

BAB 2

LANDASAN TEORI

2.1 Tinjauan Teori

Berikut dijabarkan teori-teori yang mendasari penelitian berdasarkan studi literatur.

2.1.1 *Adaptive Neuro-Fuzzy Inference System*

ANFIS merupakan kombinasi antara logika fuzzy dan jaringan saraf tiruan yang digunakan untuk memodelkan hubungan input-output secara adaptif [3]. Sistem ini terdiri dari lima lapisan utama yang memproses input hingga menghasilkan keputusan:

1. **Input Layer:** Menerima variabel input.
2. **Fuzzification Layer:** Mengubah input menjadi nilai *fuzzy* menggunakan MF.
3. **Rule Layer:** Mengaplikasikan aturan *fuzzy if-then*.
4. **Normalization Layer:** Menyesuaikan bobot dari setiap aturan.
5. **Output Layer:** Menghasilkan output sebagai keputusan akhir [5].

Salah satu keunggulan ANFIS dibanding metode lain adalah kemampuannya belajar dari data dan menyesuaikan parameter MF agar sesuai dengan pola yang ditemukan. ANFIS juga mampu menangani data *non-linear* dan bekerja baik dalam kondisi tidak pasti [1].

A Rumus ANFIS

Rangkaian perhitungan ANFIS terdiri dari lima lapisan utama yang menggambarkan proses inferensi *fuzzy* dan pembelajaran jaringan saraf. Struktur ini diadaptasi dari Jang [3].

$$O_i^1 = \mu_{A_i}(x_i) = \frac{1}{1 + \left| \frac{x_i - c_i}{a_i} \right|^{2b_i}} \quad (2.1)$$

Pada lapisan ini, setiap input x_i diproses melalui MF *fuzzy*. Fungsi ini menentukan "seberapa besar input termasuk ke dalam suatu kategori ilmiah, seperti rendah, sedang atau tinggi. Rumus ini menggunakan bentuk generalisasi dari fungsi sigmoid atau *bell-shaped curve* yang berarti merujuk pada bentuk grafik yang menyerupai lonceng, di mana nilai-nilai data terkonsentrasi di sekitar rata-rata dan secara simetris menurun ke arah nilai yang lebih ekstrem di kedua sisinya. Nilai outputnya berkisar antara 0 dan 1, yang menggambarkan derajat MF suatu nilai ke dalam himpunan *fuzzy*.

- a_i , b_i , dan c_i adalah parameter yang dilatih secara otomatis oleh sistem. - Semakin dekat nilai x_i ke pusat c_i , semakin besar derajat MF.

$$w_i = \prod_{j=1}^5 \mu_{A_{ij}}(x_j) \quad (2.2)$$

Lapisan ini menghitung kekuatan aktivasi masing-masing *fuzzy rules*. Setiap aturan merepresentasikan hubungan antara beberapa kondisi input dan output tertentu. Kekuatan aturan dihitung dengan mengalikan semua derajat MF dari input-input terkait prinsip fuzzy AND.

Contoh: Jika input adalah musuh dekat, energi banyak dan darah tinggi, maka kekuatan aturan dihitung dari produk derajat MF masing-masing kondisi tersebut.

$$\bar{w}_i = \frac{w_i}{\sum_{k=1}^N w_k} \quad (2.3)$$

Layer ini menormalisasi kekuatan masing-masing aturan dengan membaginya dengan total kekuatan semua aturan. Hal ini dilakukan untuk menghitung kontribusi relatif setiap aturan terhadap output akhir. Karena kekuatan aturan bisa bernilai berapa saja, maka dilakukan normalisasi agar proporsinya adil. Setiap aturan dinormalisasi terhadap total kekuatan semua aturan. Ini seperti mengubah bobot mentah menjadi persentase kontribusi.

Output dari lapisan ini digunakan sebagai bobot dalam menghasilkan *output* akhir.

$$f_i = p_{i1}x_1 + p_{i2}x_2 + p_{i3}x_3 + p_{i4}x_4 + p_{i5}x_5 + r_i \quad (2.4)$$

Di lapisan ini, setiap aturan menghasilkan sebuah output berdasarkan fungsi linier dari semua input. Ini disebut sebagai konsekuen dari aturan *fuzzy*. Parameter p_{ij} dan r_i dilatih menggunakan metode regresi (least squares method) untuk mendekati

target output sebaik mungkin.

Fungsinya mirip seperti regresi linier yang artinya sebuah metode statistik yang digunakan untuk memodelkan hubungan antara variabel dependen dengan satu atau lebih variabel independen, tetapi berbobot.

$$y = \sum_{i=1}^N \bar{w}_i f_i \quad (2.5)$$

Terakhir, ANFIS menghitung *output* akhir sebagai hasil penjumlahan semua output aturan yang sudah diberi bobot. Ini adalah proses *defuzzifikasi* menggunakan metode rata-rata berbobot (*weighted average*).

Output akhir ini digunakan sebagai keputusan sistem, misalnya: jarak untuk menyerang musuh, atau nilai *decision* AI untuk pemanggilan pasukan.

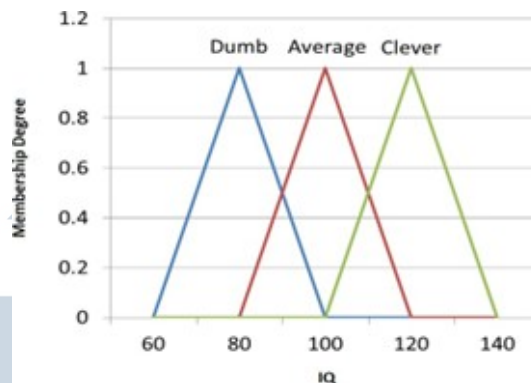
2.1.2 Fuzzy Logic

Fuzzy logic dikembangkan oleh Zadeh [4] sebagai sistem logika yang memungkinkan nilai kebenaran bersifat kontinu antara 0 dan 1. *Fuzzy logic* berguna dalam sistem pengambilan keputusan ketika data bersifat ambigu atau tidak pasti dan dapat digunakan untuk mendefinisikan aturan secara ilmiah seperti “jauh”, “sedikit”, atau “tinggi”. Hal yang tidak pasti, seperti “jarak dekat”, “energi rendah”, atau “darah sedang” [6].

Fuzzy logic banyak digunakan pada sistem kontrol, pengenalan pola dan AI dalam game karena mampu meniru cara manusia berpikir secara intuitif [2].

2.1.3 Membership Function

MF adalah fungsi matematis yang digunakan dalam logika fuzzy untuk menentukan tingkat *membership* suatu nilai input dalam himpunan fuzzy tertentu [4]. MF merupakan elemen krusial dalam sistem *fuzzy* dan ANFIS karena menentukan seberapa besar suatu input termasuk dalam kategori seperti “rendah”, “sedang” atau “tinggi” [6].



Gambar 2.1. Contoh *Fuzzy Set* pembagian kategori *Dumb*, *Average* atau *Clever* berdasarkan IQ

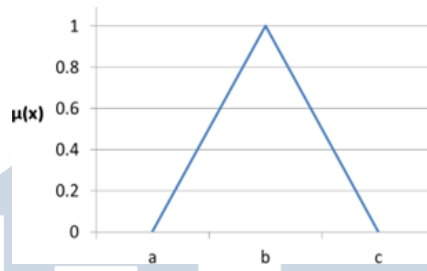
Gambar 2.1 menunjukkan contoh MF *fuzzy* dengan tiga kategori ilmiah untuk nilai IQ, yaitu *Dumb*, *Average*, dan *Clever*. Setiap kategori direpresentasikan dengan MF berbentuk segitiga (*triangular membership function*) yang saling tumpang tindih [1].

Karakteristik dari grafik ini adalah:

- ***Dumb*** memiliki puncak *membership* maksimal ($\mu = 1$) di sekitar IQ 80, dan *membership* mulai dari IQ 60 hingga 100.
- ***Average*** memiliki puncak di IQ 100, dengan batas bawah di 80 dan batas atas di 120.
- ***Clever*** memiliki puncak pada IQ 120, dengan rentang *membership* dari IQ 100 hingga 140.

Kurva ini mencerminkan sifat *fuzzy* bahwa satu nilai IQ dapat memiliki derajat *membership* lebih dari satu kategori. Sebagai contoh, seseorang dengan IQ 90 mungkin termasuk dalam kategori *Dumb* dan *Average* secara bersamaan, namun dengan tingkat *membership* yang berbeda [1].

MF seperti ini banyak digunakan dalam sistem *fuzzy* karena kemampuannya dalam menangani ketidakpastian dan ambiguitas dalam data input, yang juga diterapkan dalam sistem ANFIS untuk pengambilan keputusan berbasis kondisi dinamis di dalam game RTS [6].



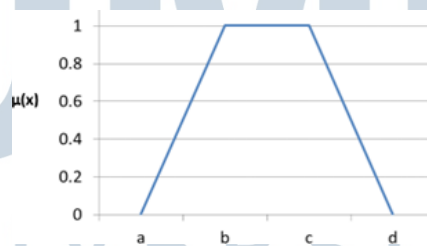
Gambar 2.2. Contoh *Triangular Fuzzy Set*

MF *triangular* merupakan bentuk paling sederhana dari MF *fuzzy*. Fungsi ini didefinisikan oleh tiga parameter: titik awal (a), titik puncak (b) dan titik akhir (c). Fungsi ini memiliki bentuk simetris dan cocok digunakan untuk representasi nilai seperti *Enemy Range* atau *Energy Amount* [1].

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \geq c \end{cases} \quad (2.6)$$

Fungsi ini digunakan pada penelitian Andy [1] untuk menangkap tingkat *membership* sederhana dalam variabel fuzzy seperti *Health*.

2.1.4 Trapezoidal Membership Function



Gambar 2.3. Contoh *Trapezoid Fuzzy Set*

MF *trapezoid* lebih fleksibel dibandingkan fungsi segitiga karena memiliki bagian puncak yang datar. Fungsi ini terdiri dari empat parameter: a , b , c , dan d , dengan *membership* penuh ($\mu = 1$) di antara b dan c .

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ 1, & b < x < c \\ \frac{d-x}{d-c}, & c \leq x < d \\ 0, & x \geq d \end{cases} \quad (2.7)$$

Dalam konteks ANFIS, MF bukan hanya digunakan untuk mengubah nilai input tetapi juga dioptimasi menggunakan algoritma pembelajaran seperti backpropagation atau GA [7] [1].

A Pemanfaatan MF dalam Sistem

Dalam sistem ANFIS yang diimplementasikan pada game strategi ini, fungsi MF berperan pada **Layer 1** dari arsitektur ANFIS. Nilai input numerik dari lingkungan permainan seperti:

- x_1 : *Enemy Range*
- x_2 : *Energy*
- x_3 : *Health*
- x_4 : *Ally Position*
- x_5 : *Priority*

akan dikonversi menjadi nilai fuzzy dengan menggunakan fungsi MF.

Sebagai contoh, apabila *health* bernilai 75, maka:

$$\mu_{\text{Sedang}}(75) = 0.4$$

$$\mu_{\text{Tinggi}}(75) = 0.6$$

Nilai-nilai MF ini kemudian digunakan dalam proses inferensi fuzzy untuk menghitung skor *value decision* AI yang menentukan apakah pasukan akan menyerang atau menunggu.

2.1.5 Genetic Algorithm

GA adalah metode optimasi yang terinspirasi dari proses seleksi alam dalam biologi. GA dapat digunakan untuk mengoptimalkan parameter MF dalam ANFIS agar performa sistem meningkat [7].

GA bekerja dengan proses seleksi, *crossover* dan mutasi terhadap populasi solusi dan telah terbukti efektif di berbagai aplikasi sistem cerdas [9] [10].

A Pemanfaatan GA dalam Sistem ANFIS

Dalam penelitian ini, GA digunakan untuk mengoptimalkan parameter fungsi MF fuzzy (seperti a_i , b_i , dan c_i dalam fungsi sigmoid atau bell-shaped). Tujuan dari optimasi ini adalah untuk meningkatkan kualitas keputusan yang dihasilkan oleh sistem ANFIS.

Tanpa pelatihan berbasis data (seperti pada pembelajaran *backpropagation*), nilai bobot dan parameter pada fungsi MF bisa bersifat statis dan *suboptimal*. GA memungkinkan pencarian kombinasi parameter yang menghasilkan *output* keputusan AI paling akurat berdasarkan fungsi objektif yang telah ditentukan [10] [9].

- **Input:** Parameter fungsi MF awal.
- **Fitness Function:** Minimasi selisih antara *value* keputusan AI hasil inferensi dan nilai target atau performa *gameplay*.
- **Output:** Parameter *fuzzy* yang telah dioptimasi.

Dengan pendekatan ini, ANFIS menjadi lebih adaptif terhadap kondisi permainan dan menghasilkan keputusan yang lebih baik secara generalisasi [9] [10].

Fungsinya dalam Optimalisasi:

- Optimalisasi MF dalam Fuzzy Logic atau ANFIS.
- Optimasi pergerakan *unit* dan strategi permainan dalam game RTS [9].
- Menyediakan alternatif terhadap learning ANFIS untuk kondisi yang membutuhkan adaptasi terhadap lingkungan dinamis.

GA secara umum digunakan dalam penelitian-penelitian sebelumnya untuk mengoptimalkan struktur atau parameter dalam sistem *fuzzy*, termasuk MF dan

strategi *unit* dalam game. Dalam konteks penelitian ini, GA menjadi salah satu alternatif metode optimasi yang dapat dipertimbangkan, meskipun pendekatan yang digunakan adalah dengan mendefinisikan MF secara *manual*. Dibandingkan dengan GA, sistem ANFIS yang dikembangkan mengandalkan kombinasi pembelajaran data dan *fuzzy rule*, namun ke depan GA dapat dijadikan metode pelengkap dalam peningkatan akurasi dan efisiensi sistem pengambilan keputusan [10] [9] [5].

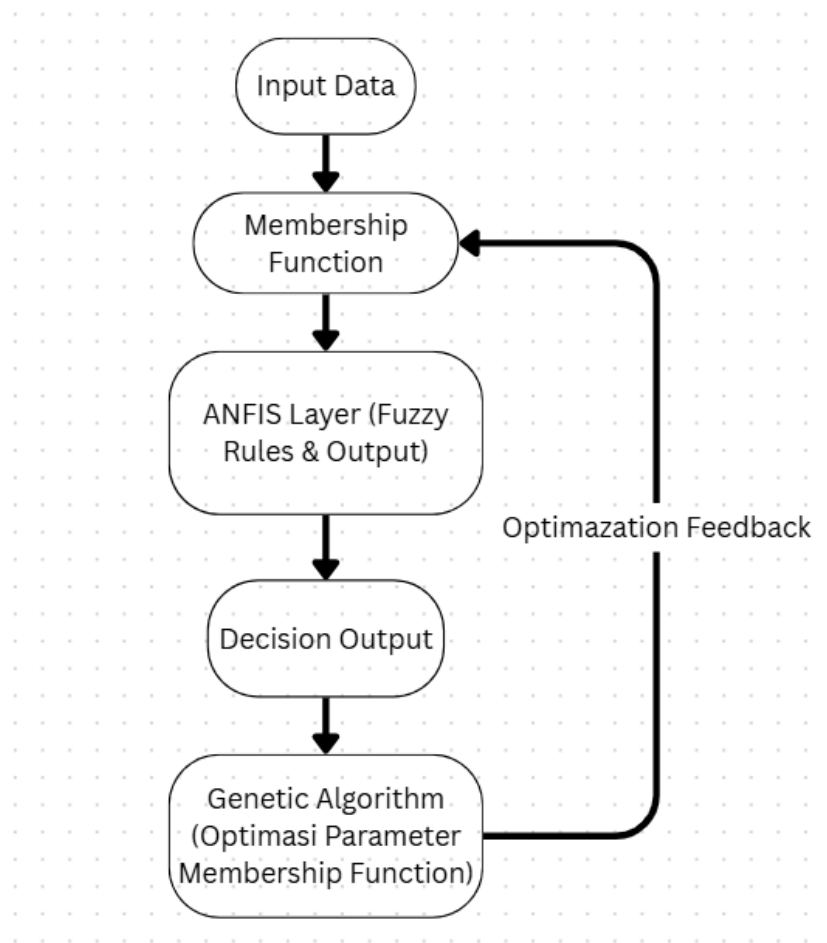
2.1.6 Keunggulan ANFIS dibandingkan GA Murni

Walau GA dapat digunakan untuk menyelesaikan berbagai masalah, ANFIS menawarkan beberapa keunggulan:

- **Struktur adaptif:** ANFIS belajar dari data, bukan hanya melakukan pencarian global seperti GA [2].
- **Kombinasi pembelajaran dan logika:** ANFIS menggabungkan *rule-based* reasoning dan pembelajaran parameter [3].
- **Hasil lebih presisi:** Karena memanfaatkan data *training* langsung [1].

Karena itu, dalam konteks game yang membutuhkan pengambilan keputusan cepat dan adaptif, ANFIS dipilih sebagai pendekatan utama dalam penelitian ini [1] [2].





Gambar 2.4. Diagram hubungan antara ANFIS, MF dan GA

Diagram di atas menunjukkan hubungan antara ANFIS, MF dan GA. Berikut ini adalah penjelasan dari setiap komponen:

1. **Input Data:** Merupakan parameter masukan dari lingkungan permainan seperti *Enemy Range*, *Energy*, *Health*, *Ally Position*, dan *Attack Priority*. Kelima parameter ini dipilih karena memiliki pengaruh signifikan dalam pengambilan keputusan AI dalam permainan strategi waktu nyata.
2. **Membership Function:** Data numerik dari lingkungan dikonversi menjadi nilai *fuzzy* menggunakan MF seperti *triangular* atau *trapezoidal*. Fungsi ini menentukan derajat *membership* variabel dalam setiap kategori *fuzzy* [1].
3. **ANFIS Layer (Fuzzy Rules & Output):** ANFIS menggabungkan pembelajaran dari jaringan saraf dan penalaran dari sistem *fuzzy* untuk memproses *input fuzzy*, menerapkan aturan *fuzzy* dan menghasilkan *output*

keputusan. Bobot dan parameter dalam struktur ANFIS diperbarui secara adaptif [3].

4. **Decision Output:** Output dari ANFIS digunakan untuk mengambil tindakan dalam game, misalnya memilih unit mana yang akan dipanggil berdasarkan nilai desirability tertinggi. Hal ini membuat perilaku AI lebih responsif dan kontekstual.
5. **Genetic Algorithm:** GA digunakan sebagai metode optimasi untuk menyempurnakan parameter MF. GA dapat menemukan konfigurasi optimal melalui proses seleksi, crossover, dan mutasi, sehingga meningkatkan akurasi sistem [1].
6. **Optimization Feedback:** Hasil evaluasi dari GA digunakan untuk memberikan umpan balik ke proses *fuzzifikasi* dan ANFIS. Parameter yang kurang optimal disesuaikan secara iteratif hingga mencapai performa maksimal.

ANFIS bertindak sebagai inti sistem inferensi dan pembelajaran. MF adalah antarmuka antara data *real* dan sistem *fuzzy*. GA bertugas menyempurnakan ANFIS dengan cara mengoptimalkan MF agar menghasilkan keputusan yang lebih baik. Sistem seperti ini sangat berguna dalam lingkungan dinamis seperti game *Real-Time Strategy*, karena mampu beradaptasi dan belajar dari lingkungan [3] [1].

2.1.7 Real-time Strategy

RTS adalah salah satu *genre* game yang menuntut pemain untuk membuat keputusan strategis secara cepat dalam lingkungan yang terus berubah [2]. Genre ini mengharuskan pemain untuk mengelola sumber daya, *unit* dan taktik dalam waktu nyata, sehingga menjadi tantangan yang kompleks baik bagi pemain manusia maupun sistem kecerdasan buatan (AI) [8].

Dalam konteks pengembangan AI untuk RTS, tantangan utama terletak pada kemampuan sistem untuk merespons secara adaptif terhadap situasi yang dinamis. Pendekatan berbasis *fuzzy* yang mampu menangani ketidakpastian, dapat menjadi solusi efektif untuk pengambilan keputusan dalam game RTS [2] [6]. Pendekatan ini memungkinkan AI untuk beroperasi dengan logika yang lebih menyerupai cara berpikir manusia seperti memperkirakan ancaman atau memprioritaskan tindakan berdasarkan kondisi permainan [6].

2.1.8 Perbandingan Penelitian Terkait

Beberapa penelitian sebelumnya telah mengimplementasikan ANFIS dalam konteks game, khususnya pada genre RTS. Salah satunya adalah penelitian oleh Andy [1], yang memanfaatkan ANFIS untuk pengambilan keputusan dalam game RTS berbasis Unity. Penelitian tersebut menggunakan hanya dua variabel input sebagai dasar inferensi, yaitu *Enemy Range* dan *Energy Amount*. Penelitian tersebut menunjukkan potensi penggunaan ANFIS, namun masih terbatas dalam kompleksitas konteks permainan yang dimodelkan.

Penelitian lain oleh Navneet [5] menggunakan ANFIS untuk pengendalian *robot navigation* dengan fokus pada penghindaran rintangan. Meskipun struktur ANFIS diterapkan secara baik, sistem tersebut belum dioptimasi menggunakan pendekatan *machine learning* atau algoritma evolusioner seperti GA.

Kontribusi utama dari penelitian ini dibandingkan kedua studi tersebut adalah sebagai berikut:

1. **Penambahan jumlah variabel input:** Sistem ANFIS yang dikembangkan dalam penelitian ini menggunakan lima variabel *input*, yaitu *Enemy Range*, *Energy*, *Health*, *Ally Position*, dan *Priority*. Hal ini meningkatkan kompleksitas dan konteks pengambilan keputusan dibandingkan hanya dua variabel pada Andy [1].
2. **Struktur ANFIS yang disesuaikan dengan data game real-time:** Inferensi fuzzy dirancang berdasarkan dinamika aktual permainan, di mana setiap nilai input merepresentasikan kondisi nyata dari medan pertempuran.
3. **Potensi optimalisasi menggunakan GA:** Sistem membuka kemungkinan untuk mengoptimalkan parameter MF seperti a_i , b_i , dan c_i menggunakan pendekatan GA, sebagaimana disarankan dalam literatur [10] [9].

Dengan peningkatan ini, sistem ANFIS yang dikembangkan diharapkan mampu menghasilkan keputusan yang lebih adaptif dan akurat dalam konteks permainan RTS yang kompleks [1] [5] [10].

A Terhadap Pustaka Lain

Posisi terhadap Pustaka Lain: Jang adalah dasar arsitektur ANFIS yang digunakan dalam penelitian ini [3].

Zadeh dan Volna memberi dasar teori *fuzzy* dan MF yang diterapkan di penelitian ini dalam layer pertama ANFIS [4] [6].

Hao dan Kristo membahas penggunaan GA untuk optimasi parameter *fuzzy*, yang dijadikan inspirasi dalam penelitian yang sedang dikembangkan untuk pengembangan ANFIS ke depan secara otomatis [10] [9].

Andy adalah yang paling relevan sebagai pembanding langsung. Dalam penelitian ini sedang mengembangkan karyanya dengan memperbanyak input dan kompleksitas situasi game [1].

Navneet juga menggunakan ANFIS, tetapi konteksnya lebih ke sistem kontrol, bukan lingkungan dinamis seperti game [5].

Lucas dan Anderson membahas konsep AI dalam game, menjadi dasar pemikiran mengapa pemilihan AI seperti ANFIS penting dalam strategi adaptif [2] [8].

Lopez menunjukkan fleksibilitas fuzzy dalam lingkungan kompleks [7].

Kaban memberi referensi dalam implementasi ANFIS berbasis Unity, namun dalam genre RPG [11].

