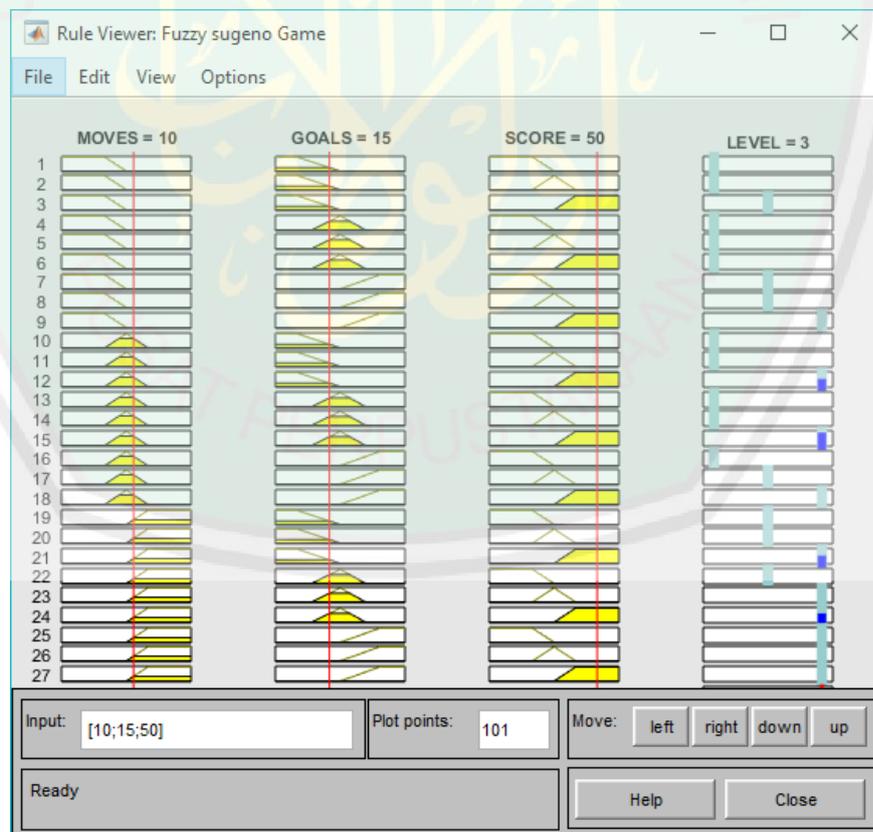


Gambar 3.13: Matlab: Himpunan *Fuzzy* untuk Variabel *Goal*

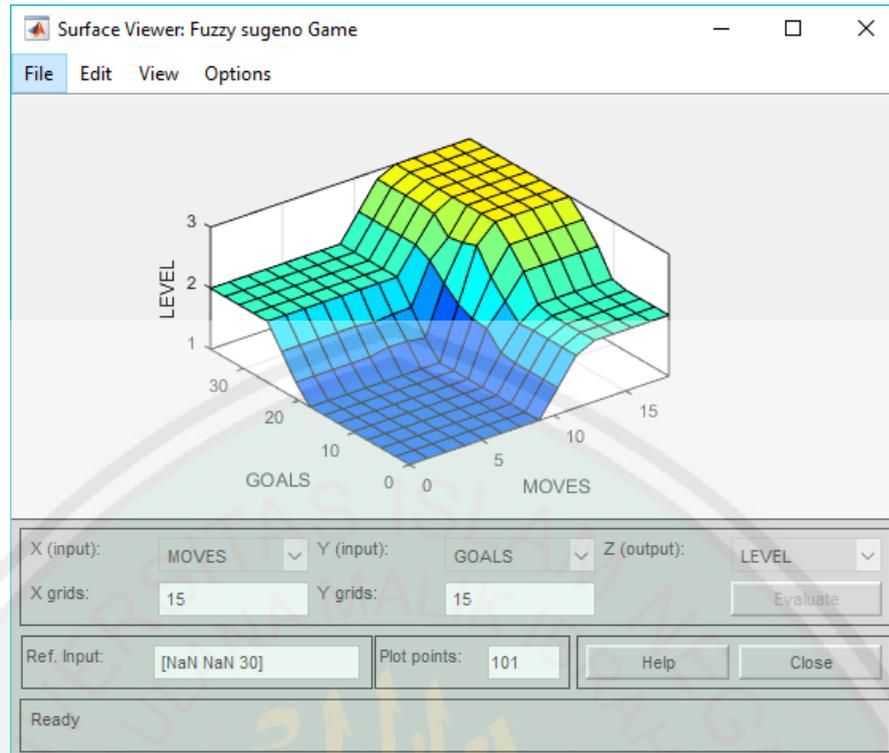
Gambar 3.14: Matlab: Himpunan *Fuzzy* untuk Variabel *Score*Gambar 3.15: Matlab: Himpunan *Fuzzy* untuk Variabel *Level*



Gambar 3.16: Matlab: Fuzzy Rule



Gambar 3.17: Matlab: Hasil

Gambar 3.18: Matlab: Grafik Permukaan *Level*

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Implementasi Desain Sistem

Pada bab ini menjelaskan mengenai implementasi algoritma fuzzy sugeno dari perencanaan yang telah dibuat. Pengujian dilakukan untuk mengetahui apakah permainan berjalan sesuai dengan tujuan penelitian yang diharapkan. Sebelum diimplementasikan, terlebih dahulu dijelaskan spesifikasi sistem perangkat keras (*hardware*) dan perangkat lunak (*software*).

#### 4.1.1 Kebutuhan Perangkat Keras

Perangkat keras yang diperlukan untuk mengimplementasikan perangkat lunak dari aplikasi permainan ini, yakni:

Tabel 4.1: Kebutuhan Perangkat Keras

No.	Perangkat Keras	Spesifikasi
1.	<i>Processor</i>	Intel® Core™ i3-3110M CPU @ 2.40Ghz
2.	<i>RAM</i>	4.00 GB
3.	<i>HDD</i>	500 GB
4.	<i>Monitor</i>	14'
7.	<i>Speaker</i>	On
8.	<i>Mouse &amp; Keyboard</i>	On

#### 4.1.2 Kebutuhan Perangkat Lunak

Perangkat lunak yang diperlukan untuk mengimplementasikan permainan *Coloration* pada tabel 4.2 berikut.

Tabel 4.2: Kebutuhan Perangkat Lunak

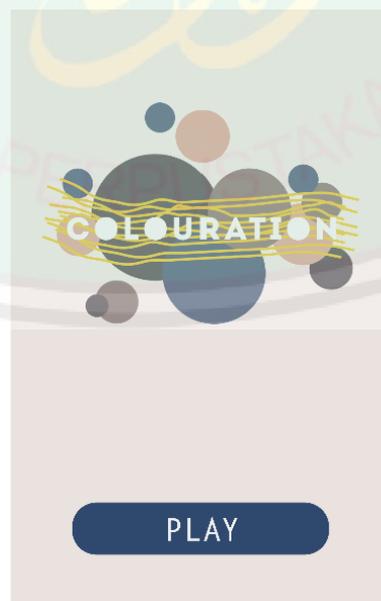
No.	Perangkat Keras	Spesifikasi
1.	Sistem Operasi	<i>Windows 10 Pro</i>
2.	<i>Game Engine</i>	<i>Unity 2017 2.3</i>
3.	Desain 2D	<i>Corel Draw X7</i>
4.	<i>Script Writer</i>	<i>MonoDevelop</i>

## 4.2 Implementasi Permainan

Pada bab pembahasan ini merupakan hasil implementasi dari rancangan aplikasi permainan *Colouration* ke dalam perangkat *mobile* berbasis *android* dan penjelasan tiap *scene*. Berikut adalah tampilan hasil implementasi yang telah selesai dibuat.

### 4.2.1 Tampilan Main Menu

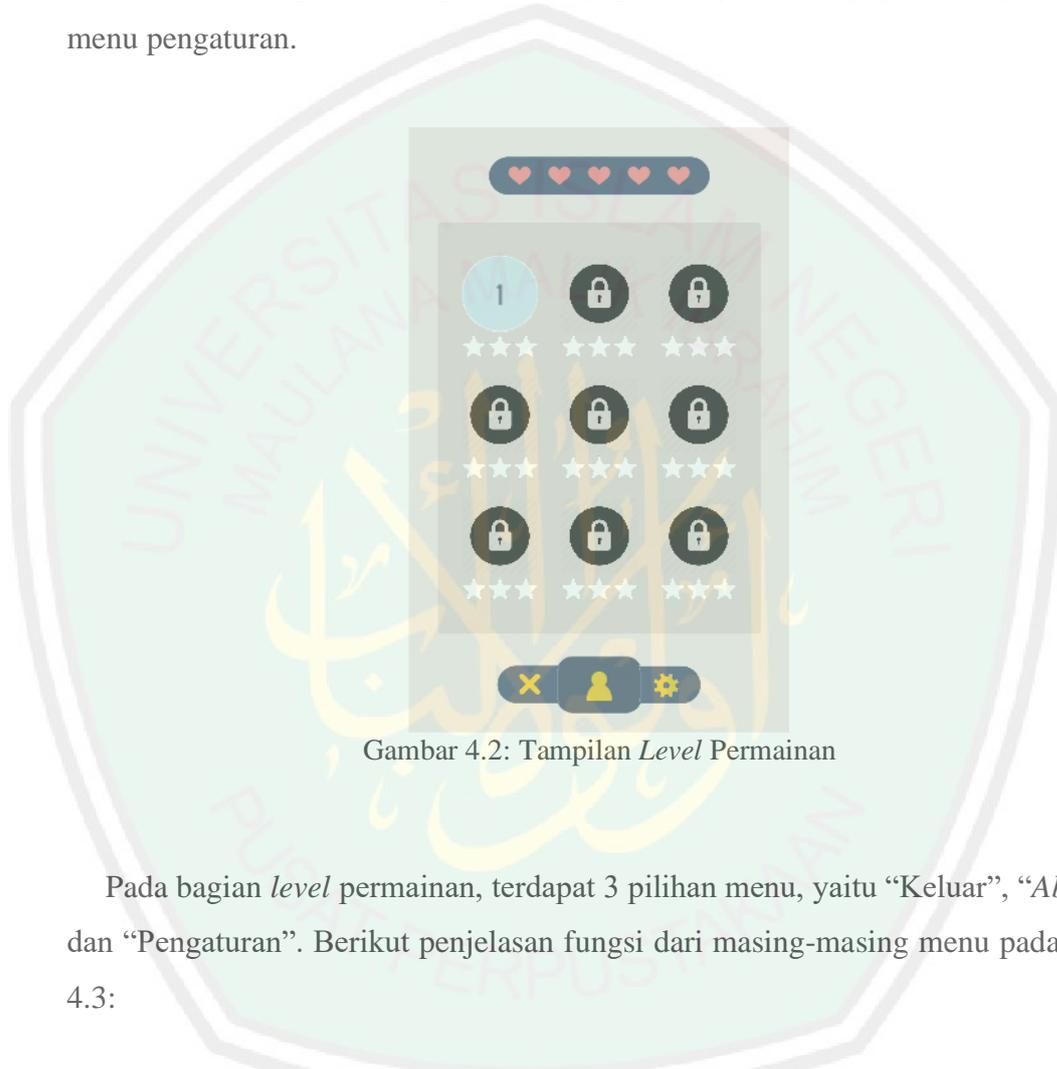
Pada saat memulai permainan, bagian yang pertama kali tampil adalah main menu. Seperti gambar 4.1, pada main menu terdapat pilihan tombol *Play* yang berfungsi untuk memulai permainan. Ketika tombol *Play* di klik maka pemain di arahkan menuju ke *level* permainan.



Gambar 4.1: Tampilan Main Menu

#### 4.2.2 Tampilan *Level Permainan*

Setelah menekan tombol *Play* pada main menu pemain akan memasuki bagian *level* permainan, dimana pada bagian ini pemain hanya dapat bermain di *level* 1 terlebih dahulu kemudian naik ke *level* 2, 3, dan seterusnya. *Level* 2 sampai *level* 9 terkunci dan akan terbuka apabila pemain berhasil menyelesaikan semua misi di *level* 1. Pada tampilan *level* permainan juga terdapat bintang, jumlah nyawa, dan menu pengaturan.



Gambar 4.2: Tampilan *Level Permainan*

Pada bagian *level* permainan, terdapat 3 pilihan menu, yaitu “Keluar”, “About”, dan “Pengaturan”. Berikut penjelasan fungsi dari masing-masing menu pada tabel 4.3:

Tabel 4.3: Keterangan Menu pada *Level Permainan*

No.	Menu	Keterangan
1.		Berfungsi untuk keluar dari <i>level</i> permainan

2.		Berfungsi untuk menampilkan informasi <i>credit game</i>
3.		Berfungsi untuk ke bagian pengaturan permainan <i>Colouration</i>

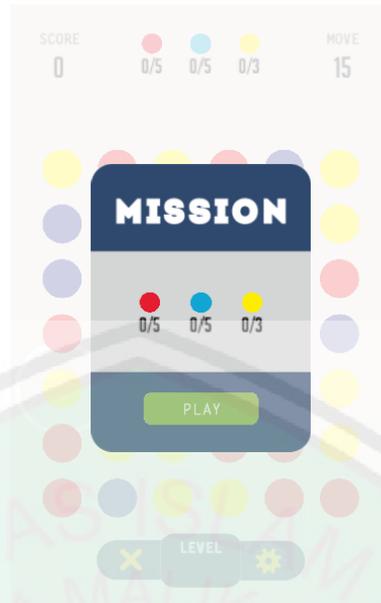
Ketika pemain memilih *level 1*, maka akan muncul scene seperti gambar 4.3 yang menampilkan informasi meliputi nilai dari total *score* permainan, jumlah bintang dari total keseluruhan score dan misi permainan yang tercapai, tombol keluar, dan tombol *play*.



Gambar 4.3: Tampilan *Scene Nilai*

#### 4.2.3 Tampilan Permainan

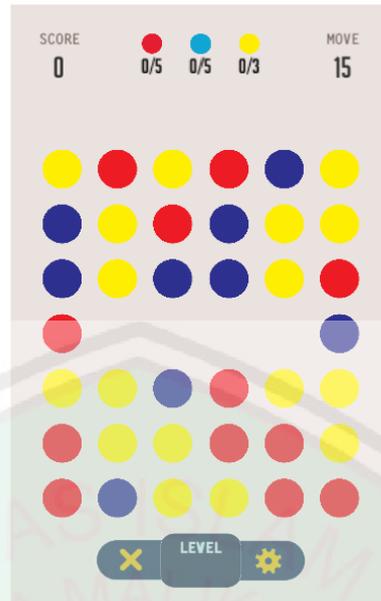
Sebelum pemain menjalankan permainan terdapat tampilan gambar 4.4 yang menampilkan informasi tentang misi berupa *dot* beserta jumlah yang harus diselesaikan pada permainan tersebut.



Gambar 4.4: Tampilan *Scene Mission*

Setelah pemain memilih tombol *play*, pemain akan dihadapkan dengan misi *item-item dot* yang harus diselesaikan. Untuk menyelesaikannya, pemain hanya memilih *item dot* yang berwarna sama dengan minimal jumlah *dot* sebanyak dua *item*. Pemain dapat menggeser *item dot* ke arah horizontal maupun vertikal, namun tidak bisa menggeser *item* secara diagonal. Ketika pemain menggeser *item dot*, secara otomatis *score* akan menjumlah banyaknya *item dot* yang menghilang dengan nilai per-*item dot* sebanyak 1. Ketika pemain dapat menghilang banyak *item dot* akan ada bonus. Apabila pemain tidak kunjung menemukan *item* yang berwarna sama, *game* otomatis akan memberikan *hint* pada *item dot* yang berwarna sama setelah pemain tidak menemukannya selama 1 menit.

Pada pojok atas kiri *game* terdapat *score* dan di pojok atas kanan terdapat *moves*. Pemain hanya dapat bermain tidak lebih dari jumlah *moves* yang telah diberikan.



Gambar 4.5: Tampilan Permainan

Pada bagian *level* permainan, terdapat 2 pilihan menu, yaitu “Keluar” dan “Pengaturan” serta keterangan di *level* mana pemain sedang bermain. Fungsi dari menu keluar adalah untuk keluar dari permainan dan kembali ke tampilan *level* permainan, sedangkan di menu pengaturan terdapat tombol musik yang dapat di aktifkan atau di nonaktifkan jika pemain tidak ingin ada latar belakang musik saat bermain *game*.



Gambar 4.6: Tampilan Tombol Menu Musik Aktif



Gambar 4.7: Tampilan Tombol Menu Musik Nonaktif

#### 4.2.4 Tampilan *Game Success*

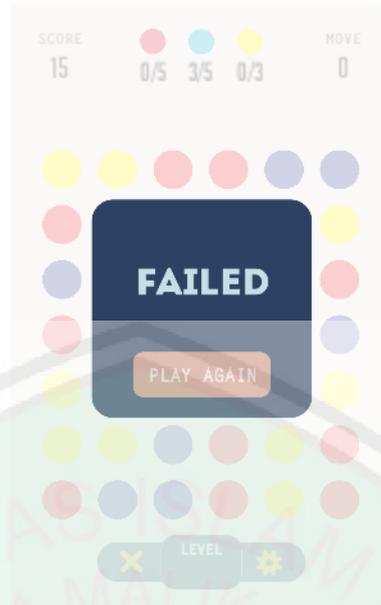
Ketika pemain berhasil menyelesaikan misi yang telah diberikan maka gambar 4.8 akan muncul. Pemain dapat memilih tombol *continue* untuk melanjutkan permainan ke *level* selanjutnya.



Gambar 4.8: Tampilan *Game Success*

#### 4.2.5 Tampilan *Game Failed*

Namun ketika pemain tidak dapat menyelesaikan misi maka gambar 4.9 akan muncul dan pemain tidak dapat melanjutkan ke *level* selanjutnya. Pemain dapat memilih tombol *play again* untuk memainkan kembali *game* di *level* tersebut.



Gambar 4.9: Tampilan *Game Failed*

### 4.3 Implementasi Algoritma *Fuzzy Sugeno*

Berdasarkan rancangan *fuzzy* yang telah dibuat pada permainan *Colouration*, algoritma *Fuzzy Sugeno* di implementasikan untuk menentukan *level* permainan. Variable yang digunakan ada empat, tiga variabel *input* dan satu variabel *output*. Variabel *input* terdiri dari batas gerak (*Move*), jumlah misi (*Goal*), dan nilai (*Score*) sedangkan variabel *output* adalah *level*.

Proses implementasi *Fuzzy Sugeno* dimulai dengan mencari derajat keanggotaan variabel *move* terhadap himpunan *fuzzy* variabel *move* yang telah dirancang sebelumnya. Himpunan *fuzzy* pada variabel *move* adalah himpunan '*moveSedikit*', himpunan '*moveSedang*', dan himpunan '*moveBanyak*'. Berikut implementasinya:

```
public static float[] Move(float move, float range) {
    float moveSedikit = 0, moveSedang = 0, moveBanyak = 0;
    float[] rangeSedikit = Sedikit(range);
    float[] rangeSedang = Sedang(range);
    float[] rangeBanyak = Banyak(range);
}
```

```

//sedikit
if (move <= rangeSedikit[0]) {
    moveSedikit = 1;
} else if (rangeSedikit[0] <= move && move <= rangeSedikit[1]) {
    moveSedikit = (rangeSedikit[1] - move) / (rangeSedikit[1] -
    rangeSedikit[0]);
} else if (move >= rangeSedikit[1]) {
    moveSedikit = 0;
}

//sedang
if (move <= rangeSedang[0]) {
    moveSedang = 1;} else if (rangeSedang[0] <= move && move <=
    rangeSedang[1]) {
    moveSedang = (move - rangeSedang[0]) / (rangeSedang[1] -
    rangeSedang[0]);
} else if (rangeSedang[0] <= move && move <= rangeSedang[2]) {
    moveSedang = (rangeSedang[2] - move) / (rangeSedang[2] -
    rangeSedang[1]);
} else if (move <= rangeSedang[0] || move >= rangeSedang[2]) {
    moveSedang = 0;
}

//banyak
if (move <= rangeBanyak[0]) {
    moveBanyak = 0;
} else if (rangeBanyak[0] <= move && move <= rangeBanyak[1]) {
    moveBanyak = (move - rangeBanyak[0]) / (rangeBanyak[1] -
    rangeBanyak[0]);
} else if (move >= rangeBanyak[1]) {
    moveBanyak = 1;
}

float[] hasil = {moveSedikit, moveSedang, moveBanyak};
return hasil;
}

```

Gambar 4.10: Source Code Derajat Keanggotaan Variabel *Moves*

Kemudian mencari derajat keanggotaan variabel *goal* terhadap himpunan *fuzzy* variabel *goal* yang telah dirancang sebelumnya. Himpunan *fuzzy* pada variabel *goal* adalah himpunan ‘*goalSedikit*’, himpunan ‘*goalSedang*’, dan himpunan ‘*goalBanyak*’. Berikut implementasinya:

```

public static float[] Goal(float goal, float range) {
    float goalSedikit = 0, goalSedang = 0, goalBanyak = 0;
    float[] rangeSedikit = Sedikit(range);
    float[] rangeSedang = Sedang(range);
    float[] rangeBanyak = Banyak(range);

    //sedikit
    if (goal <= rangeSedikit[0]) {
        goalSedikit = 1;
    } else if (rangeSedikit[0] <= goal && goal <= rangeSedikit[1]) {
        goalSedikit = (rangeSedikit[1] - goal) / (rangeSedikit[1] -
        rangeSedikit[0]);
    } else if (goal >= rangeSedikit[1]) {
        goalSedikit = 0;
    }

    // sedang
    if (goal <= rangeSedang[0]) {
        goalSedang = 1;
    } else if (rangeSedang[0] <= goal && goal <= rangeSedang[1]) {
        goalSedang = (goal - rangeSedang[0]) / (rangeSedang[1] -
        rangeSedang[0]);
    } else if (rangeSedang[0] <= goal && goal <= rangeSedang[2]) {
        goalSedang = (rangeSedang[2] - goal) / (rangeSedang[2] -
        rangeSedang[1]);
    } else if (goal <= rangeSedang[0] || goal >= rangeSedang[2]) {
        goalSedang = 0;
    }

    // banyak
    if (goal <= rangeBanyak[0]) {
        goalBanyak = 0;
    } else if (rangeBanyak[0] <= goal && goal <= rangeBanyak[1]) {
        goalBanyak = (goal - rangeBanyak[0]) / (rangeBanyak[1] -
        rangeBanyak[0]);
    } else if (goal >= rangeBanyak[1]) {
        goalBanyak = 1;
    }

    float[] hasil = {goalSedikit, goalSedang, goalBanyak};
    return hasil;
}

```

Gambar 4.11: Source Code Derajat Keanggotaan Variabel Goal

Kemudian mencari derajat keanggotaan variabel *score* terhadap himpunan *fuzzy* variabel *score* yang telah dirancang sebelumnya. Himpunan *fuzzy* pada variabel *score* adalah himpunan ‘*scoreSedikit*’, himpunan ‘*scoreSedang*’, dan himpunan ‘*scoreBanyak*’. Berikut implementasinya:

```
public static float[] Score(float score, float range) {
    float scoreSedikit = 0, scoreSedang = 0, scoreBanyak = 0;
    float[] rangeSedikit = Sedikit(range);
    float[] rangeSedang = Sedang(range);
    float[] rangeBanyak = Banyak(range);

    // sedikit
    if (score <= rangeSedikit[0]) {
        scoreSedikit = 1;
    } else if (rangeSedikit[0] <= score && score <= rangeSedikit[1]) {
        scoreSedikit = (rangeSedikit[1] - score) / (rangeSedikit[1] -
            rangeSedikit[0]);
    } else if (score >= rangeSedikit[1]) {
        scoreSedikit = 0;
    }

    // sedang
    if (score <= rangeSedang[0]) {
        scoreSedang = 1;
    } else if (rangeSedang[0] <= score && score <= rangeSedang[1]) {
        scoreSedang = (score - rangeSedang[0]) / (rangeSedang[1] -
            rangeSedang[0]);
    } else if (rangeSedang[0] <= score && score <= rangeSedang[2]) {
        scoreSedang = (rangeSedang[2] - score) / (rangeSedang[2] -
            rangeSedang[1]);
    } else if (score <= rangeSedang[0] || score >= rangeSedang[2]) {
        scoreSedang = 0;
    }

    // banyak
    if (score <= rangeBanyak[0]) {
        scoreBanyak = 0;
    } else if (rangeBanyak[0] <= score && score <= rangeBanyak[1]) {
        scoreBanyak = (score - rangeBanyak[0]) / (rangeBanyak[1] -
            rangeBanyak[0]);
    } else if (score >= rangeBanyak[1]) {
        scoreBanyak = 1;
    }
}
```

```

float[] hasil = {scoreSedikit, scoreSedang, scoreBanyak};
return hasil;
}

```

Gambar 4.12: *Source Code* Derajat Keanggotaan Variabel *Score*

Setelah mendapatkan nilai derajat keanggotaan dari ketiga variabel, selanjutnya mencari nilai  $\alpha$ -predikat yang dimasukkan ke dalam *fuzzy rules* dengan cara menerapkan fungsi implikasi MIN pada setiap aturan. kemudian akan dilakukan proses *defuzzifikasi* yang mana akan menentukan aksi pada setiap *fuzzy rules*.

```

public static int Rules(float moveSedikit, float moveSedang, float
moveBanyak, float goalSedikit, float goalSedang, float goalBanyak,
float scoreSedikit, float scoreSedang, float scoreBanyak) {
float[] minimal = new float[27];
minimal[0] = Min(moveSedikit, goalSedikit, scoreSedikit);
minimal[1] = Min(moveSedikit, goalSedikit, scoreSedang);
minimal[2] = Min(moveSedikit, goalSedikit, scoreBanyak);
minimal[3] = Min(moveSedikit, goalSedang, scoreSedikit);
minimal[4] = Min(moveSedikit, goalSedang, scoreSedang);
minimal[5] = Min(moveSedikit, goalSedang, scoreBanyak);
minimal[6] = Min(moveSedikit, goalBanyak, scoreSedikit);
minimal[7] = Min(moveSedikit, goalBanyak, scoreSedang);
minimal[8] = Min(moveSedikit, goalBanyak, scoreBanyak);
minimal[9] = Min(moveSedang, goalSedikit, scoreSedikit);
minimal[10] = Min(moveSedang, goalSedikit, scoreSedang);
minimal[11] = Min(moveSedang, goalSedikit, scoreBanyak);
minimal[12] = Min(moveSedang, goalSedang, scoreSedikit);
minimal[13] = Min(moveSedang, goalSedang, scoreSedang);
minimal[14] = Min(moveSedang, goalSedang, scoreBanyak);
minimal[15] = Min(moveSedang, goalBanyak, scoreSedikit);
minimal[16] = Min(moveSedang, goalBanyak, scoreSedang);
minimal[17] = Min(moveSedang, goalBanyak, scoreBanyak);
minimal[18] = Min(moveBanyak, goalSedikit, scoreSedikit);
minimal[19] = Min(moveBanyak, goalSedikit, scoreSedang);
minimal[20] = Min(moveBanyak, goalSedikit, scoreBanyak);
minimal[21] = Min(moveBanyak, goalSedang, scoreSedikit);
minimal[22] = Min(moveBanyak, goalSedang, scoreSedang);
minimal[23] = Min(moveBanyak, goalSedang, scoreBanyak);
minimal[24] = Min(moveBanyak, goalBanyak, scoreSedikit);
}

```

```

minimal[25] = Min(moveBanyak, goalBanyak, scoreSedang);
minimal[26] = Min(moveBanyak, goalBanyak, scoreBanyak);

int hasil = (int) Defuzzy(minimal);
    return hasil;
}

```

Gambar 4.13: *Source Code Implementasi Fuzzy Rule*

```

static float Defuzzy(float[] minimal) {
float nilai, jumlah, hasil;
nilai = minimal[0] * 3 + minimal[1] * 3 + minimal[2] * 2 +
    minimal[3] * 3 + minimal[4] * 3 + minimal[5] * 3 +
    minimal[6] * 2 + minimal[7] * 2 + minimal[8] * 2 +
    minimal[9] * 3 + minimal[10] * 3 + minimal[11] * 2 + minimal[12] *
3 + minimal[13] * 3 + minimal[14] * 2 +    minimal[15] * 3 +
minimal[16] * 2 + minimal[17] * 1 +    minimal[18] * 2 + minimal[19] * 2
+ minimal[20] * 1 +    minimal[21] * 2 + minimal[22] * 1 + minimal[23] *
1 +    minimal[24] * 1 + minimal[25] * 1 + minimal[26] * 1;
jumlah = minimal[0] + minimal[1] + minimal[2] + minimal[3] +
    minimal[4] + minimal[5] + minimal[6] + minimal[7] +
    minimal[8] + minimal[9] + minimal[10] + minimal[11] +    minimal[12]
+ minimal[13] + minimal[14] + minimal[15] +    minimal[16] + minimal[17]
+ minimal[18] + minimal[19] +    minimal[20] + minimal[21] +
minimal[22] + minimal[23] +    minimal[24] + minimal[25] + minimal[26];
hasil = nilai / jumlah;
return Mathf.Round(hasil);
}

static float Min(float a, float b, float c) {
float[] minimal = new float[3];
minimal[0] = a;
minimal[1] = b;
minimal[2] = c;
float kecil = minimal[0];
    for (int i = 0; i < minimal.Length; i++) {
        if (minimal[i] < kecil) {
            kecil = minimal[i];
        }
    }
    return kecil;
}

```

Gambar 4.14: *Source Code Penerapan fungsi implikasi MIN dan Proses defuzzifikasi*

### 4.3.1 Implementasi Algoritma *Fuzzy Sugeno* pada Kenaikan Tingkat Permainan

Pada sub-bab ini membahas mengenai implementasi *fuzzy sugeno* pada kenaikan tingkat permainan. Saat pertama kali bermain, pemain akan diarahkan ke *level 1* terlebih dahulu dengan tingkat kesulitan mudah. *Output* dari hasil permainan tergantung seberapa banyak pemain menyelesaikan misi yang telah diberikan, jika berhasil menyelesaikan semua misi dan *output fuzzy sugeno* adalah 3 maka pemain dapat melanjutkan ke *level 3* dengan tingkat kesulitan “Sulit” tanpa harus melewati *level 2* dengan tingkat kesulitan “Normal”. Berikut ini *source code* untuk menentukan kenaikan tingkat permainan pada *game Colouration*:

```
public static class LevelController{
public static int levelNext = 0;
public static void UpdateLevel(Board board){
int hasil = MissionManager.hasil;
int[] mudah = {0, 3, 6};
int[] normal = {1, 4, 7};
int[] sulit = {2, 5, 8};

if(hasil == 1){
for(int i=0; i<mudah.Length; i++){
if(board.level < mudah[i]){
levelNext = mudah[i];
break;
}
}
}else if(hasil == 2){
for(int i=0; i<normal.Length; i++){
if(board.level < normal[i]){
levelNext = normal[i];
break;
}
}
}else if(hasil == 3){
for(int i=0; i<sulit.Length; i++){
if(board.level < sulit[i]){
levelNext = sulit[i];
break;
}
}
}}}}}
```

Gambar 4.15: *Source Code* Kenaikan Level

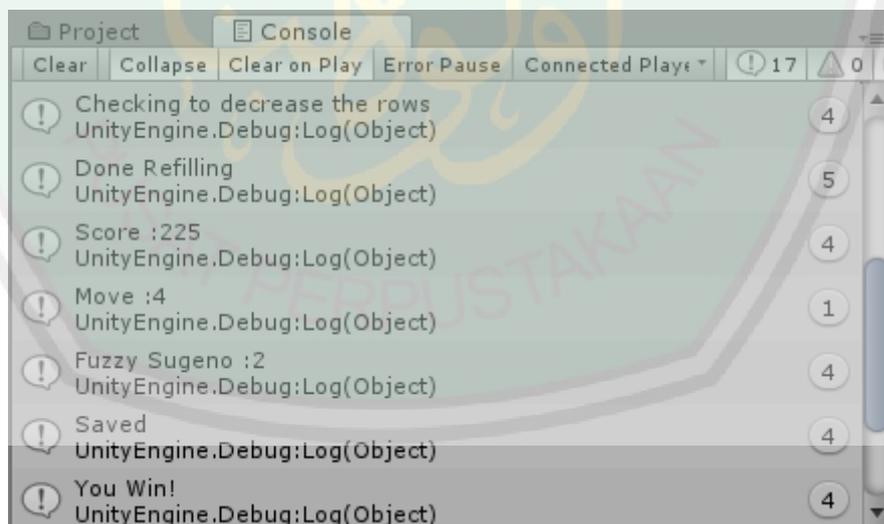
#### 4.4 Uji Coba

Pada sub bab ini membahas tentang uji coba yang dilakukan untuk mengetahui sejauh mana algoritma *Fuzzy Sugeno* berhasil di implementasikan pada *game Colouration*. Berikut pembahasan uji coba tersebut.

##### 4.4.1 Uji Coba Algoritma *Fuzzy Sugeno* pada Kenaikan Tingkat Permainan

Pada pengujian algoritma *fuzzy sugeno* ini digunakan untuk menentukan tingkat permainan yang akan dimainkan dengan aturan-aturan yang telah dirancang sebelumnya. Pengujian dilakukan berulang kali dengan nilai *variable input* yang berbeda-beda. Dari nilai tersebut didapat variabel masukan yang digunakan pada proses *Fuzzy Sugeno*. Nilai tersebut akan menentukan di tingkat permainan dan kesulitan mana pemain akan bermain selanjutnya.

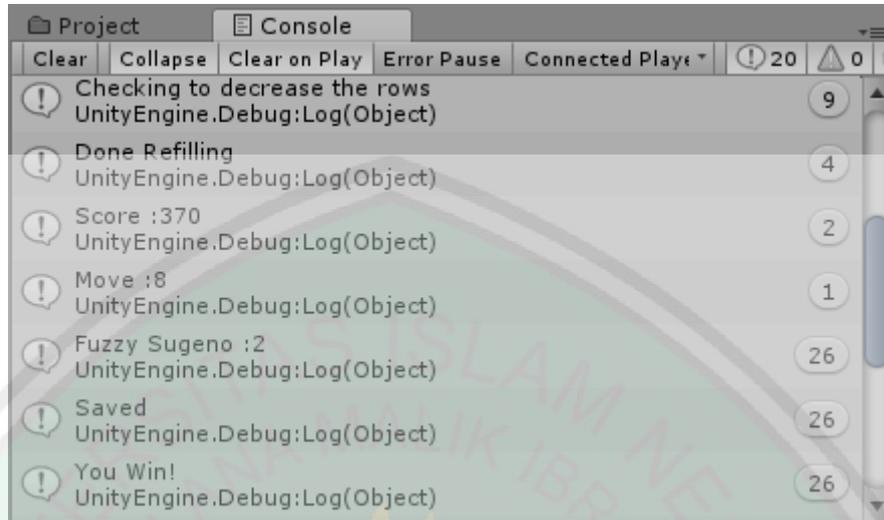
Pengujian pertama dilakukan pada kelompok warna Primer di *level 1* tingkat kesulitan “Mudah” dengan *goals* total sebanyak 20 item, *move* yang telah dipakai sebanyak 5 dari 9 *move* dan *score* yang didapat sebanyak 225. *Output* yang didapatkan adalah 2 yang berarti tingkat kesulitan di *level* selanjutnya adalah “Normal”.



Gambar 4.16: Unity: *Console* Nilai Variabel *Input* dan *Output* Pengujian Pertama

Pengujian kedua dilakukan pada kelompok warna Primer di *level 2* tingkat kesulitan “Normal” dengan *goals* sebanyak 30 item, *move* yang telah dipakai sebanyak 4 dari 12 *move* dan *score* yang didapat sebanyak 37. *Output* yang

didapatkan adalah 1 yang berarti tingkat kesulitan di *level* selanjutnya adalah “Normal”.



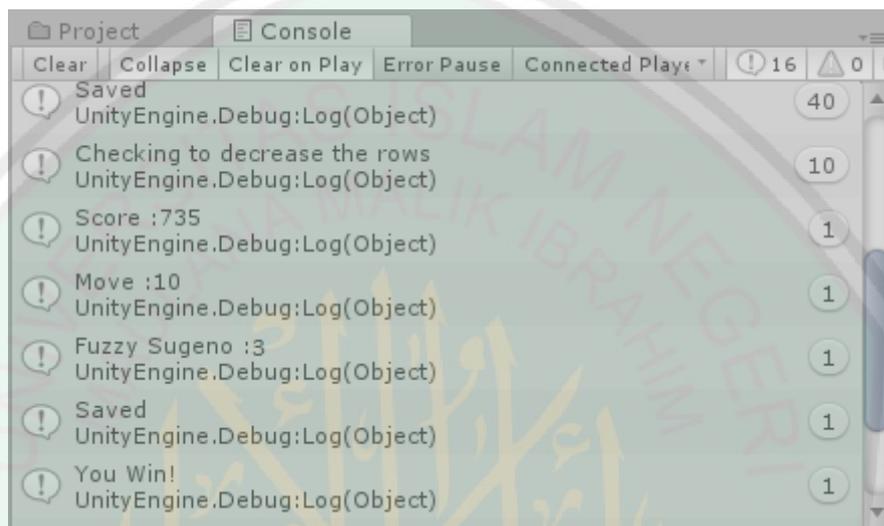
Gambar 4.17: Unity: Console Nilai Variabel *Input* dan *Output* Pengujian Kedua

Pengujian ketiga dilakukan pada kelompok warna Sekunder di *level* 5 tingkat kesulitan “Normal” dengan *goals* sebanyak 30 item, *move* yang telah dipakai sebanyak 8 dari 12 *move* dan *score* yang didapat sebanyak 380. *Output* yang didapatkan adalah 3 yang berarti tingkat kesulitan di *level* selanjutnya adalah “Sulit”.



Gambar 4.18: Unity: Console Nilai Variabel *Input* dan *Output* Pengujian Ketiga

Pengujian keempat dilakukan pada kelompok warna Tersier di *level* 9 tingkat kesulitan “Sulit” dengan *goals* total sebanyak 40 item, *moves* yang telah dipakai sebanyak 8 dari 18 *move* dan *score* yang didapat sebanyak 735. *Output* yang didapatkan adalah 3 yang berarti tingkat kesulitan di *level* selanjutnya adalah “Sulit”.



Gambar 4.19: Unity: *Console* Nilai Variabel *Input* dan *Output* Pengujian Keempat

Berikut merupakan rangkuman hasil uji coba dari penelitian yang telah dilakukan pada permainan *Colouration* dengan berbagai masukan nilai variabel *input* dan keterangan hasil uji coba yang telah tercatat pada Tabel 4.4 dibawah ini:

Tabel 4.4: Hasil Pengujian Implementasi Algoritma *Fuzzy Sugeno*

Percobaan Ke-	Puzzle			Input			Output	Hasil Uji coba	Keterangan
	Kelompok Warna	No	Level	Goals	Moves	Score	Level		
1.	Primer	1	Mudah	20	5	225	Normal	Normal	Berhasil
2.	Primer	2	Normal	30	4	370	Normal	Normal	Berhasil
3.	Sekunder	5	Normal	30	8	380	Sulit	Sulit	Berhasil

4.	Tersier	9	Sulit	40	8	735	Sulit	Sulit	Berhasil
5.	Primer	1	Mudah	20	5	135	Normal	Normal	Berhasil
6.	Primer	2	Normal	30	3	290	Normal	Normal	Berhasil
7.	Sekunder	5	Normal	30	10	310	Normal	Normal	Berhasil
8.	Tersier	8	Normal	30	11	350	Normal	Normal	Berhasil
9.	Primer	1	Mudah	20	9	400	Sulit	Sulit	Berhasil
10.	Primer	3	Sulit	40	17	275	Sulit	Sulit	Berhasil
11.	Sekunder	6	Sulit	40	10	300	Mudah	Mudah	Berhasil
12.	Tersier	7	Mudah	20	6	145	Mudah	Mudah	Berhasil
13.	Primer	1	Mudah	20	3	495	Mudah	Mudah	Berhasil
14.	Sekunder	4	Mudah	20	4	388	Mudah	Mudah	Berhasil
15.	Tersier	7	Mudah	20	9	485	Sulit	Sulit	Berhasil
16.	Tersier	9	Sulit	40	11	210	Normal	Normal	Berhasil
17.	Primer	1	Mudah	20	2	180	Normal	Normal	Berhasil
18.	Primer	2	Normal	30	5	360	Normal	Normal	Berhasil
19.	Sekunder	5	Normal	30	2	375	Normal	Normal	Berhasil
20.	Tersier	8	Normal	30	8	350	Normal	Normal	Berhasil
21.	Primer	1	Mudah	20	2	145	Normal	Normal	Berhasil
22.	Primer	2	Normal	30	5	375	Normal	Normal	Berhasil
23.	Sekunder	5	Normal	30	6	385	Normal	Normal	Berhasil
24.	Tersier	8	Normal	30	8	470	Sulit	Sulit	Berhasil
25.	Primer	1	Mudah	20	4	145	Sulit	Sulit	Berhasil
26.	Primer	3	Sulit	40	8	275	Sulit	Sulit	Berhasil
27.	Sekunder	6	Sulit	40	12	210	Sulit	Sulit	Berhasil
28.	Tersier	9	Sulit	30	5	810	Sulit	Sulit	Berhasil
29.	Primer	1	Mudah	20	2	125	Mudah	Mudah	Berhasil

30.	Sekunder	4	Mudah	20	7	305	Mudah	Mudah	Berhasil
31.	Tersier	7	Mudah	20	9	370	Normal	Normal	Berhasil
32.	Tersier	8	Normal	30	11	175	Normal	Normal	Berhasil
33.	Primer	1	Mudah	20	4	350	Mudah	Mudah	Berhasil
34.	Sekunder	4	Mudah	20	2	240	Mudah	Mudah	Berhasil
35.	Tersier	7	Mudah	20	8	175	Mudah	Mudah	Berhasil
36.	Primer	1	Mudah	20	1	530	Mudah	Mudah	Berhasil
37.	Sekunder	4	Mudah	20	8	1760	Normal	Normal	Berhasil
38.	Sekunder	5	Normal	30	4	350	Mudah	Mudah	Berhasil
39.	Tersier	7	Mudah	20	7	225	Mudah	Mudah	Berhasil
40.	Primer	1	Mudah	20	1	335	Mudah	Mudah	Berhasil
41.	Primer	2	Normal	30	4	515	Sulit	Sulit	Berhasil
42.	Primer	3	Sulit	40	5	490	Sulit	Sulit	Berhasil
43.	Sekunder	6	Sulit	40	12	810	Sulit	Sulit	Berhasil
44.	Tersier	9	Sulit	40	11	410	Sulit	Sulit	Berhasil
45.	Primer	1	Mudah	20	6	355	Mudah	Mudah	Berhasil
46.	Sekunder	4	Mudah	20	8	170	Mudah	Mudah	Berhasil
47.	Tersier	7	Mudah	20	6	250	Mudah	Mudah	Berhasil
48.	Primer	1	Mudah	20	2	400	Mudah	Mudah	Berhasil
49.	Sekunder	4	Mudah	20	4	375	Mudah	Mudah	Berhasil
50.	Tersier	7	Mudah	20	5	270	Mudah	Mudah	Berhasil

Berdasarkan dari tabel 4.4 tersebut dapat diketahui bahwa hasil akurasi menunjukkan sebanyak 100%. Hasil tersebut dinyatakan baik karena dari percobaan tersebut sesuai dengan *rules fuzzy* yang telah dirancang sehingga tidak

mengurangi hasil akurasi pada kenaikan tingkat permainan *game Colouration*. Sedangkan prosentase akurasi dari *output* tingkat kesulitan yang dihasilkan adalah dengan tingkat kesulitan Mudah 38%, tingkat kesulitan Normal 34 %, dan tingkat kesulitan Sulit 28 %.

#### 4.5 Integrasi *Game Colouration* dengan Islam

Islam merupakan agama yang realistis dengan kehidupan, bukan juga agama yang membelenggu yang memperlakukan manusia seolah-olah malaikat yang tidak memiliki keinginan atau nafsu sama sekali. Islam tidak mewajibkan pemeluknya untuk menjadikan semua omongannya dzikir, semua diamnya berfikir, segala imajinasinya ilmu, semua kekosongannya ibadah, akan tetapi islam memperlakukan manusia sesuai porsi fitrahnya yang bisa senang, sedih, jenuh, bercanda dan bermain selama semua masih dalam batasan yang baik dan tidak berlebih-lebihan.

Keluasan serta kelapangan manusia untuk merasakan kenikmatan hidup dapat diambil pelajarannya sebagaimana penjelasan dari hadits riwayat Muslim berikut:

حَدَّثَنَا يَحْيَى بْنُ يَحْيَى التَّمِيمِيُّ وَقَطْنُ بْنُ نُسَيْرٍ - وَاللَّفْظُ لِيَحْيَى -: أَخْبَرَنَا جَعْفَرُ بْنُ سُلَيْمَانَ، عَنْ سَعِيدِ بْنِ إِيَّاسِ الْجُرَيْرِيِّ، عَنْ أَبِي عُثْمَانَ النَّهْدِيِّ، عَنْ حَنْظَلَةَ الْأَسَدِيِّ قَالَ: وَكَانَ مِنْ كُتَابِ رَسُولِ اللَّهِ ﷺ قَالَ: لَقَيْتَنِي أَبُو بَكْرٍ فَقَالَ: كَيْفَ أَنْتَ يَا حَنْظَلَةُ؟ قَالَ: قُلْتُ: نَافِقٌ حَنْظَلَةُ. قَالَ: سُبْحَانَ اللَّهِ، مَا تَقُولُ؟ قَالَ: قُلْتُ: نَكُونُ عِنْدَ رَسُولِ اللَّهِ ﷺ. يُدَكِّرُنَا بِالنَّارِ وَالْجَنَّةِ. حَتَّى كَأَنَّ رَأْيِي عَيْنٍ، فَإِذَا خَرَجْنَا مِنْ عِنْدِ رَسُولِ اللَّهِ ﷺ، عَافَسْنَا الْأَزْوَاجَ وَالْأَوْلَادَ وَالصَّبِيغَاتِ، فَتَسِينَا كَثِيرًا. قَالَ أَبُو بَكْرٍ: فَوَاللَّهِ، إِنَّا لَنَلْقَى مِثْلَ هَذَا. فَأَنْطَلَقْتُ أَنَا وَأَبُو بَكْرٍ، حَتَّى دَخَلْنَا عَلَى رَسُولِ اللَّهِ ﷺ. قُلْتُ: نَافِقٌ حَنْظَلَةُ، يَا رَسُولَ اللَّهِ! فَقَالَ رَسُولُ اللَّهِ ﷺ: (وَمَا ذَاكَ؟) قُلْتُ: يَا رَسُولَ اللَّهِ، نَكُونُ عِنْدَكَ. نُدَكِّرُنَا بِالنَّارِ وَالْجَنَّةِ، حَتَّى كَأَنَّ رَأْيِي عَيْنٍ. فَإِذَا خَرَجْنَا مِنْ عِنْدِكَ، عَافَسْنَا الْأَزْوَاجَ وَالْأَوْلَادَ وَالصَّبِيغَاتِ. نَسِينَا كَثِيرًا. فَقَالَ رَسُولُ اللَّهِ ﷺ: (وَالَّذِي نَفْسِي بِيَدِهِ، إِنْ لَوْ تَدُومُونَ عَلَيَّ مَا تَكُونُونَ عِنْدِي، وَفِي الذِّكْرِ، لَصَافَحْتَكُمْ الْمَلَائِكَةُ عَلَى فُرُشِكُمْ وَفِي طُرُقِكُمْ. وَلَكِنْ، يَا حَنْظَلَةُ، سَاعَةٌ وَسَاعَةٌ) ثَلَاثَ مَرَّاتٍ [رواه مسلم]

Artinya:

“Telah menceritakan kepada kami Yahya bin Yahya At-Taimi dan Qathan bin Nusair telah menceritakan kepada kami. Lafal ini milik Yahya. Ja’far bin Sulaiman mengabarkan kepada kami dari Sa’id bin Iyas Al-Juraiiri, dari Abu ‘Utsman An-Nahdi, dari Hanzhalah Al-Usayyidi. Beliau termasuk juru tulis Rasulullah *shallallahu ‘alaihi wa sallam*, beliau mengatakan:

Abu Bakr menemuiku dan bertanya: “Apa kabarmu wahai Hanzhalah?” Hanzhalah berkata: Aku katakan: “Hanzhalah telah berbuat nifak”. Abu Bakr mengatakan: “Mahasuci Allah, apa yang engkau ucapkan?” Hanzhalah berkata: Aku berkata: “Kami apabila berada di dekat Rasulullah *shallallahu ‘alaihi wa sallam*, lalu beliau mengingatkan kami dengan neraka dan surga, sehingga seakan-akan kami melihat dengan mata kepala. Namun jika kami telah keluar dari sisi Rasulullah *shallallahu ‘alaihi wa sallam*, kami pun mengurus istri-istri, anak-anak, dan mata pencaharian kami, sehingga kami banyak lalai.” Abu Bakr mengatakan: “Demi Allah, sesungguhnya kami juga mengalami hal semisal ini”.

Aku dan Abu Bakr berangkat pergi menemui Rasulullah *shallallahu ‘alaihi wa sallam*. Aku mengatakan: “Hanzhalah telah berbuat nifak wahai Rasulullah.” Rasulullah *shallallahu ‘alaihi wa sallam* bertanya, “Kenapa begitu?” Aku berkata: “Wahai Rasulullah, kami apabila berada di dekatmu, lalu engkau ingatkan kami dengan neraka dan surga, sampai seakan-akan kami melihat dengan mata kepala. Namun, jika kami sudah keluar pergi dari sisimu, kami pun mengurus istri-istri, anak-anak, dan mata pencaharian kami, sehingga kami sering lalai.”

Rasulullah *shallallahu ‘alaihi wa sallam* bersabda, “*Demi Allah yang jiwaku berada di tangan-Nya, kalau kalian terus-menerus seperti ketika kalian berada di dekatku dan selalu zikir, niscaya malaikat akan menjabat tangan kalian di atas ranjang-ranjang dan di jalan-jalan kalian. Tetapi, wahai Hanzhalah, ada saatnya begini dan ada saatnya begini.*” Sebanyak tiga kali.” [H.R Muslim]

Hadits tersebut menunjukkan bahwa kesenangan psikologis dan hiburan merupakan dua hal yang natural dalam diri manusia. Rasulullah saw bahkan mengatakan orang yang di dalam dirinya tidak ada hal tersebut, ia akan disalami

Malaikat. Maksud dari disalami Malaikat adalah menunjukkan suatu hal yang mustahil terjadi. Karena islam tidak mengajarkan agar seseorang menjauhi kesenangan dan hiburan. Namun sebaliknya, islam justru mengajarkan bahwa mencari ketenangan, beristirahat, mencari hiburan bisa dilakukan dan tidak mengharamkan hiburan sama sekali.

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تُحَرِّمُوا طَيِّبَاتِ مَا أَحَلَّ اللَّهُ لَكُمْ وَلَا تَعْتَدُوا إِنَّ اللَّهَ لَا يُحِبُّ الْمُعْتَدِينَ

Artinya:

*“Wahai orang-orang yang beriman! Janganlah kamu mengharamkan apa yang baik yang telah dihalalkan Allah kepadamu, dan janganlah kamu melampaui batas. Sesungguhnya Allah tidak menyukai orang-orang yang melampaui batas.”*  
(QS. Al-Maidah 5:87)

Penggalan hadits dan ayat di atas menjelaskan bahwa menghibur diri dalam Islam tidak dilarang, karena permainan sesungguhnya adalah bagian dari sarana hiburan dan sarana melepas lelah. Baiknya *game* dimainkan sesuai dengan porsinya, tidak berlebihan, tidak menyita seluruh waktu, menghalangi dari aktifitas dan mengambil waktu-waktu belajar serta bekerja.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil uji coba implementasi algoritma *Fuzzy Sugeno* pada game *Colouration*, dapat disimpulkan bahwa penggunaan metode ini untuk tingkatan permainan dapat diterapkan cukup baik dan sesuai dengan akurasi sebanyak 100% di mana prosentase akurasi dari *output* tingkat kesulitan mudah 38%, tingkat kesulitan normal 34%, dan tingkat kesulitan sulit 28%. Hasil pengujian telah dilakukan pada penelitian ini dengan hasil yang telah dirangkum pada Tabel 4.4. *Game* ini menggunakan variabel *input Fuzzy Sugeno* sebanyak tiga variabel, yakni variabel *moves*, variabel *goals*, dan variabel *score* dengan masing-masing memiliki nilai linguistik sedikit, sedang dan banyak. Serta *output* berupa tingkat permainan mudah, normal, dan sulit.

Hal ini menunjukkan bahwa metode algoritma *Fuzzy Sugeno* dapat menentukan tingkat permainan sesuai dengan nilai *variable input* yang didapat pemain. Ketika total *variable input* bagus maka pemain dapat langsung bermain ke tingkat permainan sulit tanpa melalui tingkat permainan mudah dan normal atau pemain dapat langsung bermain di tingkat permainan normal tanpa harus melalui tingkat permainan mudah terlebih dahulu. Akan tetapi ketika *variable input* pemain kurang bagus bahkan tidak bisa menyelesaikan misi yang diberikan sehingga gagal, maka pemain hanya dapat mengulang tingkat permainan yang sama sampai berhasil terlebih dahulu kemudian dapat bermain di tingkat permainan selanjutnya.

#### 5.2 Saran

Pembuatan permainan *Colouration* masih jauh dari sempurna, maka ada beberapa saran yang diajukan untuk pengembangan selanjutnya, yakni:

1. Menambahkan *obstacle* baru di setiap *level* yang lebih beragam pada permainan sehingga lebih menantang.
2. Meningkatkan tampilan permainan dengan menambahkan animasi dan *backsound* agar lebih menarik.

## DAFTAR PUSTAKA

- Acevedo, Edmund O; Ekkekakis, Panteleimon (ed.). 2006. *Psychobiology of Physical Activity*. Champaign, IL: Human Kinetics.
- Andri. 2017. "Setiap Orang Punya Respons yang Berbeda terhadap Stres". *Kompasiana*, [https://www.kompasiana.com/psikosomatik\\_andri/setiap-orang-punya-respons-yang-berbeda-terhadap-stres\\_588155426023bd07048b4567](https://www.kompasiana.com/psikosomatik_andri/setiap-orang-punya-respons-yang-berbeda-terhadap-stres_588155426023bd07048b4567).
- Bagus Harisa, Ardiawan, dkk. 2016. *Model Tingkat Kesulitan Dinamis Berbasis Logika Fuzzy pada Game Wayang Ramayana*. SEMNASTIM 2016, ISSN: 2302-3805.
- Candy Crush Saga. <https://discover.king.com/>.
- Crunchbase. <https://www.crunchbase.com/organization/dots#/entity>.
- Hadits riwayat Imam Muslim no. 2750.
- Hadits riwayat Imam Muslim no.7307.
- Hermawan, Latus., Novita Putri, Astrid. 2014. *Penerapan Algoritma Fuzzy Mamdani untuk Mengatur Game Scoring pada Game Helitap*. SEMANTIK 2014, ISBN: 979-26-0276-3.
- Kusumadewi, Sri. 2002. *Analisis & Desain Sistem Fuzzy menggunakan Toolbox Matlab*. Yogyakarta: Graha Ilmu.
- Kusumadewi, Sri. 2003. *Artificial Intelligence*. Yogyakarta: Graha Ilmu.
- Kusumadewi, Sri dan Hari Purnomo. 2004. *Aplikasi Logika Fuzzy untuk Pendukung Keputusan*. Yogyakarta: Graha Ilmu.
- Li, Y., Musilek, P., dan Wyard-Scott, L. 2004. Fuzzy Logic in Agent-Based Game Design (pp. 734-739). IEEE

Mardhiyah, Siti. 2014. *Upaya Peningkatan Kemampuan Mengenal Warna Melalui Metode Eksperimen Kelompok A RA Tamanagung 3 Muntilan Tahun Pelajaran 2013/2014*. Skripsi pada Fakultas Ilmu Tarbiyah dan Keguruan UIN Sunan Kalijaga.

Meilani. 2013. *Teori Warna: Penerapan Lingkaran Warna dalam Berbusana*. Jurnal *Humaniora* Vol. 4 No. 1, April 2013: 326-338.

Naba, Agus. 2009. *Belajar Fuzzy Logic Menggunakan Matlab*. Yogyakarta: CV. Andi Offset.

Purba, Kristo Radion, Rini Nur Hasanah, M. Azis Muslim. 2013. *Implementasi Logika Fuzzy Untuk Mengatur Perilaku Musuh dalam Game Bertipe Action-RPG*. Jurnal *EECCIS*, Vol. 7 No. 1, Juni 2013.

Rahayu, Adhifa. 2012. *Peran Warna dalam Arsitektur Sebagai Salah Satu Kebutuhan Manusia*. Skripsi pada Fakultas Teknik Universitas Indonesia.

Schneier, Bruce. 1996. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Second Edition, John Wiley & Sons. ISBN 0-471-11709-9

Susanti, Afriani. 2017. "Mantap! Menurut Peneliti, Bermain Video Game Dapat Redakan Stres bagi Para Pekerja". *Okezone*, 30 Juli 2017, dilihat 23 Oktober 2017, <https://techno.okezone.com/read/2017/07/29/56/1746316/mantap-menurut-peneliti-bermain-video-game-dapat-redakan-stres-bagi-para-pekerja>.

Two Dots. <https://www.dots.co/about/>.

Utomo. 2008. *Hubungan Antara Model-Model Coping Stres pada Mahasiswa Tahun Pertama Fakultas Psikologi UIN Malang*. Skripsi pada Fakultas Psikologi UIN Malang.

Wardana, Kusuma. 2015. "Membuat Himpunan Fuzzy dan Input Fuzzy". Website, 22 November 2015, <https://tutorkeren.com/artikel/membuat-himpunan-fuzzy-dan-input-fuzzy.html>.

Wibawa, Shierine Wangsa. 2017. "Studi Generasi Milenium Paling Stres di Tempat Kerja". *Kompas*, 10 Februari 2017, <http://female.kompas.com/read/2017/02/10/170400720/studi.generasi.milenium.paling.stres.di.tempat.kerja>.