

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

2.1.1 Algoritma Genetika

2.1.1.1 Pengertian Algoritma Genetika

Algoritma Genetika adalah metode optimasi yang didasarkan pada teori evolusi biologi, terutama prinsip seleksi alam oleh Charles Darwin. Proses ini melibatkan mekanisme seleksi, *crossover* dan mutasi untuk mencari solusi optimal dalam permasalahan kompleks (SILITONGA, 2021). Selain itu, *Genetic Algorithm* (GA) juga diterapkan dalam pengoptimalan parameter sistem *fuzzy*, seperti *Fuzzy Tsukamoto*, melalui evolusi populasi secara bertahap untuk memperbaiki parameter dan menemukan solusi terbaik (Herdi Juningsih & Aziz, 2024).

2.1.1.2 Struktur Dasar dalam Algoritma Genetika

1. Populasi

Sekumpulan individu atau solusi potensial. Setiap individu di dalam populasi disebut kromosom.

2. Kromosom

Representasi dari solusi yang terdiri dari sekumpulan gen. Kromosom bisa berbentuk *string biner*, *string real*, atau pengkodean lainnya tergantung masalah yang dihadapi.

3. Gen

Elemen dasar dari kromosom yang menyimpan nilai atau karakteristik tertentu. Gen dapat berupa bit, angka, atau *string* yang mewakili variabel-variabel dari solusi.

2.1.1.3 Teknik Pengkodean dalam Algoritma Genetika

Pengkodean adalah proses representasi solusi dalam bentuk kromosom, teknik pengkodean yang digunakan sangat memengaruhi kinerja algoritma genetika karena menentukan bagaimana solusi dapat dieksplorasi dan dieksploitasi dalam ruang pencarian. Berdasarkan buku berjudul *Genetic Algorithms in Search, Optimization, and Machine Learning* oleh David E. Goldberg (1989), teknik pengkodean dalam algoritma genetika dapat disesuaikan dengan masalah yang dihadapi. Teknik pengkodean ini mencakup *binary encoding*, *permutation encoding*, *value encoding*, dan *nonbinary encoding*.

Tabel 2.1 Contoh Teknik Pengkodean yang Umum Digunakan

Teknik Pengkodean	Deskripsi	Contoh Kromosom
<i>Binary Encoding</i>	Menggunakan <i>string</i> biner untuk representasi.	101010
<i>Permutation Encoding</i>	Setiap kromosom adalah serangkaian angka yang menyatakan urutan solusi tertentu.	3-1-4-2-5
<i>Value Encoding</i>	Memanfaatkan nilai <i>numerik</i> yang sesuai dengan variabel solusi.	2.45 5.32 6.21
<i>Nonbinary Encoding</i>	Memanfaatkan alfabet yang lebih luas untuk merepresentasikan kromosom, misalnya huruf A-Z atau angka 0-9.	A B C D

Berdasarkan Tabel 2.1, berbagai teknik pengkodean dapat digunakan untuk merepresentasikan solusi dalam algoritma genetika. Teknik seperti *binary*

encoding sering digunakan dalam masalah kombinatorial, sementara *value encoding* lebih sesuai untuk masalah dengan variabel kontinu. Penggunaan teknik pengkodean yang tepat dapat meningkatkan kemampuan algoritma genetika dalam menemukan solusi optimal sebagaimana dijelaskan oleh David E. Goldberg (1989) bahwa representasi solusi yang baik akan memengaruhi efisiensi algoritma genetika.

2.1.1.4 Proses dalam Algoritma Genetika (AG)

1. Inisialisasi Populasi

Populasi awal dihasilkan secara acak untuk menciptakan variasi dalam solusi yang akan diproses, variasi ini penting agar AG dapat mengeksplorasi berbagai kemungkinan solusi dalam ruang pencarian.

2. Evaluasi *Fitness*

Kromosom dalam populasi dievaluasi menggunakan fungsi *fitness* untuk menilai seberapa baik solusi yang diwakili oleh kromosom tersebut. Nilai *fitness* menentukan kualitas kromosom dan semakin tinggi nilai *fitness* semakin besar peluang kromosom tersebut untuk dipilih dalam proses seleksi.

3. Seleksi

Seleksi adalah proses pemilihan individu terbaik dari populasi berdasarkan nilai *fitness*, beberapa metode seleksi yang umum digunakan dalam AG adalah.

a. *Roulette Wheel Selection*

Individu dipilih berdasarkan probabilitas yang berbanding lurus dengan nilai *fitness*, semakin tinggi *fitness* semakin besar probabilitas individu

tersebut untuk dipilih.

b. *Linear Ranking Selection*

Setiap individu diurutkan berdasarkan nilai *fitness* dan probabilitas pemilihan diatur sesuai peringkat.

c. *Tournament Selection*

Sekelompok individu dipilih secara acak dan individu dengan nilai *fitness* tertinggi dalam kelompok tersebut dipilih untuk melanjutkan ke generasi berikutnya.

2.1.1.5 Proses *Crossover* dalam Algoritma Genetika

Crossover adalah proses kombinasi gen dari dua *parent* untuk menghasilkan *offspring* baru, proses ini bertujuan menciptakan variasi genetik dalam populasi sehingga solusi yang lebih baik dapat ditemukan. Berdasarkan buku berjudul Genetic Algorithms in Search, Optimization, and Machine Learning oleh David E. Goldberg (1989), *crossover* adalah salah satu operator penting dalam algoritma genetika. Goldberg menjelaskan bahwa probabilitas *crossover* (*pcross*) menentukan apakah pasangan *parent* akan menjalani proses *crossover* atau tidak. Jika probabilitas terpenuhi, maka sebuah titik potong akan dipilih secara acak dalam kromosom untuk melaksanakan *crossover*.

Tabel 2.2 Teknik *Crossover*

Jenis <i>Crossover</i>	Deskripsi	Contoh <i>Parent</i> dan <i>Offspring</i>
<i>One-Point Crossover</i>	Memilih satu titik potong dan menukar gen setelah titik tersebut antara dua <i>parent</i> .	<i>Parent</i> 1: `10101

Jenis <i>Crossover</i>	Deskripsi	Contoh <i>Parent</i> dan <i>Offspring</i>
<i>Two-Point Crossover</i>	Memilih dua titik potong dan menukar gen di antara kedua titik tersebut.	<i>Parent 1</i> : `10
<i>Uniform Crossover</i>	Setiap gen <i>offspring</i> diambil dari salah satu <i>parent</i> dengan probabilitas yang sama.	<i>Parent 1</i> : 1010110, <i>Parent 2</i> : 0101101 <i>Offspring</i> : 1101110
<i>Shuffle Crossover</i>	Mengacak gen pada <i>parent</i> sebelum memilih titik potong.	<i>Parent 1</i> : `1
<i>K-Point Crossover</i>	Memilih lebih dari dua titik potong, dengan variasi yang lebih tinggi dari <i>crossover</i> lainnya.	<i>Parent 1</i> : `10

Berdasarkan Tabel 2.2, terdapat beberapa teknik *crossover* yang digunakan dalam algoritma genetika, seperti *One-Point Crossover* dan *Uniform Crossover* yang masing-masing memiliki metode penggabungan gen yang berbeda. Teknik-teknik ini memungkinkan variasi genetik yang lebih luas dalam proses pembentukan *offspring*.

2.1.1.6 Proses Mutasi dalam Algoritma Genetika

Mutasi adalah proses perubahan gen secara acak dalam kromosom, mutasi bertujuan untuk menjaga keberagaman genetik dalam populasi dan mencegah algoritma terjebak dalam solusi lokal. Berdasarkan buku Goldberg (1989), probabilitas mutasi (*pmutation*) menentukan apakah suatu gen akan dimutasi atau tidak. Mutasi dilakukan dengan menggunakan fungsi acak untuk memutuskan apakah nilai gen diubah atau dibiarkan tetap.

Tabel 2.3 Teknik Mutasi

Jenis Mutasi	Deskripsi	Contoh Kromosom Sebelum & Sesudah Mutasi
<i>Bit-Flip Mutation</i>	Mengubah nilai bit dari 0 ke 1 atau sebaliknya.	Sebelum: 101010, Sesudah: 101110
<i>Swap Mutation</i>	Menukar dua gen dalam kromosom.	Sebelum: 12345, Sesudah: 14325
<i>Inversion Mutation</i>	Memutar balik urutan gen dalam segmen tertentu pada kromosom.	Sebelum: 12345, Sesudah: 14325
<i>Insertion Mutation</i>	Memasukkan gen ke posisi lain di dalam kromosom.	Sebelum: 12345, Sesudah: 12435
<i>Gaussian Mutation</i>	Digunakan dalam <i>encoding</i> nilai <i>real</i> dengan menambahkan nilai acak dari distribusi Gaussian.	Sebelum: 3.45, Sesudah: 3.47

Berdasarkan Tabel 2.3, terdapat berbagai jenis teknik mutasi yang digunakan dalam algoritma genetika. Teknik-teknik tersebut seperti *Bit-Flip Mutation* dan *Gaussian Mutation* berfungsi untuk menjaga keberagaman genetik dan mencegah algoritma dari konvergensi prematur.

2.1.1.7 Proses Crossover dan Mutasi

1. Crossover dengan Rata-Rata Tertimbang

Teknik *crossover* yang digunakan dalam implementasi algoritma genetika pada penelitian ini mengombinasikan gen dari dua individu (*parent*) menggunakan rata-rata tertimbang (*weighted average*), nilai gen *offspring* dihasilkan dengan bobot yang telah ditentukan, misalnya:

- a. Nilai gen untuk skor *soft skills* dihitung sebagai 60% dari *parent* 1 dan 40% dari *parent* 2.

- b. Nilai gen untuk skor wawancara dihitung sebagai 40% dari *parent* 1 dan 60% dari *parent* 2.

2. Mutasi dengan Penyesuaian Nilai Acak

Teknik mutasi dalam kode program dilakukan dengan menambahkan nilai acak ke gen individu, rentang nilai acak ditentukan untuk menghindari mutasi yang menghasilkan gen di luar batas yang diperbolehkan (0–100). Sebagai contoh:

- a. Mutasi untuk skor *soft skills* menggunakan rentang ± 3 .
- b. Mutasi untuk skor wawancara menggunakan rentang ± 4 .

Hal ini bertujuan untuk mempertahankan keberagaman genetik dalam populasi dan menjaga nilai gen tetap valid.

2.1.1.8 Alur Kerja Algoritma Genetika

Proses kerja algoritma genetika secara umum dapat digambarkan dalam siklus.

1. Inisialisasi Populasi: Populasi awal yang terdiri dari kromosom dihasilkan secara acak.
2. Evaluasi *fitness*: Setiap kromosom dievaluasi untuk menentukan kualitasnya.
3. Seleksi: Individu dengan nilai *fitness* terbaik dipilih sebagai *parent*.
4. *Crossover*: *Parent* yang terpilih menghasilkan *offspring* melalui proses *crossover*.
5. Mutasi: *Offspring* mengalami mutasi untuk menjaga keberagaman genetik.
 - a. Evaluasi Ulang dan Pengulangan

Langkah a - e diulangi hingga kondisi berhenti terpenuhi (misalnya, batas generasi atau nilai *fitness* maksimum).

2.1.1.9 Keunggulan dan Kelemahan Algoritma Genetika

1. Keunggulan

a. Kemampuan Mengatasi Ruang Solusi yang Luas

Algoritma genetika mampu mengeksplorasi ruang solusi yang sangat besar, sehingga memiliki kemungkinan yang tinggi untuk menemukan solusi optimal atau mendekati optimal dalam masalah yang kompleks.

b. Adaptabilitas

Algoritma genetika dapat diadaptasi untuk berbagai jenis pengkodean dan masalah, mulai dari biner hingga *real number encoding*, serta dapat digunakan untuk berbagai aplikasi seperti pemrosesan citra, penjadwalan, dan optimasi jaringan.

c. Toleransi terhadap Ketidaktepatan Data

Algoritma genetika memiliki toleransi terhadap ketidakpastian dan ketidaktepatan dalam data, yang berarti algoritma ini tetap efektif meskipun terdapat variasi atau *noise* dalam data.

d. Kemampuan Paralelisasi

Tahapan seleksi, *crossover*, dan mutasi dalam algoritma genetika dapat dilakukan secara paralel, sehingga algoritma ini dapat dioptimalkan lebih lanjut dengan menggunakan pendekatan pemrograman paralel, seperti pada sistem komputer *multicore* atau komputasi awan.

2. Kelemahan

a. Potensi Konvergensi Prematur

Algoritma genetika cenderung dapat mengalami konvergensi prematur pada solusi lokal, terutama jika parameter seleksi, *crossover*, dan mutasi tidak diatur dengan baik.

b. Waktu Eksekusi yang Panjang

Proses iteratif dalam algoritma genetika, terutama jika ruang solusi besar atau kompleks dapat memakan waktu cukup lama dan memerlukan sumber daya komputasi yang tinggi.

c. Kesulitan dalam Menentukan Parameter yang Optimal

Keefektifan algoritma genetika sangat bergantung pada parameter seperti ukuran populasi, tingkat *crossover* dan tingkat mutasi. Menentukan kombinasi parameter yang optimal sering kali memerlukan eksperimen berulang.

d. Kurang Efisien untuk Masalah yang Mudah atau Deterministik

Pada masalah yang memiliki solusi deterministik dan sederhana, AG sering kali kurang efisien dibandingkan dengan metode konvensional seperti pemrograman *linier* atau metode *numerik*.

2.1.1.10 Contoh Implementasi

Contoh implementasi sederhana pada masalah optimasi dalam pemilihan jalur terpendek (*Shortest Path Problem*) dalam graf yang relevan misalnya pada aplikasi jaringan atau pemetaan.

1. Inisialisasi Populasi

Contoh memiliki graf yang terdiri dari beberapa titik dan setiap titik terhubung oleh lintasan dengan bobot tertentu (jarak atau waktu), populasi awal dihasilkan secara acak di mana setiap kromosom merepresentasikan urutan titik yang dihubungkan untuk mencapai tujuan.

2. Pengkodean Kromosom

Berdasarkan kasus ini, setiap kromosom adalah urutan titik dari titik awal ke titik tujuan, misalnya kromosom [A, B, D, E] menunjukkan urutan titik yang dilewati untuk mencapai tujuan.

3. Evaluasi *Fitness*

Evaluasi *fitness* dilakukan dengan total jarak yang ditempuh pada jalur yang direpresentasikan oleh setiap kromosom, kromosom dengan total jarak lebih pendek memiliki nilai *fitness* yang lebih tinggi.

4. Seleksi

Seleksi dilakukan dengan memilih kromosom berdasarkan nilai *fitness*, individu dengan *fitness* lebih tinggi memiliki peluang lebih besar untuk dipilih sebagai *parent* dalam tahap *crossover*.

5. *Crossover*

Contoh memilih *one-point crossover* untuk menghasilkan *offspring* dari dua *parent*, titik potong dipilih secara acak dan *segment* kromosom setelah titik potong ditukar antara dua *parent*.

1. *Parent* 1: [A, B, C, D]

2. *Parent* 2: [A, C, B, D]

3. *Offspring*: [A, B, B, D] dan [A, C, C, D]

6. Mutasi

Mutasi dilakukan untuk mengubah posisi titik secara acak dalam kromosom *offspring*. Misalnya, pada *offspring* [A, B, C, D], titik B dan C dipertukarkan menjadi [A, C, B, D] untuk menciptakan jalur baru.

7. Evaluasi Ulang dan Konvergensi

Setelah *crossover* dan mutasi, setiap *offspring* dievaluasi kembali untuk menghitung *fitness*-nya. Proses ini diulang hingga ditemukan solusi terbaik atau hingga batas iterasi terpenuhi.

Tabel 2.4 Contoh Hasil Akhir

Generasi	Kromosom Terbaik	Total Jarak
1	[A, B, C, D]	15
2	[A, B, D, C]	13
3	[A, C, B, D]	11
...
n	[A, D, C, B]	9

Berdasarkan Tabel 2.4, hasil evaluasi proses *crossover* dan mutasi yang dilakukan secara iteratif, tabel tersebut menampilkan kromosom terbaik pada setiap generasi beserta total jarak yang dihasilkan. Evaluasi ini bertujuan untuk menentukan jalur dengan total jarak terpendek yang merepresentasikan solusi optimal dalam algoritma genetika.

2.1.1.11 Fungsi Objektif dalam Algoritma Genetika

Fungsi objektif dalam algoritma genetika menilai kualitas individu dalam populasi dan mengukur sejauh mana solusi memenuhi kriteria optimasi. Berdasarkan penelitian *Enhanced Genetic Algorithm for Indoor Low-Illumination Stereo Matching Energy Function Optimization* (Hongjin & Hui, 2025), fungsi ini umumnya berbentuk fungsi energi yang terdiri dari data *term* dan *smoothing term*, dengan formula:

$$ENER = \sum_{i \in \Omega} \{Weight_i * Data_i\} + \sum_{i,j \in \Omega} \{Corr_{i,j} * Smooth_{i,j}\} \quad (1)$$

Penjelasan :

- ENER* adalah fungsi energi yang akan diminimalkan dalam proses optimasi.
- $Data_i$ adalah nilai kesalahan yang mengukur perbedaan hasil optimasi dan *ground truth*.
- $Smooth_{i,j}$ adalah *regularization term* yang memastikan kelancaran hasil optimasi dengan mempertimbangkan korelasi antar-piksel.
- W_{ei} dan $Corr_{i,j}$ adalah bobot dari masing-masing *term*, yang digunakan untuk mengatur keseimbangan antara akurasi data dan kehalusan hasil.

Berdasarkan konteks seleksi kandidat, fungsi objektif dapat disesuaikan dengan perhitungan berbobot dari berbagai aspek evaluasi.

$$F(x) = w1 \times score_test + w2 \times sof_skills + w3 \times score_interviews \quad (1)$$

Keterangan:

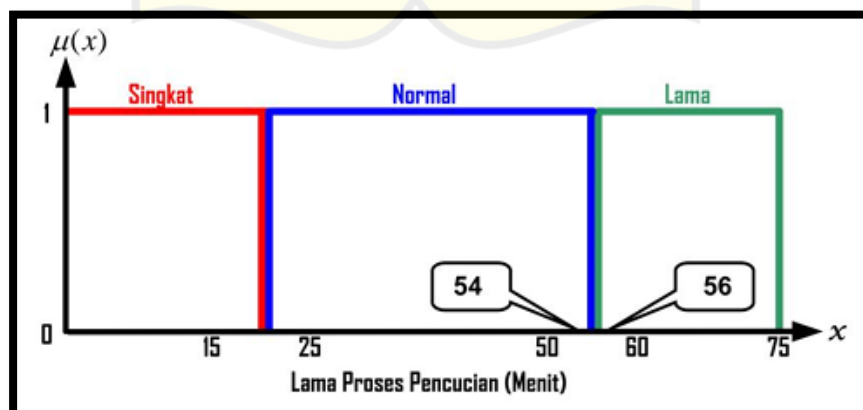
- $w1, w2, w3$ adalah bobot kepentingan dari masing-masing kriteria.
- $score_test, soft_skills, score_interviews$ merupakan nilai yang diperoleh dari hasil evaluasi kandidat.

2.1.2 Logika *Fuzzy*

2.1.2.1 Pengertian Logika *Fuzzy*

Logika *Fuzzy* dikembangkan oleh Lotfi Zadeh pada tahun 1965 untuk menangani ketidakpastian dalam data dan informasi, terutama untuk menangkap kompleksitas yang tidak dapat digambarkan dengan tegas dalam sistem logika klasik berbasis nilai biner. Berbeda dengan logika tradisional yang hanya mengenal dua nilai absolut (benar atau salah), logika *fuzzy* memungkinkan adanya rentang nilai kebenaran antara 0 dan 1, seperti "mungkin" atau "cukup benar," sehingga lebih akurat dalam menangani informasi yang bersifat ambigu atau tidak pasti (Sulandari & Yudhanto, 2020),(Dr. Eng. Anik Nur H dkk., 2021).

Berdasarkan konteks sistem kompleks, logika *fuzzy* menawarkan cara yang fleksibel untuk memetakan ruang *input* ke ruang *output*, terutama ketika parameter sistem sulit didefinisikan secara matematis. Penambahan *input* baru dalam sistem *fuzzy* hanya memerlukan fungsi keanggotaan dan aturan tambahan, tanpa memengaruhi struktur sistem secara keseluruhan, menjadikannya solusi yang ideal untuk sistem dengan ketidakpastian tinggi (Wijaya dkk., 2021).



Gambar 2.1 Keanggotaan Masukan dalam Logika *Boolean*

Berdasarkan contoh dari Gambar 2.1, dalam penentuan lama proses pencucian, logika *boolean* hanya bisa menentukan waktu secara absolut sebagai “singkat”, “normal”, atau “lama” tanpa adanya perbedaan nilai keanggotaan yang terletak di antara kategori tersebut. Sedangkan pada Logika *Fuzzy*, tingkat keanggotaan suatu nilai bisa berkisar dari 0 hingga 1, yang menggambarkan derajat keanggotaan pada kategori tertentu. Dengan demikian, Logika *Fuzzy* memungkinkan deskripsi yang lebih fleksibel dan realistis untuk data yang bersifat kontinu.

2.1.2.2 Metode *Fuzzy Sugeno*

Fuzzy Sugeno adalah metode dalam sistem inferensi *fuzzy* yang menghasilkan *output* berupa konstanta atau persamaan linier. Berbeda dengan metode Mamdani yang menghasilkan *output* dalam bentuk himpunan *fuzzy*, *Fuzzy Sugeno* langsung menghasilkan nilai tegas tanpa memerlukan proses defuzzifikasi tambahan. Metode ini sangat sesuai untuk pemodelan sistem, identifikasi sistem, prediksi, dan kontrol otomatis yang membutuhkan keluaran *numerik*.

Berdasarkan implementasinya, metode Sugeno menggunakan aturan berbasis *IF-THEN* dengan konsekuen berbentuk fungsi matematis, *output* dihitung berdasarkan bobot atau konstanta yang menghubungkan nilai keanggotaan variabel *input* dengan *output*. Hal ini membuat metode *Sugeno* lebih efisien untuk sistem yang kompleks karena setiap tambahan *input* baru hanya memerlukan fungsi keanggotaan dan aturan baru tanpa memengaruhi struktur yang sudah ada (Sihombing, 2024).

2.1.2.3 Operator Dasar dalam Logika *Fuzzy*

Beberapa operator yang digunakan untuk melakukan operasi logika.

1. *Operator AND (Minimum)*

Nilai keanggotaan pada himpunan *fuzzy* hasil ditentukan oleh nilai minimum dari nilai keanggotaan kedua himpunan *fuzzy*.

2. *Operator OR (Maximum)*

Nilai keanggotaan pada himpunan *fuzzy* hasil ditentukan oleh nilai maksimum dari kedua nilai keanggotaan himpunan *fuzzy*.

3. Negasi (Komplemen)

Digunakan untuk membalik nilai keanggotaan, di mana nilai keanggotaan 0 menjadi 1 dan sebaliknya.

2.1.2.4 Grafik Fungsi Keanggotaan *Fuzzy Sugeno*

Fungsi keanggotaan *fuzzy* digunakan untuk merepresentasikan derajat keanggotaan suatu nilai *input* ke dalam kategori *fuzzy* tertentu, berdasarkan penelitian Implementasi *Artificial Intelligence* Rekrutmen Karyawan Perusahaan dengan Konsep *Fuzzy Sugeno* oleh (Yendrizar, 2023), metode Sugeno menggunakan fungsi keanggotaan berbentuk linear, yaitu segitiga dan trapesium. Fungsi segitiga dipilih karena kesederhanaannya dalam merepresentasikan nilai sedang, sedangkan fungsi trapesium digunakan untuk merepresentasikan nilai ekstrem (rendah dan tinggi).

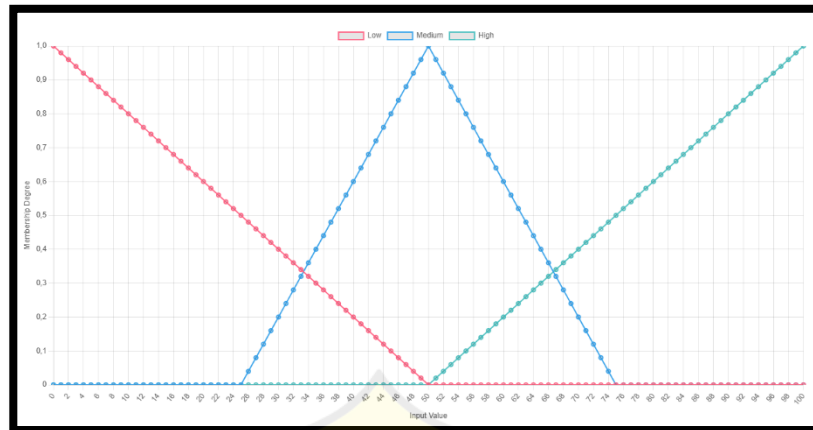
Berdasarkan penelitian ini, sistem logika *fuzzy* dirancang menggunakan rentang nilai untuk kategori *Low*, *Medium*, dan *High* yang diterapkan pada tiga

variabel utama: Skor Kompetensi, *Soft Skills*, dan Wawancara. Sebagaimana terlihat pada Gambar 2.2 berikut, fungsi keanggotaan *fuzzy* dalam sistem logika *fuzzy* penelitian ini direpresentasikan dengan rentang berikut:

- a. *Low*: Nilai derajat keanggotaan menurun secara linear dari 1 ke 0 untuk nilai input dalam rentang tertentu.
- b. *Medium*: Fungsi keanggotaan segitiga menunjukkan nilai derajat keanggotaan yang naik dari 0 ke 1 dan kembali turun ke 0, sesuai dengan nilai *input* yang berada di rentang menengah.
- c. *High*: Nilai derajat keanggotaan meningkat secara linear dari 0 ke 1 untuk nilai input tinggi.

Rentang nilai ini ditentukan berdasarkan hasil implementasi dari sistem *fuzzy* dan penelitian sebelumnya, penentuan rentang nilai dilakukan dengan mengacu pada pendekatan yang dijelaskan dalam penelitian Implementasi *Artificial Intelligence* Rekrutmen Karyawan Perusahaan dengan Konsep *Fuzzy* Sugeno, serta disesuaikan dengan konteks penelitian ini untuk memastikan akurasi hasil.

Gambar grafik fungsi keanggotaan dihasilkan berdasarkan implementasi menggunakan PHP dan Chart.js untuk memvisualisasikan derajat keanggotaan *fuzzy* dari ketiga variabel utama penelitian.



Gambar 2.2 Grafik Fungsi Keanggotaan *Fuzzy* Berdasarkan Sistem Penelitian

2.1.2.5 Sistem Inferensi *Fuzzy*

Sistem inferensi *fuzzy* adalah kerangka untuk mengambil keputusan atau melakukan penalaran berbasis aturan *fuzzy*. Sistem ini mencakup tiga tahap utama.

1. Fuzzifikasi

Input crisp diubah menjadi nilai *fuzzy* melalui fungsi keanggotaan, dalam penelitian ini *input* meliputi nilai kompetensi, soft skills, dan wawancara yang masing-masing diklasifikasikan menjadi kategori Low, Medium, dan High.

2. Inferensi

Menerapkan aturan *fuzzy* berbasis *IF-THEN* untuk menentukan *output fuzzy*.

Sebagai contoh:

- a. IF Kompetensi = High AND Soft Skills = Medium AND Wawancara = High THEN Output = High.
- b. IF Kompetensi = Low OR Wawancara = Low THEN Output = Low.

3. Defuzzifikasi

Proses defuzzifikasi dalam penelitian ini menggunakan metode *Weighted*

Average, yang dinyatakan dengan formula:

$$Z = \frac{\sum_{i=1}^n (\alpha_i \times z_i)}{\sum_{i=1}^n \alpha_i} \quad (1)$$

Penjelasan :

- a. Z = nilai crisp hasil defuzzifikasi.
- b. α_i = derajat keanggotaan fuzzy untuk aturan ke- i
- c. z_i = nilai konsekuensi crisp dari aturan fuzzy ke- i
- d. n = jumlah aturan fuzzy.

Rumus ini digunakan untuk menghitung skor akhir berdasarkan kontribusi berbobot dari setiap kategori fuzzy dengan bobot tertentu. Penyesuaian bobot dilakukan berdasarkan relevansi tiap kategori terhadap hasil akhir, seperti 0.2 (*Low*), 0.5 (*Medium*), dan 1.0 (*High*).

Berdasarkan penelitian *Enhanced Sugeno Fuzzy Inference System with Fuzzy AHP and Coefficient of Variation to Diagnose Cardiovascular Disease During Pregnancy* (Mariadoss & Augustin, 2023), metode *Alpha-Cut Defuzzification* juga dapat digunakan. Metode ini lebih cocok jika hasil fuzzy dinyatakan dengan rumus berikut:

$$ACD = \frac{\alpha + 2b + 2c + d}{6} \quad (1)$$

Penjelasan:

- a. a, b, c, d adalah parameter bilangan fuzzy trapezoidal.
- b. ACD adalah nilai *crisp* hasil defuzzifikasi.

2.1.2.6 Penerapan Logika *Fuzzy Sugeno*

Metode *Fuzzy Sugeno* sering diterapkan dalam berbagai bidang, termasuk pemodelan sistem, prediksi, dan kontrol otomatis, karena kemampuannya menghasilkan *output* berupa nilai *numerik* yang langsung dapat digunakan. Salah satu penerapan yang umum adalah pada sistem pendukung keputusan (*Decision Support System*) untuk menentukan jumlah produksi berdasarkan variabel biaya produksi dan permintaan.

Sebagai contoh, berdasarkan penelitian (Sihombing, 2024), metode *Sugeno* digunakan untuk menghitung jumlah produksi makanan tertentu. Jika biaya produksi dinaikkan dan jumlah permintaan diketahui, metode *Sugeno* mampu memberikan hasil berupa jumlah produksi yang optimal berdasarkan fungsi keanggotaan dan aturan yang telah ditentukan. Dengan efisiensi dan fleksibilitasnya, logika *fuzzy Sugeno* telah terbukti menjadi solusi yang ideal untuk sistem dengan tingkat ketidakpastian tinggi.

2.1.2.7 Keunggulan dan Kelemahan

Keunggulan logika *fuzzy* adalah kemampuannya untuk mengatasi ketidakpastian, menjadikannya ideal untuk permasalahan dengan data yang ambigu atau kurang pasti. Sedangkan kelemahan logika *fuzzy* terletak pada kompleksitas dalam penentuan fungsi keanggotaan dan aturan inferensi yang memerlukan keahlian serta pemahaman mendalam mengenai domain masalah.

2.1.2.8 Contoh Implementasi dalam Pengambilan Keputusan

Sebagai contoh, logika *fuzzy* digunakan untuk menilai kandidat dalam proses seleksi karyawan. Setiap kandidat dievaluasi berdasarkan skor kompetensi, kemampuan *soft skills*, dan wawancara, dengan hasil akhir dihitung menggunakan metode Sugeno. Proses ini memastikan bahwa kandidat terbaik dipilih berdasarkan kontribusi berbobot dari setiap aspek evaluasi.

2.1.3 Machine Learning

Menurut buku berjudul “Machine Learning: Concepts, Techniques and Applications” *machine learning* merupakan bagian dari AI yang melibatkan penerapan algoritma yang mampu belajar dari data atau contoh sebelumnya dan mampu melakukan tugas tanpa instruksi atau pemrograman yang jelas. Prosedur untuk belajar dari data melibatkan pengenalan pola secara statistik dan penyesuaian model untuk mengevaluasi data secara lebih akurat dan memberikan hasil yang tepat. (T V Geetha, 2023)

2.1.3.1 Cakupan Machine Learning

Machine Learning adalah cabang dari kecerdasan buatan (*Artificial Intelligence/AI*) yang memungkinkan sistem untuk belajar dan membuat keputusan atau prediksi berdasarkan data. Berikut adalah cakupan utama dari *Machine Learning*:

1. *Supervised Learning* (Pembelajaran Terawasi):
 - a. Definisi: Pendekatan di mana model dilatih menggunakan data yang telah

diberi label, yaitu data yang memiliki *input* dan *output* yang jelas.

- b. Algoritma Utama: *Regresi Linear*, *Regresi Logistik*, *Random Forest* dan *Support Vector Machine (SVM)*
2. *Unsupervised Learning* (Pembelajaran Tak Terawasi):
 - a. Definisi: Model dilatih pada data tanpa label untuk menemukan pola atau struktur yang tersembunyi.
 - b. Algoritma Utama: *Clustering* (misalnya *K-Means*, *Hierarchical Clustering*), *Principal Component Analysis (PCA)* dan *Autoencoders*.
 3. *Reinforcement Learning* (Pembelajaran Penguatan):
 - a. Definisi: Model dilatih melalui interaksi dengan lingkungan dan menggunakan sistem penghargaan (*reward*) dan hukuman (*penalty*) untuk meningkatkan kinerjanya.
 - b. Komponen Utama: *Agent* (pelaku pembelajaran), *Environment* (lingkungan tempat agent beroperasi) dan *Reward System* (penghargaan atas tindakan yang benar)

2.1.4 CRISP-DM sebagai Tahap Merancang Sistem Berbasis *Machine Learning*:

a. Pengertian CRISP-DM

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) adalah metodologi standar yang digunakan untuk merancang dan melaksanakan proyek berbasis data mining atau *machine learning*, metodologi ini terdiri dari enam tahap

utama yang bertujuan untuk memberikan panduan yang terstruktur dalam mengolah data menjadi solusi yang bermanfaat. Tahapan dalam CRISP-DM meliputi:

1. *Business Understanding*: Memahami tujuan bisnis atau permasalahan yang ingin diselesaikan.
2. *Data Understanding*: Mengumpulkan dan mengeksplorasi data untuk memahami karakteristiknya.
3. *Data Preparation*: Membersihkan dan mempersiapkan data untuk digunakan dalam model.
4. *Modeling*: Membangun model menggunakan algoritma tertentu.
5. *Evaluation*: Mengevaluasi hasil model berdasarkan metrik tertentu.
6. *Deployment*: Mengimplementasikan hasil model ke dalam sistem yang digunakan oleh pengguna.

b. Adaptasi CRISP-DM dalam Penelitian

Meskipun penelitian ini tidak melibatkan pelatihan model *machine learning*, CRISP-DM tetap dapat digunakan sebagai kerangka kerja untuk merancang sistem berbasis data. Penyesuaian tahapan CRISP-DM pada penelitian ini adalah sebagai berikut:

1. *Business Understanding*:

Tujuan penelitian ini adalah mengoptimasi proses seleksi karyawan melalui sistem berbasis *web* yang menggabungkan algoritma genetika dan logika *fuzzy*. Sistem ini dirancang untuk memberikan hasil seleksi yang akurat dan objektif dengan memanfaatkan data pelamar sebagai *input*.

2. *Data Understanding*:

Data pelamar diimpor dalam *format* *xlsx* (Microsoft Excel) yang berisi informasi seperti nama, posisi, cv dan dokumen lainnya. Data ini kemudian dimasukkan ke dalam *database* menggunakan PHP dan MySQL. Pada tahap ini data dieksplorasi untuk memastikan kelengkapan dan keakuratan nilai sebelum diolah oleh algoritma.

3. *Data Preparation*:

Data yang sudah diunggah diproses agar sesuai dengan kebutuhan sistem. Tahap ini mencakup:

- a. Validasi data, seperti memeriksa *format* dan mengatasi data kosong.
- b. Normalisasi nilai skor agar sesuai dengan rentang yang dapat digunakan oleh logika *fuzzy*.

4. *Modeling* (Adaptasi):

Penelitian ini mencakup implementasi Algoritma Genetika untuk mengoptimasi peringkat pelamar dan Logika *Fuzzy* untuk menentukan skor seleksi berbasis aturan (*rule-based*). Tidak ada proses pelatihan model yang dilakukan, tetapi logika sistem sepenuhnya bergantung pada aturan *fuzzy* dan mekanisme evolusi genetika.

5. *Evaluation*:

Evaluasi dilakukan dengan membandingkan hasil seleksi yang dihasilkan oleh sistem dengan hasil manual yang sebelumnya dilakukan oleh tim *Human Capital*. Metode evaluasi ini memastikan bahwa sistem memberikan hasil yang objektif dan sesuai ekspektasi.

6. *Deployment*:

Sistem berbasis *web* ini diimplementasikan pada *server* dan dapat diakses oleh tim *Human Capital* PT Gobel International. Data pelamar yang sudah dimasukkan ke *database* akan diproses oleh sistem dan hasil seleksi dapat langsung diakses melalui antarmuka *web*.

2.1.5 Pemodelan Sistem UML

2.1.5.1 Pengertian UML

Unified Modeling Language (UML) adalah standar pemodelan visual yang digunakan untuk menggambarkan, merancang, dan mendokumentasikan sistem perangkat lunak. UML menyediakan berbagai jenis diagram untuk merepresentasikan struktur, perilaku, dan interaksi dalam sistem.

Dalam penelitian ini, UML digunakan untuk memodelkan sistem rekrutmen karyawan berbasis Algoritma Genetika dan Logika *Fuzzy*, dengan fokus pada interaksi pengguna dan alur kerja sistem.

2.1.5.2 Jenis Diagram UML yang Digunakan

1. *Use Case* Diagram

Use Case Diagram adalah diagram UML yang digunakan untuk menggambarkan interaksi antara aktor (pengguna) dengan sistem, diagram ini menunjukkan fungsi-fungsi utama sistem yang dapat digunakan oleh pengguna.

Komponen Utama:

- a. Aktor: Entitas yang berinteraksi dengan sistem, seperti pelamar kerja atau

admin.

- b. *Use Case*: Fungsi atau layanan yang disediakan oleh sistem, seperti input data pelamar, proses seleksi, dan hasil peringkat.
- c. Relasi: Hubungan antara aktor dan use case.

Fungsi dalam penelitian:

Use Case Diagram digunakan untuk memberikan gambaran menyeluruh tentang fitur utama sistem rekrutmen berbasis *web*, serta bagaimana setiap fitur melibatkan aktor tertentu.

2. *Activity Diagram*

Activity Diagram adalah diagram UML yang digunakan untuk memodelkan alur kerja (*workflow*) atau aktivitas dalam sistem, diagram ini menunjukkan bagaimana data atau proses mengalir dari satu tahap ke tahap lainnya.

Komponen Utama:

- a. *Swimlane*: Mengelompokkan aktivitas berdasarkan aktor atau modul sistem.
- b. *Activity Node*: Representasi aktivitas atau langkah tertentu dalam alur kerja.
- c. *Decision Node*: Simbol untuk menunjukkan percabangan atau pengambilan keputusan.

Fungsi dalam penelitian:

Activity Diagram digunakan untuk menggambarkan alur kerja sistem rekrutmen, mulai dari *input* data pelamar hingga pengolahan data menggunakan algoritma genetika dan logika *fuzzy*, serta penyajian hasil seleksi.

2.1.6 Software dan Pemrograman Terkait

2.1.6.1 Software yang Digunakan:

1. FileZilla

FileZilla adalah aplikasi *File Transfer Protocol* (FTP) yang digunakan untuk mengelola transfer *file* antara komputer lokal dengan *server*.

2. Visual Studio Code

Visual Studio Code (VS Code) adalah teks editor yang dikembangkan oleh Microsoft dan dirancang untuk mendukung pengembangan perangkat lunak.

2.1.6.2 Bahasa Pemrograman yang Digunakan:

1. Pemrograman PHP (Hypertext Preprocessor)

PHP adalah bahasa pemrograman skrip yang terutama digunakan dalam pengembangan situs *web* dinamis, menurut (Elgamar, 2020) PHP telah menjadi pilihan populer di kalangan pengembang *web* karena kemudahan penggunaannya dalam mengelola konten *web*. Awalnya, PHP diciptakan pada tahun 1994 oleh Rasmus Lerdorf dengan nama *Personal Home Page*. Seiring perkembangan teknologi, nama tersebut berubah menjadi PHP, yang mengacu pada kegunaannya dalam pemrosesan data hypertext. Saat ini, pengelolaan PHP dilakukan oleh *The PHP Group*, yang terus mengembangkan bahasa ini agar tetap relevan dengan kebutuhan *web modern*.

a. Sifat dan Cara Kerja PHP sebagai Interpreter

PHP adalah bahasa pemrograman yang bersifat *interpreter*, artinya setiap

instruksi dalam kode PHP dijalankan langsung baris demi baris saat program mulai dieksekusi, interpreter dalam PHP bekerja dengan membaca dan menjalankan setiap baris kode secara berurutan tanpa proses kompilasi terlebih dahulu.

b. Sejarah dan Perkembangan Versi PHP

Sejak pertama kali dirilis, PHP telah mengalami banyak perubahan dan peningkatan versi untuk memenuhi kebutuhan yang terus berkembang di industri *web*. Dimulai dari PHP 1.0 pada tahun 1994, setiap versi berikutnya menambahkan berbagai fitur baru seperti Tabel 2.5 berikut, peningkatan performa, dan perbaikan keamanan. Berikut adalah beberapa versi utama PHP dan tahun rilisnya:

Tabel 2.5 Perkembangan Bahasa Pemrograman PHP

Versi <i>PHP</i>	Tahun Rilis
PHP 1.0	1994
PHP 2.0	1996
PHP 3.0	1998
PHP 4.0	2000
PHP 5.0	2004
PHP 6.0	2005
PHP 7.0	2015
PHP 8.0	2020

c. Keunggulan PHP sebagai Bahasa *Open Source*

Salah satu keunggulan utama PHP adalah sifatnya yang *open source*, artinya siapa saja dapat menggunakan dan memodifikasinya tanpa perlu membayar biaya lisensi, bahasa ini memiliki komunitas pengguna yang luas, yang berkontribusi

dalam pengembangan fitur baru dan menyediakan berbagai sumber daya seperti dokumentasi, modul tambahan, dan solusi untuk permasalahan teknis.

2. Pemrograman JavaScript

Menurut buku "Semua Bisa Menjadi Programmer JavaScript & Node.js", JavaScript merupakan bahasa pemrograman tingkat tinggi (*High-Level Language*) dan dinamis. JavaScript populer di internet dan dapat bekerja di sebagian besar penjelajah (*browser*) *web* populer, seperti Google Chrome, Internet Explorer, Mozilla Firefox, Netscape, dan Opera (Ir. Yuniar Supardi, 2020).

a. Keunggulan JavaScript

1. Interaktif: JavaScript memungkinkan pengembang untuk membuat halaman *web* yang dinamis dan interaktif, seperti validasi formulir, animasi, dan manipulasi elemen DOM (*Document Object Model*).
2. Kompatibilitas: Dapat dijalankan pada berbagai browser tanpa perlu instalasi tambahan.
3. Cepat: Karena berjalan langsung pada *browser*, JavaScript dapat meningkatkan kecepatan eksekusi dibandingkan bahasa pemrograman lain yang membutuhkan komunikasi dengan *server*.
4. Komunitas Besar: Dengan komunitas pengembang yang luas, banyak sumber daya, dokumentasi, dan pustaka (*library*) yang tersedia untuk mempermudah pengembangan.

b. Elemen Dasar JavaScript

1. Variabel: Tempat menyimpan data yang digunakan dalam program.
Deklarasi variabel menggunakan kata kunci *var*, *let*, atau *const*.
2. Tipe Data: JavaScript memiliki berbagai tipe data seperti:
 - a. *String*: Teks, contohnya "Hello World".
 - b. *Number*: Angka, contohnya 123.
 - c. *Boolean*: Nilai benar/salah, contohnya *true* atau *false*.
 - d. *Array*: Kumpulan data, contohnya [1, 2, 3].
 - e. *Object*: Data kompleks dengan pasangan kunci-nilai, contohnya {nama: "John", umur: 25}.
3. Operator: Simbol untuk melakukan operasi matematika, logika, atau manipulasi data.
 - a. Contoh operator aritmatika: +, -, *, /.
 - b. Contoh operator logika: && (dan), || (atau), ! (tidak).
4. Fungsi: Blok kode yang dirancang untuk menjalankan tugas tertentu.
5. Kondisi (*If-Else*): Digunakan untuk membuat keputusan berdasarkan kondisi tertentu.
6. Perulangan (*Loop*): Digunakan untuk mengulangi eksekusi kode.

c. Pustaka dan Framework JavaScript

1. Pustaka: Seperti jQuery, digunakan untuk mempermudah penulisan JavaScript.
2. Framework: Seperti React.js dan Angular.js, digunakan untuk membangun aplikasi *web* yang lebih kompleks dan terstruktur.

2.2 Kajian Penelitian Terdahulu

Paper 1 : Rancang Bangun Sistem Penunjang Keputusan Penentuan Beasiswa Menggunakan Metode *Fuzzy Tsukamoto* Dengan Optimasi *Genetic Algorithm*, (Yanto, Eka Herdit Juningsih, Faruq Aziz), JATI (Jurnal Mahasiswa Teknik Informatika), 2024, Sinta 5.

Tujuan Penelitian: Tujuan penelitian ini untuk memberikan solusi dalam penentuan penerima beasiswa dengan lebih efisien, khususnya dalam kondisi di mana terdapat ketidakpastian pada kriteria penilaian seperti prestasi akademik, kondisi ekonomi, dan faktor penunjang lainnya. Penelitian ini menggunakan kombinasi *Fuzzy Tsukamoto*, yang berfungsi untuk menangani data dengan nilai linguistik, dan algoritma genetika, yang diimplementasikan untuk mengoptimalkan nilai-nilai fuzzy berdasarkan kriteria yang diberikan.

Metodologi yang digunakan: Metodologi penelitian yang digunakan dalam jurnal ini melibatkan dua metode utama, yaitu *Fuzzy Tsukamoto* dan Algoritma Genetika. Berikut adalah penjelasan rinci mengenai kedua metode tersebut.

1. Metodologi *Fuzzy Tsukamoto*:

a. Pendekatan Logika *Fuzzy*

Metode ini efektif dalam menangani ketidakpastian dan kompleksitas, memungkinkan evaluasi berdasarkan nilai linguistik. *Fuzzy Tsukamoto* menggunakan *fuzzy inference system* untuk memproses data masukan dan

menghasilkan keputusan yang terstruktur.

b. Proses Perhitungan

Proses perhitungan melibatkan pembuatan fungsi keanggotaan (misalnya, rendah, sedang, tinggi) dan basis aturan (contoh: "Jika Nilai Rata-rata rendah dan Aktivitas Ekstrakurikuler rendah, maka Beasiswa rendah"). Derajat keanggotaan tiap aturan dihitung, dan nilai akhir diperoleh dengan formula.

$$Beasiswa = \frac{\sum(\alpha_i \times Z_i)}{\sum \alpha_i} \quad (1)$$

Berdasarkan rumus di atas α_i adalah *firing strength* setiap aturan, dan Z_i adalah konsekuen yang dihasilkan.

2. Metodologi Algoritma Genetika (GA):

a. Prinsip Dasar

Algoritma genetika adalah algoritma evolusi yang terinspirasi oleh seleksi alam, digunakan untuk optimasi. Dalam penelitian ini, algoritma genetika membantu mengoptimalkan parameter *fuzzy* seperti bobot aturan.

b. Langkah-Langkah Algoritma Genetika.

1. Inisialisasi Populasi: Membuat populasi awal kromosom secara acak.
2. Evaluasi *fitness*: Menilai kualitas solusi berdasarkan hasil yang diintegrasikan dengan *Fuzzy Tsukamoto*.
3. Operasi Genetika: Menggunakan seleksi, *crossover*, dan mutasi untuk menghasilkan generasi baru.
4. Iterasi: Proses ini diulang hingga konvergensi tercapai atau iterasi

maksimum dilampaui.

- c. Integrasi dengan *Fuzzy Tsukamoto*: Parameter hasil optimasi GA digunakan untuk memperbaiki model *Fuzzy Tsukamoto*, sehingga meningkatkan akurasi dan efisiensi sistem secara keseluruhan.

Integrasi *Fuzzy Tsukamoto* dengan Algoritma Genetika memungkinkan pengolahan data yang lebih adaptif dan optimal, terbukti dengan hasil korelasi evaluasi yang positif (nilai *Spearman* 0.88). Hal ini mendukung transparansi dan efisiensi dalam proses penentuan beasiswa.

Temuan Utama: Temuan utama dari penelitian ini adalah keberhasilan kombinasi kedua metode dalam menciptakan sistem yang lebih adaptif dan efisien, algoritma genetika memainkan peran kunci dengan mengoptimalkan aturan fuzzy melalui mekanisme evolusi yang melibatkan seleksi, crossover, dan mutasi. Proses ini dilakukan secara iteratif hingga ditemukan solusi optimal, hasil evaluasi menunjukkan bahwa integrasi ini mampu menghasilkan sistem yang lebih tepat, dengan bukti korelasi positif antara hasil sistem dan data aktual (nilai korelasi *Spearman* sebesar 0.88).

Berdasarkan pembahasan sebelumnya penelitian ini memberikan kontribusi signifikan pada pengembangan sistem penunjang keputusan yang efektif dan transparan. Dengan menggunakan metode *Fuzzy Tsukamoto* yang dioptimalkan oleh Algoritma Genetika, sistem ini mampu mengatasi tantangan kompleksitas dan ketidakpastian data.

Kelemahan Penelitian: Kelemahan penelitian ini terletak pada keterbatasan cakupan variabel kriteria yang digunakan dalam sistem penunjang keputusan, yang berfokus hanya pada beberapa parameter seperti UKT (Uang Kuliah Tunggal) dan IPK (Indeks Prestasi Kumulatif). Hal ini dapat membatasi kemampuan sistem dalam menilai kelayakan penerima beasiswa secara lebih komprehensif.

Beberapa langkah yang kemungkinan dapat diambil oleh penulis untuk mengembangkan penelitian ini.

1. Penambahan Kriteria Penilaian

Tambahkan variabel baru seperti prestasi non-akademis, latar belakang ekonomi, dan aktivitas sosial untuk menghasilkan penilaian yang lebih menyeluruh.

2. Penggunaan Infrastruktur Komputasi yang Lebih Cepat

Implementasikan sistem pada *platform cloud* atau dengan dukungan komputasi paralel untuk mengurangi waktu komputasi.

3. Peningkatan Antarmuka Pengguna (UI/UX)

Kembangkan antarmuka pengguna yang lebih intuitif dan ramah pengguna, sehingga memudahkan pengoperasian oleh pengguna.

4. Penyediaan Modul Pelatihan Pengguna

Tambahkan modul pelatihan atau panduan untuk pengguna, supaya lebih memahami cara menggunakan sistem secara optimal.

Paper 2 : Implementasi *Artificial Intelligence* Rekrutmen Karyawan Perusahaan dengan Konsep *Fuzzy Sugeno*, Yendrizal, Jurnal Teknik Informatika Unika Santo Thomas (JTIUST), 2023, Sinta 4.

Tujuan Penelitian: Tujuan dari penelitian ini adalah untuk membantu perusahaan dalam menyeleksi calon karyawan secara lebih objektif dengan mempertimbangkan kriteria yang bersifat subjektif dan kuantitatif, seperti pendidikan, pengalaman kerja, keterampilan teknis, serta soft skills.

Metodologi yang digunakan: Penelitian ini menjelaskan penggunaan metode Fuzzy Sugeno dalam pengembangan sistem rekrutmen karyawan di perusahaan. Berikut penjelasan mengenai metodologi yang digunakan.

1. Metodologi *Fuzzy Sugeno*:

- a. Digunakan untuk menangani ketidakpastian dan kompleksitas dalam pengambilan keputusan rekrutmen.
- b. Proses melibatkan empat tahap utama: *fuzzifikasi*, aplikasi fungsi implikasi, mesin inferensi, dan *defuzzifikasi*.
- c. *Fuzzifikasi*: Konversi *input numerik* (seperti nilai pendidikan, pengalaman kerja, keahlian, dan penampilan) ke dalam bentuk himpunan *fuzzy*.
- d. Aplikasi fungsi implikasi: Mengaplikasikan aturan *fuzzy* untuk memproses data yang telah di *fuzzifikasi*.
- e. Mesin inferensi: Menentukan *output* berdasarkan kombinasi aturan *IF-THEN* yang menghubungkan variabel *input* dengan *output*.

- f. *Defuzzifikasi*: Mengubah hasil *fuzzy* menjadi *output numerik* untuk keputusan diterima atau tidak.
2. Variabel *Input* dan *Output*
- a. Empat variabel *input* digunakan: Nilai pendidikan, pengalaman kerja, keahlian, dan penampilan.
 - b. Satu variabel *output* berupa keputusan akhir (diterima atau tidak).
3. Fungsi Keanggotaan
- a. Fungsi keanggotaan yang digunakan meliputi kurva segitiga dan trapesium untuk memodelkan derajat keanggotaan variabel *input*.
 - b. *Domain* untuk setiap variabel *input* diatur pada skala 0–100.
4. Aturan Inferensi
- a. Aturan inferensi dirancang untuk menghubungkan variabel *input* dengan *output* berdasarkan skenario tertentu. Contoh aturan adalah, "Jika Nilai Pendidikan Bagus dan Pengalaman Kerja Banyak, maka Keputusan Diterima."
 - b. *Operator AND* digunakan untuk menggabungkan kondisi.
5. *Defuzzifikasi*
- a. Hasil akhir dihitung menggunakan rumus rata-rata berbobot, di mana *output* keputusan diberikan dalam rentang 0 hingga 1. Jika hasilnya lebih besar dari 0,5, maka pelamar diterima, dan sebaliknya, ditolak.

Temuan Utama: Temuan utama penelitian ini menunjukkan bahwa metode Fuzzy Sugeno efektif mendukung pengambilan keputusan dalam rekrutmen karyawan, dengan menangani ketidakpastian data dan mengubah input numerik, seperti pendidikan, pengalaman, dan keahlian, menjadi keputusan diterima atau tidak. Fungsi keanggotaan berbentuk kurva segitiga dan trapesium berhasil memetakan variabel input, sementara aturan IF-THEN mempermudah proses inferensi. Tahap defuzzifikasi memungkinkan hasil fuzzy dikonversi menjadi keputusan numerik yang mudah dipahami, seperti nilai di atas 0,5 yang berarti pelamar diterima. Metode ini meningkatkan efisiensi dan objektivitas dalam menilai kandidat.

Kelemahan Penelitian: Kelemahan penelitian ini terletak pada keterbatasan aturan fuzzy yang digunakan tidak selalu relevan untuk semua situasi rekrutmen, aturan yang dibentuk mungkin kurang fleksibel dalam menangani kasus-kasus khusus atau variabel yang belum dipertimbangkan. Selain itu, meskipun metode Fuzzy Sugeno cukup efektif, penelitian ini masih bergantung pada input manual untuk memasukkan data, sehingga ada potensi kesalahan dan kurangnya efisiensi.

Beberapa langkah yang kemungkinan dapat diambil oleh penulis untuk mengembangkan penelitian ini.

1. Penambahan Aturan *Fuzzy* yang Lebih Kompleks

Kembangkan aturan *fuzzy* yang lebih komprehensif untuk mencakup skenario rekrutmen yang lebih beragam, seperti aspek kepribadian dan keterampilan interpersonal.

2. Otomatisasi *Penginputan Data*

Kembangkan fitur otomatisasi *input* data untuk mengurangi kesalahan manual, misalnya dengan menggunakan integrasi data dari CV digital atau formulir *online*.

3. Penggunaan *Platform* yang Lebih *User-Friendly*

Alihkan implementasi dari Matlab ke *platform* berbasis *web* atau *software* lain yang lebih *user-friendly* dan tidak memerlukan latar belakang teknis yang mendalam.

4. Pelatihan dan Dokumentasi Lengkap

Sediakan dokumentasi dan pelatihan untuk pengguna perusahaan agar sistem dapat diimplementasikan dengan mudah dan efisien tanpa memerlukan bantuan teknis yang intensif.

5. Pengujian dengan Dataset yang Lebih Luas

Lakukan pengujian sistem menggunakan dataset yang lebih besar dan beragam untuk memastikan keandalan dan fleksibilitas dalam berbagai kondisi rekrutmen.

Paper 3 : Penerapan *Fuzzy Tahani* untuk Pemilihan Perangkat *Smartphone* Berbasis *Website* berdasarkan Kriteria Membeli, (Hendra Wijaya, Maria Virginia, Lukman Hakim), Jurnal Teknik Informatika Unika Santo Thomas (JTIUST), 2021, Sinta 4.

Tujuan Penelitian: Tujuan penelitian ini untuk mengembangkan sebuah sistem rekomendasi berbasis Fuzzy Tahani yang membantu calon konsumen dalam memilih smartphone yang sesuai dengan kriteria, dengan banyaknya jenis dan variasi smartphone di pasaran, calon pembeli sering merasa kebingungan dalam menentukan pilihan terbaik.

Metodologi yang digunakan:

a. Pendekatan *Fuzzy Tahani*

Bedasarkan penelitian ini metode yang digunakan untuk memberikan rekomendasi pemilihan *smartphone* dengan mempertimbangkan berbagai kriteria, *Fuzzy Tahani* memanfaatkan teori himpunan *fuzzy* untuk memproses *query* data, di mana *input* dinyatakan dalam variabel linguistik seperti "murah", "sedang", dan "mahal" untuk kategori harga. Sistem berbasis *Fuzzy Tahani* ini cocok untuk menangani pencarian data yang memerlukan fleksibilitas dalam menginterpretasikan *input* berbahasa sehari-hari.

b. Kriteria yang Dipertimbangkan

Penelitian ini menggunakan beberapa variabel *input* yang menjadi dasar rekomendasi.

- a. Harga: diklasifikasikan sebagai MURAH, SEDANG, dan MAHAL.
- b. Berat: dikelompokkan menjadi RINGAN, SEDANG, dan BERAT.
- c. Resolusi Kamera: dibagi menjadi RENDAH, NORMAL, dan TINGGI.
- d. RAM: dikategorikan sebagai KECIL, SEDANG, dan BESAR.
- e. Memori Internal: diklasifikasikan ke dalam SEDIKIT, SEDANG, dan

BANYAK.

c. Proses Metodologi

a. Pembentukan Himpunan *Fuzzy*

Setiap kategori variabel memiliki fungsi keanggotaan yang ditentukan, seperti fungsi berbentuk bahu untuk nilai "murah" dan "mahal" serta fungsi segitiga untuk nilai "sedang".

b. Evaluasi Keanggotaan

Derajat keanggotaan dihitung untuk setiap kriteria *smartphone* berdasarkan data dalam *database*.

c. Proses *Query*

Input dari pengguna diolah melalui *query fuzzy* untuk mencari rekomendasi *smartphone* yang paling sesuai dengan kriteria yang diberikan.

d. Perhitungan Nilai *Fire Strength*

Berdasarkan derajat keanggotaan, sistem menghitung kekuatan rekomendasi (*fire strength*) untuk mengurutkan *smartphone* yang paling direkomendasikan.

e. Hasil Rekomendasi

Nilai *fire strength* yang tertinggi menunjukkan produk yang paling cocok sesuai kriteria *input*.

Temuan Utama: Penelitian ini menunjukkan bahwa metode *Fuzzy Tahani* efektif dalam merekomendasikan *smartphone* sesuai kriteria pengguna, seperti harga, RAM, memori internal, dan resolusi kamera. Sistem ini memproses data input berbasis variabel linguistik, seperti "murah" atau "tinggi," untuk menghitung

derajat keanggotaan setiap variabel dengan akurat. Hasil pengolahan digunakan untuk menentukan kekuatan rekomendasi (*fire strength*), menghasilkan daftar *smartphone* yang diurutkan berdasarkan kesesuaian, dengan rekomendasi utama memiliki nilai *fire strength* tertinggi.

Kelemahan Penelitian: Kelemahan penelitian ini terletak pada cakupan kriteria yang masih cukup terbatas dan penggunaan data yang masih bersifat umum, meskipun metode *Fuzzy Tahani* telah terbukti efektif untuk merekomendasikan *smartphone*, variasi kriteria yang digunakan dalam sistem belum mencakup semua faktor yang mungkin relevan bagi pengguna. Selain itu, paper ini belum mengeksplorasi penggunaan data *real-time* atau kemampuan untuk menangani pembaruan data *smartphone* secara dinamis, hal ini membatasi kemampuan sistem dalam menyesuaikan rekomendasi dengan tren pasar yang terus berkembang dan kebutuhan spesifik pengguna.

Beberapa langkah yang kemungkinan dapat diambil oleh penulis untuk mengembangkan penelitian ini.

1. Penambahan Kriteria yang Lebih Luas

Tambahkan kriteria baru seperti daya tahan baterai, fitur keamanan, kualitas layar, dan fitur tambahan lainnya agar sistem dapat memberikan rekomendasi yang lebih komprehensif.

2. Penggunaan Data *Real-Time*

Integrasikan data *real-time* dari sumber *online* agar sistem dapat menyajikan informasi yang selalu diperbarui dan relevan dengan kondisi pasar terkini.

3. Optimalisasi Algoritma

Tingkatkan efisiensi algoritma *Fuzzy Tahani* untuk mempercepat proses perhitungan dan meningkatkan respons sistem, terutama untuk penggunaan dengan basis data besar.

4. Peningkatan Akurasi Melalui Pengujian yang Lebih Luas

Lakukan pengujian dengan dataset yang lebih beragam dan besar untuk memastikan bahwa sistem dapat beradaptasi dengan berbagai jenis preferensi pengguna dan tipe *smartphone* yang berbeda.

5. Pengembangan Antarmuka Pengguna yang Lebih Interaktif

Perbarui antarmuka pengguna agar lebih ramah pengguna dan interaktif, memungkinkan pengguna untuk dengan mudah menyesuaikan kriteria dan melihat dampaknya pada hasil rekomendasi.