



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 *Game*

Game didefinisikan sesuatu yang digunakan untuk bermain atau pun dimainkan dan dibatasi dengan peraturan tertentu yang menentukan tindakan pemain dalam game tersebut (Pangkatodi, et al., 2016). Menurut Costikyan, game adalah bentuk karya seni dimana peserta membuat keputusan untuk menggunakan sumber daya yang dimiliki pada *game* tersebut demi mencapai tujuan (Costikyan, 2013).

Dalam sebuah *game*, adanya sebuah tantangan pada *game* tersebut berperan besar dalam keseruan dan membuat orang tertarik. Karena tanpa adanya sebuah tantangan, game tidak akan terlihat begitu menarik (Juul, 2011).

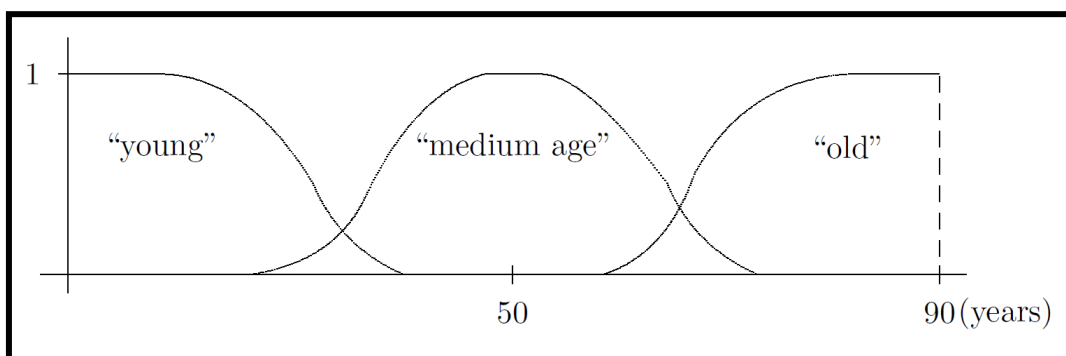
2.2 *Dynamic Difficulty Adjustment (DDA)*

Secara umum, terdapat sebuah nilai ketidakpuasan dari pemain dalam memainkan sebuah *game*. Hal ini dikarenakan keterbatasan tantangan yang terdapat dalam *game* kurang menyesuaikan dengan *skill* yang dimiliki oleh setiap pemain, seperti kecepatan reaksi pemain, *learning and adaptibility* setiap pemain. Kesulitan dalam sebuah *game* yang secara manual dapat dipilih oleh pemain tidak dapat menghindar dari kebosanan karena pemain tidak bisa memutuskan tingkat tantangan atau kesulitan yang sesuai dengan *skill* pemain tersebut (Zohaib, 2018).

Dynamic Difficulty Adjustment (DDA) adalah proses penyesuaian tingkat kesulitan (atau tantangan) secara dinamis atau *real-time* berdasarkan kemampuan pemain (Zohaib, 2018). Tingkat kesulitan akan diatur sesuai dengan kemampuan pemain berdasarkan elemen-elemen tertentu yang telah ditentukan sehingga game tidak menjadi terlalu sulit atau terlalu mudah untuk dimainkan. Inti dari *dynamic difficulty adjustment* adalah untuk membuat pemain tetap tertarik selama *game* dimainkan dan memberikan tingkat kesulitan yang sesuai dengan pemain (Zohaib, 2018).

2.3 Fuzzy Logic

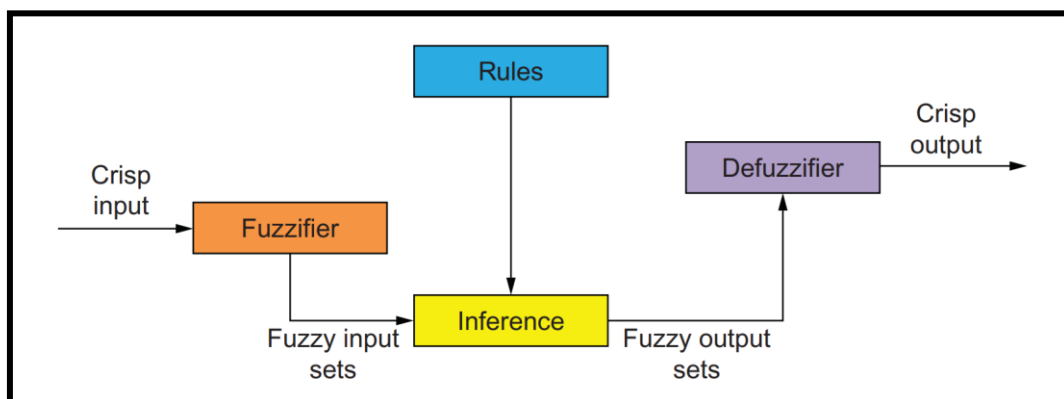
Fuzzy Logic merupakan sebuah peningkatan dari logika klasik yang dimana menggunakan tingkat kebenaran daripada kebenaran boolean (Novak, et al., 1999). *Fuzzy Logic* memungkinkan adanya nilai di antara 0 dan 1, konsep *true*, *very true*, *more or less true*, *rather true*, *not true*, *false*, *not very true*, dan *not very false* (Zadeh, 1975).



Gambar 2.1 Contoh Fuzzy Logic (Novak, et al., 1999)

Gambar 2.1 merupakan sebuah contoh sederhana mengenai *Fuzzy Logic*. Jika seseorang berusia dibawah lima puluh dan terletak di pertemuan antara *young* dan *medium age*, orang tersebut bisa dianggap *medium age* dan *young*. Seseorang juga dapat hanya dianggap *young* atau *medium age* atau *old* jika umurnya hanya dalam cakupan usia tersebut dan tidak mengiris cakupan usia lain.

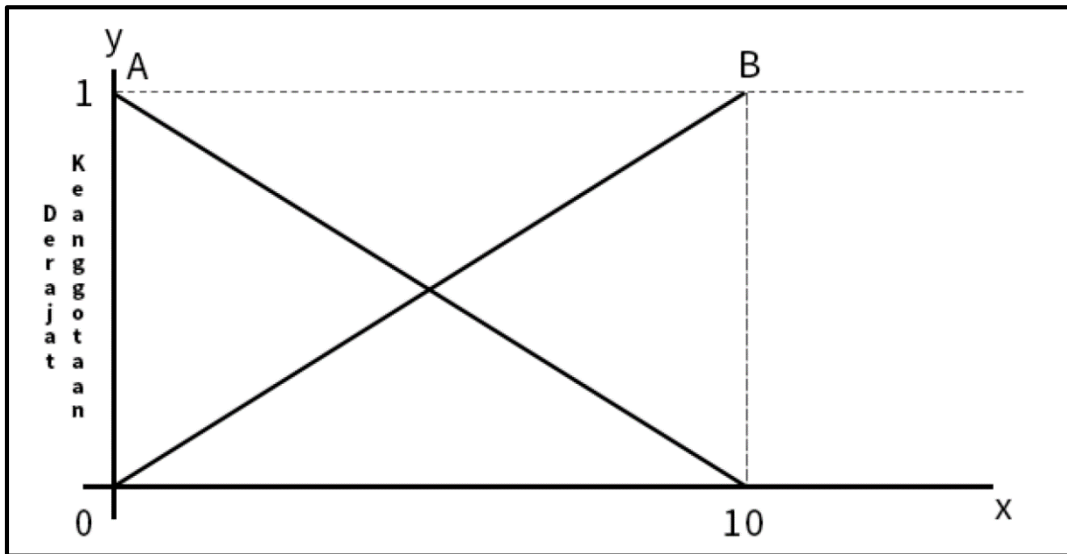
Gambar 2.2 menunjukkan terdapat tiga proses utama dalam penggunaan algoritma *fuzzy logic* pada sebuah perangkat, yaitu proses *fuzzifier* atau *fuzzification*, kemudian dilanjutkan dengan penyesuaian terhadap aturan atau *rule evaluation*, dan hasil dari *rule evaluation* akan dilakukan proses *defuzzifier* atau *defuzzification* (Zadeh, 1996).



Gambar 2.2 Arsitektur Fuzzy Logic (Kayacan & Khanesar, 2016)

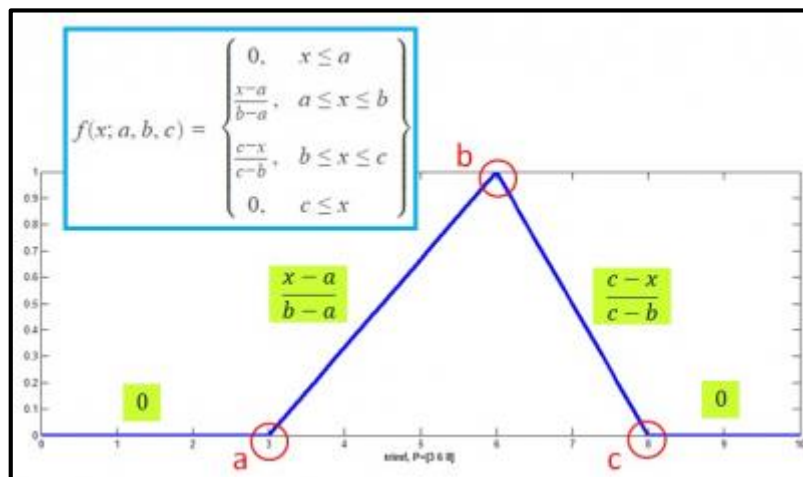
2.3.1 Fuzzification

Fuzzification merupakan sebuah proses perubahan nilai input dari bentuk tegas (*crisp*) menjadi *fuzzy* yang secara umum disajikan dalam bentuk himpunan-himpunan *fuzzy* yang masing-masing memiliki fungsi keanggotaan (Zadeh, 1996).



Gambar 2.3 Grafik Fungsi Keanggotaan (Zadeh, 1996)

Gambar 2.3 merupakan contoh fungsi keanggotaan sebuah himpunan *fuzzy*. Dalam fungsi ini, terdapat dua himpunan yaitu A dan B. Sumbu x merupakan input dari algoritma *fuzzy* dan y merupakan derajat keanggotaan atau nilai *fuzzy*. Dalam contoh ini input *fuzzy* berkisar antara 0 sampai 10 dan derajat keanggotaan memiliki nilai antara 0 sampai 1.



Gambar 2.4 Rumus Persamaan Garis (Kubelka, 2000)

Gambar 2.4 merupakan rumus untuk mendapatkan nilai dari persamaan garis yang melalui titik potong sebuah himpunan *fuzzy*. Nilai-nilai derajat keanggotaan dari setiap input akan diterapkan operator-operator logika atau aturan-aturan atau *rules* yang telah dibuat. Contoh bentuk dari operator-operator logika ini dapat dilihat pada gambar 2.3.

2.3.2 Inference System (Rule Evaluation)

Aturan-aturan atau *rules* yang telah ditentukan sebelumnya akan digunakan sebagai acuan untuk menjelaskan hubungan antara variabel masukan dan keluaran. Metode Tsukamoto digunakan untuk menjelaskan hubungan antara masukan dan keluar variabel-variabel ini (Zadeh, 1996).

Dalam metode Tsukamoto, implikasi dari setiap aturan memiliki bentuk implikasi “Sebab-Akibat”. Terdapat dua cara untuk menentukan hubungan antar variabel:

1. Konjungsi Fuzzy

Konjungsi fuzzy dari A dan B dilambangkan dengan $A \cap B$ dan didefinisikan dengan:

$$\begin{aligned} \alpha = \mu_{A \wedge B} &= \mu_A(x) \cap \mu_B(y) \\ &= \min(\mu_A(x), \mu_B(y)) \end{aligned} \quad \dots(2.1)$$

2. Disjungsi Fuzzy

Disjungsi fuzzy dari A dan B dilambangkan dengan $A \cup B$ dan didefinisikan dengan:

$$\begin{aligned} \alpha = \mu_{A \vee B} &= \mu_A(x) \cup \mu_B(y) \\ &= \max(\mu_A(x), \mu_B(y)) \end{aligned} \quad \dots(2.2)$$

Dalam konjungsi, perhitungan dengan logika ‘dan’, atau ‘*and*’ adalah dengan mencari nilai yang paling kecil. Sedangkan dalam disjungsi, perhitungan dengan logika ‘atau’, atau ‘*or*’ adalah dengan mencari nilai yang paling besar.

Hasil dari evaluasi aturan dengan variabel adalah derajat keanggotaan pada fungsi keanggotaan *output*. Nilai ini masih dalam berbentuk *fuzzy* dan selanjutnya akan dilakukan *defuzzification* untuk menentukan nilai tegas (*crisp*) pada fungsi keanggotaan *output*.

2.3.3 Defuzzification

Proses *defuzzification* merupakan proses perubahan variabel yang berbentuk *fuzzy* menjadi variabel tegas (*crisp*) (Zadeh, 1996). Salah satu cara yang dapat digunakan adalah metode *weighted average*. Metode *defuzzification* ini paling sering digunakan dalam aplikasi-aplikasi yang menggunakan algoritma *fuzzy logic* karena efisien dalam hal mengkomputasi. Metode *weighted average* memiliki rumus sebagai berikut.

$$z^* = \frac{\sum \mu_i z_i}{\sum \mu_i} \quad \dots(2.3)$$

2.4 Algoritma A*

Algoritma A Star (A*) merupakan salah satu algoritma heuristik yang sering digunakan dalam perencanaan jalur terpendek dengan perhitungan terkecil (Wang, et al., 2017). Algoritma ini menggunakan *array* bersifat dua dimensi untuk

menentukan simpul-simpul yang akan dilewati. Perhitungan Algoritma Star (A*) ditentukan dengan rumus berikut (Wang, et al., 2017).

$$F(x) = G(x) + H(x) \quad \dots(2.4)$$

Dimana:

- a. $G(x)$ adalah nilai pada pergerakan simpul awal menuju simpul berikutnya.
- b. $H(x)$ adalah perkiraan nilai untuk pergerakan dari simpul awal menuju simpul akhir.
- c. $F(x)$ adalah jumlah nilai dari fungsi $G(x)$ dan $H(x)$ dengan nilai terkecil dari fungsi $F(x)$ adalah jalur terpendek dari simpul awal menuju simpul akhir.

2.5 Game Design Document

Dalam industri *game* tidak terdapat acuan standar penulisan untuk *Game Design Document*, yang dimana merupakan bagian penting dari proses pembentukan *game* (Carvalho & Gomes, 2016). *Game Design Document* dapat dibagi menjadi sembilan bagian yaitu (Gonzalez, 2016):

1. Characters: merupakan karakter-karakter yang digunakan dalam *game*.
2. Story: merupakan bagian yang akan terjadi pada karakter-karakter pada *game*.
3. Story Progression: merupakan perkembangan alur cerita seiring berjalannya *game*.
4. Gameplay: merupakan bagian penting pada *Game Design Document* yang menjelaskan:
 - a. Goals: objektif yang harus dicapai oleh pemain dalam *game*.

- b. User Skills: kemampuan pemain dalam memainkan *game*.
 - c. Game Mechanics: menjelaskan bagaimana *game* seharusnya bekerja.
 - d. Item & Power-Ups: menjelaskan alat atau benda yang dapat membantu pemain dalam mencapai *goals* pada *game*.
 - e. Losing: kondisi-kondisi yang menyebabkan pemain dianggap kalah.
5. Art Style: merupakan penggunaan tipe *art* pada keseluruhan *game*.
 6. Music & Sound: merupakan bagian yang menjelaskan musik dan efek suara yang digunakan dalam *game*.
 7. Technical Description: merupakan bagian yang menjelaskan *platform* apa saja yang dapat menjalankan *game*.
 8. Marketing & Funding: merupakan bagian yang menjelaskan cara yang digunakan untuk memasarkan *game* dan meraih pemasukan dari *game*.
Bagian ini tidak wajib.
 9. Other Ideas: merupakan kumpulan-kumpulan ide yang tidak yakin untuk diimplementasikan pada *game*.

Dalam penulisan *Game Design Document*, tidak semua bagian harus ditulis. Bila dalam *game* terdapat bagian yang tidak ada maka bagian tersebut dapat dihilangkan, misalnya *Marketing & Funding*, maka bagian tersebut tidak ditulis (Gonzalez, 2016).

2.6 Game User Experience Satisfaction Scale (GUESS)

Salah satu metode pengukuran kepuasan sebuah game adalah *Game User Experience Satisfaction Scale* yang menggunakan *seven point likert scale* (Phan, et al., 2016). Metode ini memiliki 55 pertanyaan yang dibagi dalam sembilan kategori, yaitu *usability / playability, narratives, play engrossment, enjoyment, creative freedom, audio aesthetics, personal gratification, social connectivity, and visual aesthetic* (Phan, et al., 2016).

Penilaian dengan metode *GUESS* bila dalam *game* tidak terdapat elemen seperti *social connectivity*, maka *subscale* dapat dihilangkan dari penilaian (Phan, et al., 2016). Hasil dari penilaian metode *GUESS* dapat dilakukan dengan mencari rata-rata dari setiap *subscale* yang digunakan, kemudian menghitung rata-rata dari gabungan rata-rata *subscale* untuk mendapatkan nilai akhir sebagai *video game satisfaction* (Phan, et al., 2016).

Pertanyaan dari *GUESS* menggunakan skala likert tujuh poin. Nilai terendah yang ada pada skala likert ini adalah 1 dan disebut *Strongly Disagree*, nilai tertinggi adalah 7 dan disebut *Strongly Agree*. Berikut adalah tabel *range* (100) dengan jumlah interval tujuh.

Tabel 2.1 Tabel Interval 7 Poin

Range Nilai	Keterangan
0% - 14.825%	Strongly Disagree
14.286% - 28.571%	Disagree
28.572% - 42.857%	Somewhat Disagree
42.858% - 57.142%	Neither
57.143% - 71.428%	Somewhat Agree
71.429% - 85.714%	Agree
85.715% - 100%	Strongly Agree