



**OPTIMASI FUNGSI KEANGGOTAAN FUZZY MAMDANI
MENGUNAKAN ALGORITME GENETIKA UNTUK
PENENTUAN KESESUAIAN LAHAN TANAM TEMBAKAU**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Fikri Hilman

135150201111038



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2017

PENGESAHAN

OPTIMASI FUNGSI KEANGGOTAAN FUZZY MAMDANI MENGGUNAKAN
ALGORITME GENETIKA UNTUK PENENTUAN KESESUAIAN LAHAN TANAM
TEMBAKAU

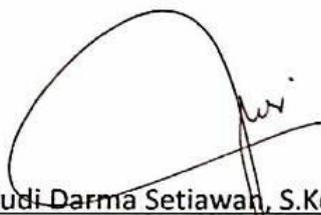
SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Fikri Hilman
135150201111038

Skripsi ini telah diuji dan dinyatakan lulus pada
31 Juli 2017
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Budi Darma Setiawan, S.Kom, M.Cs
NIP: 19841015 201404 1 002

Dosen Pembimbing II



Randy Cahya Wihandika, S.ST., M.Kom
NIK: 201405 880206 1 001

Mengetahui
Ketua Jurusan Teknik Informatika



Tn Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 3 Agustus 2017



Fikri Hilman

135150201111038



KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi dengan judul “Optimasi Fungsi Keanggotaan Fuzzy Mamdani Menggunakan Algoritme Genetika Untuk Penentuan Kesesuaian Lahan Tanam Tembakau”.

Melalui kesempatan ini penulis ingin menyampaikan terima kasih kepada semua pihak yang telah memberikan bantuan serta dukungan kepada penulis dalam menyelesaikan skripsi ini, diantaranya:

1. Kedua orang tua penulis, Alm. H. Khozin Mu'tamar & Hj. Siti Zaenab yang selalu memberikan do'a, kasih sayang, dan semangat kepada penulis.
2. Bapak Budi Darma Setiawan, S.Kom, M.Cs selaku dosen pembimbing I yang telah memberikan ilmu serta saran selama proses pengerjaan skripsi ini.
3. Bapak Randy Cahya Wihandika, S.ST., M.Kom selaku dosen pembimbing II yang juga telah memberikan ilmu serta saran selama proses pengerjaan skripsi ini.
4. Ibu Diah Priharsari, S.T, M.T selaku dosen penasihat akademik yang telah memberikan banyak kritik dan saran selama proses perkuliahan.
5. Bapak Wayan Firdaus Mahmudy, S. Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
7. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
8. Semua teman FILKOM khususnya angkatan 2013 yang telah memberikan bantuan serta dukungan selama masa perkuliahan dan pengerjaan skripsi.
9. Semua pihak yang tidak dapat disebutkan satu persatu yang terlibat baik secara langsung maupun tidak langsung yang telah membantu penulis dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan, sehingga penulis dengan terbuka menerima kritik serta saran yang membangun. Akhir kata penulis berharap skripsi ini dapat bermanfaat dan berguna bagi semua pihak yang menggunakannya.

Malang, 3 Agustus 2017

Penulis

philman94@gmail.com



ABSTRAK

Salah satu syarat agar tembakau yang dihasilkan berkualitas baik adalah penggunaan lahan yang sesuai. Kualitas lahan yang semakin baik akan berakibat pada peningkatan kualitas serta kuantitas tanaman tembakau yang dihasilkan. Kendala utama yang dialami oleh para petani tembakau dalam menentukan kesesuaian lahan adalah terbatasnya pengetahuan serta sulitnya memperoleh data yang benar tentang kualitas lahan yang sesuai untuk tanaman tembakau. Oleh karena itu, diperlukan suatu sistem yang terkomputerisasi untuk membantu petani dalam membuat keputusan terhadap calon lahan yang akan digunakan. Sistem ini diimplementasikan menggunakan *Fuzzy Inference System (FIS) Mamdani* dan dioptimasi menggunakan Algoritme Genetika. Beberapa faktor yang digunakan dalam sistem ini antara lain persentase lahan terkena penyakit, keterbukaan wilayah, derajat berat tanah, ketebalan lapis olah, kemudahan irigasi, kondisi medan, dan pH tanah. Metode reproduksi yang digunakan adalah *extended intermediate crossover* dan *random mutation*, sedangkan metode seleksi yang digunakan adalah *elitism*. Berdasarkan hasil pengujian yang telah dilakukan, solusi yang paling optimum didapatkan pada jumlah populasi sebesar 90, kombinasi nilai *cr* dan *mr* berturut-turut sebesar 0,2 dan 0,8, dan jumlah generasi sebesar 500, dengan rata-rata nilai *fitness* yang dihasilkan adalah 0,917. Akurasi yang dihasilkan oleh sistem ini adalah sebesar 80% dari 10 data uji yang digunakan.

Kata kunci: *Fuzzy Inference System (FIS) Mamdani*, Algoritme Genetika, Kesesuaian Lahan, Optimasi

ABSTRACT

One of the requirements for good quality tobacco is a good land use. Improved quality of the land will result in increased quality and quantity of tobacco plants produced. The main constraints experienced by tobacco farmers in determining land suitability are the limited knowledge and difficulty of obtaining correct data on the quality of land suitable for tobacco plants. Therefore, a computerized system is needed to assist farmers in making decisions on prospective land to be used. This system is implemented using Fuzzy Inference System (FIS) Mamdani and optimized using Genetic Algorithm. Some of the factors used in this system include the percentage of land affected by the disease, the openness of the region, the degree of weight of the soil, the thickness of the layer, the ease of irrigation, terrain conditions, and soil pH. The reproduction method used is extended intermediate crossover and random mutation, while the selection method used is elitism. Based on the results of the tests that have been done, the most optimum solution obtained on the total number for 90 population, the combination of cr and mr value of 0.2 and 0.8 respectively and the number of generations of 500, with the average of fitness value generated of 0,917. The Accuracy generated by this system is 80 % using 10 test data.

Keywords: *Fuzzy Inference System (FIS) Mamdani, Genetic Algorithm, Land Suitability, Optimization*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR KODE PROGRAM	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	3
1.3 Tujuan	4
1.4 Manfaat.....	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Kajian Pustaka.....	6
2.2 Evaluasi Kesesuaian Lahan.....	8
2.3 Logika Fuzzy	8
2.3.1 Himpunan Fuzzy.....	10
2.3.2 Fungsi Keanggotaan	10
2.4 Fuzzy Inference System Mamdani	13
2.5 Algoritme Genetika.....	16
2.5.1 Struktur Algoritme Genetika	16
2.5.2 Komponen Algoritme Genetika	16
2.5.2.1 Representasi Chromosome.....	17
2.5.2.2 Inisialisasi	17
2.5.2.3 Reproduksi	17



2.5.2.4 Evaluasi.....	19
2.5.2.5 Seleksi.....	20
2.6 Fuzzy Inference System Mamdani – Algoritme Genetika.....	20
BAB 3 METODOLOGI	21
3.1 Tahapan Penelitian	21
3.2 Studi Literatur	21
3.3 Pengumpulan Data	22
3.4 Analisis dan Perancangan	25
3.5 Implementasi	25
3.6 Pengujian dan Analisis	26
3.6.1 Pengujian Jumlah Populasi (popsiize).....	26
3.6.2 Pengujian Kombinasi Nilai Cr dan Mr.....	27
3.6.3 Pengujian Jumlah Generasi	27
3.7 Penarikan Kesimpulan	28
BAB 4 PERANCANGAN.....	29
4.1 Desain Algoritme.....	29
4.2 Siklus Penyelesaian Masalah Menggunakan Fuzzy Mamdani	30
4.3 Siklus Optimasi Fungsi Keanggotaan Fuzzy Mamdani Menggunakan Algoritme Genetika	41
4.3.1 Representasi Kromosom	41
4.3.2 Inisialisasi Parameter Algoritme Genetika.....	42
4.3.3 Inisialisasi Populasi Awal	42
4.3.4 Reproduksi	43
4.3.5 Evaluasi.....	47
4.3.6 Seleksi.....	50
4.4 Perancangan User Interface	53
4.4.1 Halaman Proses Optimasi Fungsi Keanggotaan.....	53
4.4.2 Halaman Proses Pengujian Akurasi Sistem	54
BAB 5 IMPLEMENTASI	55
5.1 Implementasi Program	55
5.1.1 Implementasi Proses Inisialiasi Populasi Awal.....	55
5.1.2 Implementasi Proses Crossover	57
5.1.3 Implementasi Proses Mutasi.....	59



5.1.4 Implementasi Proses Perhitungan Nilai Fitness.....	62
5.1.5 Implementasi Proses Seleksi Elitism	62
5.1.6 Implementasi Proses Fuzzifikasi.....	63
5.1.7 Implementasi Proses Implikasi.....	64
5.1.8 Implementasi Proses Komposisi Aturan	65
5.2 Implementasi User Interface	65
5.2.1 Implementasi Halaman Proses Optimasi Fungsi Keanggotaan...	65
5.2.2 Implementasi Halaman Proses Pengujian Akurasi Sistem.....	66
BAB 6 PENGUJIAN DAN ANALISIS.....	67
6.1 Hasil dan Analisis Pengujian Jumlah Populasi (Popsize).....	67
6.2 Hasil dan Analisis Pengujian Kombinasi Nilai Cr dan Mr	68
6.3 Hasil dan Analisis Pengujian Jumlah Generasi.....	70
6.4 Hasil Pengujian Akurasi Sistem	71
BAB 7 KESIMPULAN.....	73
7.1 Kesimpulan.....	73
7.2 Saran	73
DAFTAR PUSTAKA.....	74
LAMPIRAN	76

**DAFTAR TABEL**

Tabel 2.1 Kajian Pustaka	6
Tabel 3.1 Data Latih Lahan Tembakau	22
Tabel 3.2 Data Uji Lahan Tembakau	24
Tabel 3.3 Rancangan Pengujian Jumlah Populasi (Popsize).....	26
Tabel 3.4 Rancangan Pengujian Kombinasi Nilai Cr dan Mr	27
Tabel 3.5 Rancangan Pengujian Jumlah Generasi.....	28
Tabel 4.1 Rentang Nilai Setiap Kriteria	31
Tabel 4.2 Aturan Fuzzy	37
Tabel 4.3 Contoh Representasi Kromosom	41
Tabel 4.4 Insialisasi Populasi Awal	43
Tabel 4.5 Hasil Offspring Metode Extended Intermediate Crossover	44
Tabel 4.6 Hasil Offspring Metode Random Mutation.....	47
Tabel 4.7 Akurasi Perbandingan Kelas Pada Parent 1.....	47
Tabel 4.8 Hasil Evaluasi Nilai Fitness Setiap Individu	49
Tabel 4.9 Hasil Sorting Individu Berdasarkan Nilai Fitness	51
Tabel 4.10 Hasil Seleksi Menggunakan Metode Elitism	52
Tabel 6.1 Hasil Pengujian Jumlah Populasi (Popsize).....	67
Tabel 6.2 Hasil Pengujian Kombinasi Nilai Cr dan Mr	69
Tabel 6.3 Hasil Pengujian Jumlah Generasi.....	70
Tabel 6.4 Hasil Pengujian Akurasi Sistem.....	72



DAFTAR KODE PROGRAM

Kode Program 5.1 Proses Inialisasi Populasi Awal	56
Kode Program 5.2 Proses Crossover	58
Kode Program 5.3 Proses Mutasi	61
Kode Program 5.4 Proses Perhitungan Fitness	62
Kode Program 5.5 Proses Seleksi Elitism	63
Kode Program 5.6 Proses Fuzzifikasi	63
Kode Program 5.7 Proses Implikasi	64
Kode Program 5.8 Proses Komposisi Aturan	65



DAFTAR GAMBAR

Gambar 2.1 Representasi Linear Naik.....	11
Gambar 2.2 Representasi Linear Turun	11
Gambar 2.3 Representasi Kurva Segitiga	12
Gambar 2.4 Representasi Kurva Trapesium.....	12
Gambar 2.5 Representasi Kurva Bentuk Bahu	13
Gambar 2.6 Contoh Penerapan Metode One-Cut Point Crossover.....	18
Gambar 2.7 Contoh Penerapan Metode Reciprocal Exchange Mutation	19
Gambar 3.1 Diagram Alir Tahapan Penelitian.....	21
Gambar 3.2 Diagram Alir Hybrid Fuzzy Mamdani – Algoritme Genetika	25
Gambar 4.1 Diagram Alir Fuzzy Mamdani	29
Gambar 4.2 Diagram Alir Algoritme Genetika	30
Gambar 4.3 Grafik Fuzzy Kriteria Persentase Lahan Terkena Penyakit.....	32
Gambar 4.4 Grafik Fuzzy Kriteria Keterbukaan Wilayah.....	33
Gambar 4.5 Grafik Fuzzy Kriteria Derajat Berat Tanah	33
Gambar 4.6 Grafik Fuzzy Kriteria Ketebalan Lapis Olah	34
Gambar 4.7 Grafik Fuzzy Kriteria Kemudahan Irigasi	35
Gambar 4.8 Grafik Fuzzy Kriteria Kondisi Medan	36
Gambar 4.9 Grafik Fuzzy Kriteria pH Tanah	36
Gambar 4.10 Diagram Alir Inisialisasi Populasi Awal	42
Gambar 4.11 Diagram Alir Proses Extended Intermediate Crossover.....	45
Gambar 4.12 Diagram Alir Proses Random Mutation	46
Gambar 4.13 Diagram Alir Proses Seleksi Elitism	50
Gambar 4.14 Rancangan Halaman Proses Optimasi.....	53
Gambar 4.15 Rancangan Halaman Proses Pengujian Akurasi Sistem	54
Gambar 5.1 Antarmuka Halaman Proses Optimasi	66
Gambar 5.2 Antarmuka Halaman Proses Pengujian Akurasi Sistem	66
Gambar 6.1 Grafik Rata-rata Nilai Fitness Pengujian Jumlah Populasi.....	68
Gambar 6.2 Grafik Rata-rata Nilai Fitness Pengujian Kombinasi Nilai Cr dan Mr	69
Gambar 6.3 Grafik Rata-rata Nilai Fitness Pengujian Jumlah Generasi.....	71



DAFTAR LAMPIRAN

LAMPIRAN A ATURAN FUZZY	76
-------------------------------	----



BAB 1 PENDAHULUAN

1.1 Latar belakang

Lahan adalah suatu lingkungan fisik yang memiliki potensi untuk digunakan baik dalam bidang pertanian maupun non pertanian. Untuk mengoptimalkan penggunaan suatu lahan maka perlu dilakukan sebuah evaluasi untuk menentukan kesesuaian dari lahan tersebut. Kriteria yang digunakan dalam menentukan kesesuaian lahan adalah persyaratan penggunaan lahan terkait atau jika dalam bidang pertanian adalah syarat tumbuh suatu tanaman yang selanjutnya dihubungkan dengan data dari lahan suatu wilayah (Laksono & Apriyanti, 2009). Dalam bidang pertanian, penggunaan lahan yang sesuai akan mengoptimalkan tanaman yang dihasilkan, baik dari segi kualitas maupun kuantitas. Penentuan kesesuaian lahan dalam bidang pertanian bertujuan untuk menentukan kecocokan lahan untuk ditanami suatu jenis tanaman tertentu, seperti tembakau.

Budidaya tanaman tembakau di Indonesia telah ada sejak lama, bahkan sampai saat ini komoditas ini mampu menjadi sumber pendapatan yang cukup besar bagi petani maupun bagi negara. Sebagai komoditas perkebunan unggulan Indonesia, kualitas dari tembakau yang dihasilkan harus baik. Salah satu syarat agar tembakau yang dihasilkan berkualitas baik adalah penggunaan lahan yang sesuai. Kualitas lahan yang semakin baik akan berakibat pada peningkatan kualitas serta kuantitas tanaman tembakau yang dihasilkan. Jika lahan yang digunakan tidak sesuai, maka kemungkinan yang bisa terjadi adalah kerusakan pada tembakau, kualitas dan kuantitas tembakau yang rendah atau bahkan juga bisa gagal panen (Nandadiri et al., 2014).

Penentuan kesesuaian lahan yang akan ditanami tembakau sangat diperlukan untuk mengoptimalkan hasil produksi dari tanaman tembakau serta untuk menghindari dampak negatif dari penggunaan lahan yang tidak sesuai. Kendala utama yang dialami oleh para petani tembakau dalam menentukan kesesuaian lahan adalah terbatasnya pengetahuan serta sulitnya memperoleh data yang benar tentang kualitas lahan yang sesuai untuk tanaman tembakau. Dibutuhkan waktu yang cukup lama untuk memperoleh semua pengetahuan yang diperlukan, sehingga perlu adanya suatu sistem yang terkomputerisasi untuk membantu petani dalam membuat keputusan terhadap calon lahan yang akan digunakan (Sevani et al., 2009).

Penentuan kesesuaian calon lahan tembakau dilakukan sebelum proses tanam dilakukan dengan tujuan untuk menilai apakah lahan tersebut cocok untuk ditanami tembakau atau tidak. Kriteria-kriteria atau parameter yang digunakan dalam menentukan kesesuaian lahan tanam tembakau yaitu persentase lahan terkena penyakit, keterbukaan wilayah, derajat berat tanah, ketebalan lapis olah, kemudahan irigasi, kondisi medan, dan pH tanah. Kriteria-kriteria tersebut bersifat tidak pasti karena dapat memiliki dua kemungkinan sekaligus seperti baik dan buruk secara bersamaan, sehingga konsep logika *fuzzy* dapat digunakan untuk memecahkan masalah tersebut (Hidayat dalam Azizah et al., 2015). Dalam logika



fuzzy, segala sesuatu tidak bisa dinyatakan hanya dengan satu kemungkinan seperti “ya atau tidak”, “baik atau buruk”, “tinggi atau rendah”, dan lain-lain. Logika *fuzzy* memungkinkan segala sesuatu memiliki dua kemungkinan sekaligus seperti baik dan buruk secara bersamaan, namun dibedakan dengan nilai keanggotaan yang dimiliki pada masing-masing kemungkinan. Untuk merepresentasikan penggunaan logika *fuzzy*, digunakan sistem inferensi *fuzzy* dengan berdasarkan pada teori himpunan *fuzzy*, aturan *fuzzy*, dan prinsip penalaran *fuzzy* (Nisak et al., 2015). Metode sistem inferensi *fuzzy* yang dapat digunakan antara lain sistem inferensi *fuzzy tsukamoto*, sistem inferensi *fuzzy mamdani*, dan sistem inferensi *fuzzy sugeno*.

Dalam beberapa penelitian sebelumnya, logika *fuzzy* telah digunakan untuk menyelesaikan permasalahan penentuan kesesuaian calon lahan tanam tembakau. Salah satu penelitian yang berkaitan dengan permasalahan tersebut adalah penelitian Imawati dan Abadi (2016) yang menggunakan sistem inferensi *fuzzy sugeno* untuk melakukan analisis kesesuaian lahan tembakau di Kabupaten Temanggung. Parameter yang digunakan dalam penelitian tersebut adalah tingkat curah hujan, ketinggian, dan kemiringan lahan. Hasil akhir yang didapatkan dalam penelitian tersebut adalah tingkat kesesuaian lahan untuk tanaman tembakau. Penelitian selanjutnya yang juga berkaitan dengan permasalahan tersebut adalah penelitian Auliya dan Mahmudy (2016). Penelitian tersebut berhasil mengimplementasikan *Fuzzy Tsukamoto* untuk melakukan pemilihan lahan tanam yang optimum untuk tanaman tembakau. Parameter yang digunakan dalam penelitian tersebut antara lain persentase lahan yang terkena penyakit, bentuk wilayah, derajat berat, ketebalan lapis olah, kemudahan irigasi, kondisi medan dan pH tanah. Dari parameter-parameter tersebut kemudian diperoleh keputusan tentang calon lahan tanam berupa diterima atau ditolak untuk ditanami tembakau. Penerapan *Fuzzy Tsukamoto* untuk permasalahan tersebut menghasilkan akurasi sistem sebesar 70%.

Logika *fuzzy* dengan sistem inferensi *tsukamoto* dan *sugeno* telah terbukti berhasil diterapkan dalam permasalahan penentuan kesesuaian calon lahan tanam tembakau. Dalam penelitian ini sistem inferensi *fuzzy* yang lain yaitu sistem inferensi *fuzzy mamdani* digunakan untuk menyelesaikan permasalahan yang sama. Pemilihan metode ini dikarenakan kriteria atau parameter yang digunakan dalam menentukan kesesuaian lahan tanam tembakau bersifat tidak pasti, sehingga konsep dari logika *fuzzy* dapat digunakan. Pada penelitian ini juga dilakukan optimasi fungsi keanggotaan *fuzzy mamdani* dengan menggunakan algoritme genetika.

Algoritme genetika merupakan salah satu jenis algoritme evolusi yang paling populer dan dapat digunakan untuk menyelesaikan permasalahan optimasi dengan model matematikanya yang kompleks (Mahmudy, 2015). Dalam penelitian ini, algoritme genetika digunakan untuk melakukan optimasi fungsi keanggotaan dengan tujuan untuk mendapatkan akurasi yang baik dari sistem. Salah satu kelebihan dari algoritme genetika sebagai metode optimasi adalah



kemampuannya untuk digunakan dalam permasalahan optimasi dengan ruang pencarian yang sangat luas dan kompleks (Gen & Cheng dalam Mahmudy, 2015).

Algoritme genetika telah berhasil digunakan untuk melakukan optimasi fungsi keanggotaan dalam beberapa penelitian sebelumnya. Penelitian yang dilakukan oleh Armanda dan Mahmudy (2016) berhasil menerapkan algoritme genetika untuk melakukan optimasi fungsi keanggotaan *fuzzy tsukamoto* pada studi kasus peramalan permintaan barang. Penelitian ini menghasilkan nilai yang optimal jika dibandingkan hanya dengan menggunakan *fuzzy tsukamoto* tanpa dilakukan optimasi. Penelitian selanjutnya yang dilakukan oleh Restuputri, Mahmudy, dan Cholissoddin (2015) telah berhasil melakukan optimasi fungsi keanggotaan *fuzzy tsukamoto* dua tahap menggunakan algoritme genetika pada pemilihan calon penerima beasiswa. Akurasi sistem yang dihasilkan dalam penelitian ini adalah sebesar 98,9%, mengalami peningkatan dari penelitian sebelumnya yang dilakukan tanpa optimasi.

Dalam penelitian ini tidak terdapat batasan fungsi keanggotaan awal *fuzzy mamdani* yang diperoleh dari pakar. Hal ini dikarenakan adanya keterbatasan dari pakar dalam merepresentasikan *range* input secara akurat kedalam parameter angka. Oleh karena itu algoritme genetika dalam penelitian ini digunakan untuk menentukan sekaligus mengoptimasi batasan fungsi keanggotaan dari *fuzzy mamdani* sehingga fungsi keanggotaan yang didapatkan adalah yang paling optimal.

Berdasarkan uraian yang telah dijelaskan diatas, penulis mengusulkan sebuah penelitian dengan judul “Optimasi Fungsi Keanggotaan *Fuzzy Mamdani* Menggunakan Algoritme Genetika Untuk Penentuan Kesesuaian Lahan Tanam Tembakau”. Metode *Fuzzy Inference System Mamdani* digunakan untuk melakukan penentuan kesesuaian lahan tanam tembakau, sementara Algoritme Genetika digunakan untuk menentukan sekaligus mengoptimasi batasan fungsi keanggotaan dari *fuzzy mamdani* sehingga fungsi keanggotaan yang didapatkan adalah yang paling optimal dan akurasi sistem yang dihasilkan menjadi baik. Penelitian ini nantinya diharapkan mampu memberikan solusi dalam penentuan kesesuaian lahan tanam tembakau dengan tingkat akurasi yang baik.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah diuraikan diatas, dapat ditarik rumusan masalah sebagai berikut:

1. Bagaimana pengaruh nilai parameter algoritme genetika dalam optimasi fungsi keanggotaan *fuzzy mamdani* terhadap hasil penentuan kesesuaian lahan tanam tembakau?
2. Bagaimana akurasi sistem yang dihasilkan dalam optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau?



1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah sebagai berikut :

1. Mengetahui pengaruh nilai parameter algoritme genetika dalam optimasi fungsi keanggotaan *fuzzy mamdani* terhadap hasil penentuan kesesuaian lahan tanam tembakau.
2. Mengetahui akurasi sistem yang dihasilkan dalam optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau.

1.4 Manfaat

Penelitian yang dilakukan ini diharapkan dapat memberikan manfaat untuk berbagai pihak. Manfaat yang bisa diperoleh dari penelitian ini antara lain:

1. Memberikan hasil yang lebih akurat dalam penentuan kesesuaian lahan tanam tembakau.
2. Hasil dari penentuan kesesuaian lahan tanam tembakau dalam penelitian ini dapat membantu petani dalam tujuan mengoptimalkan hasil produksi baik dari segi kualitas maupun kuantitas.
3. Secara teoritis, manfaat bagi peneliti adalah memperluas wawasan serta mengembangkan ilmu pengetahuan yang didapatkan selama perkuliahan.

1.5 Batasan masalah

Untuk menghindari melebarnya permasalahan, diberikan batasan masalah sebagai berikut :

1. Data lahan, parameter atau kriteria, serta atura-aturan *fuzzy (rule)* yang digunakan dalam penentuan kesesuaian lahan tanam tembakau didapatkan dari PTPN X Jember.
2. Kriteria atau parameter yang digunakan dalam menentukan kesesuaian lahan tanam tembakau adalah persentase lahan terkena penyakit, keterbukaan wilayah, derajat berat tanah, ketebalan lapis olah, kemudahan irigasi, kondisi medan, dan pH tanah.
3. Metode yang digunakan adalah *fuzzy inference system mamdani* dan dioptimasi menggunakan algoritme genetika.
4. Metode *crossover* yang digunakan adalah *extended intermediate crossover*, sedangkan metode mutasi yang digunakan adalah *random mutation*, dan metode seleksi yang digunakan adalah *elitism*.

1.6 Sistematika pembahasan

Sistematika pembahasan dalam penulisan laporan penelitian ini adalah sebagai berikut :



BAB 1 : PENDAHULUAN

Pada bab pendahuluan ini berisi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan terkait penelitian optimasi *fuzzy inference system mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau.

BAB 2 : LANDASAN KEPUSTAKAAN

Pada bab landasan kepastakaan ini berisi tentang kajian pustaka yang merupakan pembahasan review penelitian-penelitian terdahulu yang berkaitan dengan permasalahan dalam penelitian ini, dan teori-teori yang diperlukan dalam penelitian seperti evaluasi lahan, *fuzzy mamdani* dan algoritme genetika.

BAB 3 : METODOLOGI PENELITIAN

Pada bab metodologi penelitian ini berisi tentang penjelasan langkah-langkah yang digunakan dalam melakukan penelitian, mulai dari studi literatur, pengumpulan data, perancangan, implementasi, pengujian, dan kesimpulan.

BAB 4 : PERANCANGAN

Pada bab ini akan berisi penjelasan dari serangkaian proses yang dilakukan dalam merancang sistem sesuai dengan permasalahan dan algoritme yang digunakan.

BAB 5 : IMPLEMENTASI

Pada bab ini akan berisi penjelasan dari proses implementasi dengan berdasarkan pada perancangan yang telah dilakukan sebelumnya.

BAB 6 : PENGUJIAN DAN ANALISIS

Pada bab ini akan berisi tentang hasil pengujian serta analisis terhadap implementasi yang telah dilakukan dalam penelitian.

BAB 7 : KESIMPULAN

Pada bab ini akan berisi tentang pengambilan kesimpulan dan saran dari hasil serangkaian proses yang telah dilakukan dalam penelitian.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi tentang kajian pustaka dan juga dasar teori yang berhubungan dengan penelitian ini. Dalam kajian pustaka akan dijelaskan tentang review penelitian terdahulu yang berhubungan dengan objek maupun metode dalam permasalahan yang diangkat. Selain itu dalam bab ini juga dijelaskan tentang teori-teori pendukung dalam penelitian ini seperti *fuzzy inference system mamdani* dan algoritme genetika.

2.1 Kajian Pustaka

Pada beberapa penelitian terdahulu, sistem inferensi *fuzzy mamdani* telah berhasil digunakan untuk menyelesaikan permasalahan klasifikasi dan juga peramalan. Sistem inferensi *fuzzy mamdani* telah digunakan untuk menentukan kualitas air sungai oleh Zulfa dan Mahmudy (2016). Parameter yang digunakan sebagai variabel input dalam penelitian tersebut adalah nilai TSS, BOD, COD, DO, pH, Fenol, Minyak dan Lemak. Hasil akhir yang didapatkan dalam penelitian tersebut adalah kualitas air (memenuhi baku mutu, tercemar ringan, tercemar ringan, tercemar sedang, tercemar berat).

Penelitian tentang implementasi algoritme genetika untuk optimasi fungsi keanggotaan sistem *fuzzy* juga telah dilakukan dalam beberapa penelitian terdahulu. Penelitian yang dilakukan oleh Pramesti dan Mahmudy (2016) berhasil mengimplementasikan algoritme genetika untuk melakukan optimasi fungsi keanggotaan *fuzzy mamdani* dalam prediksi nilai tukar rupiah terhadap dollar amerika.

Berdasarkan pada beberapa penelitian terdahulu, penulis mengusulkan sebuah penelitian yang memiliki keterkaitan dari segi objek maupun metode yang digunakan. Penelitian yang dilakukan oleh penulis adalah optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau. Beberapa penelitian terdahulu yang dimaksud disajikan pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

No	Judul	Objek	Metode	Perbandingan	
				Kajian Pustaka	Skripsi Peneliti
1	Optimasi Model Fuzzy Mamdani Dalam Penentuan Kualitas Air Sungai Dengan	Kualitas air sungai	Fuzzy Inference System Mamdani dan Algoritma	Objek yang diteliti adalah kualitas air sungai, metode yang digunakan adalah fuzzy inference	Objek yang diteliti adalah kesesuaian lahan tanam tembakau, metode yang digunakan adalah fuzzy



No	Judul	Objek	Metode	Perbandingan	
				Kajian Pustaka	Skripsi Peneliti
	Menggunakan Algoritma Evolution Strategies (Zulfa, I., Mahmudy, W. F., 2016)		<i>Evolution Strategies</i>	system mamdani dan dioptimasi menggunakan algoritma <i>evolution strategies</i>	inference system mamdani dan dioptimasi menggunakan algoritma genetika
2	Optimasi Fuzzy Inference System Mamdani untuk Prediksi Nilai Tukar Rupiah Terhadap Dollar Amerika Menggunakan Algoritma Genetika (Pramesti, R. A., Mahmudy, W. F., 2016)	Nilai tukar rupiah terhadap dollar amerika	Fuzzy Inference System Mamdani dan Algoritma Genetika	Objek yang diteliti adalah nilai tukar rupiah terhadap dollar amerika	Objek yang diteliti adalah kesesuaian lahan tanam tembakau
3	Optimasi Fungsi Keanggotaan Fuzy Tsukamoto Menggunakan Algoritma Genetika Untuk Penentuan Harga Jual Rumah (Azizah et al., 2015)	Harga jual rumah	Fuzzy Inference System Tsukamoto dan Algoritma Genetika	Objek yang diteliti adalah harga jual rumah, metode yang digunakan adalah fuzzy inference system tsukamoto dan dioptimasi menggunakan algoritma genetika	Objek yang diteliti adalah kesesuaian lahan tanam tembakau, metode yang digunakan adalah fuzzy inferene system mamdani dan dioptimasi menggunakan algoritma genetika



2.2 Evaluasi Kesesuaian Lahan

Evaluasi lahan adalah sebuah kegiatan untuk melihat kecocokan atau kesesuaian lahan untuk digunakan baik dalam bidang pertanian maupun non pertanian. Kriteria yang digunakan dalam melakukan evaluasi lahan adalah persyaratan penggunaan lahan terkait atau syarat tumbuh suatu tanaman yang selanjutnya dihubungkan dengan data dari lahan suatu wilayah. Hasil yang diperoleh dari evaluasi lahan berupa kecocokan lahan untuk penggunaan tertentu (Laksono & Apriyanti, 2009). Dalam bidang pertanian, evaluasi kesesuaian lahan bisa dikatakan sebagai proses pemilihan lahan yang sesuai untuk jenis tanaman tertentu.

Penentuan kesesuaian lahan dapat dilakukan untuk kondisi sekarang (kesesuaian lahan aktual) atau setelah dilakukan perbaikan (kesesuaian lahan potensial). Penentuan kesesuaian lahan aktual dilakukan sebelum lahan tersebut diberikan masukan untuk mengatasi kendala-kendala yang mungkin terjadi. Sebaliknya, kesesuaian lahan potensial akan didapatkan setelah mendapatkan masukan untuk mengatasi kendala yang mungkin terjadi dan telah dilakukan perbaikan-perbaikan. Lahan yang dapat dievaluasi antara lain hutan konversi, lahan yang tidak produktif atau terlantar, dan lahan yang produktivitasnya kurang memuaskan (Ritung et al., 2007).

Klasifikasi kesesuaian lahan dapat dibedakan menurut tingkatannya, yaitu tingkat Ordo, Kelas, Subkelas, dan Unit. Tingkat ordo melakukan klasifikasi kesesuaian lahan dengan membedakan antara lahan yang sesuai (S=Suitable) dan lahan yang tidak sesuai (N=Not Suitable). Kelas merupakan tingkat kesesuaian lahan yang berada dalam tingkat ordo. Dalam tingkat kelas, lahan yang tergolong ordo sesuai (S) diklasifikasikan ke dalam tiga kelas, yaitu: lahan sangat sesuai (S1), cukup sesuai (S2), dan sesuai marginal (S3). Lahan yang tergolong dalam ordo tidak sesuai (N) tidak dibedakan ke dalam kelas-kelas. Subkelas adalah tingkatan dalam tingkat kelas kesesuaian lahan. Tingkat kelas melakukan klasifikasi ke dalam subkelas berdasarkan kualitas dan karakteristik lahan yang menjadi faktor pembatas terberat, misal subkelas S3rc yang artinya kelas sesuai marginal dengan pembatasnya adalah kondisi perakaran ($rc=$ rooting condition). Unit adalah tingkatan dalam tingkat subkelas yang pengklasifikasiannya didasarkan pada sifat tambahan yang berpengaruh dalam pengelolaannya. Contoh S3rc1 (unit 1) dan S3rc2 (unit 2) mempunyai kelas, subkelas, dan faktor penghambat yang sama yaitu kondisi perakaran terutama faktor kedalaman efektif tanah. Dalam contoh tersebut unit 1 memiliki kedalaman efektif sedang (50-75cm) dan unit 2 memiliki kedalaman efektif dangkal (<50cm) (Ritung et al., 2007).

2.3 Logika Fuzzy

Logika *fuzzy* pertama kali diperkenalkan oleh Prof. Lotfi A. Zadeh pada tahun 1965. Teori yang mendasari logika *fuzzy* adalah himpunan *fuzzy*. Dalam himpunan *fuzzy* derajat keanggotaan memiliki peranan yang sangat penting sebagai penentu keberadaan suatu elemen pada suatu himpunan. Derajat keanggotaan atau



membership function itulah yang menjadi ciri utama dari penalaran logika *fuzzy* (Kusumadewi & Purnomo, 2013).

Beberapa hal yang menjadi alasan digunakannya logika *fuzzy* antara lain (Cox dalam Kusumadewi & Purnomo, 2013):

1. Konsep logika *fuzzy* mudah dimengerti karena menggunakan dasar teori himpunan.
2. Logika *fuzzy* bersifat fleksibel yang artinya mampu menyesuaikan dengan perubahan dan ketidakpastian yang ada dalam suatu permasalahan.
3. Logika *fuzzy* bersifat toleran terhadap data yang tidak tepat.
4. Logika *fuzzy* memiliki kemampuan untuk memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
5. Logika *fuzzy* dapat membangun dan mengaplikasikan pengalaman pakar secara langsung tanpa adanya proses pelatihan.
6. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional.
7. Logika *fuzzy* mudah dimengerti karena menggunakan bahasa alami atau bahasa sehari-hari.

Logika *fuzzy* merupakan unsur pokok dari prinsip *soft computing* dalam berbagai penelitian. *Soft computing* memberikan toleransi terhadap ketidaktepatan dan ketidakpastian data. Tujuan utama *soft computing* adalah mengambil keuntungan dari toleransi ini agar mendapatkan kepatuhan dari sistem, tingkat kecerdasan sistem yang tinggi, dan biaya yang lebih rendah (Robandi, 2006). Dalam memahami sistem *fuzzy*, beberapa hal yang perlu diketahui antara lain (Kusumadewi & Purnomo, 2013):

a. Variabel *fuzzy*

Variabel *fuzzy* adalah variabel yang hendak dibahas dalam suatu sistem *fuzzy*.

b. Himpunan *fuzzy*

Himpunan *fuzzy* adalah sebuah kelompok yang digunakan untuk mewakili sebuah keadaan atau kondisi tertentu dari variabel *fuzzy*.

c. Semesta pembicaraan

Semesta pembicaraan adalah semua nilai yang boleh dioperasikan dalam suatu variabel *fuzzy*.

d. Domain

Domain adalah semua nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam variabel *fuzzy*.



2.3.1 Himpunan Fuzzy

Pada himpunan tegas (*crisp*), nilai keanggotaan suatu elemen dalam suatu himpunan memiliki dua kemungkinan, yaitu satu (1) yang berarti bahwa elemen tersebut menjadi anggota dalam himpunan, atau nol (0) yang berarti elemen tersebut tidak menjadi anggota dalam himpunan. Berbeda dengan himpunan tegas, dalam himpunan *fuzzy* suatu elemen dapat masuk ke dalam dua himpunan yang berbeda sekaligus. Seberapa besar eksistensi elemen dalam himpunan tersebut dapat dilihat dari nilai keanggotaannya. Nilai keanggotaan pada himpunan *fuzzy* terletak pada rentang 0 sampai 1. Jika $\mu_A(x)=0$ maka dapat dikatakan nilai keanggotaan x pada himpunan *fuzzy* A adalah 0 atau dapat dikatakan x tidak menjadi anggota himpunan A , demikian pula jika $\mu_A(x)=1$ maka nilai keanggotaan x pada himpunan *fuzzy* A adalah 1 atau dapat dikatakan x menjadi anggota penuh pada himpunan A (Kusumadewi & Purnomo, 2013).

Kemiripan antara keanggotaan *fuzzy* dengan probabilitas terkadang menimbulkan kerancuan. Keduanya memiliki nilai pada interval 0 sampai 1 [0,1], namun interpretasi nilai kedua kasus tersebut sangatlah berbeda. Nilai keanggotaan *fuzzy* memberikan ukuran terhadap pendapat atau keputusan, sedangkan nilai probabilitas merupakan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Sebagai contoh, jika suatu elemen x memiliki nilai keanggotaan 0,8 pada himpunan *fuzzy* MUDA, maka tidak penting seberapa sering nilai itu diulang untuk hasil yang hampir pasti muda. Di sisi yang lain, nilai probabilitas 0,8 muda berarti 20% dari himpunan tersebut diharapkan tidak muda (Kusumadewi & Purnomo, 2013).

Himpunan *fuzzy* memiliki 2 atribut, yaitu (Kusumadewi & Purnomo, 2013):

- Linguistik, yaitu pemberian nama suatu grup yang mewakili kondisi atau keadaan tertentu dengan menggunakan bahasa yang alami, seperti: PANAS, HANGAT, NORMAL, SEJUK, dan DINGIN.
- Numeris, yaitu nilai yang menunjukkan ukuran dari variabel dalam himpunan *fuzzy*, seperti: 40, 25, 50, dsb.

2.3.2 Fungsi Keanggotaan

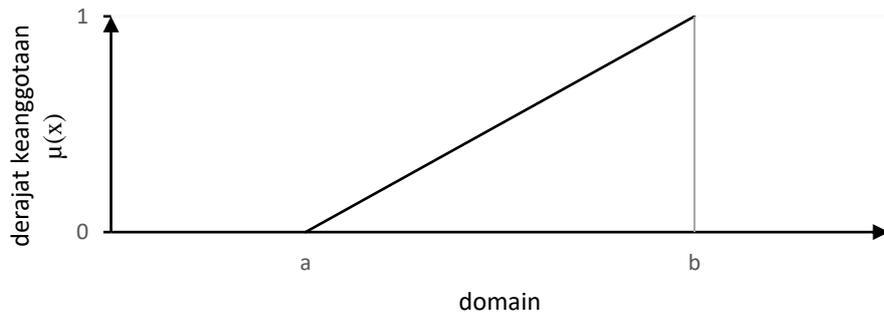
Untuk mendapatkan nilai keanggotaan dari suatu elemen pada himpunan tertentu dapat menggunakan pendekatan fungsi. Fungsi keanggotaan (*membership function*) sendiri adalah sebuah kurva yang menunjukkan pemetaan titik-titik input ke dalam nilai keanggotaannya atau yang biasa disebut dengan derajat keanggotaan. Nilai keanggotaan yang didapatkan memiliki nilai dengan interval antara 0 sampai 1. Beberapa fungsi yang biasa digunakan antara lain (Kusumadewi & Purnomo, 2013):

A. Representasi Linear

Representasi linear merupakan bentuk yang paling sederhana dan menjadi pilihan yang baik untuk mendekati konsep yang kurang jelas. Pada representasi linear, data yang digunakan sebagai input dipetakan ke dalam



derajat keanggotaan dengan digambarkan sebagai suatu garis lurus. Terdapat dua keadaan pada himpunan *fuzzy* yang linear. Pertama, himpunan *fuzzy* yang kenaikan himpunannya dimulai dari nilai domain yang derajat keanggotaannya bernilai nol (0) kemudian bergerak ke kanan menuju ke nilai domain yang derajat keanggotaannya lebih tinggi. Representasi kurva untuk keadaan pertama pada himpunan *fuzzy* yang linear ditunjukkan pada Gambar 2.1.



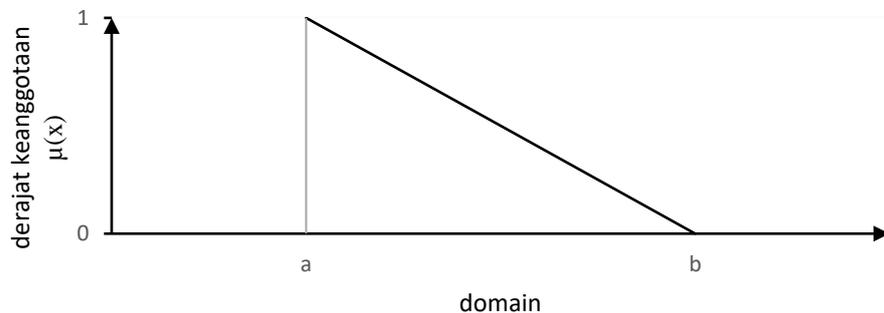
Gambar 2.1 Representasi Linear Naik

Sumber : (Kusumadewi & Purnomo, 2013)

Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \\ (x-a)/(b-a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.1)$$

Keadaan kedua pada himpunan *fuzzy* yang linear merupakan kebalikan dari keadaan yang pertama. Dalam keadaan ini, garis lurus dimulai dari nilai domain yang memiliki derajat keanggotaan tertinggi pada sisi kiri kemudian bergerak menurun menuju ke nilai domain yang derajat keanggotaannya lebih rendah. Representasi kurva untuk keadaan kedua pada himpunan *fuzzy* yang linear ditunjukkan pada Gambar 2.2.



Gambar 2.2 Representasi Linear Turun

Sumber : (Kusumadewi & Purnomo, 2013)

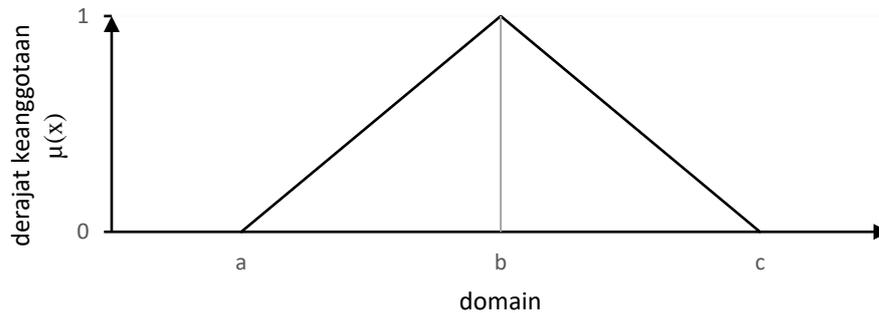


Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 1; & x \leq a \\ (b-x)/(b-a); & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.2)$$

B. Representasi Kurva Segitiga

Pada dasarnya, representasi kurva segitiga adalah gabungan dari 2 garis atau linear. Representasi kurva segitiga ditunjukkan pada Gambar 2.3



Gambar 2.3 Representasi Kurva Segitiga

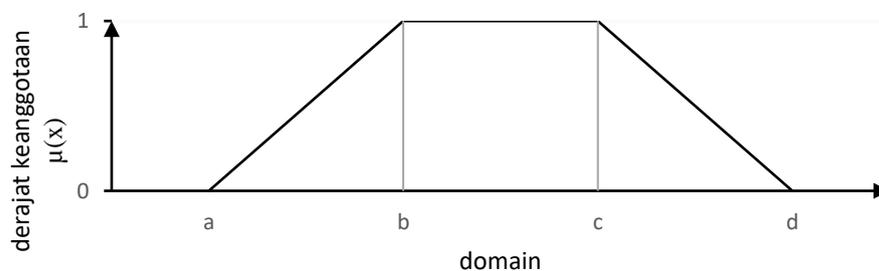
Sumber : (Kusumadewi & Purnomo, 2013)

Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x-a)/(b-a) & a \leq x \leq b \\ (c-x)/(c-b) & b \leq x \leq c \end{cases} \quad (2.3)$$

C. Representasi Kurva Trapesium

Representasi kurva trapesium hampir mirip dengan representasi kurva segitiga, hanya saja terdapat beberapa titik yang mempunyai nilai keanggotaan 1. Representasi kurva trapesium ditunjukkan pada Gambar 2.4



Gambar 2.4 Representasi Kurva Trapesium

Sumber : (Kusumadewi & Purnomo, 2013)

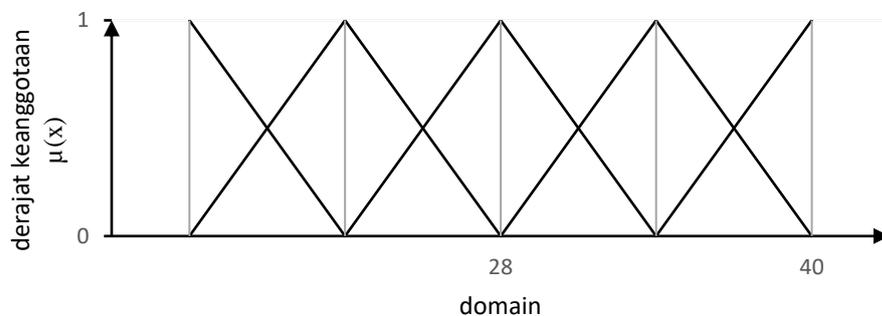


Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x-a)/(b-a) & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d-x)/(d-c) & c \leq x \leq d \end{cases} \quad (2.4)$$

D. Representasi Kurva Bentuk Bahu

Representasi kurva bahu menggambarkan naik dan turunnya sisi kanan dan kiri daerah yang terletak di tengah-tengah variabel pada bentuk segitiga. Bahu kiri akan bergerak dari benar ke salah, begitu juga sebaliknya bahu kanan akan bergerak dari salah ke benar. Representasi kurva bahu ditunjukkan pada Gambar 2.5.



Gambar 2.5 Representasi Kurva Bentuk Bahu

Sumber : (Kusumadewi & Purnomo, 2013)

2.4 Fuzzy Inference System Mamdani

Metode mamdani atau yang biasa disebut dengan metode Max-Min diperkenalkan oleh Ebrahim Mamdani pada tahun 1975. Dalam metode mamdani, untuk mendapatkan output diperlukan 4 tahapan, yaitu (Kusumadewi & Purnomo, 2013):

1. Pembentukan himpunan *fuzzy*

Tahapan ini diperlukan untuk membagi variabel input maupun variabel output menjadi satu atau lebih himpunan *fuzzy*.

2. Aplikasi fungsi implikasi

Fungsi implikasi yang digunakan dalam metode mamdani adalah Min.

3. Komposisi aturan

Berbeda dari penalaran monoton, sebuah sistem *fuzzy* yang memiliki beberapa aturan maka inferensi diperoleh dari kumpulan dan korelasi antar aturan. Dalam melakukan inferensi sistem *fuzzy*, terdapat tiga metode yang dapat digunakan, yaitu yaitu max, additive dan probabilistik OR (probor).



a. Metode Max (Maximum)

Dalam metode max, solusi dari himpunan *fuzzy* didapatkan dengan mengambil nilai maksimum dari aturan yang ada. Solusi tersebut kemudian digunakan untuk memodifikasi daerah *fuzzy* dan selanjutnya diaplikasikan ke output menggunakan operator OR (union). Secara umum persamaan dari metode max dapat dituliskan:

$$\mu_{sf}(x_i) = \max(\mu_{sf}(x_i), \mu_{kf}(x_i)) \quad (2.5)$$

Keterangan:

x_i = aturan *fuzzy* ke-i

$\mu_{sf}(x_i)$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}(x_i)$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

b. Metode Additive (Sum)

Dalam metode additive, dilakukan *bounded-sum* terhadap semua output daerah *fuzzy* untuk mendapatkan solusi himpunan *fuzzy*. Secara umum persamaan dari metode additive dapat dituliskan:

$$\mu_{sf}(x_i) = \min(1, \mu_{sf}(x_i) + \mu_{kf}(x_i)) \quad (2.6)$$

Keterangan:

x_i = aturan *fuzzy* ke-i

$\mu_{sf}(x_i)$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}(x_i)$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

c. Metode Probabilistik OR (probor)

Metode probor mendapatkan solusi himpunan *fuzzy* dengan melakukan *product* terhadap semua output daerah *fuzzy*. Secara umum persamaan dari metode probor dapat dituliskan:

$$\mu_{sf}(x_i) = (\mu_{sf}(x_i) + \mu_{kf}(x_i)) - (\mu_{sf}(x_i) * \mu_{kf}(x_i)) \quad (2.7)$$

Keterangan:

x_i = aturan *fuzzy* ke-i

$\mu_{sf}(x_i)$ = nilai keanggotaan solusi *fuzzy* sampai aturan ke-i

$\mu_{kf}(x_i)$ = nilai keanggotaan konsekuen *fuzzy* aturan ke-i

4. Penegasan (*defuzzy*)

Langkah terakhir dalam metode mamdani adalah penegasan (*defuzzy*). *Output* dari komposisi aturan yang berupa himpunan *fuzzy* digunakan sebagai input pada proses defuzzifikasi. Proses defuzzifikasi menghasilkan *output* suatu bilangan pada domain himpunan *fuzzy* tersebut. Dengan demikian apabila diberikan himpunan *fuzzy* dalam *range* tertentu maka harus dapat diambil sebuah nilai *crisp* yang berada pada domain himpunan *fuzzy* tersebut untuk dijadikan



sebagai *output*. Beberapa metode defuzzifikasi yang ada pada metode mamdani antara lain:

a. Metode *Centroid (Composite Moment)*

Dalam metode centroid, solusi nilai *crisp* didapatkan dengan cara mengambil titik pusat (z^*) daerah *fuzzy*. Secara umum persamaan metode ini dapat dituliskan:

$$z^* = \frac{\int_z z\mu(z)dz}{\int_z \mu(z)dz} \quad \text{untuk variabel kontinu, atau} \quad (2.8)$$

z^* = titik pusat *fuzzy (output)*

$\mu(z)$ = derajat keanggotaan titik ke- i

z = nilai domain ke- i

$$z^* = \frac{\sum_{j=1}^n z_j\mu(z_j)}{\sum_{j=1}^n \mu(z_j)} \quad \text{untuk variabel diskret} \quad (2.9)$$

Keterangan:

z^* = titik pusat *fuzzy (output)*

$\mu(z_j)$ = derajat keanggotaan titik ke- j

z_j = nilai domain ke- j

b. Metode *Bisektor*

Dalam metode bisektor, untuk mendapatkan solusi *crisp* dilakukan dengan cara mengambil nilai pada domain *fuzzy* yang mempunyai nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah *fuzzy*.

c. Metode *Mean of Maximum (MOM)*

Dalam metode *mean of maximum*, nilai rata-rata domain yang memiliki nilai keanggotaan maksimum diambil untuk mendapatkan solusi *crisp*.

d. Metode *Largest of Maximum (LOM)*

Dalam metode *largest of maximum*, nilai terbesar dari domain yang memiliki nilai keanggotaan maksimum diambil untuk mendapatkan solusi *crisp*.

e. Metode *Smallest of Maximum (SOM)*

Dalam metode *smallest of maximum*, nilai terkecil dari domain yang memiliki nilai keanggotaan maksimum diambil untuk mendapatkan solusi *crisp*.

Dalam penelitian ini, *fuzzy mamdani* digunakan untuk menentukan (klasifikasi) kesesuaian lahan tanam tembakau. Sistem klasifikasi *fuzzy* adalah sistem *fuzzy* berbasis aturan yang dirancang untuk melakukan tugas klasifikasi. Langkah-langkah dalam menggunakan *fuzzy mamdani* untuk proses klasifikasi adalah sebagai berikut (Setiawan & Subanar, 2010):

1. Melakukan fungsi implikasi pada setiap aturan atau *rule* dengan menggunakan metode *MIN*.



2. Melakukan komposisi aturan menggunakan metode MAX. Dalam metode MAX, solusi *fuzzy* diperoleh dengan cara mengambil nilai terbesar dari seluruh aturan pada masing-masing kelasnya.
3. Mengambil kelas dari kesesuaian lahan yang memiliki nilai keanggotaan terbesar untuk dijadikan hasil akhir klasifikasi.

2.5 Algoritme Genetika

Algoritme genetika (*Genetic Algorithms*, GAS) merupakan salah satu cabang dari teknik optimasi yang didasarkan pada proses evolusi biologi, yaitu *evolutionary algorithms*. Algoritme ini memiliki kemampuan untuk menyelesaikan permasalahan yang kompleks sehingga banyak digunakan untuk penyelesaian masalah optimasi dengan model matematika yang kompleks atau bahkan sulit untuk dibangun (Mahmudy, 2015). Siklus dari algoritme genetika diawali dengan inisialisasi atau pembentukan himpunan individu baru yang kemudian ditempatkan pada penampungan atau biasa disebut populasi. Individu dalam algoritme genetika merepresentasikan solusi dari permasalahan yang diangkat. Individu-individu terpilih yang ada dalam populasi selanjutnya dikembangkan (reproduksi) menggunakan operator genetik seperti pindah silang (*crossover*) dan mutasi (*mutation*) untuk menghasilkan individu baru. Selanjutnya proses evaluasi dilakukan untuk perbaikan populasi, sehingga individu-individu baru yang lolos seleksi akan menggantikan himpunan populasi asal dan dipertahankan pada generasi selanjutnya. Siklus tersebut akan terus dilakukan sampai keturunan yang dihasilkan tidak lebih baik dari sebelumnya atau sampai kriteria optimum telah ditemukan (Widodo & Mahmudy, 2010).

2.5.1 Struktur Algoritme Genetika

Dalam menggunakan algoritme genetika, solusi dari permasalahan yang akan diselesaikan harus dipetakan (*encoding*) menjadi string *chromosome* yang tersusun dari sejumlah gen. String *chromosome* inilah yang menggambarkan variabel keputusan dan digunakan dalam solusi. Representasi *string chromosome* tersebut dimasukkan ke dalam algoritme genetika bersamaan dengan fungsi *fitness* yang digunakan untuk menilai kualitas dari *chromosome* (untuk menjadi solusi yang layak). Algoritme genetika akan menghasilkan *chromosome* terbaik setelah melewati beberapa generasi dengan menirukan proses evolusi biologi. Selanjutnya *chromosome* terbaik tersebut harus diuraikan (*decoding*) sehingga solusi yang diharapkan mendekati optimum bisa didapatkan (Mahmudy, 2015).

2.5.2 Komponen Algoritme Genetika

Terdapat beberapa proses yang ada dalam siklus algoritme genetika dalam pencarian solusi (*chromosome*) terbaik untuk menyelesaikan suatu permasalahan. Proses-proses tersebut antara lain (Mahmudy, 2015):



2.5.2.1 Representasi *Chromosome*

Representasi *chromosome* yang terdapat dalam algoritme genetika yaitu representasi bilangan *biner*, *real* atau desimal, dan *integer*. Representasi *chromosome* yang akan digunakan dalam algoritme genetika harus disesuaikan dengan permasalahan yang akan diselesaikan. Misalnya, pada penyelesaian masalah maksimasi fungsi digunakan representasi *chromosome biner*, kemudian pada penyelesaian *Travelling Salesman Problem (TSP)* dan *Job-Shop Scheduling Problem (JSP)* dalam masalah kombinatorial digunakan representasi *chromosome integer*, dan pada penyelesaian *Flexible Job-Shop Scheduling Problem (FJSP)* digunakan representasi *chromosome* bilangan *real*.

2.5.2.2 Inisialisasi

Inisialisasi adalah tahap untuk membangkitkan himpunan solusi baru yang dilakukan secara acak/random dan terdiri dari sejumlah *string chromosome*. Himpunan solusi baru ini selanjutnya ditempatkan dalam penampungan atau yang disebut sebagai populasi. Dalam tahapan ini harus ditentukan ukuran dari populasi (*popsiz*) untuk menentukan berapa banyak individu atau *chromosome* yang ditampung dalam populasi.

2.5.2.3 Reproduksi

Reproduksi merupakan tahap yang dilakukan untuk menghasilkan keturunan (*offspring*) dari individu-individu terpilih yang ada dalam populasi. Tahap reproduksi dapat dilakukan dengan menggunakan dua operator genetika yaitu tukar silang (*crossover*) dan mutasi (*mutation*).

1. *Crossover*

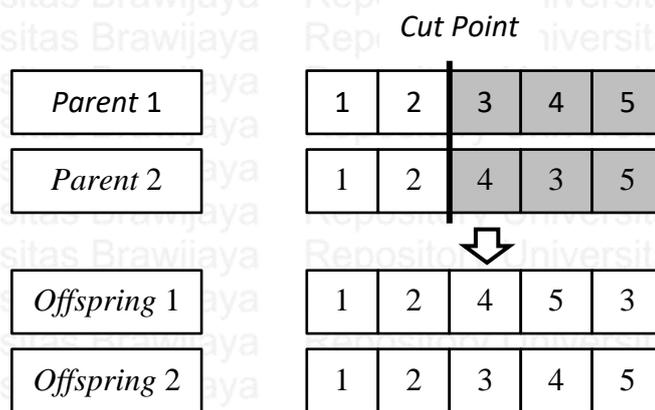
Crossover adalah tahap reproduksi yang dilakukan dengan cara perkawinan silang antara dua induk (*parent*) terpilih yang ada dalam populasi. Tingkat *crossover* (*crossover rate/cr*) harus ditentukan untuk menyatakan rasio *offspring* yang akan dihasilkan proses *crossover* terhadap ukuran populasi. Rumus untuk menghitung *offspring* yang dihasilkan pada *crossover* adalah sebagai berikut:

$$\text{Offspring} = cr \times \text{popsiz} \quad (2.10)$$

Terdapat beberapa metode *crossover* yang dapat digunakan dalam proses reproduksi, antara lain (Mahmudy, 2015):

a. *One-Cut Point Crossover*

Dalam metode *crossover* ini, *offspring* akan dihasilkan dengan cara menukarkan bagian kanan dari masing-masing induk terpilih berdasarkan satu titik potong yang dipilih secara acak. Misalkan dalam sebuah permasalahan ditentukan nilai $cr = 0,5$ dan $popsiz = 4$ maka akan ada $0,5 \times 4 = 2$ *offspring* yang dihasilkan dalam setiap kali proses *crossover*. Jika dalam setiap *crossover* telah ditentukan akan menghasilkan dua *offspring* maka operasi *crossover* hanya akan dilakukan satu kali. Sebagai contoh, penerapan metode *one-cut point crossover* dengan titik potong (*cut-point*) pada titik ke-2 dapat dilihat pada Gambar 2.6.



Gambar 2.6 Contoh Penerapan Metode One-Cut Point Crossover

b. *Extended Intermediate Crossover*

Dalam metode *crossover* ini, *offspring* akan dihasilkan dengan cara mengkombinasikan nilai dua induk yang terpilih secara acak. Jika dimisalkan P_1 dan P_2 adalah dua induk (*parent*) yang terpilih untuk digunakan dalam *crossover*, maka *offspring* (C) dapat dibangkitkan dengan rumus sebagai berikut:

$$C_1 = P_1 + a (P_2 - P_1) \tag{2.11}$$

$$C_2 = P_2 + a (P_1 - P_2) \tag{2.12}$$

a dipilih secara acak pada interval tertentu, misalkan interval a adalah $[0,1, 1,25]$.

Misalkan dalam sebuah permasalahan ditentukan nilai $cr = 0,5$ dan $popsiz=4$ maka akan ada $0,5 \times 4 = 2$ *offspring* yang dihasilkan dalam setiap kali proses *crossover*. Jika dalam setiap *crossover* telah ditentukan akan menghasilkan dua *offspring* maka operasi *crossover* hanya akan dilakukan satu kali. Sebagai contoh, misalkan *parent* yang terpilih adalah $P_1 (2,3, 5,2)$ dan $P_2 (4,8, 3,1)$, $a [0,1, 0,2]$ maka akan dihasilkan dua *offspring* (C_1 dan C_2) sebagai berikut:

$$C_1: \quad x1 = 2,3 + 0,1 (4,8 - 2,3) = 2,55$$

$$x2 = 5,2 + 0,2 (3,1 - 5,2) = 4,78$$

$$C_2: \quad x1 = 4,8 + 0,1 (2,3 - 4,8) = 4,55$$

$$x2 = 3,1 + 0,2 (5,2 - 3,1) = 3,52$$

2. Mutasi

Mutasi adalah salah satu operator genetika dalam reproduksi yang bekerja dengan mengubah gen yang terpilih secara acak dalam satu induk terpilih. Induk yang digunakan dalam tahap reproduksi ini juga dipilih secara acak dari populasi. Untuk menyatakan rasio *offspring* yang akan dihasilkan proses mutasi terhadap ukuran populasi maka tingkat mutasi (*mutation rate/mr*) harus ditentukan terlebih dahulu. Rumus untuk menghitung *offspring* yang dihasilkan pada mutasi adalah sebagai berikut:

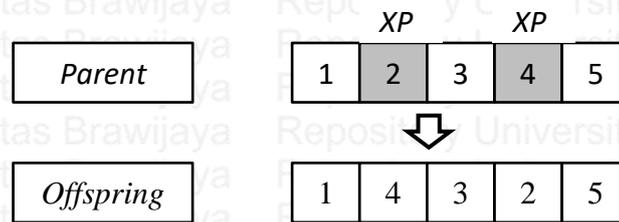


$$Offspring = mr \times popsize \tag{2.13}$$

Terdapat beberapa metode mutasi yang dapat digunakan dalam proses reproduksi, antara lain (Mahmudy, 2015):

a. *Reciprocal Exchange Mutation*

Metode mutasi yang paling sederhana adalah *reciprocal exchange mutation*. Dalam metode ini, dua posisi (*exchange point/XP*) gen dalam kromosom dipilih secara random kemudian dilakukan penukaran posisi gen tersebut. Misalkan dalam sebuah permasalahan ditentukan nilai $mr = 0,2$ dan $popsize=4$ maka akan ada $0,2 \times 4 = 0,8$ (dibulatkan menjadi 1) *offspring* yang dihasilkan dalam setiap kali proses mutasi. Sebagai contoh, penerapan metode *reciprocal exchange mutation* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Contoh Penerapan Metode *Reciprocal Exchange Mutation*

b. *Random Mutation*

Metode *random mutation* dilakukan dengan cara menambah atau mengurangi nilai dari gen yang terpilih dari salah satu *parent* dengan bilangan *random* yang kecil. Jika dimisalkan domain variabel x_j adalah $[min_j, max_j]$ dan *offspring* yang dihasilkan adalah $C=[x'_1...x'_n]$ maka nilai gen *offspring* dapat dibangkitkan dengan rumus sebagai berikut:

$$C = x'_i + r (max_j - min_j) \tag{2.14}$$

r dipilih secara acak pada interval tertentu, misalkan interval r adalah $[-0,1, 0,1]$.

Misalkan dalam sebuah permasalahan ditentukan nilai $mr = 0,2$ dan $popsize=4$ maka akan ada $0,2 \times 4 = 0,8$ (dibulatkan menjadi 1) *offspring* yang dihasilkan dalam setiap kali proses mutasi. Sebagai contoh, misalkan *parent* yang terpilih adalah $P_1 (2,3, 5,2)$, domain variabel $[0, 7,3]$, gen yang terpilih adalah gen ke-2 (x_2) dan $r = -0,0584$ maka akan dihasilkan *offspring* sebagai berikut:

$$C : \begin{aligned} x_1 &= 2,3 \text{ (tetap)} \\ x_2 &= 5,2 - 0,0584 (7,3 - 0,0) = 4,77368 \end{aligned}$$

2.5.2.4 Evaluasi

Evaluasi dilakukan untuk menghitung nilai *fitness* setiap individu (termasuk *offspring*). Nilai *fitness* digunakan untuk menentukan kualitas dari individu (untuk menjadi solusi yang layak). Semakin besar nilai *fitness* suatu individu, maka semakin baik pula individu tersebut untuk dijadikan calon solusi dari permasalahan. Terdapat perbedaan dalam penghitungan nilai *fitness* antara



permasalahan untuk mencari nilai maksimum dengan permasalahan untuk mencari nilai minimum. Untuk permasalahan pencarian nilai maksimum, nilai *fitness* dapat dihitung menggunakan rumus:

$$fitness = f(x) \quad (2.15)$$

sedangkan untuk permasalahan pencarian nilai minum, nilai *fitness* dapat dihitung menggunakan rumus:

$$fitness = C - f(x) \quad (2.16)$$

$$fitness = 1/f(x) \quad (2.17)$$

2.5.2.5 Seleksi

Nilai *fitness* dari sebuah individu sangat menentukan peluang terpilih untuk lolos atau hidup pada generasi berikutnya. Pemilihan individu dalam himpunan populasi dan *offspring* yang dipertahankan hidup pada generasi berikutnya dilakukan pada tahapan seleksi. Pada dasarnya, individu yang dipertahankan hidup merupakan individu yang memiliki nilai *fitness* lebih tinggi daripada individu yang lain. Metode seleksi yang sering digunakan dalam algoritme genetika adalah *roulette wheel*, *binary tournament*, dan *elitism*.

2.6 Fuzzy Inference System Mamdani – Algoritme Genetika

Dalam penelitian ini, metode fuzzy mamdani digunakan untuk melakukan penentuan kesesuaian lahan tanam tembakau sedangkan algoritme genetika digunakan untuk mengoptimasi fungsi keanggotaan dari *fuzzy mamdani*. Siklus penyelesaian masalah menggunakan *fuzzy mamdani* dan algoritme genetika adalah sebagai berikut (Azizah, 2015):

1. Penentuan parameter dari algoritme genetika yaitu ukuran populasi (*popsize*), jumlah generasi, *crossover rate/cr*, dan *mutation rate/mr*.
2. Pembangkitan populasi awal yang dilakukan secara acak sesuai dengan ukuran populasi (*popsize*) yang telah ditentukan.
3. Pembentukan individu baru atau menghasilkan keturunan (*offspring*) melalui proses reproduksi dengan menggunakan operator genetika yaitu tukar silang (*crossover*) dan mutasi (*mutation*).
4. Melakukan perhitungan menggunakan metode *fuzzy mamdani* sampai ditemukan hasil akhir dengan ketentuan batasan fungsi keanggotaan pada masing-masing individu.
5. Melakukan perhitungan nilai *fitness* untuk masing-masing individu. Nilai *fitness* dalam penelitian ini dihitung menggunakan rumus berikut:

$$Fitness = \frac{\text{Jumlah data kelas yang sama}}{\text{Total seluruh data}} \quad (2.18)$$

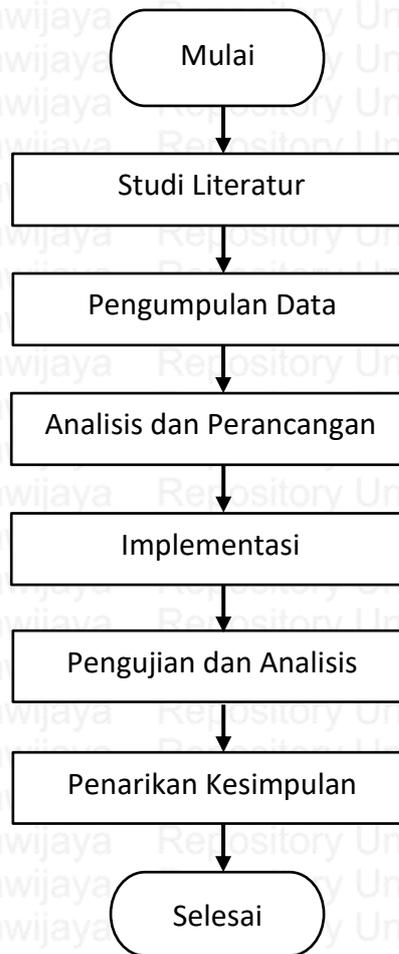
6. Melakukan seleksi terhadap individu-individu yang ada dalam populasi beserta *offspring* sehingga didapatkan individu baru sejumlah *popsize* untuk selanjutnya digunakan dalam iterasi berikutnya.

BAB 3 METODOLOGI

Bab ini membahas langkah-langkah yang digunakan dalam melakukan penelitian optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau.

3.1 Tahapan Penelitian

Tahapan penelitian memberikan gambaran tentang langkah-langkah yang digunakan dalam melakukan penelitian ini. Secara lengkap, tahapan penelitian yang dikerjakan ditunjukkan dalam Gambar 3.1.



Gambar 3.1 Diagram Alir Tahapan Penelitian

3.2 Studi Literatur

Studi literatur adalah tahap pengumpulan dasar teori yang diperlukan untuk menunjang penelitian yang sedang dilakukan. Tahap ini dilakukan untuk mendapatkan pemahaman yang lebih tentang teori yang berkaitan dengan penelitian. Pengumpulan teori berasal dari buku, jurnal, ataupun penelitian



terdahulu. Teori yang dimaksud berkaitan dengan penelitian ini adalah tentang evaluasi lahan, logika *fuzzy*, dan algoritme genetika.

3.3 Pengumpulan Data

Penelitian ini menggunakan data lahan tembakau sebanyak 40 data yang didapatkan dari PTPN X Jember. 30 data lahan tembakau akan digunakan sebagai data latih dalam optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk menghasilkan batasan fungsi keanggotaan yang optimum. Selanjutnya sebanyak 10 data lahan tembakau akan digunakan sebagai data uji. Tabel 3.1 menunjukkan data latih yang digunakan dalam penelitian ini, sedangkan data uji yang digunakan ditunjukkan dalam Tabel 3.2.

Tabel 3.1 Data Latih Lahan Tembakau

No	Lahan	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	Kelas
1	Alas	0	3	3	30	3	4	6,7	Sangat Sesuai
2	Klatakan	0	3	3	30	3	4	6,1	Sangat Sesuai
3	Parseh	0	3	2	30	3	3	6,7	Kurang Sesuai
4	Tugusari	2	3	2	30	3	3	6,7	Kurang Sesuai
5	Grintingan	3	3	3	30	3	4	6,8	Sangat Sesuai
6	Pecoro Barat	2	3	2	30	3	4	6,7	Cukup Sesuai
7	Botosari	0	3	2	30	3	2	6,7	Kurang Sesuai
8	Cora 1	3	3	3	30	3	2	6,7	Cukup Sesuai
9	KUD	2	3	3	30	3	4	6,7	Sangat Sesuai
10	Curah Kates	1	3	3	30	3	2	6,8	Cukup Sesuai
11	Dukuhsia Selatan	8	3	2	30	3	3	6,8	Kurang Sesuai
12	Dukuhsia	10	2	2	20	3	4	6,7	Kurang Sesuai



No	Lahan	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	Kelas
13	Ambulu	17	3	3	30	3	4	6,8	Cukup Sesuai
14	Darungan	3	3	3	30	3	1	6,8	Cukup Sesuai
15	Spadia	2	3	3	30	3	4	6,8	Sangat Sesuai
16	Curah Buntu	0	3	2	30	3	4	6,8	Cukup Sesuai
17	Gayasan	0	3	2	30	3	4	6,8	Cukup Sesuai
18	Pondok Labu	0	3	2	30	3	4	6,8	Cukup Sesuai
19	Cora 2	15	3	1	30	2	4	5,5	Tidak Sesuai
20	Dam 1	2	3	2	30	3	4	6,8	Cukup Sesuai
21	Dam 2	1	3	2	30	3	4	6,8	Cukup Sesuai
22	Andongsari	4	2	2	30	3	4	6,8	Kurang Sesuai
23	Mandilis	5	3	2	30	3	4	6,8	Cukup Sesuai
24	Pontang	3	3	2	45	3	4	6,8	Cukup Sesuai
25	Randu	0	3	2	30	3	4	6,8	Cukup Sesuai
26	Besuk	0	3	2	30	3	4	6,8	Cukup Sesuai
27	Manggar	3	3	2	30	3	4	6,8	Cukup Sesuai
28	Sumber Rejeki 1	15	3	2	10	3	2	6,8	Tidak Sesuai
29	Sumber Rejeki 2	2	3	2	30	2	4	6,8	Kurang Sesuai
30	Sumber Waru	0	3	2	30	1	4	7,8	Tidak Sesuai



Tabel 3.2 Data Uji Lahan Tembakau

No	Nama Lahan	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	Kelas
1	Nogosari	0	3	3	30	3	4	6,8	Sangat Sesuai
2	Karang Odeng	3	2	1	30	3	3	6,7	Kurang Sesuai
3	Mayang	0	3	3	30	3	4	6,8	Sangat Sesuai
4	Jubung	0	3	2	30	2	4	6,7	Cukup Sesuai
5	Wirowongso	5	3	2	30	3	4	6,8	Cukup Sesuai
6	Banjarsari Barat	2	3	3	30	3	1	6,8	Cukup Sesuai
7	Banjarsari Tengah	13	2	1	30	2	4	5,8	Tidak Sesuai
8	Banjar Sengon	4	2	2	30	3	4	5,5	Kurang Sesuai
9	Bedadung	2	3	2	30	2	4	6,8	Kurang Sesuai
10	Plalangan	8	3	1	30	1	4	7,7	Tidak Sesuai

Keterangan :

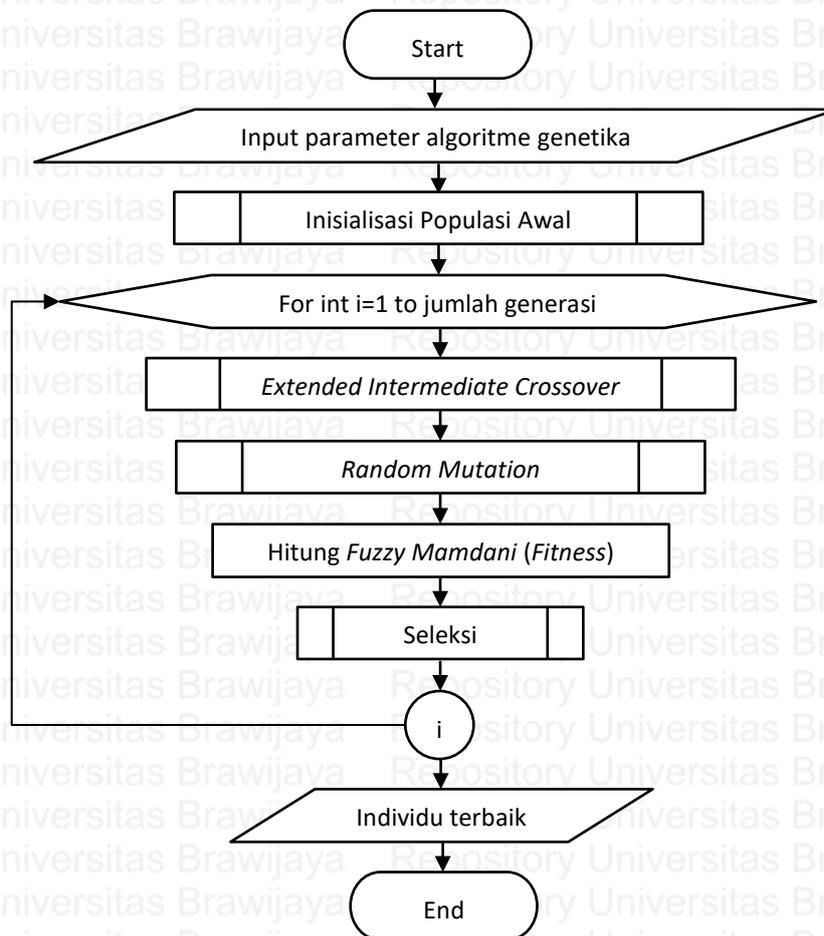
X₁ : Kriteria persentase lahan terkena penyakitX₂ : Kriteria keterbukaan wilayahX₃ : Kriteria derajat berat tanahX₄ : Kriteria ketebalan lapis olahX₅ : Kriteria kemudahan irigasiX₆ : Kriteria kondisi medanX₇ : Kriteria pH tanah



3.4 Analisis dan Perancangan

Analisis kebutuhan sistem dalam tahapan ini bertujuan untuk mendapatkan segala sesuatu yang dibutuhkan dalam melakukan penelitian ini. Dalam penelitian ini, perangkat keras yang digunakan adalah komputer PC dengan spesifikasi processor AMD A6-7310, RAM 4 GB, dan Harddisk 500 GB. Sedangkan perangkat lunak yang digunakan adalah sistem operasi Windows 10 Pro 64-bit, bahasa pemrograman Java dengan editor Netbeans IDE 8.2, Microsoft Word 2016 yang digunakan dalam pembuatan dokumen, dan Microsoft Excel 2016 yang digunakan untuk melakukan perhitungan manualisasi dari algoritme yang digunakan.

Dalam tahapan ini juga dilakukan perancangan sistem baik perancangan algoritme maupun perancangan antarmuka yang dijadikan dasar dalam implementasi. Siklus optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika seperti yang terlihat pada Gambar 3.2.



Gambar 3.2 Diagram Alir *Hybrid Fuzzy Mamdani* – Algoritme Genetika

3.5 Implementasi

Pada tahap ini dibahas tentang pembangunan sistem optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau. Tahap ini dilakukan berdasarkan dari hasil



analisis dan perancangan yang telah dilakukan sebelumnya. Implementasi ini meliputi implementasi algoritme dan implementasi antarmuka.

3.6 Pengujian dan Analisis

Pada tahap ini dilakukan pengujian algoritme yang digunakan dan selanjutnya dilakukan analisis terhadap hasil pengujian terkait penelitian optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau yang telah diimplementasikan. Tidak adanya metode yang pasti dalam menentukan parameter dari algoritme genetika mengharuskan dilakukan pengujian terhadap semua kombinasi parameter yang mungkin agar mendapatkan parameter yang optimal. Parameter yang diujikan dalam penelitian ini adalah jumlah *popsi* atau populasi, kombinasi nilai C_r dan M_r , dan jumlah generasi. Pengujian ini dilakukan untuk mengetahui pengaruh dari masing-masing parameter terhadap nilai *fitness* yang dihasilkan. Skenario pengujian yang akan dilakukan antara lain:

1. Pengujian jumlah populasis atau *popsi* dengan angka kelipatan 10
2. Pengujian kombinasi nilai C_r dan M_r dengan bilangan antara 0 sampai 1
3. Pengujian jumlah generasi yang dilakukan dengan angka kelipatan 50

3.6.1 Pengujian Jumlah Populasi (*popsi*)

Pengujian ini dilakukan dengan tujuan untuk mendapatkan jumlah atau ukuran populasi terbaik yang dapat menghasilkan batasan fungsi keanggotaan paling optimal dalam *Fuzzy Inference System Mamdani* untuk penentuan kesesuaian lahan tanam tembakau. Pengujian ini dilakukan sebanyak 10 kali pada setiap jumlah populasi dengan nilai C_r , M_r , dan jumlah generasi yang sama. Jumlah populasi yang akan diujikan merupakan kelipatan 10 mulai dari 10 sampai 100. Rancangan pengujian jumlah populasi ditunjukkan pada Tabel 3.3.

Tabel 3.3 Rancangan Pengujian Jumlah Populasi (*Popsi*)

Jumlah Populasi	Nilai <i>Fitness</i> Percobaan Ke-i										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10											
20											
30											
40											
50											
60											
70											



Jumlah Populasi	Nilai <i>Fitness</i> Percobaan Ke-i										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
80											
90											
100											

3.6.2 Pengujian Kombinasi Nilai *Cr* dan *Mr*

Pengujian ini dilakukan dengan tujuan untuk mendapatkan kombinasi nilai *Cr* dan *Mr* yang paling baik sehingga dapat menghasilkan batasan fungsi keanggotaan paling optimal. Uji coba dilakukan sebanyak 10 kali pada setiap kombinasi dengan menggunakan jumlah populasi dan jumlah generasi yang sama. Jumlah populasi yang digunakan adalah nilai yang paling optimal dari hasil pengujian jumlah populasi yang telah dilakukan sebelumnya. Nilai *Cr* dan *Mr* yang digunakan dalam pengujian ini mulai dari 0 sampai 1. Rancangan pengujian kombinasi nilai *Cr* dan *Mr* ditunjukkan pada Tabel 3.4.

Tabel 3.4 Rancangan Pengujian Kombinasi Nilai *Cr* dan *Mr*

Nilai <i>Crossover Rate</i> (<i>Cr</i>)	Nilai <i>Mutation Rate</i> (<i>Mr</i>)	Nilai <i>Fitness</i> Percobaan Ke-i										Rata-rata Nilai <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0	1											
0,1	0,9											
0,2	0,8											
0,3	0,7											
0,4	0,6											
0,5	0,5											
0,6	0,4											
0,7	0,3											
0,8	0,2											
0,9	0,1											
1	0											

3.6.3 Pengujian Jumlah Generasi

Pengujian ini dilakukan untuk mendapatkan jumlah generasi yang paling terbaik sehingga dapat menghasilkan batasan fungsi keanggotaan paling optimal



dalam *Fuzzy Inference System Mamdani* untuk penentuan kesesuaian lahan tanam tembakau. Pengujian ini dilakukan sebanyak 10 kali pada setiap jumlah generasi dengan menggunakan jumlah populasi, nilai Cr dan Mr yang tetap. Jumlah populasi dan kombinasi nilai Cr dan Mr yang digunakan adalah nilai yang paling optimal dari hasil pengujian yang telah dilakukan sebelumnya. Jumlah generasi yang akan diujikan merupakan kelipatan 50 mulai dari 150 sampai dengan 600. Rancangan pengujian jumlah generasi ditunjukkan pada Tabel 3.5.

Tabel 3.5 Rancangan Pengujian Jumlah Generasi

Jumlah Generasi	Nilai <i>Fitness</i> Percobaan Ke-i										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
150											
200											
250											
300											
350											
400											
450											
500											
550											
600											

3.7 Penarikan Kesimpulan

Tahap ini membahas tentang kesimpulan dari serangkaian tahapan yang telah dilakukan dalam penelitian ini. Bagian ini juga berisi tentang saran penelitian selanjutnya yang dapat dilakukan untuk mengembangkan penelitian ini.

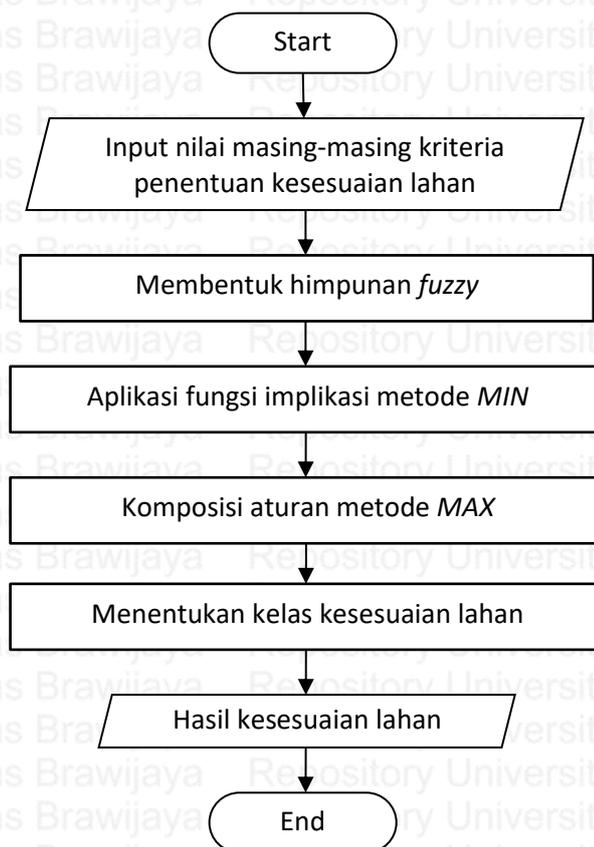


BAB 4 PERANCANGAN

4.1 Desain Algoritme

Dalam penelitian ini, digunakan *hybrid fuzzy mamdani* dengan algoritme genetika untuk menyelesaikan permasalahan. *Fuzzy mamdani* digunakan untuk melakukan penentuan kesesuaian lahan tanam tembakau, sedangkan algoritme genetika digunakan untuk mengoptimasi batas fungsi keanggotaan pada *fuzzy mamdani*.

Dalam penelitian ini, proses *fuzzy mamdani* untuk mendapatkan output yang diinginkan harus melalui empat tahapan. Langkah pertama yang dilakukan dalam *fuzzy mamdani* setelah mendapatkan inputan adalah melakukan fuzzifikasi untuk masing-masing nilai yang telah diinputkan. Tahap selanjutnya adalah aplikasi fungsi implikasi menggunakan metode *MIN*, dan dilanjutkan dengan melakukan komposisi aturan menggunakan metode *MAX*. Langkah yang terakhir adalah menentukan kelas kesesuaian lahan. Secara umum, diagram alir dari *fuzzy mamdani* dalam penelitian ini ditunjukkan dalam Gambar 4.1.

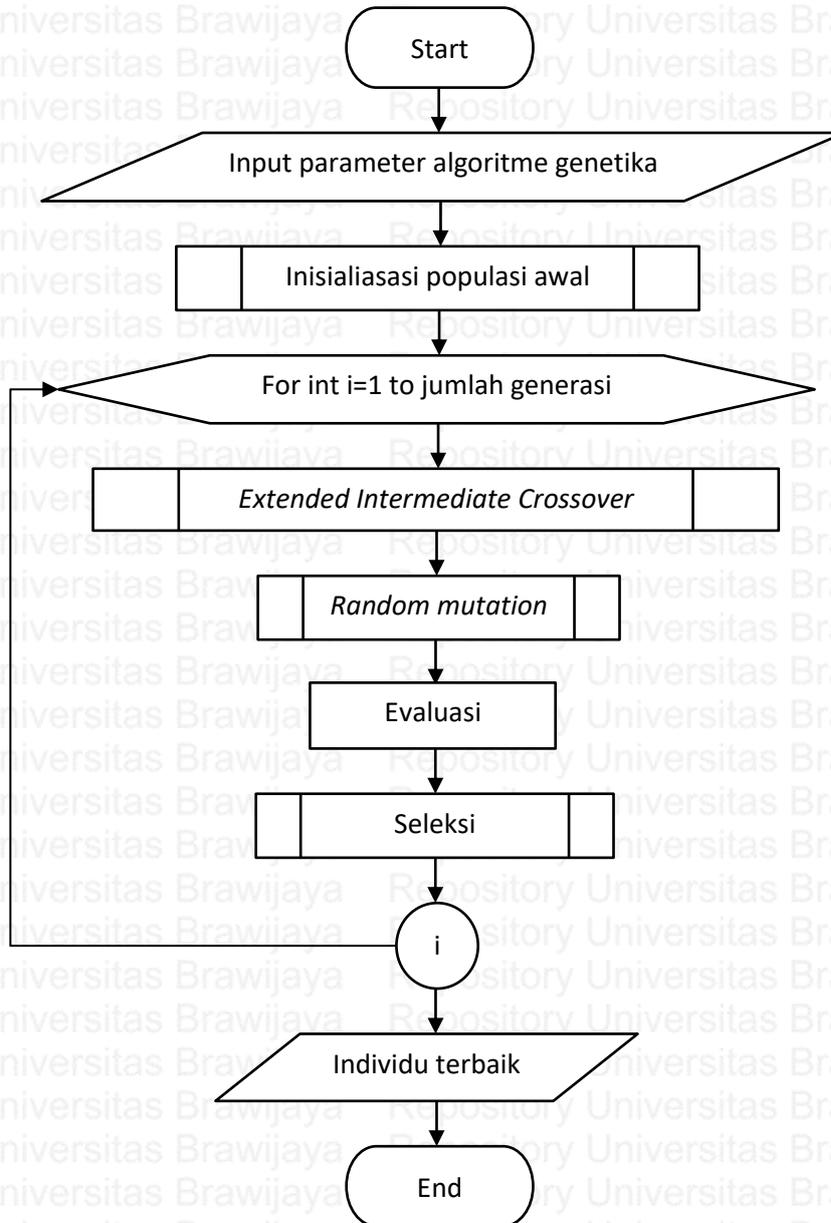


Gambar 4.1 Diagram Alir *Fuzzy Mamdani*

Algoritme genetika digunakan untuk melakukan optimasi fungsi keanggotaan pada *fuzzy mamdani*, dengan tujuan untuk mendapatkan akurasi sistem yang baik. Terdapat empat proses utama dalam algoritme genetika, yaitu inialisasi populasi



awal, reproduksi, evaluasi, dan seleksi. Diagram alir algoritme genetika yang digunakan dalam penelitian ini secara umum ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Diagram Alir Algoritme Genetika

4.2 Siklus Penyelesaian Masalah Menggunakan *Fuzzy Mamdani*

Dari salah satu data lahan tembakau (Ambulu) pada tabel 3.1, siklus penyelesaian masalah menggunakan *fuzzy mamdani* dalam penelitian ini dapat dilakukan melalui 4 tahapan berikut:

1. Pembentukan himpunan *fuzzy* (*fuzzifikasi*)

Terdapat 8 variabel *fuzzy*, dengan ketentuan 7 variabel *input* (persentase lahan terkena penyakit, keterbukaan wilayah, derajat berat tanah, ketebalan lapis



olah, kemudahan irigasi, kondisi medan, dan pH tanah) dan 1 variabel *ouput* (kesesuaian lahan).

- Variabel *input*:

Persentase lahan terkena penyakit : 17

Keterbukaan wilayah : 3

Derajat berat tanah : 3

Ketebalan lapis olah : 30

Kemudahan irigasi : 3

Kondisi medan : 4

pH tanah : 6,8

- Variabel *output*:

Kesesuaian lahan : Sangat sesuai/Cukup Sesuai/Kurang sesuai/Tidak sesuai

Berdasarkan data yang telah didapatkan, rentang nilai *fuzzy* masing-masing kriteria dapat dilihat pada Tabel 4.1.

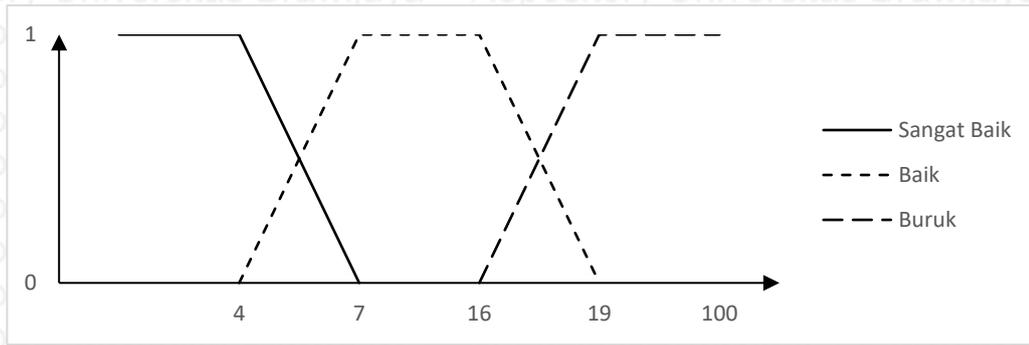
Tabel 4.1 Rentang Nilai Setiap Kriteria

Kriteria	Rentang Nilai
Persentase lahan terkena penyakit	0-100
Keterbukaan wilayah	0-3
Derajat berat tanah	0-3
Ketebalan lapis olah	0-3
Kemudahan irigasi	0-60
Kondisi medan	0-4
pH tanah	0-14

Pada penelitian ini kromosom individu yang dihasilkan dalam algoritme genetika digunakan sebagai batasan fungsi keanggotaan pada masing-masing kriteria (variabel) untuk melakukan *fuzzifikasi* (pembentukan himpunan *fuzzy*). Batasan fungsi keanggotaan pada Gambar 4.4 sampai dengan 4.9 hanya digunakan sebagai contoh dalam pembentukan himpunan *fuzzy* dalam penelitian ini.

a. Kriteria 1 (persentase lahan terkena penyakit)

Nilai dari kriteria persentase lahan terkena penyakit dikelompokkan menjadi tiga himpunan *fuzzy* yaitu sangat baik, baik, dan buruk. Rentang nilai *fuzzy* untuk kriteria persentase lahan terkena penyakit ditunjukkan pada Gambar 4.3.



Gambar 4.3 Grafik Fuzzy Kriteria Persentase Lahan Terkena Penyakit

Selanjutnya dari Gambar 4.3 dapat dibentuk persamaan himpunan fuzzy sebagai berikut:

$$\mu_{\text{Sangat Baik}}(x_1) = \begin{cases} 1; & x \leq 4 \\ (7-x)/3; & 4 < x < 7 \\ 0; & x \geq 7 \end{cases} \quad (4.1)$$

$$\mu_{\text{Baik}}(x_1) = \begin{cases} 0; & x \leq 4 \\ (x-4)/3; & 4 < x < 7 \\ 1; & 7 \leq x \leq 16 \\ (19-x)/3; & 16 < x < 19 \\ 0; & x \geq 17 \end{cases} \quad (4.2)$$

$$\mu_{\text{Buruk}}(x_1) = \begin{cases} 0; & x \leq 16 \\ (x-16)/3; & 16 < x < 19 \\ 1; & x \geq 19 \end{cases} \quad (4.3)$$

Sehingga jika nilai persentase lahan terkena penyakit=17, maka:

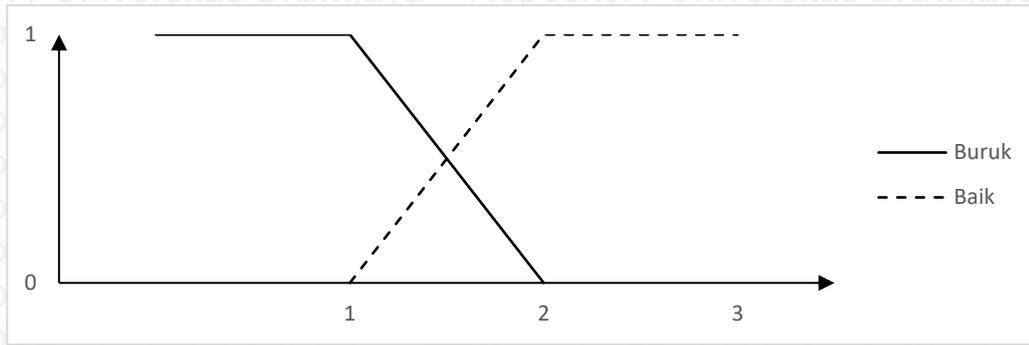
$$\mu_{\text{Sangat Baik}}(x_1) = 0$$

$$\mu_{\text{Baik}}(x_1) = 0,67$$

$$\mu_{\text{Buruk}}(x_1) = 0,33$$

b. Kriteria 2 (keterbukaan wilayah)

Nilai dari kriteria keterbukaan wilayah dikelompokkan menjadi dua himpunan fuzzy yaitu baik dan buruk. Rentang nilai fuzzy untuk kriteria keterbukaan wilayah ditunjukkan pada Gambar 4.4.



Gambar 4.4 Grafik Fuzzy Kriteria Keterbukaan Wilayah

Selanjutnya dari Gambar 4.4 dapat dibentuk persamaan himpunan fuzzy sebagai berikut:

$$\mu_{\text{Buruk}}(x_2) = \begin{cases} 1; & x \leq 1 \\ (2-x)/1; & 1 < x < 2 \\ 0; & x \geq 2 \end{cases} \quad (4.4)$$

$$\mu_{\text{Baik}}(x_2) = \begin{cases} 0; & x \leq 1 \\ (x-1)/1; & 1 < x < 2 \\ 1; & x \geq 2 \end{cases} \quad (4.5)$$

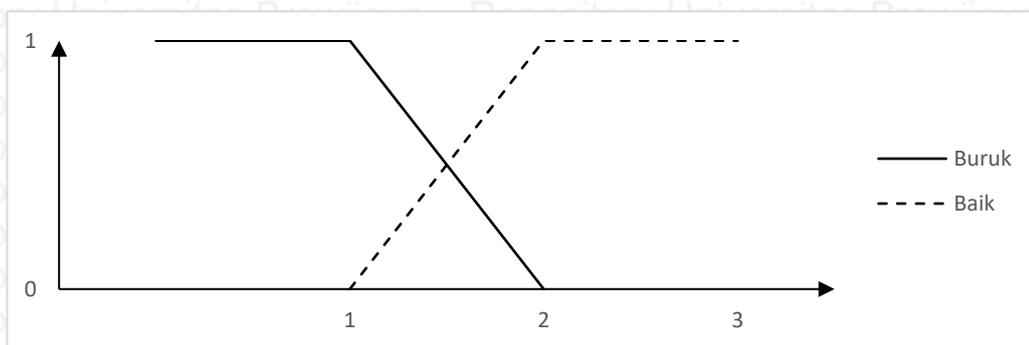
Sehingga jika nilai keterbukaan wilayah=3, maka:

$$\mu_{\text{Buruk}}(x_2) = 0$$

$$\mu_{\text{Baik}}(x_2) = 1$$

c. Kriteria 3 (derajat berat tanah)

Nilai dari kriteria derajat berat tanah dikelompokkan menjadi dua himpunan fuzzy yaitu baik dan buruk. Rentang nilai fuzzy untuk kriteria derajat berat tanah ditunjukkan pada Gambar 4.5.



Gambar 4.5 Grafik Fuzzy Kriteria Derajat Berat Tanah

Selanjutnya dari Gambar 4.5 dapat dibentuk persamaan himpunan fuzzy sebagai berikut:



$$\mu_{\text{Buruk}}(x_1) = \begin{cases} 1; & x \leq 1 \\ (2-x)/1; & 1 < x < 2 \\ 0; & x \geq 2 \end{cases} \quad (4.6)$$

$$\mu_{\text{Baik}}(x_1) = \begin{cases} 0; & x \leq 1 \\ (x-1)/1; & 1 < x < 2 \\ 1; & x \geq 2 \end{cases} \quad (4.7)$$

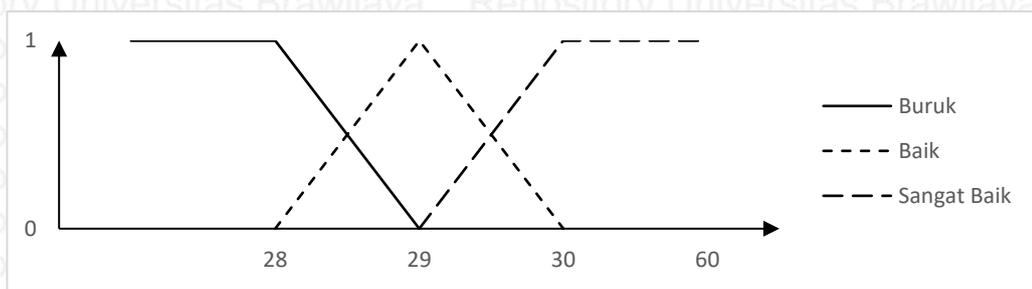
Sehingga jika nilai derajat berat tanah=3, maka:

$$\mu_{\text{Buruk}}(x_3) = 0$$

$$\mu_{\text{Baik}}(x_3) = 1$$

d. Kriteria 4 (ketebalan lapis olah)

Nilai dari kriteria ketebalan lapis olah dikelompokkan menjadi tiga himpunan fuzzy yaitu sangat baik, baik, dan buruk. Rentang nilai fuzzy untuk kriteria ketebalan lapis olah ditunjukkan pada Gambar 4.6.



Gambar 4.6 Grafik Fuzzy Kriteria Ketebalan Lapis Olah

Selanjutnya dari Gambar 4.6 dapat dibentuk persamaan himpunan fuzzy sebagai berikut:

$$\mu_{\text{Sangat Baik}}(x_4) = \begin{cases} 0; & x \leq 29 \\ (x-29)/1; & 29 < x < 30 \\ 1; & x \geq 30 \end{cases} \quad (4.8)$$

$$\mu_{\text{Baik}}(x_4) = \begin{cases} 0; & x \leq 28 \\ (x-28)/1; & 28 < x < 29 \\ (30-x)/1; & 29 < x < 30 \\ 0; & x \geq 30 \end{cases} \quad (4.9)$$

$$\mu_{\text{Buruk}}(x_4) = \begin{cases} 1; & x \leq 28 \\ (29-x)/1; & 28 < x < 29 \\ 0; & x \geq 29 \end{cases} \quad (4.10)$$



Sehingga jika nilai ketebalan lapis olah=30, maka:

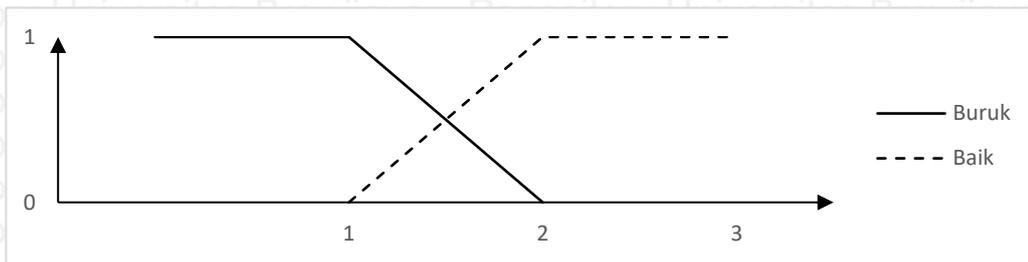
$$\mu_{\text{Sangat Baik}}(x_4) = 1$$

$$\mu_{\text{Baik}}(x_4) = 0$$

$$\mu_{\text{Buruk}}(x_4) = 0$$

e. Kriteria 5 (kemudahan irigasi)

Nilai dari kriteria kemudahan irigasi dikelompokkan menjadi dua himpunan *fuzzy* yaitu baik dan buruk. Rentang nilai *fuzzy* untuk kriteria kemudahan irigasi ditunjukkan pada Gambar 4.7.



Gambar 4.7 Grafik *Fuzzy* Kriteria Kemudahan Irigasi

Selanjutnya dari Gambar 4.7 dapat dibentuk persamaan himpunan *fuzzy* sebagai berikut:

$$\mu_{\text{Buruk}}(x_5) = \begin{cases} 1; & x \leq 1 \\ (2-x)/1; & 1 < x < 2 \\ 0; & x \geq 2 \end{cases} \quad (4.11)$$

$$\mu_{\text{Baik}}(x_5) = \begin{cases} 0; & x \leq 1 \\ (x-1)/1; & 1 < x < 2 \\ 1; & x \geq 2 \end{cases} \quad (4.12)$$

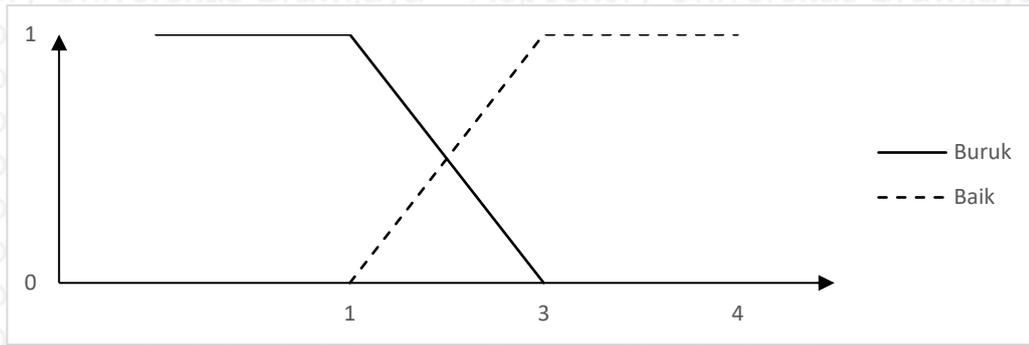
Sehingga jika nilai kemudahan irigasi=3, maka:

$$\mu_{\text{Buruk}}(x_5) = 0$$

$$\mu_{\text{Baik}}(x_5) = 1$$

f. Kriteria 6 (kondisi medan)

Berdasarkan data yang telah didapatkan, nilai dari kondisi medan dikelompokkan menjadi dua himpunan *fuzzy* yaitu baik dan buruk. Rentang nilai *fuzzy* untuk kriteria kondisi medan ditunjukkan pada Gambar 4.8.



Gambar 4.8 Grafik Fuzzy Kriteria Kondisi Medan

Selanjutnya dari Gambar 4.8 dapat dibentuk persamaan himpunan fuzzy sebagai berikut:

$$\mu_{\text{Buruk}}(x_6) = \begin{cases} 1; & x \leq 1 \\ (3-x)/1; & 1 < x < 3 \\ 0; & x \geq 3 \end{cases} \quad (4.13)$$

$$\mu_{\text{Baik}}(x_6) = \begin{cases} 0; & x \leq 1 \\ (x-1)/2; & 1 < x < 3 \\ 1; & x \geq 3 \end{cases} \quad (4.14)$$

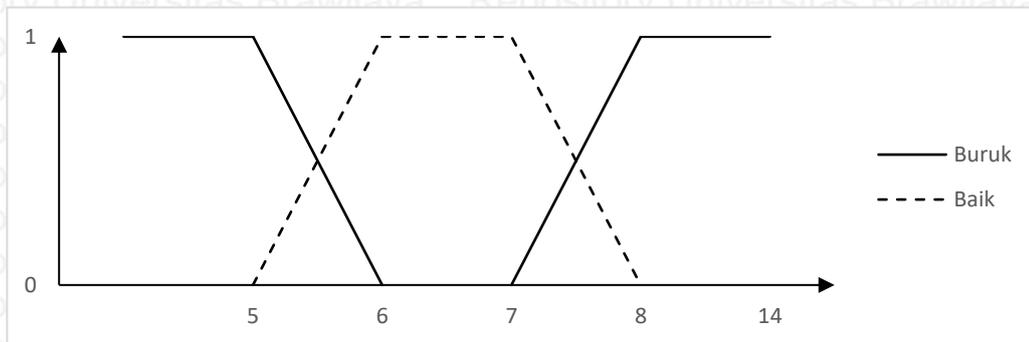
Sehingga jika nilai kondisi medan=4, maka:

$$\mu_{\text{Buruk}}(x_6) = 0$$

$$\mu_{\text{Baik}}(x_6) = 1$$

g. Kriteria 7 (pH tanah)

Berdasarkan data yang telah didapatkan, nilai dari pH tanah dikelompokkan menjadi dua himpunan fuzzy yaitu baik dan buruk. Rentang nilai fuzzy untuk kriteria kondisi medan ditunjukkan pada Gambar 4.9.



Gambar 4.9 Grafik Fuzzy Kriteria pH Tanah

Selanjutnya dari Gambar 4.9 dapat dibentuk persamaan himpunan fuzzy sebagai berikut:



$$\mu_{\text{Baik}}(x_7) = \begin{cases} 0; & x \leq 5 \\ (x-5)/1; & 5 < x < 6 \\ 1; & 6 \leq x \leq 7 \\ (8-x)/1 & 7 < x < 8 \\ 0; & x \geq 8 \end{cases} \quad (4.15)$$

$$\mu_{\text{Buruk}}(x_7) = \begin{cases} 1; & x \leq 5 \\ (6-x)/1; & 5 < x < 6 \\ 0; & 6 \leq x \leq 7 \\ (x-7)/1; & 7 < x < 8 \\ 1; & x \geq 8 \end{cases} \quad (4.16)$$

Sehingga jika nilai pH tanah=6,8, maka:

$$\mu_{\text{Baik}}(x_7) = 1$$

$$\mu_{\text{Buruk}}(x_7) = 0$$

2. Aplikasi fungsi implikasi

Terdapat 7 variabel input, dengan ketentuan variabel persentase lahan terkena penyakit memiliki 3 himpunan (sangat baik, baik, buruk), variabel keterbukaan wilayah memiliki 2 himpunan (buruk, baik), variabel derajat berat tanah memiliki 2 himpunan (buruk, baik), variabel ketebalan lapis olah memiliki 3 himpunan (sangat baik, baik, buruk), variabel kemudahan irigasi memiliki 2 himpunan (buruk, baik), dan variabel pH tanah memiliki 2 himpunan (baik, buruk). Oleh karena itu, jumlah aturan yang dapat dibentuk sebanyak = 3 x 2 x 2 x 3 x 2 x 2 x 2 = 288. Beberapa aturan yang terbentuk ditunjukkan dalam Tabel 4.2. Aturan fuzzy selengkapnya dapat dilihat pada Lampiran A Aturan Fuzzy.

Tabel 4.2 Aturan Fuzzy

No	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	Keputusan
1	Sangat baik	Baik	Baik	Sangat baik	Baik	Baik	Baik	Sangat sesuai
2	Sangat baik	Baik	Baik	Sangat baik	Baik	Baik	Buruk	Cukup sesuai
4	Sangat baik	Baik	Baik	Sangat baik	Baik	Buruk	Buruk	Kurang sesuai
7	Sangat baik	Baik	Baik	Sangat baik	Buruk	Buruk	Baik	Kurang sesuai
8	Sangat baik	Baik	Baik	Sangat baik	Buruk	Buruk	Buruk	Tidak sesuai

Tabel 4.2 Aturan Fuzzy

No	X ₁	X ₂	X ₃	X ₄	X ₅	X ₆	X ₇	Keputusan
97	Baik	Baik	Baik	Sangat baik	Baik	Baik	Baik	Sangat sesuai
101	Baik	Baik	Baik	Sangat baik	Buruk	Baik	Baik	Cukup sesuai
173	Baik	Buruk	Buruk	Sangat baik	Buruk	Baik	Baik	Tidak sesuai
197	Buruk	Baik	Baik	Sangat baik	Buruk	Baik	Baik	Kurang sesuai
201	Buruk	Baik	Baik	Baik	Baik	Baik	Baik	Kurang sesuai
215	Buruk	Baik	Baik	Buruk	Buruk	Buruk	Baik	Tidak sesuai
288	Buruk	Tidak sesuai						

Berdasarkan aturan-aturan yang telah terbentuk pada Tabel 4.4, selanjutnya dapat dilakukan perhitungan pada lahan Ambulu dengan menggunakan fungsi *min* atau operator *AND* sebagai berikut:

$$[R1] = \text{MIN} (\mu_{\text{Sangat Baik}}(x_1) \cap \mu_{\text{Baik}}(x_2) \cap \mu_{\text{Baik}}(x_3) \cap \mu_{\text{Sangat Baik}}(x_4) \cap \mu_{\text{Baik}}(x_5) \cap \mu_{\text{Baik}}(x_6) \cap \mu_{\text{Baik}}(x_7))$$

$$= \text{MIN} (0, 1, 1, 1, 1, 1, 1)$$

$$= 0$$

$$[R2] = \text{MIN} (\mu_{\text{Sangat Baik}}(x_1) \cap \mu_{\text{Baik}}(x_2) \cap \mu_{\text{Baik}}(x_3) \cap \mu_{\text{Sangat Baik}}(x_4) \cap \mu_{\text{Baik}}(x_5) \cap \mu_{\text{Baik}}(x_6) \cap \mu_{\text{Buruk}}(x_7))$$

$$= \text{MIN} (0, 1, 1, 1, 1, 1, 0)$$

$$= 0$$

$$[R4] = \text{MIN} (\mu_{\text{Sangat Baik}}(x_1) \cap \mu_{\text{Baik}}(x_2) \cap \mu_{\text{Baik}}(x_3) \cap \mu_{\text{Sangat Baik}}(x_4) \cap \mu_{\text{Baik}}(x_5) \cap \mu_{\text{Buruk}}(x_6) \cap \mu_{\text{Buruk}}(x_7))$$

$$= \text{MIN} (0, 1, 1, 1, 1, 0, 0)$$

$$= 0$$



$$[R7] = \text{MIN} (\mu_{\text{Sangat Baik}}(X_1) \cap \mu_{\text{Baik}}(X_2) \cap \mu_{\text{Baik}}(X_3) \cap \mu_{\text{Sangat Baik}}(X_4) \cap \mu_{\text{Buruk}}(X_5) \cap \mu_{\text{Buruk}}(X_6) \cap \mu_{\text{Baik}}(X_7))$$

$$= \text{MIN} (0, 1, 1, 1, 0, 0, 1)$$

$$= 0$$

$$[R8] = \text{MIN} (\mu_{\text{Sangat Baik}}(X_1) \cap \mu_{\text{Baik}}(X_2) \cap \mu_{\text{Baik}}(X_3) \cap \mu_{\text{Sangat Baik}}(X_4) \cap \mu_{\text{Buruk}}(X_5) \cap \mu_{\text{Buruk}}(X_6) \cap \mu_{\text{Buruk}}(X_7))$$

$$= \text{MIN} (0, 1, 1, 1, 0, 0, 0)$$

$$= 0$$

$$[R97] = \text{MIN} (\mu_{\text{Baik}}(X_1) \cap \mu_{\text{Baik}}(X_2) \cap \mu_{\text{Baik}}(X_3) \cap \mu_{\text{Sangat Baik}}(X_4) \cap \mu_{\text{Baik}}(X_5) \cap \mu_{\text{Baik}}(X_6) \cap \mu_{\text{Baik}}(X_7))$$

$$= \text{MIN} (0,67, 1, 1, 1, 1, 1)$$

$$= 0,67$$

$$[R101] = \text{MIN} (\mu_{\text{Baik}}(X_1) \cap \mu_{\text{Baik}}(X_2) \cap \mu_{\text{Baik}}(X_3) \cap \mu_{\text{Sangat Baik}}(X_4) \cap \mu_{\text{Buruk}}(X_5) \cap \mu_{\text{Baik}}(X_6) \cap \mu_{\text{Baik}}(X_7))$$

$$= \text{MIN} (0,67, 1, 1, 1, 0, 1, 1)$$

$$= 0$$

$$[R173] = \text{MIN} (\mu_{\text{Baik}}(X_1) \cap \mu_{\text{Buruk}}(X_2) \cap \mu_{\text{Buruk}}(X_3) \cap \mu_{\text{Sangat Baik}}(X_4) \cap \mu_{\text{Buruk}}(X_5) \cap \mu_{\text{Baik}}(X_6) \cap \mu_{\text{Baik}}(X_7))$$

$$= \text{MIN} (0,67, 0, 0, 1, 0, 1, 1)$$

$$= 0$$

$$[R197] = \text{MIN} (\mu_{\text{Buruk}}(X_1) \cap \mu_{\text{Baik}}(X_2) \cap \mu_{\text{Baik}}(X_3) \cap \mu_{\text{Sangat Baik}}(X_4) \cap \mu_{\text{Buruk}}(X_5) \cap \mu_{\text{Baik}}(X_6) \cap \mu_{\text{Baik}}(X_7))$$

$$= \text{MIN} (0,33, 1, 1, 1, 0, 1, 1)$$

$$= 0$$

$$[R201] = \text{MIN} (\mu_{\text{Buruk}}(X_1) \cap \mu_{\text{Baik}}(X_2) \cap \mu_{\text{Baik}}(X_3) \cap \mu_{\text{Baik}}(X_4) \cap \mu_{\text{Baik}}(X_5) \cap \mu_{\text{Baik}}(X_6) \cap \mu_{\text{Baik}}(X_7))$$

$$= \text{MIN} (0,33, 1, 1, 0, 1, 1, 1)$$

$$= 0$$



$$[R215] = \text{MIN} (\mu_{\text{Buruk}}(x_1) \cap \mu_{\text{Baik}}(x_2) \cap \mu_{\text{Baik}}(x_3) \cap \mu_{\text{Buruk}}(x_4) \cap \mu_{\text{Buruk}}(x_5) \cap \mu_{\text{Buruk}}(x_6) \cap \mu_{\text{Baik}}(x_7))$$

$$= \text{MIN} (0,33, 1, 1, 0, 0, 0, 1)$$

$$= 0$$

$$[R288] = \text{MIN} (\mu_{\text{Buruk}}(x_1) \cap \mu_{\text{Buruk}}(x_2) \cap \mu_{\text{Buruk}}(x_3) \cap \mu_{\text{Buruk}}(x_4) \cap \mu_{\text{Buruk}}(x_5) \cap \mu_{\text{Buruk}}(x_6) \cap \mu_{\text{Buruk}}(x_7))$$

$$= \text{MIN} (0,33, 0, 0, 0, 0, 0, 0)$$

$$= 0$$

3. Komposisi aturan

Setelah mendapatkan hasil dari aplikasi fungsi implikasi pada masing-masing aturan, tahap selanjutnya adalah melakukan komposisi aturan menggunakan metode *max*. Dalam metode *max*, solusi *fuzzy* akan diperoleh dengan cara mengambil nilai terbesar dari seluruh aturan pada masing-masing kelasnya.

- Sangat sesuai = $\text{MAX} ([R1] ; [R97])$
= $\text{MAX} (0 ; 0,67)$
= 0,67
- Cukup sesuai = $\text{MAX} ([R2] ; [R101])$
= $\text{MAX} (0 ; 0)$
= 0
- Kurang sesuai = $\text{MAX} ([R4] ; [R7] ; [R197] ; [R201])$
= $\text{MAX} (0 ; 0 ; 0 ; 0)$
= 0
- Tidak sesuai = $\text{MAX} ([R8] ; [R173] ; [R215] ; [R288])$
= $\text{MAX} (0 ; 0 ; 0 ; 0)$
= 0

4. Setelah nilai maksimum derajat keanggotaan untuk masing-masing kelas didapatkan, langkah selanjutnya adalah membandingkan nilai dari masing-masing kelas tersebut. Kelas yang memiliki nilai derajat keanggotaan terbesar akan dijadikan sebagai kelas akhir dari data yang dihitung. Berdasarkan data



lahan tembakau (Ambulu) pada Tabel 3.1 yang telah dihitung maka lahan tersebut “Sangat Sesuai” untuk ditanami tanaman tembakau. Selanjutnya untuk seluruh data lahan yang ada dilakukan proses yang sama seperti yang telah dilakukan sebelumnya sehingga didapatkan keputusan tentang kesesuaian lahan tersebut untuk ditanami tembakau.

4.3 Siklus Optimasi Fungsi Keanggotaan Fuzzy Mamdani Menggunakan Algoritme Genetika

Algoritme genetika dalam penelitian ini digunakan untuk melakukan optimasi fungsi keanggotaan *fuzzy mamdani* dengan tujuan agar nilai akurasi yang dihasilkan menjadi baik. Algoritme genetika terdiri dari empat proses utama, yaitu inialisasi populasi awal, reproduksi, evaluasi, dan seleksi. Sebelum melakukan keempat proses utama dalam algoritme genetika, terlebih dahulu ditentukan representasi kromosom yang nantinya akan digunakan dalam algoritme genetika. Selanjutnya dilakukan inialisasi parameter algoritme genetika seperti jumlah populasi (*popsize*), nilai *crossover rate/cr*, nilai *mutation rate/mr*, dan jumlah generasi.

4.3.1 Representasi Kromosom

Representasi kromosom yang digunakan dalam penelitian ini adalah *real-coded* (pengkodean *real*). Penggunaan representasi kromosom ini dikarenakan kemampuannya dalam melakukan optimasi fungsi yang dapat menjangkau beberapa titik solusi yang mungkin berada dalam daerah kontinu (Mahmudy, 2015). Kromosom dalam permasalahan ini terdiri dari sejumlah bilangan yang merupakan batas dari masing-masing himpunan pada setiap kriteria penentuan kesesuaian lahan, sehingga terdapat 19 gen penyusun dalam satu kromosom. Contoh representasi kromosom yang digunakan ditunjukkan pada Tabel 4.3.

Tabel 4.3 Contoh Representasi Kromosom

Ind	X ₁			X ₂		X ₃		X ₄			X ₅		X ₆		X ₇				
P / C	4	7	14	17	1	2	1	2	29	30	31	1	2	1	3	5	6	7	8

Keterangan :

Ind : Individu

P / C : *Parent / Child*

X₁ : Kriteria persentase lahan terkena penyakit

X₂ : Kriteria keterbukaan wilayah

X₃ : Kriteria derajat berat tanah

X₄ : Kriteria ketebalan lapis olah



X_5 : Kriteria kemudahan irigasi

X_6 : Kriteria Kondisi medan

X_7 : Kriteria pH tanah

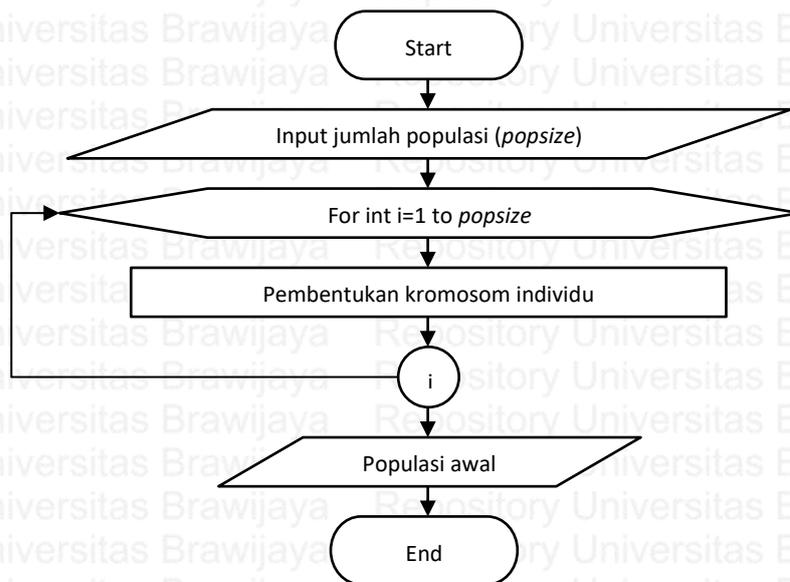
4.3.2 Inisialisasi Parameter Algoritme Genetika

Parameter algoritme genetika yang digunakan dalam menyelesaikan permasalahan optimasi fungsi keanggotaan pada *fuzzy mamdani* adalah jumlah populasi (*popsize*), nilai *crossover rate/cr*, nilai *mutation rate/mr*, dan jumlah generasi. Contoh inisialisasi parameter algoritme genetika yang digunakan adalah sebagai berikut:

- Jumlah populasi (*popsize*) = 4
- Nilai *crossover rate* = 0,4
- Nilai *mutation rate* = 0,2
- Jumlah generasi = 1

4.3.3 Inisialisasi Populasi Awal

Inisialisasi populasi awal dilakukan untuk membangkitkan individu (solusi) dalam populasi sejumlah *popsize* yang telah ditentukan. Proses pembangkitan populasi awal dalam permasalahan ini dilakukan secara *random*. Susunan gen dalam masing-masing kromosom individu dibangkitkan secara acak dengan batasan *range* pada setiap kriteria. Nilai *random/acak* yang telah dibangkitkan pada masing-masing kriteria nantinya harus diurutkan dari yang terkecil pada sisi paling kiri hingga terbesar pada sisi paling kanan. Diagram alir proses inisialisasi populasi awal terlihat seperti pada Gambar 4.10.



Gambar 4.10 Diagram Alir Inisialisasi Populasi Awal



Setelah dilakukan proses pembangkitan populasi awal secara random, akan didapatkan keluaran berupa populasi awal yang berisi kumpulan individu sejumlah *popsize* sebagai representasi dari solusi permasalahan. Individu-individu yang ada dalam populasi awal inilah yang selanjutnya menjadi *parent* dan digunakan pada tahapan berikutnya. Contoh populasi awal yang telah terbentuk ditunjukkan pada Tabel 4.4.

Tabel 4.4 Insialisasi Populasi Awal

Ind	Kromosom																		
	4	7	16	19	1	2	1	2	28	29	30	1	2	1	3	5	6	7	9
P ₁	4	7	16	19	1	2	1	2	28	29	30	1	2	1	3	5	6	7	9
P ₂	5	6	15	16	1	1,5	1	1,5	30	31	32	1	1,5	1	2	5,5	6	7	9
P ₃	6	7	13	15	1,3	2	1	2,5	30	32	34	1	3	1	3	5,5	6,5	7	9
P ₄	3	5	12	13	1	3	2	2,5	28	30	31	2	3	2	3	5	7	8	9

4.3.4 Reproduksi

Reproduksi dilakukan dengan menggunakan dua operator genetika yaitu *crossover* dan mutasi, dengan tujuan untuk mendapatkan keturunan/*offspring*. *Crossover* dilakukan dengan cara melakukan perkawinan silang antara dua induk (*parent*) terpilih yang ada dalam populasi. Rasio *offspring* yang akan dihasilkan dalam proses *crossover* didapatkan dari hasil perkalian *crossover rate/cr* dengan jumlah populasi (*popsize*). Sedangkan mutasi dilakukan dengan cara merubah susunan gen yang ada pada kromosom dari salah satu individu yang terpilih secara acak dalam populasi. Rasio *offspring* yang akan dihasilkan dalam proses mutasi didapatkan dari hasil perkalian *mutation rate/mr* dengan jumlah populasi (*popsize*).

a. *Crossover*

Dalam pembahasan sebelumnya, Proses inialisasi parameter algoritme genetika telah dilakukan dengan ketentuan jumlah populasi (*popsize*) sebanyak 4 dan *crossover rate/cr* sebesar 0,4 sehingga jumlah total *offspring crossover* yang dihasilkan dalam proses *crossover* adalah sebanyak $4 \times 0,4 = 1,6$ (dibulatkan menjadi 2). Satu kali proses *crossover* akan menghasilkan jumlah *offspring* sebanyak 2, sehingga dalam contoh kasus ini hanya akan ada satu kali proses *crossover* yang dilakukan dalam setiap generasi. Metode *crossover* yang digunakan dalam penelitian ini adalah *extended intermediate crossover*, yang menghasilkan *offspring* dari kombinasi nilai dua induk yang dipilih secara acak.

Jika dimisalkan *parent* yang terpilih secara acak untuk digunakan dalam *crossover* adalah *parent* ke-1 dan *parent* ke-2, dengan nilai *alpha* (*a*) 0,1 dan 0,5, maka *offspring* (*C*) yang akan dihasilkan berdasarkan persamaan 2.11 dan 2.12 adalah sebagai berikut:

$$\begin{aligned}
 C_1: \quad X_1 &= 4 + 0,1 (5 - 4) = 4,1 & X_2 &= 7 + 0,1 (6 - 7) = 6,9 \\
 X_3 &= 16 + 0,1 (15 - 16) = 15,9 & X_4 &= 19 + 0,1 (16 - 19) = 18,7
 \end{aligned}$$



$$X_5 = 1 + 0,1 (1 - 1) = 1$$

$$X_7 = 1 + 0,1 (1 - 1) = 1$$

$$X_9 = 28 + 0,1 (30 - 28) = 28,2$$

$$X_{11} = 30 + 0,1 (32 - 30) = 30,2$$

$$X_{13} = 2 + 0,1 (1,5 - 2) = 1,95$$

$$X_{15} = 3 + 0,1 (2 - 3) = 2,9$$

$$X_{17} = 6 + 0,1 (6 - 6) = 6$$

$$X_{19} = 8 + 0,1 (9 - 8) = 8,1$$

$$C_2: X_1 = 5 + 0,5 (4 - 5) = 4,5$$

$$X_3 = 15 + 0,5 (16 - 15) = 15,5$$

$$X_5 = 1 + 0,5 (1 - 1) = 1$$

$$X_7 = 1 + 0,5 (1 - 1) = 1$$

$$X_9 = 30 + 0,5 (28 - 30) = 29$$

$$X_{11} = 32 + 0,5 (30 - 32) = 31$$

$$X_{13} = 1,5 + 0,5 (2 - 1,5) = 1,75$$

$$X_{15} = 2 + 0,5 (3 - 2) = 2,5$$

$$X_{17} = 6 + 0,5 (6 - 6) = 6$$

$$X_{19} = 9 + 0,5 (8 - 9) = 8,5$$

$$X_6 = 2 + 0,1 (1,5 - 2) = 1,95$$

$$X_8 = 2 + 0,1 (1,5 - 2) = 1,95$$

$$X_{10} = 29 + 0,1 (31 - 29) = 29,2$$

$$X_{12} = 1 + 0,1 (1 - 1) = 1$$

$$X_{14} = 1 + 0,1 (1 - 1) = 1$$

$$X_{16} = 5 + 0,1 (5,5 - 5) = 5,05$$

$$X_{18} = 7 + 0,1 (7 - 7) = 7$$

$$X_2 = 6 + 0,5 (7 - 6) = 6,5$$

$$X_4 = 16 + 0,5 (19 - 16) = 17,5$$

$$X_6 = 1,5 + 0,5 (2 - 1,5) = 1,75$$

$$X_8 = 1,5 + 0,5 (2 - 1,5) = 1,75$$

$$X_{10} = 31 + 0,5 (29 - 31) = 30$$

$$X_{12} = 1 + 0,5 (1 - 1) = 1$$

$$X_{14} = 1 + 0,5 (1 - 1) = 1$$

$$X_{16} = 5,5 + 0,5 (5 - 5,5) = 5,25$$

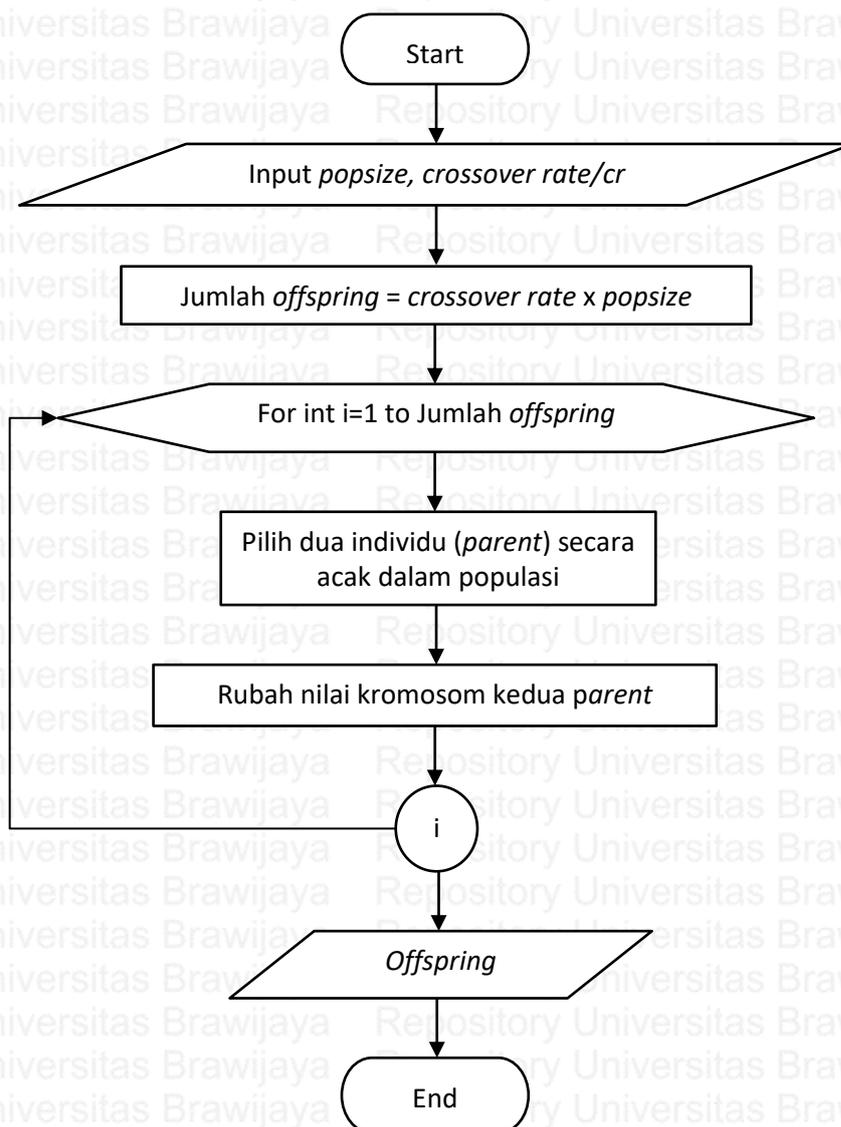
$$X_{18} = 7 + 0,5 (7 - 7) = 7$$

Sehingga kromosom *offspring* yang dihasilkan seperti yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Hasil *Offspring* Metode *Extended Intermediate Crossover*

Ind	Kromosom																		
	4	7	16	19	1	2	1	2	28	29	30	1	2	1	3	5	6	7	9
P ₁	4	7	16	19	1	2	1	2	28	29	30	1	2	1	3	5	6	7	9
P ₂	5	6	15	16	1	1,5	1	1,5	30	31	32	1	1,5	1	2	5,5	6	7	9
C ₁	4,1	6,9	15,9	18,7	1	1,95	1	1,95	28,2	29,2	30,2	1	1,95	1	2,9	5,05	6	7	8,1
C ₂	4,5	6,5	15,5	17,5	1	1,75	1	1,75	29	30	31	1	1,75	1	2,5	5,25	6	7	8,5

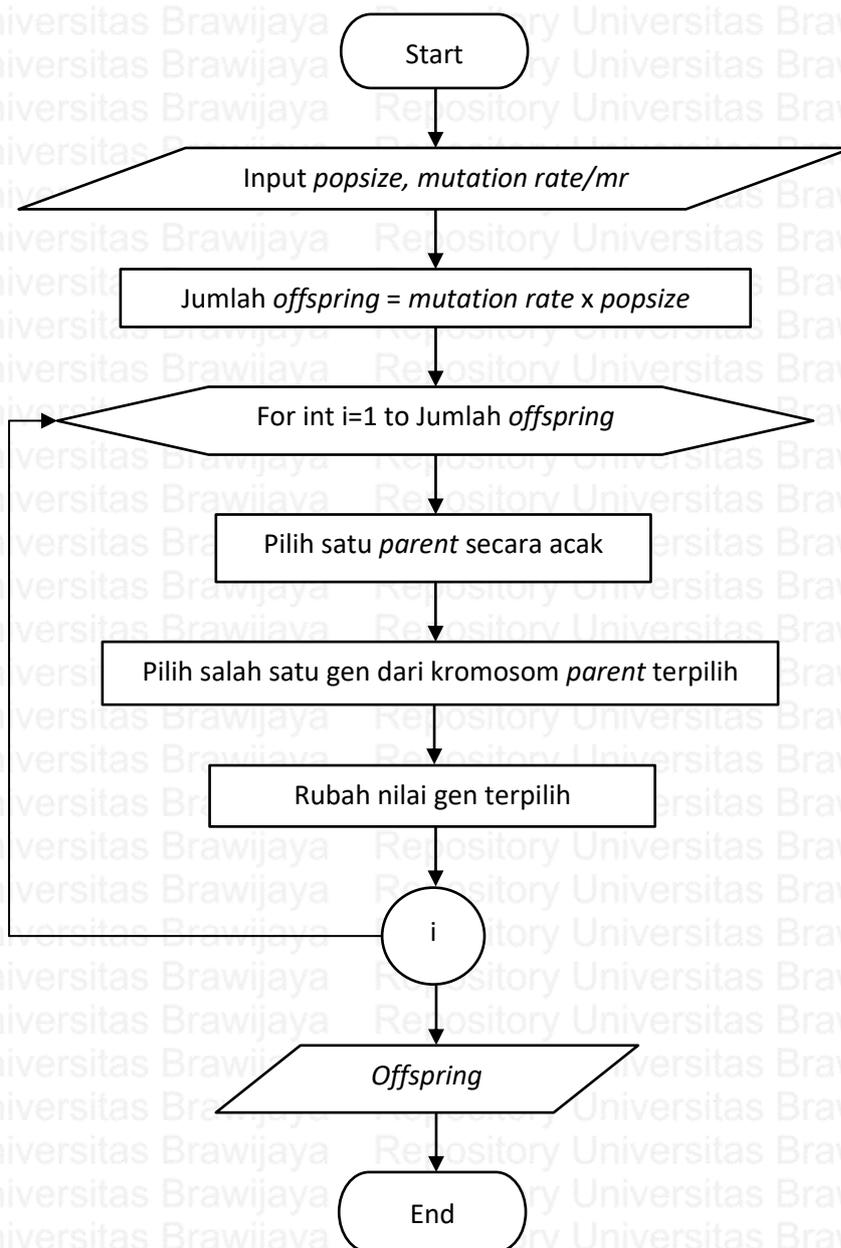
Diagram alir dari proses *crossover* menggunakan metode *extended intermediate crossover* terlihat seperti pada Gambar 4.11.



Gambar 4.11 Diagram Alir Proses *Extended Intermediate Crossover*

b. Mutasi

Jumlah populasi (*popsize*) sebanyak 4 dan *mutation rate/mr* sebesar 0,2 sehingga jumlah *offspring* yang akan dihasilkan dalam satu kali proses mutasi adalah sebanyak $4 \times 0,2 = 0,8$ (dibulatkan menjadi 1). Metode mutasi yang digunakan dalam penelitian ini adalah *random mutation*, yang menghasilkan *offspring* dengan cara menambah atau mengurangi nilai dari gen yang terpilih dari salah satu *parent* dengan bilangan *random* yang kecil. Diagram alir dari proses mutasi menggunakan metode *random mutation* terlihat seperti pada Gambar 4.12.



Gambar 4.12 Diagram Alir Proses *Random Mutation*

Jika dimisalkan *parent* yang terpilih secara acak untuk digunakan dalam mutasi adalah *parent* ke-4, gen terpilih yang akan dirubah nilainya adalah gen ke-9, domain variabel [0, 60], dan $r = -0,0584$, maka *offspring* yang akan dihasilkan berdasarkan persamaan 2.14 adalah sebagai berikut:

$$C_3 : X_9 = 28 - 0,0584 (60 - 0) = 24,4$$

Sehingga kromosom *offspring* yang dihasilkan seperti yang ditunjukkan pada Tabel 4.6.

Tabel 4.6 Hasil Offspring Metode Random Mutation

Ind	Kromosom																		
	P ₄	3	5	12	13	1	3	2	2,5	28	30	31	2	3	2	3	5	7	8
C ₃	3	5	12	13	1	3	2	2,5	24,4	30	31	2	3	2	3	5	7	8	9

4.3.5 Evaluasi

Setelah seluruh individu (*parent* dan *offspring*) terbentuk, langkah selanjutnya adalah menghitung nilai *fitness* untuk masing-masing individu. Nilai *fitness* dalam penelitian ini didapatkan dari hasil akurasi perbandingan antara kelas kesesuaian lahan tanam tembakau yang didapatkan dari pakar dengan kelas kesesuaian lahan tanam tembakau yang dihitung menggunakan metode *fuzzy mamdani* yang telah dioptimasi.

Langkah awal yang dilakukan adalah menentukan kesesuaian setiap lahan tanam tembakau menggunakan *fuzzy mamdani*, dengan ketentuan batasan fungsi keanggotaan pada masing-masing individu. Proses perhitungan *fuzzy mamdani* pada salah satu lahan (Ambulu) pada Tabel 3.1 dengan menggunakan batas fungsi keanggotaan pada kromosom *parent* 1 telah dijelaskan dalam sub-bab 4.2 yang menghasilkan kelas kesesuaian lahan yaitu “Sangat sesuai”. Kemudian dengan cara yang sama dilakukan perhitungan pada seluruh lahan tanam yang digunakan sebagai data latih, sehingga didapatkan hasil akurasi perbandingan antara kelas kesesuaian lahan tanam tembakau yang didapatkan dari pakar dengan kelas kesesuaian lahan tanam tembakau yang dihitung menggunakan metode *fuzzy mamdani* seperti yang ditunjukkan dalam Tabel 4.7.

Tabel 4.7 Akurasi Perbandingan Kelas Pada Parent 1

No	Lahan	Kelas (pakar)	Kelas (<i>fuzzy mamdani</i>)	Akurasi
1	Alas	Sangat Sesuai	Sangat Sesuai	1
2	Klatakan	Sangat Sesuai	Sangat Sesuai	1
3	Parseh	Kurang Sesuai	Sangat Sesuai	0
4	Tugusari	Kurang Sesuai	Kurang Sesuai	1
5	Grintingan	Sangat Sesuai	Sangat Sesuai	1
6	Pecoro Barat	Cukup Sesuai	Sangat Sesuai	0
7	Botosari	Kurang Sesuai	Kurang Sesuai	1
8	Cora 1	Cukup Sesuai	Kurang Sesuai	0
9	KUD	Sangat Sesuai	Sangat Sesuai	1
10	Curah Kates	Cukup Sesuai	Sangat Sesuai	0
11	Dukuhsia Selatan	Kurang Sesuai	Kurang Sesuai	1



No	Lahan	Kelas (pakar)	Kelas (<i>fuzzy mamdani</i>)	Akurasi
12	Dukuhsia	Kurang Sesuai	Kurang Sesuai	1
13	Ambulu	Sangat Sesuai	Sangat Sesuai	1
14	Darungan	Cukup Sesuai	Kurang Sesuai	0
15	Spadia	Sangat Sesuai	Cukup Sesuai	0
16	Curah Buntu	Cukup Sesuai	Cukup Sesuai	1
17	Gayasan	Cukup Sesuai	Cukup Sesuai	1
18	Pondok Labu	Cukup Sesuai	Cukup Sesuai	1
19	Cora 2	Tidak Sesuai	Tidak Sesuai	1
20	Dam 1	Cukup Sesuai	Cukup Sesuai	1
21	Dam 2	Cukup Sesuai	Cukup Sesuai	1
22	Andongsari	Kurang Sesuai	Sangat Sesuai	0
23	Mandilis	Kurang Sesuai	Kurang Sesuai	1
24	Pontang	Cukup Sesuai	Sangat Sesuai	0
25	Randu	Cukup Sesuai	Cukup Sesuai	1
26	Besuk	Cukup Sesuai	Cukup Sesuai	1
27	Manggar	Cukup Sesuai	Cukup Sesuai	1
28	Sumber Rejeki 1	Tidak Sesuai	Tidak Sesuai	1
29	Sumber Rejeki 2	Kurang Sesuai	Kurang Sesuai	1
30	Sumber Waru	Tidak Sesuai	Tidak Sesuai	1

Berdasarkan Tabel 4.7 selanjutnya dihitung nilai *fitness* melalui perbandingan antara kelas kesesuaian lahan tanam tembakau yang didapatkan dari pakar dengan kelas kesesuaian lahan tanam tembakau yang dihitung menggunakan metode *fuzzy mamdani* melalui persamaan 2.18, sehingga nilai *fitness* yang dihasilkan adalah $\frac{22}{30} = 0,73$. Dari hasil perhitungan tersebut dapat diketahui bahwa nilai *fitness* dari *parent 1* adalah 0,73. Kemudian dengan cara yang sama yaitu penentuan kesesuaian seluruh lahan tanam tembakau yang digunakan sebagai data latih dihitung menggunakan *fuzzy mamdani* dengan ketentuan batasan fungsi keanggotaan pada masing-masing individu, nilai *fitness* dari masing-masing individu ditunjukkan dalam Tabel 4.8.

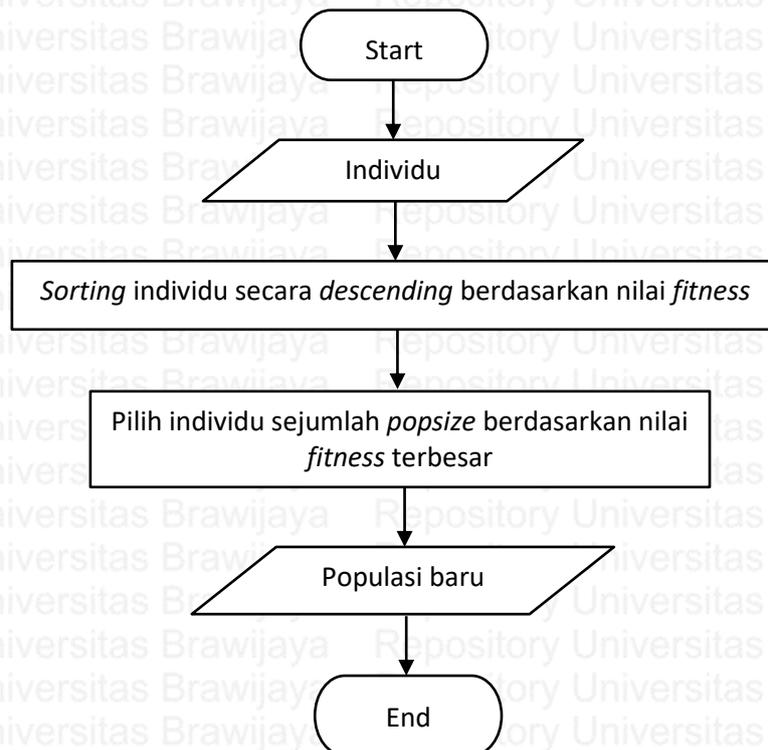
Tabel 4.8 Hasil Evaluasi Nilai *Fitness* Setiap Individu

Ind	Kromosom												<i>Fitness</i>										
	4	5	6	7	8	9	10	11	12	13	14	15											
P ₁	4	5	6	7	16	19	1	2	1	2	28	29	30	1	2	1	3	3	5	6	7	9	0,73
P ₂	5	6	7	15	16	1	1,5	1	1,5	30	31	32	1	1,5	1	2	2	5,5	6	7	9	0,23	
P ₃	6	7	13	15	13	1,3	2	1	2,5	30	32	34	1	3	1	3	3	5,5	6	7	9	0,3	
P ₄	3	5	12	13	1	1	3	2	2,5	28	30	31	2	3	2	3	3	5	7	8	9	0,53	
C ₁	4	6,9	15,9	18,7	1	1	1,95	1	1,95	28,2	29,2	30,2	1	1,95	1	2,9	5,05	6	7	8,1	0,6		
C ₂	4	6,5	15,5	17,5	1	1	1,75	1	1,75	29	30	31	1	1,75	1	2,5	5,25	6	7	8,5	0,47		
C ₃	3	5	12	13	1	1	3	2	2,5	24,4	30	31	2	3	2	3	5	7	8	9	0,43		



4.3.6 Seleksi

Setelah masing-masing individu baik *parent* maupun *offspring* dihitung nilai fitnessnya, tahapan selanjutnya adalah seleksi. Dalam tahapan ini, semua individu diurutkan mulai dari individu yang memiliki nilai *fitness* terbesar sampai dengan individu yang memiliki nilai *fitness* terkecil. Diagram alir dari proses seleksi terlihat seperti pada Gambar 4.13.



Gambar 4.13 Diagram Alir Proses Seleksi Elitism

Berdasarkan Gambar 4.13, proses seleksi *elitism* dilakukan dengan cara mengurutkan seluruh individu baik *parent* maupun *offspring* berdasarkan nilai *fitness* dari yang terbesar sampai yang terkecil. Setelah seluruh individu diurutkan, langkah selanjutnya adalah melakukan seleksi atau memilih individu dengan nilai *fitness* terbesar sejumlah *popsize* untuk digunakan menjadi *parent* pada generasi selanjutnya. Hasil pengurutan individu berdasarkan nilai *fitness* ditunjukkan dalam Tabel 4.9. Proses algoritme genetika ini akan diulang sebanyak jumlah generasi yang telah ditentukan, dan jika jumlah generasi yang ditentukan telah tercapai atau tidak menghasilkan nilai *fitness* yang lebih baik maka proses algoritme akan dihentikan. Hasil seleksi individu ditunjukkan dalam Tabel 4.10.

Tabel 4.9 Hasil *Sorting Individu* Berdasarkan Nilai *Fitness*

Ind	Kromosom														Fitness					
	4	7	16	19	1	2	1	2	28	29	30	1	2	1		3	5	6	7	9
P ₁	4	7	16	19	1	2	1	2	28	29	30	1	2	1	3	5	6	7	9	0,73
C ₁	4	6,9	15,9	18,7	1	1,95	1	1,95	28,2	29,2	30,2	1	1,95	1	2,9	5,05	6	7	8,1	0,6
P ₄	3	5	12	13	1	3	2	2,5	28	30	31	2	3	2	3	5	7	8	9	0,53
C ₂	4	6,5	15,5	17,5	1	1,75	1	1,75	29	30	31	1	1,75	1	2,5	5,25	6	7	8,5	0,47
C ₃	3	5	12	13	1	3	2	2,5	24,4	30	31	2	3	2	3	5	7	8	9	0,43
P ₃	6	7	13	15	1,3	2	1	2,5	30	32	34	1	3	1	3	5,5	6	7	9	0,3
P ₂	5	6	15	16	1	1,5	1	1,5	30	31	32	1	1,5	1	2	5,5	6	7	9	0,23

Tabel 4.10 Hasil Seleksi Menggunakan Metode *Elitism*

Ind	Kromosom											Fitness								
	4	7	16	19	1	2	1	2	28	29	30		1	2	1	3	5	6	7	9
P ₁	4	7	16	19	1	2	1	2	28	29	30	1	2	1	3	5	6	7	9	0,73
P ₂	4	6,9	15,9	18,7	1	1,95	1	1,95	28,2	29,2	30,2	1	1,95	1	2,9	5,05	6	7	8,1	0,6
P ₃	3	5	12	13	1	3	2	2,5	28	30	31	2	3	2	3	5	7	8	9	0,53
P ₄	4	6,5	15,5	17,5	1	1,75	1	1,75	29	30	31	1	1,75	1	2,5	5,25	6	7	8,5	0,47



4.4 Perancangan *User Interface*

Program yang dibuat terdiri dari dua halaman yaitu halaman optimasi untuk mendapatkan fungsi keanggotaan (individu) terbaik dan halaman pengujian untuk mengukur tingkat akurasi yang dihasilkan oleh sistem.

4.4.1 Halaman Proses Optimasi Fungsi Keanggotaan

Halaman optimasi digunakan untuk menampilkan hasil dari serangkaian proses optimasi menggunakan algoritme genetika. Proses optimasi dilakukan untuk mendapatkan hasil akhir berupa individu terbaik setelah beberapa generasi. Rancangan *user interface* untuk halaman optimasi terlihat seperti pada Gambar 4.14.

Gambar 4.14 Rancangan Halaman Proses Optimasi

Keterangan :

- A. Judul program yang dibuat.
- B. *TextField* untuk memasukkan jumlah populasi.
- C. *TextField* untuk memasukkan jumlah generasi.
- D. *TextField* untuk memasukkan nilai probabilitas *crossover*.
- E. *TextField* untuk memasukkan nilai probabilitas mutasi.
- F. *Button* untuk melakukan proses optimasi.
- G. *Button* untuk mereset semua nilai input dan output.
- H. *Table* untuk menampilkan kromosom individu pada masing-masing generasi
- I. *Table* untuk menampilkan nilai *fitness* dari setiap individu
- J. *Table* untuk menampilkan kromosom individu terbaik



4.4.2 Halaman Proses Pengujian Akurasi Sistem

Halaman pengujian digunakan untuk menampilkan hasil dari proses pengujian akurasi sistem yang dilakukan. Pengujian dilakukan untuk mengetahui tingkat akurasi yang dihasilkan oleh sistem setelah mendapatkan fungsi keanggotaan paling optimum. Rancangan *user interface* untuk halaman pengujian akurasi sistem terlihat seperti pada Gambar 4.15.

Gambar 4.15 Rancangan Halaman Proses Pengujian Akurasi Sistem

Keterangan :

- A. *Table* untuk menampilkan fungsi keanggotaan paling optimum
- B. *Table* untuk menampilkan data hasil pengujian
- C. *Button* untuk melakukan proses testing
- D. Bagian untuk menampilkan nilai akurasi sistem



BAB 5 IMPLEMENTASI

5.1 Implementasi Program

Dalam bab ini dijelaskan tentang implementasi program optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau. Proses implementasi dilakukan berdasarkan perancangan yang telah dilakukan pada bab sebelumnya. Bahasa pemrograman yang digunakan dalam implementasi program adalah Java.

5.1.1 Implementasi Proses Inisialisasi Populasi Awal

Proses inisialisasi populasi awal diimplementasikan dengan nama method inisialisasi. Inisialisasi dilakukan dengan melakukan *random* bilangan pada *range* tertentu untuk masing-masing kriteria yang kemudian dilakukan pengurutan. *Source code* dari proses inisialisasi populasi awal ditunjukkan dalam Kode Program 5.1.

```

1 public double[][] inisialisasi(int p, double[][] individu) {
2     for (int i = 0; i < p; i++) {
3         for (int j = 0; j < 19; j++) {
4             double btsBawah, btsAtas;
5             if (j < 4) {
6                 btsBawah = 0;
7                 btsAtas = 100;
8                 BigDecimal individu2 = new BigDecimal(btsBawah +
9                 (random.nextDouble() * (btsAtas - btsBawah))).setScale(1,
10                RoundingMode.CEILING);
11                String indiv = individu2.toString();
12                individu[i][j] = Double.parseDouble(indiv);
13            } else if (j > 3 && j < 6) {
14                btsBawah = 0;
15                btsAtas = 3;
16                BigDecimal individu2 = new BigDecimal(btsBawah +
17                (random.nextDouble() * (btsAtas - btsBawah))).setScale(1,
18                RoundingMode.CEILING);
19                String indiv = individu2.toString();
20                individu[i][j] = Double.parseDouble(indiv);
21            } else if (j > 5 && j < 8) {
22                btsBawah = 0;
23                btsAtas = 3;
24                BigDecimal individu2 = new BigDecimal(btsBawah +
25                (random.nextDouble() * (btsAtas - btsBawah))).setScale(1,
26                RoundingMode.CEILING);
27                String indiv = individu2.toString();
28                individu[i][j] = Double.parseDouble(indiv);
29            } else if (j > 7 && j < 11) {
30                btsBawah = 0;
31                btsAtas = 60;
32                BigDecimal individu2 = new BigDecimal(btsBawah +
33                (random.nextDouble() * (btsAtas - btsBawah))).setScale(1,
34                RoundingMode.CEILING);
35                String indiv = individu2.toString();
36                individu[i][j] = Double.parseDouble(indiv);
37            } else if (j > 10 && j < 13) {
38                btsBawah = 0;
39                btsAtas = 3;
40            }
41        }
42    }
43 }

```



```

42         BigDecimal individu2 = new BigDecimal(btsBawah +
43 (random.nextDouble() * (btsAtas - btsBawah))).setScale(1,
44 RoundingMode.CEILING);
45         String indiv = individu2.toString();
46         individu[i][j] = Double.parseDouble(indiv);
47     } else if (j > 12 && j < 15) {
48         btsBawah = 0;
49         btsAtas = 4;
50         BigDecimal individu2 = new BigDecimal(btsBawah +
51 (random.nextDouble() * (btsAtas - btsBawah))).setScale(1,
52 RoundingMode.CEILING);
53         String indiv = individu2.toString();
54         individu[i][j] = Double.parseDouble(indiv);
55     } else if (j > 14 && j < 19) {
56         btsBawah = 0;
57         btsAtas = 14;
58         BigDecimal individu2 = new BigDecimal(btsBawah +
59 (random.nextDouble() * (btsAtas - btsBawah))).setScale(1,
60 RoundingMode.CEILING);
61         String indiv = individu2.toString();
62         individu[i][j] = Double.parseDouble(indiv);
63     }
64     }
65 }
66 return individu;
67 }

```

Kode Program 5.1 Proses Inisialisasi Populasi Awal

Penjelasan:

1. Baris 1 adalah deklarasi method dengan nama inisialisasi.
2. Baris 2-3 adalah perulangan untuk melakukan inisialisasi.
3. Baris 4 adalah deklarasi variabel dengan nama btsAtas dan btsBawah.
4. Baris 5-12 adalah proses inisialisasi populasi awal dengan kondisi jika kromosom yang akan diinisialisasi adalah kromosom 0-3.
5. Baris 13-20 adalah proses inisialisasi populasi awal dengan kondisi jika kromosom yang akan diinisialisasi adalah kromosom 4-5.
6. Baris 21-28 adalah proses inisialisasi populasi awal dengan kondisi jika kromosom yang akan diinisialisasi adalah kromosom 6-7.
7. Baris 29-36 adalah proses inisialisasi populasi awal dengan kondisi jika kromosom yang akan diinisialisasi adalah kromosom 8-10.
8. Baris 37-46 adalah proses inisialisasi populasi awal dengan kondisi jika kromosom yang akan diinisialisasi adalah kromosom kromosom 11-12.
9. Baris 47-54 adalah proses inisialisasi populasi awal dengan kondisi jika kromosom yang akan diinisialisasi adalah kromosom 13-14.



10. Baris 55-62 adalah proses inialisasi populasi awal dengan kondisi jika kromosom yang akan diinisialisasi adalah kromosom 15-18.

11. Baris 66 adalah pengembalian nilai hasil proses inialisasi populasi awal.

5.1.2 Implementasi Proses *Crossover*

Proses *crossover* diimplementasikan dengan nama method prosesCrossover. Proses ini dilakukan sampai jumlah total *offspring crossover* dihasilkan, dengan ketentuan *offspring* yang dihasilkan sebanyak 2 dalam satu kali proses *crossover*. *Source code* dari proses *crossover* ditunjukkan dalam Kode Program 5.2.

```

1 public double[][] randomCrossover(double[][] childCR, double[][]
2 individu, int pop, int jmlCross) {
3     double[] alpha = new double[19];
4     int idx1, idx2;
5
6     double[][] ind = individu;
7     int jCross = 0;
8
9     for (int i = 0; i < 19; i++) {
10        BigDecimal a = new BigDecimal(0.25 + Math.random() * (1.25 -
11 0.25)).setScale(1, RoundingMode.CEILING);
12        String al = a.toString();
13        alpha[i] = Double.parseDouble(al);
14    }
15
16    double urut;
17    for (int j = 0; j < 19; j++) {
18        for (int l = 0; l < 19 - 1; l++) {
19            if (alpha[l] > alpha[l + 1]) {
20                urut = alpha[l];
21                alpha[l] = alpha[l + 1];
22                alpha[l + 1] = urut;
23            }
24        }
25    }
26
27    int proses = jmlCross / 2;
28    for (int i = 0; i < proses; i++) {
29        idx1 = random.nextInt(pop);
30        idx2 = random.nextInt(pop);
31        int j = 0;
32        while (j < 1) {
33            if (idx1 == idx2) {
34                idx1 = random.nextInt(pop);
35                idx2 = random.nextInt(pop);
36            } else if (idx1 != idx2) {
37                double[][] childc = new double[2][19];
38                for (int jk = 0; jk < 19; jk++) {
39                    BigDecimal hasil1 = new BigDecimal(individu[idx1][jk] +
40 (alpha[jk] * (individu[idx2][jk] -
41 individu[idx1][jk])).setScale(1, RoundingMode.CEILING);
42                    BigDecimal hasil2 = new BigDecimal(individu[idx2][jk] +
43 (alpha[jk] * (individu[idx1][jk] -
44 individu[idx2][jk])).setScale(1, RoundingMode.CEILING);
45                    String anak1 = hasil1.toString();
46                    String anak2 = hasil2.toString();
47                    childc[0][jk] = Double.parseDouble(anak1);
48                    childc[1][jk] = Double.parseDouble(anak2);
49                }
50                for (int k = 0; k < 2; k++) {

```



```

51         childCR[jCross++] = childc[k];
52     }
53 }
54     j++;
55 }
56 }
57     return childCR;
58 }

```

Kode Program 5.2 Proses Crossover

Penjelasan:

1. Baris 1 adalah deklarasi method dengan nama prosesCrossover.
2. Baris 3 adalah deklarasi array dengan nama alpha.
3. Baris 4 adalah deklarasi variabel dengan nama idx1, idx2.
4. Baris 6 adalah deklarasi array dua dimensi dengan nama ind.
5. Baris 7 adalah deklarasi inialisasi variabel jCross dengan nilai 0.
6. Baris 9-14 adalah perulangan yang berisi proses dalam melakukan random nilai *alpha*.
7. Baris 16 adalah deklarasi variabel dengan namaurut.
8. Baris 17-25 adalah perulangan yang berisi proses pengurutan nilai *alpha* yang sudah didapatkan.
9. Baris 27-28 adalah perulangan yang berisi proses *crossover*.
10. Baris 29-30 adalah proses *random parent* yang akan digunakan dalam *crossover*.
11. Baris 33-35 adalah kondisi jika *parent* hasil random sama, sehingga dilakukan proses *random* ulang.
12. Baris 36-38 adalah kondisi jika hasil random berbeda, sehingga dilanjutkan ke proses *crossover*.
13. Baris 15-26 adalah perulangan yang berisi proses *crossover*.
14. Baris 39-44 adalah proses persilangan dua *parent* (*crossover*) untuk menghasilkan offspring yang disimpan dalam variabel hasil1 dan hasil2.
15. Baris 45-46 adalah proses untuk merubah tipe data hasil *crossover* menjadi *String*.
16. Baris 47-48 adalah proses untuk merubah tipe data *String* hasil *crossover* menjadi *double* dan disimpan dalam array childc.



17. 50-51 adalah perulangan yang berisi proses penyimpanan *child* hasil *crossover* ke dalam variabel *childCR*.

18. Baris 57 adalah pengembalian nilai *childCR* hasil proses *crossover*.

5.1.3 Implementasi Proses Mutasi

Proses mutasi diimplementasikan dengan nama method *prosesMutasi*. Mutasi dilakukan sampai jumlah total *offspring* mutasi terpenuhi. *Source code* dari proses mutasi ditunjukkan dalam Kode Program 5.3.

```

1 public double[][] randomMutasi(double[][] childMT, double[][]
2 individu, int pop, int jmlMutasi, int jmlCross) {
3     int randomParent, randomKromosom;
4     double[][] ind = individu;
5     for (int i = 0; i < jmlMutasi; i++) {
6         double[] childm = new double[19];
7         double[] mts = new double[19];
8         randomParent = random.nextInt(pop);
9         for (int j = 0; j < 19; j++) {
10            mts[j] = ind[randomParent][j];
11            childm[j] = ind[randomParent][j];
12        }
13        randomKromosom = random.nextInt(19);
14        BigDecimal a = new BigDecimal(0.25 + Math.random() * (1.25 -
15 0.25)).setScale(1, RoundingMode.CEILING);
16        String a1 = a.toString();
17        double alpha = Double.parseDouble(a1);
18        double tamp1 = mts[randomKromosom];
19        if (randomKromosom < 4) {
20            BigDecimal hasil1 = new BigDecimal(tamp1 + (alpha * (100 -
21 0))).setScale(1, RoundingMode.CEILING);
22            String anak1 = hasil1.toString();
23            childm[randomKromosom] = Double.parseDouble(anak1);
24        }
25        if (childm[randomKromosom] < 0) {
26            childm[randomKromosom] = 0;
27        } else if (childm[randomKromosom] > 100) {
28            childm[randomKromosom] = 100;
29        } else {
30            childm[randomKromosom] = childm[randomKromosom];
31        }
32    }
33    }
34    if (randomKromosom > 3 && randomKromosom < 6) {
35        BigDecimal hasil1 = new BigDecimal(tamp1 + (alpha * (3 -
36 0))).setScale(1, RoundingMode.CEILING);
37        String anak1 = hasil1.toString();
38        childm[randomKromosom] = Double.parseDouble(anak1);
39    }
40    if (childm[randomKromosom] < 0) {
41        childm[randomKromosom] = 0;
42    } else if (childm[randomKromosom] > 3) {
43        childm[randomKromosom] = 3;
44    } else {
45        childm[randomKromosom] = childm[randomKromosom];
46    }
47    }
48    }
49    if (randomKromosom > 5 && randomKromosom < 8) {
50        BigDecimal hasil1 = new BigDecimal(tamp1 + (alpha * (3 -
51 0))).setScale(1, RoundingMode.CEILING);
52        String anak1 = hasil1.toString();

```



```

53     childm[randomKromosom] = Double.parseDouble(anak1);
54
55     if (childm[randomKromosom] < 0) {
56         childm[randomKromosom] = 0;
57     } else if (childm[randomKromosom] > 3) {
58         childm[randomKromosom] = 3;
59     } else {
60         childm[randomKromosom] = childm[randomKromosom];
61     }
62 }
63 }
64     if (randomKromosom > 7 && randomKromosom < 11) {
65         BigDecimal hasil1 = new BigDecimal(tamp1 + (alpha * (60 -
66 0))).setScale(1, RoundingMode.CEILING);
67         String anak1 = hasil1.toString();
68         childm[randomKromosom] = Double.parseDouble(anak1);
69
70     if (childm[randomKromosom] < 0) {
71         childm[randomKromosom] = 0;
72     } else if (childm[randomKromosom] > 60) {
73         childm[randomKromosom] = 60;
74     } else {
75         childm[randomKromosom] = childm[randomKromosom];
76     }
77 }
78 }
79     if (randomKromosom > 10 && randomKromosom < 13) {
80         BigDecimal hasil1 = new BigDecimal(tamp1 + (alpha * (3 -
81 0))).setScale(1, RoundingMode.CEILING);
82         String anak1 = hasil1.toString();
83         childm[randomKromosom] = Double.parseDouble(anak1);
84
85     if (childm[randomKromosom] < 0) {
86         childm[randomKromosom] = 0;
87     } else if (childm[randomKromosom] > 3) {
88         childm[randomKromosom] = 3;
89     } else {
90         childm[randomKromosom] = childm[randomKromosom];
91     }
92 }
93 }
94     if (randomKromosom > 12 && randomKromosom < 15) {
95         BigDecimal hasil1 = new BigDecimal(tamp1 + (alpha * (4 -
96 0))).setScale(1, RoundingMode.CEILING);
97         String anak1 = hasil1.toString();
98         childm[randomKromosom] = Double.parseDouble(anak1);
99
100    if (childm[randomKromosom] < 0) {
101        childm[randomKromosom] = 0;
102    } else if (childm[randomKromosom] > 4) {
103        childm[randomKromosom] = 4;
104    } else {
105        childm[randomKromosom] = childm[randomKromosom];
106    }
107 }
108 }
109    if (randomKromosom > 14 && randomKromosom < 19) {
110        BigDecimal hasil1 = new BigDecimal(tamp1 + (alpha * (14 -
111 0))).setScale(1, RoundingMode.CEILING);
112        String anak1 = hasil1.toString();
113        childm[randomKromosom] = Double.parseDouble(anak1);
114
115    if (childm[randomKromosom] < 0) {
116        childm[randomKromosom] = 0;
117    } else if (childm[randomKromosom] > 14) {

```



```

118     childm[randomKromosom] = 14;
119     } else {
120     childm[randomKromosom] = childm[randomKromosom];
121
122     }
123     }
124     childMT[i] = childm;
125     }
126     return childMT;
127     }

```

Kode Program 5.3 Proses Mutasi

Penjelasan:

1. Baris 1 adalah deklarasi method dengan nama prosesMutasi.
2. Baris 3 adalah deklarasi variabel dengan nama randomParent dan randomKromosom.
3. Baris 4 adalah deklarasi array dua dimensi dengan nama ind.
4. Baris 5-12 adalah perulangan yang berisi deklarasi dan inisialisasi array childm dan mts, dan didalamnya terdapat proses *random parent*.
5. Baris 13 adalah untuk melakukan pemilihan gen secara random yang akan dirubah nilainya.
6. Baris 14-17 adalah proses *random* nilai *alpha*.
7. Baris 18 adalah inisialisasi variabel tamp1 dengan nilai gen yang telah didapatkan sebelumnya.
8. Baris 19-33 adalah proses mutasi dengan kondisi jika gen yang terpilih adalah gen 0-3.
9. Baris 34-48 adalah proses mutasi dengan kondisi jika gen yang terpilih adalah gen 4 atau 5.
10. Baris 49-63 adalah proses mutasi dengan kondisi jika gen yang terpilih adalah gen 6 atau 7.
11. Baris 64-78 adalah proses mutasi dengan kondisi jika gen yang terpilih adalah gen 8-10.
12. Baris 79-93 adalah proses mutasi dengan kondisi jika gen yang terpilih adalah gen 11 atau 12.
13. Baris 94-108 adalah proses mutasi dengan kondisi jika gen yang terpilih adalah gen 13 atau 14.



14. Baris 109-123 adalah proses mutasi dengan kondisi jika gen yang terpilih adalah gen 15-18.
15. Baris 124 adalah proses penyimpanan *child* hasil mutasi ke dalam variabel *childMT*.
16. Baris 126 adalah pengembalian nilai hasil proses mutasi.

5.1.4 Implementasi Proses Perhitungan Nilai *Fitness*

Dalam perhitungan *fitness* dilakukan pembagian jumlah data yang sesuai dengan jumlah total data keseluruhan. Jumlah data yang sesuai dijadikan sebagai parameter dalam method *fitness*. *Source code* dari proses perhitungan *fitness* ditunjukkan dalam Kode Program 5.4.

```

1 public double fitness(double a) {
2     double ftn;
3     ftn = a / 30;
4     return ftn;
5 }

```

Kode Program 5.4 Proses Perhitungan *Fitness*

Penjelasan:

1. Baris 1 adalah deklarasi method dengan nama *fitness*.
2. Baris 2 adalah deklarasi variabel dengan nama *ftn*.
3. Baris 3 merupakan proses perhitungan nilai *fitness*.
4. Baris 4 adalah pengembalian nilai hasil perhitungan *fitness*.

5.1.5 Implementasi Proses Seleksi Elitism

Proses seleksi diimplementasikan dengan nama method seleksi. Metode seleksi yang digunakan adalah *elitism*, yang dilakukan dengan cara mengurutkan individu berdasarkan nilai *fitness* terbesar dan kemudian dipilih sejumlah *popsiz*. *Source code* dari proses seleksi ditunjukkan dalam Kode Program 5.5.

```

1 public void seleksi(int popsize, double[][] individu, int
2 jmlCross, int jmlMutasi, double[] fitnessGabungan, double[][]
3 kromosomGabungan) {
4     double tukar;
5     double[][] tukar2 = new double[1][19];
6     for (int i = 0; i < popsize + jmlCross + jmlMutasi; i++) {
7         for (int j = 0; j < popsize + jmlCross + jmlMutasi - 1; j++) {
8
9             if (fitnessGabungan[j] < fitnessGabungan[j + 1]) {
10                tukar = fitnessGabungan[j];
11                fitnessGabungan[j] = fitnessGabungan[j + 1];
12                fitnessGabungan[j + 1] = tukar;
13                tukar2[0] = kromosomGabungan[j];
14                kromosomGabungan[j] = kromosomGabungan[j + 1];
15                kromosomGabungan[j + 1] = tukar2[0];
16            }
17        }

```



```

18     }
19     for (int i = 0; i < popsize; i++) {
20         individu[i] = kromosomGabungan[i];
21     }
22     return individu;
23 }

```

Kode Program 5.5 Proses Seleksi *Elitism*

Penjelasan:

1. Baris 1 adalah deklarasi method dengan nama seleksi.
2. Baris 4 adalah deklarasi variabel dengan nama tukar.
3. Baris 5 adalah deklarasi array dua dimensi dengan nama tukar2.
4. Baris 6-18 adalah perulangan yang berisi proses pengurutan individu berdasarkan nilai *fitness* mulai dari yang terbesar sampai yang terkecil.
5. Baris 19-21 adalah perulangan yang berisi proses pengambilan individu sejumlah *popsize* berdasarkan nilai *fitness* terbesar untuk dijadikan *parent* pada generasi selanjutnya.
6. Baris 22 pengembalian nilai hasil seleksi.

5.1.6 Implementasi Proses Fuzzifikasi

Fuzzifikasi dilakukan pada setiap himpunan *fuzzy* dalam masing-masing kriteria. *Source code* dari salah satu proses fuzzifikasi ditunjukkan dalam Kode Program 5.6.

```

1 public double penyakitBaik(double x) {
2     if (x <= penyakit1) {
3         penyakitBaik = 0;
4     } else if (x > penyakit1 && x < penyakit2) {
5         penyakitBaik = (x - penyakit1) / selisihPenyakit1;
6     } else if (x >= penyakit2 && x <= penyakit3) {
7         penyakitBaik = 1;
8     } else if (x > penyakit3 && x < penyakit4) {
9         penyakitBaik = (penyakit4 - x) / selisihPenyakit2;
10    } else if (x >= penyakit4) {
11        penyakitBaik = 0;
12    }
13    return penyakitBaik;
14 }

```

Kode Program 5.6 Proses Fuzzifikasi

Penjelasan:

1. Baris 1 adalah deklarasi method dengan nama penyakitBaik.
2. Baris 2-3 adalah fuzzifikasi dengan kondisi jika nilai data inputan kurang dari atau sama dengan nilai penyakit1.



3. Baris 4-5 adalah fuzzifikasi dengan kondisi jika nilai data inputan lebih dari nilai penyakit1 dan kurang dari nilai penyakit2.
4. Baris 6-7 adalah fuzzifikasi dengan kondisi jika nilai data inputan lebih dari atau sama dengan nilai penyakit2 dan kurang dari atau sama dengan nilai penyakit3.
5. Baris 8-9 adalah fuzzifikasi dengan kondisi jika nilai data inputan lebih dari nilai penyakit3 dan kurang dari nilai penyakit4.
6. Baris 10-11 adalah fuzzifikasi dengan kondisi jika nilai data inputan lebih dari atau sama dengan nilai penyakit4.
7. Baris 13 adalah proses pengembalian nilai hasil fuzzifikasi.

5.1.7 Implementasi Proses Implikasi

Implikasi dilakukan dengan menggunakan fungsi min atau operator *AND*. Proses ini dilakukan dengan cara memilih nilai minimum dari setiap rule yang ada. *Source code* dari proses implikasi ditunjukkan dalam Kode Program 5.7.

```

1 private double Min(double k1, double k2, double k3, double k4,
2 double k5, double k6, double k7) {
3     double k[] = new double[7];
4     k[0] = k1;
5     k[1] = k2;
6     k[2] = k3;
7     k[3] = k4;
8     k[4] = k5;
9     k[5] = k6;
10    k[6] = k7;
11
12    double min = k[0];
13
14    for (int i = 0; i < 7; i++) {
15        if (k[i] < min) {
16            min = k[i];
17        }
18    }
19    return min;
20 }

```

Kode Program 5.7 Proses Implikasi

Penjelasan:

7. Baris 1 adalah deklarasi method dengan nama Min.
8. Baris 3 adalah deklarasi array dengan nama k.
9. Baris 4-10 adalah proses inisialisasi array k.
10. Baris 12 adalah inisialisasi variabel min dengan nilai dari array k indeks ke-0.
11. Baris 14-18 adalah perulangan yang berisi perbandingan nilai dari semua indeks array k untuk mendapatkan nilai terkecil.



12. Baris 19 adalah pengembalian nilai terkecil yang didapatkan dari proses sebelumnya.

5.1.8 Implementasi Proses Komposisi Aturan

Komposisi aturan dilakukan dengan menggunakan fungsi max atau operator *OR*. Proses komposisi aturan dilakukan untuk masing-masing kelas, dengan tujuan untuk mencari derajat keanggotaan tertinggi pada masing-masing kelas kesesuaian lahan. *Source code* dari proses komposisi aturan salah satu kelas ditunjukkan dalam Kode Program 5.8.

```

1 public double komposisi1() {
2     ruleSangatSesuai();
3     max1 = ruleSangatSesuai[0];
4     for (int i = 0; i < 4; i++) {
5         if (ruleSangatSesuai[i] > max1) {
6             max1 = ruleSangatSesuai[i];
7         }
8     }
9     return max1;
10 }

```

Kode Program 5.8 Proses Komposisi Aturan

Penjelasan:

1. Baris 1 adalah deklarasi method dengan nama komposisi1.
2. Baris 2 adalah pemanggilan method ruleSangatSesuai.
3. Baris 3 adalah inisialisasi variabel max1 dengan nilai dari array ruleSangatSesuai indeks ke-0.
4. Baris 4-8 adalah perulangan yang berisi perbandingan nilai dari semua indeks array ruleSangatSesuai untuk mendapatkan nilai terbesar.
5. Baris 9 adalah pengembalian nilai terbesar yang didapatkan dari proses sebelumnya.

5.2 Implementasi *User Interface*

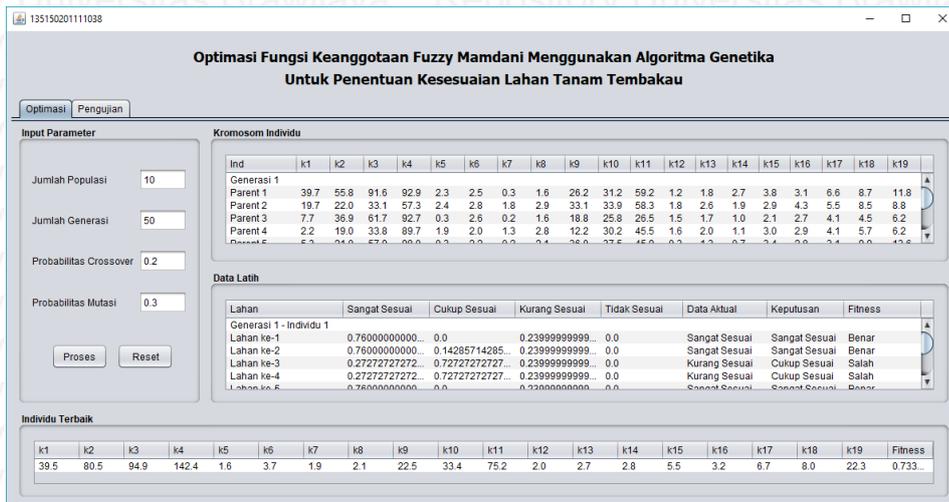
Seperti yang telah dijelaskan dalam perancangan *user interface* pada bab 4, program yang dibuat terdiri dari dua halaman yaitu halaman optimasi dan halaman pengujian. Halaman pertama digunakan untuk melakukan proses optimasi dengan menggunakan inputan parameter algoritme, sedangkan halaman kedua digunakan untuk melakukan pengujian akurasi sistem dengan menggunakan fungsi keanggotaan optimum yang didapatkan dari hasil optimasi.

5.2.1 Implementasi Halaman Proses Optimasi Fungsi Keanggotaan

Pada halaman optimasi, terdapat form untuk memasukkan inputan berupa parameter algoritme genetika. Dibawah form untuk memasukkan inputan terdapat dua buah *button* yaitu proses dan reset. Selain itu dalam halaman

optimasi juga terdapat tiga buah tabel yang digunakan untuk menampilkan hasil proses optimasi.

Tabel pertama berisi 20 kolom, dengan rincian kolom pertama berisi individu pada tiap generasi dan kolom selanjutnya adalah kromosom dari masing-masing individu. Tabel kedua memiliki 8 kolom yang berisi tentang data latih, nilai suatu lahan pada masing-masing kelas, keputusan terhadap lahan dan nilai *fitness* untuk masing-masing individu. Tabel yang terakhir dalam halaman ini berisi kromosom dari individu terbaik. Hasil implementasi halaman proses optimasi dapat dilihat pada Gambar 5.1.



Gambar 5.1 Antarmuka Halaman Proses Optimasi

5.2.2 Implementasi Halaman Proses Pengujian Akurasi Sistem

Pada halaman pengujian terdapat dua buah tabel, *button* untuk melakukan testing dan hasil akurasi yang dihasilkan. Tabel pertama dalam halaman pengujian berisi fungsi keanggotaan paling optimum yang merupakan kromosom dari individu terbaik, sedangkan tabel kedua berisi data hasil pengujian yang dilakukan. Hasil implementasi halaman proses pengujian dapat dilihat pada Gambar 5.2.



Gambar 5.2 Antarmuka Halaman Proses Pengujian Akurasi Sistem



BAB 6 PENGUJIAN DAN ANALISIS

Dalam bab ini dibahas tentang pengujian yang telah dilakukan pada penelitian ini seperti pengujian jumlah populasi, pengujian kombinasi nilai cr dan mr , pengujian jumlah generasi, dan pengujian akurasi sistem. Setelah pengujian dilakukan selanjutnya dilakukan analisis untuk mendapatkan kesimpulan tentang parameter-parameter yang menghasilkan nilai *fitness* terbaik.

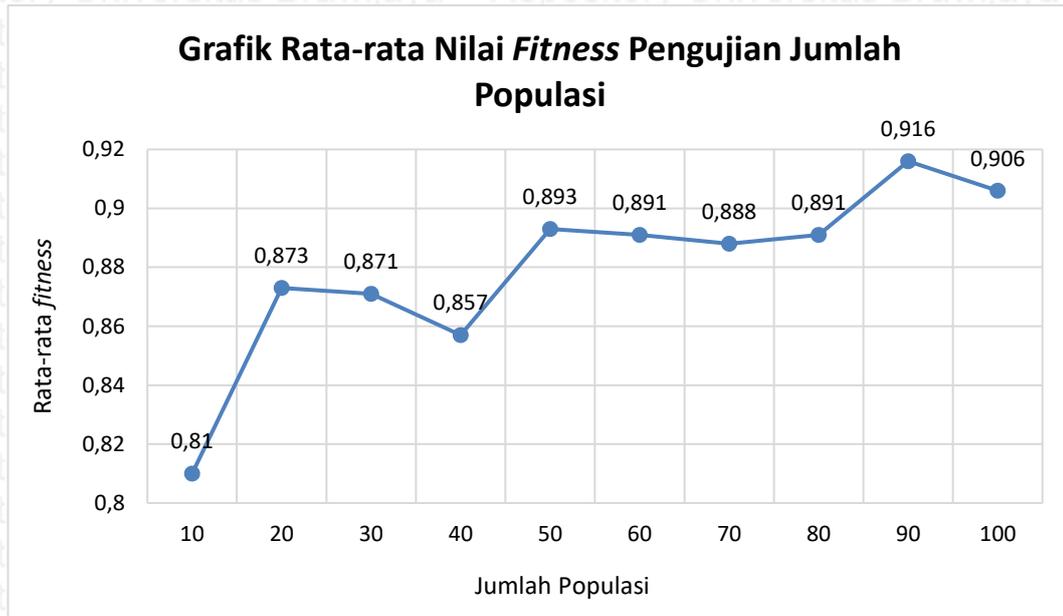
6.1 Hasil dan Analisis Pengujian Jumlah Populasi (*Popsize*)

Pengujian populasi dilakukan dengan tujuan untuk mendapatkan jumlah atau ukuran populasi terbaik yang dapat menghasilkan batasan fungsi keanggotaan paling optimal dalam *Fuzzy Inference System Mamdani* untuk penentuan kesesuaian lahan tanam tembakau. Jumlah populasi yang diujikan adalah kelipatan 10 mulai dari 10 sampai dengan 100. Pada masing-masing jumlah populasi yang diujikan didapatkan rata-rata nilai *fitness* setelah dilakukan percobaan sebanyak 10 kali. Jumlah generasi yang digunakan dalam pengujian ini adalah 500, dengan nilai *crossover rate* (cr) dan *mutation rate* (mr) berturut-turut adalah 0,5 dan 0,5. Hasil dari pengujian jumlah populasi ditunjukkan dalam Tabel 6.1.

Tabel 6.1 Hasil Pengujian Jumlah Populasi (*Popsize*)

<i>Popsize</i>	Nilai <i>Fitness</i> Percobaan Ke-i										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
10	0,73	0,77	0,8	0,87	0,8	0,73	0,87	0,83	0,83	0,87	0,81
20	0,87	0,87	0,9	0,9	0,87	0,83	0,83	0,83	0,9	0,93	0,873
30	0,87	0,87	0,9	0,83	0,9	0,9	0,9	0,77	0,87	0,9	0,871
40	0,83	0,8	0,87	0,9	0,87	0,9	0,87	0,83	0,8	0,9	0,857
50	0,83	0,9	0,93	0,9	0,87	0,9	0,9	0,97	0,83	0,9	0,893
60	0,87	0,87	0,93	0,93	0,87	0,9	0,87	0,87	0,9	0,9	0,891
70	0,87	0,87	0,87	0,87	0,9	0,9	0,9	0,9	0,87	0,93	0,888
80	0,9	0,9	0,93	0,87	0,9	0,87	0,9	0,87	0,9	0,87	0,891
90	0,93	0,9	0,93	0,9	0,9	0,9	0,93	0,97	0,9	0,9	0,916
100	0,9	0,9	0,87	0,9	0,93	0,9	0,9	0,93	0,9	0,93	0,906

Berdasarkan Tabel 6.1, selanjutnya dapat dibuat grafik perubahan rata-rata nilai *fitness* jumlah populasi yang diujikan seperti yang terlihat dalam Gambar 6.1.



Gambar 6.1 Grafik Rata-rata Nilai *Fitness* Pengujian Jumlah Populasi

Dari Gambar 6.1 dapat diamati bahwa perbedaan jumlah populasi memiliki pengaruh terhadap rata-rata nilai *fitness* yang dihasilkan. Rata-rata nilai *fitness* terkecil dihasilkan oleh jumlah populasi sebesar 10, hal ini disebabkan jumlah populasi 10 belum dapat mencapai daerah pencarian yang optimal. Sedangkan rata-rata nilai *fitness* terbesar dihasilkan oleh jumlah populasi sebesar 90, dan dalam rentang jumlah populasi 10 sampai dengan 90, rata-rata nilai *fitness* yang dihasilkan mengalami naik turun. Pada saat jumlah populasi sebesar 100, rata-rata nilai *fitness* yang dihasilkan kembali mengalami penurunan. Hal ini menunjukkan bahwa jumlah populasi yang semakin besar tidak menjamin bahwa rata-rata nilai *fitness* yang dihasilkan juga semakin baik karena jumlah populasi yang besar dapat memperluas area pencarian sehingga membuat adanya kemungkinan untuk keluar dari daerah pencarian yang optimal. Setelah melakukan pengujian jumlah populasi dalam penelitian ini, maka dapat disimpulkan bahwa jumlah populasi terbaik yang dapat digunakan adalah sebesar 90.

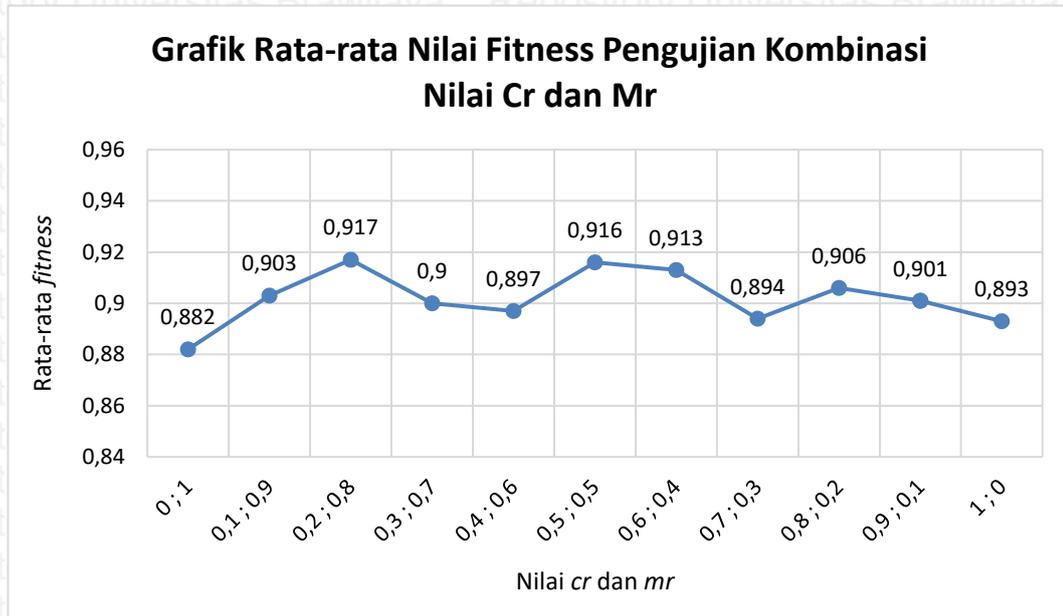
6.2 Hasil dan Analisis Pengujian Kombinasi Nilai *Cr* dan *Mr*

Pengujian ini dilakukan dengan tujuan untuk mendapatkan kombinasi nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*) yang paling baik sehingga dapat menghasilkan batasan fungsi keanggotaan paling optimal. Uji coba dilakukan sebanyak 10 kali pada setiap kombinasi dengan menggunakan jumlah populasi dan jumlah generasi yang sama. Jumlah populasi yang digunakan adalah jumlah populasi terbaik dari hasil pengujian jumlah populasi yang telah dilakukan sebelumnya, sedangkan jumlah generasi yang digunakan adalah 500. Hasil dari pengujian kombinasi nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*) ditunjukkan dalam Tabel 6.2.

Tabel 6.2 Hasil Pengujian Kombinasi Nilai *Cr* dan *Mr*

Nilai <i>cr</i>	Nilai <i>mr</i>	Nilai <i>Fitness</i> Percobaan Ke-i										Rata-rata Nilai <i>Fitness</i>
		1	2	3	4	5	6	7	8	9	10	
0	1	0,87	0,9	0,87	0,87	0,87	0,87	0,9	0,9	0,87	0,87	0,882
0,1	0,9	0,9	0,9	0,9	0,9	0,9	0,87	0,9	0,93	0,9	0,93	0,903
0,2	0,8	0,9	0,9	0,9	0,97	0,93	0,9	0,9	0,97	0,9	0,9	0,917
0,3	0,7	0,9	0,9	0,93	0,9	0,87	0,9	0,87	0,9	0,9	0,93	0,9
0,4	0,6	0,9	0,9	0,93	0,9	0,9	0,9	0,9	0,87	0,9	0,87	0,897
0,5	0,5	0,93	0,9	0,93	0,9	0,9	0,9	0,93	0,97	0,9	0,9	0,916
0,6	0,4	0,87	0,93	0,9	0,9	0,9	0,93	0,97	0,9	0,93	0,9	0,913
0,7	0,3	0,93	0,87	0,9	0,87	0,9	0,9	0,9	0,9	0,9	0,87	0,894
0,8	0,2	0,9	0,9	0,9	0,93	0,93	0,87	0,93	0,87	0,93	0,9	0,906
0,9	0,1	0,97	0,9	0,9	0,9	0,9	0,9	0,87	0,9	0,9	0,87	0,901
1	0	0,93	0,83	0,87	0,9	0,87	0,9	0,9	0,9	0,93	0,9	0,893

Berdasarkan Tabel 6.2, selanjutnya dapat dibuat grafik perubahan rata-rata nilai *fitness* kombinasi nilai *cr* dan *mr* yang diujikan seperti yang terlihat dalam Gambar 6.2.



Gambar 6.2 Grafik Rata-rata Nilai *Fitness* Pengujian Kombinasi Nilai *Cr* dan *Mr*

Pengujian kombinasi nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*) menghasilkan rata-rata nilai *fitness* yang tidak pasti atau naik turun. Berdasarkan hasil pengujian yang telah dilakukan seperti yang terlihat pada Tabel 6.2 dan juga Gambar 6.2, rata-rata nilai *fitness* terkecil didapatkan dari kombinasi nilai *cr* sebesar 0 dan *mr* sebesar 1, sedangkan rata-rata nilai *fitness* terbesar didapatkan



pada saat kombinasi nilai cr sebesar 0,2 dan mr sebesar 0,8. Nilai $fitness$ yang naik turun atau tidak pasti menandakan bahwa kombinasi antara nilai cr dan mr sangat mempengaruhi nilai $fitness$ yang dihasilkan. Jika nilai cr lebih besar dari nilai mr maka hal ini akan menurunkan kemampuan dari algoritme genetika dalam menjelajahi daerah pencarian karena sulit mendapatkan area baru dan hanya terbatas pada area yang telah dijelajahi oleh *parent* saja. Setelah melakukan pengujian kombinasi nilai cr dan mr ini dapat dikatakan bahwa reproduksi secara mutasi mampu memberikan area baru dan tidak terbatas hanya pada area yang telah dijelajahi oleh *parent*.

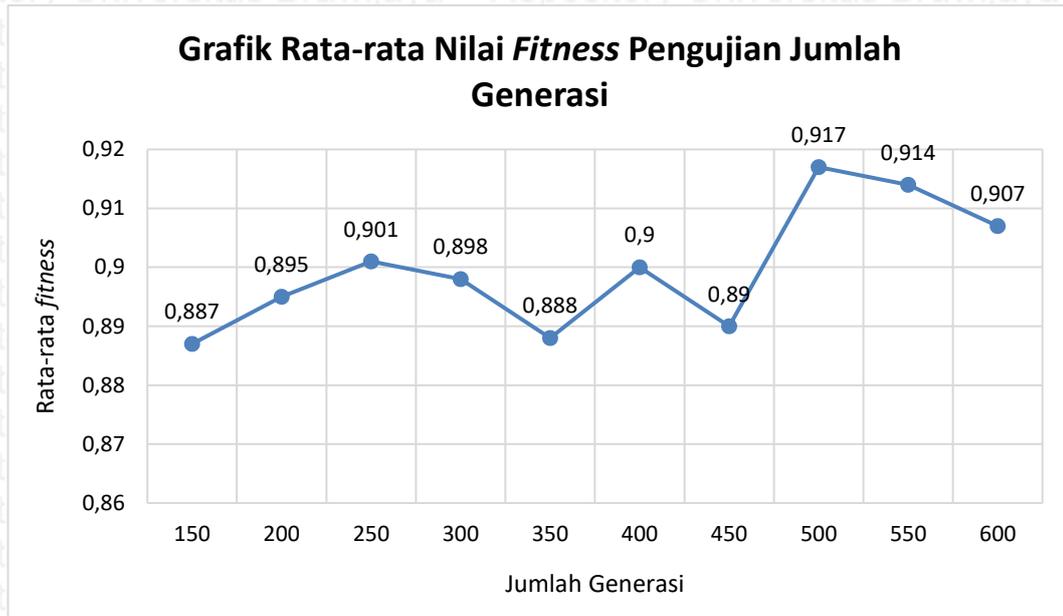
6.3 Hasil dan Analisis Pengujian Jumlah Generasi

Pengujian ini dilakukan untuk mendapatkan jumlah generasi terbaik sehingga dapat menghasilkan batasan fungsi keanggotaan paling optimal dalam *Fuzzy Inference System Mamdani* untuk penentuan kesesuaian lahan tanam tembakau. Jumlah generasi yang digunakan adalah kelipatan 50 mulai dari 150 sampai dengan 600. Rata-rata nilai $fitness$ masing-masing jumlah generasi yang diujikan dapat diketahui setelah dilakukan percobaan sebanyak 10 kali pada masing-masing jumlah generasi. Jumlah populasi, kombinasi nilai cr dan mr yang digunakan adalah nilai yang terbaik dari hasil pengujian sebelumnya. Hasil dari pengujian jumlah generasi ditunjukkan dalam Tabel 6.3.

Tabel 6.3 Hasil Pengujian Jumlah Generasi

Jumlah Generasi	Nilai <i>Fitness</i> Percobaan Ke-i										Rata-rata Nilai <i>Fitness</i>
	1	2	3	4	5	6	7	8	9	10	
150	0,83	0,93	0,87	0,9	0,87	0,9	0,93	0,87	0,87	0,9	0,887
200	0,97	0,9	0,87	0,93	0,87	0,87	0,9	0,87	0,87	0,9	0,895
250	0,9	0,87	0,9	0,97	0,9	0,9	0,9	0,9	0,87	0,9	0,901
300	0,87	0,87	0,9	0,9	0,93	0,87	0,9	0,87	0,97	0,9	0,898
350	0,93	0,87	0,87	0,93	0,9	0,9	0,87	0,87	0,87	0,87	0,888
400	0,9	0,93	0,93	0,9	0,87	0,9	0,87	0,9	0,9	0,9	0,9
450	0,87	0,9	0,87	0,9	0,83	0,87	0,9	0,93	0,9	0,93	0,89
500	0,9	0,9	0,9	0,97	0,93	0,9	0,9	0,97	0,9	0,9	0,917
550	0,87	0,9	0,9	0,9	0,9	0,97	0,97	0,9	0,9	0,93	0,914
600	0,9	0,9	0,93	0,9	0,9	0,9	0,87	0,87	0,97	0,93	0,907

Berdasarkan Tabel 6.3, selanjutnya dapat dibuat grafik perubahan rata-rata nilai $fitness$ jumlah generasi yang diujikan seperti yang terlihat dalam Gambar 6.3.



Gambar 6.3 Grafik Rata-rata Nilai *Fitness* Pengujian Jumlah Generasi

Gambar 6.3 menunjukkan pengaruh jumlah generasi terhadap rata-rata nilai *fitness* yang dihasilkan mulai dari 150 generasi sampai dengan 600 generasi. Berdasarkan hasil pengujian jumlah generasi yang telah dilakukan, rata-rata nilai *fitness* terkecil didapatkan pada saat jumlah generasi sebesar 150, sedangkan pada saat jumlah generasi sebesar 500 dihasilkan rata-rata nilai *fitness* terbesar. Jumlah generasi terkecil yang diujikan yaitu 150 membuat area pencarian masih sempit sehingga rata-rata nilai *fitness* yang dihasilkan menjadi kecil. Pada saat jumlah generasi 250 terjadi peningkatan rata-rata nilai *fitness* yang dihasilkan, kemudian rata-rata nilai *fitness* yang dihasilkan mengalami naik turun sampai pada generasi 500 yang menghasilkan rata-rata nilai *fitness* terbesar dalam pengujian. Selanjutnya pada pengujian dengan jumlah generasi 550 dan 600 nilai *fitness* yang dihasilkan terus mengalami penurunan.

Dari hasil pengujian jumlah generasi yang telah dilakukan dapat ditarik kesimpulan bahwa jumlah generasi yang besar akan menghasilkan nilai *fitness* yang baik, meskipun pada proses komputasinya membutuhkan waktu yang lebih lama. Tetapi, jumlah generasi yang semakin besar tidak akan selalu menjamin bahwa nilai *fitness* yang dihasilkan semakin baik juga. Hal ini dikarenakan pada titik tertentu jumlah generasi telah berada pada daerah pencarian optimal, sehingga jika jumlah generasi diperbesar dapat menurunkan nilai *fitness* yang dihasilkan karena telah menjauhi daerah pencarian yang optimal.

6.4 Hasil Pengujian Akurasi Sistem

Pengujian ini dilakukan untuk mengetahui tingkat akurasi yang dihasilkan oleh sistem setelah dilakukan optimasi menggunakan algoritme genetika. Dalam pengujian ini digunakan 10 data uji untuk melihat kesamaan output dalam penentuan kesesuaian lahan tanam tembakau. Parameter algoritme genetika yang digunakan dalam pengujian ini adalah parameter yang paling optimum dari



hasil pengujian sebelumnya, yaitu jumlah populasi sebesar 90, nilai cr sebesar 0,2, nilai mr sebesar 0,8, dan jumlah generasi sebesar 500. Dari hasil pengujian terhadap 10 data uji yang telah dilakukan, terdapat 8 data yang outputnya sama dan 2 data yang outputnya tidak sama sehingga dengan demikian dapat diketahui bahwa akurasi yang dihasilkan oleh sistem adalah sebesar 80%. Hasil dari pengujian akurasi sistem ditunjukkan dalam Tabel 6.4.

Tabel 6.4 Hasil Pengujian Akurasi Sistem

No	Lahan	Output (Pakar)	Output (Optimasi Fuzzy Mamdani)
1	Nogosari	Sangat Sesuai	Sangat Sesuai
2	Karang Odeng	Kurang Sesuai	Tidak Sesuai
3	Mayang	Sangat Sesuai	Sangat Sesuai
4	Jubung	Cukup Sesuai	Cukup Sesuai
5	Wirowongso	Cukup Sesuai	Cukup Sesuai
6	Banjarsari Barat	Cukup Sesuai	Cukup Sesuai
7	Banjarsari Tengah	Tidak Sesuai	Tidak Sesuai
8	Banjar Sengon	Kurang Sesuai	Kurang Sesuai
9	Bedadung	Kurang Sesuai	Kurang Sesuai
10	Plalangan	Tidak Sesuai	Kurang Sesuai



BAB 7 KESIMPULAN

7.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis yang telah dilakukan dalam penelitian optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau, beberapa kesimpulan yang didapatkan adalah sebagai berikut:

1. Perubahan pada nilai parameter algoritme genetika memiliki pengaruh terhadap nilai *fitness* yang dihasilkan. Berdasarkan hasil pengujian yang telah dilakukan, parameter terbaik dari algoritme genetika yang dapat memberikan solusi paling optimum dalam penelitian ini adalah jumlah populasi sebesar 90, kombinasi nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*) berturut-turut adalah 0,2 dan 0,8, dan jumlah generasi sebesar 500. Rata-rata nilai *fitness* yang dihasilkan pada kombinasi nilai parameter terbaik tersebut adalah 0,917.
2. Nilai akurasi yang dihasilkan dalam penelitian optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau adalah sebesar 80%. Akurasi tersebut didapatkan dari hasil perbandingan output kesesuaian lahan yang didapatkan dari pakar dengan output kesesuaian lahan hasil optimasi *fuzzy mamdani*. Dalam perbandingan tersebut, dari 10 data uji yang digunakan terdapat 8 data yang outputnya sama dan 2 data yang outputnya tidak sama.

7.2 Saran

Berdasarkan penelitian optimasi fungsi keanggotaan *fuzzy mamdani* menggunakan algoritme genetika untuk penentuan kesesuaian lahan tanam tembakau yang telah dilakukan, saran yang dapat diberikan untuk penelitian selanjutnya adalah menambahkan data yang lebih banyak dan bervariasi sehingga nilai akurasi sistem yang dihasilkan dalam penelitian ini dapat ditingkatkan lagi. Penggunaan data yang semakin banyak dan bervariasi dapat membuat pola data semakin terlihat, sehingga sistem dapat menghasilkan akurasi yang lebih baik.

DAFTAR PUSTAKA

- Armanda, R. S., Mahmudy, W. F. 2016. *Penerapan Algoritma Genetika Untuk Penentuan Batasan Fungsi Keanggotaan Fuzzy Tsukamoto Pada Kasus Peramalan Permintaan Barang*. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), Vol. 3 No. 3. Universitas Brawijaya.
- Auliya, Y. A., Mahmudy, W. F. 2016. *Pemilihan Lahan Tanam Optimum Untuk Tanaman Tembakau Menggunakan Fuzzy Inference System (FIS) Tsukamoto*. Seminar Nasional Teknologi Informasi dan Komunikasi 2016 (SENTIKA 2016). Universitas Brawijaya.
- Azizah, E. N., Cholissoddin, I., Mahmudy, W. F. 2015. *Optimasi Fungsi Keanggotaan Fuzzy Tsukamoto Menggunakan Algoritma Genetika Untuk Penentuan Harga Jual Rumah*. Journal of Environmental Engineering & Sustainable Technology, Vol. 02 No. 02. Universitas Brawijaya.
- Imawati, D., Abadi, A. M. 2016. *Analisis Sistem Inferensi Fuzzy Sugeno dalam Menentukan Kesesuaian Lahan Tembakau di Kabupaten Temanggung*. Seminar Nasional Matematika dan Pendidikan Matematika UNY 2016. Universitas Negeri Yogyakarta.
- Kusumadewi, S., Purnomo, H. 2013. *Aplikasi Logika Fuzzy Untuk Pendukung Keputusan Edisi 2*. Yogyakarta: Graha Ilmu.
- Laksono, R. A., Apriyanti, D. H. 2009. *Aplikasi Sistem Informasi Geografis Dalam Evaluasi Pemilihan Lahan Tanam*. Konservasi Flora Indonesia dalam Mengatasi Dampak Pemanasan Global.
- Mahmudy, W. F. 2015. *Algoritma Evolusi*. Universitas Brawijaya. Malang.
- Nandadiri, Annisa., Swasono, D. I., Adiwijaya, N. O. 2014. *Sistem Informasi Geografis Pemilihan Lahan Tembakau Di Kabupaten Jember Berbasis Web Menggunakan Metode Topsis-Ahp*. Artikel Ilmiah Hasil Penelitian Mahasiswa Tahun 2014. Universitas Jember.
- Nisak, A., Soebroto, A. A., Furqon, M. T. 2015. *Sistem Pakar Diagnosa Penyakit Pada Sapi Potong Dengan Metode Fuzzy Inference System (FIS) Tsukamoto (Studi Kasus: Pos Keswan Kabupaten Nganjuk)*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol. 6 No. 5.
- Pramesti, R. A., Mahmudy, W. F. 2016. *Optimasi Fuzzy Inference System Mamdani Untuk Memprediksi Nilai Tukar Rupiah Terhadap Dollar Amerika Menggunakan Algoritma Genetika*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol. 7 No. 27.
- Restuputri, B. A., Mahmudy, W. F., Cholissoddin, I. 2015. *Optimasi Fungsi Keanggotaan Fuzzy Tsukamoto Dua Tahap Menggunakan Algoritma Genetika Pada Pemilihan Calon Penerima Beasiswa dan BBP-PPA (Studi Kasus: PTIIK Universitas Brawijaya Malang)*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol. 5 No. 11.



Ritung, S., Wahyunto., Agus, F., Hidayat, H. 2007. *Panduan Evaluasi Kesesuaian Lahan dengan Contoh Peta Arahan Penggunaan Lahan Kabupaten Aceh Barat*. Balai Penelitian Tanah dan World Agroforestry Centre (ICRAF). Bogor, Indonesia.

Robandi, I. 2006. *Desain Sistem Tenaga Modern, Optimasi, Logika Fuzzy, dan Algoritma Genetika*. Yogyakarta: Penerbit Andi.

Setiawan, B. D., Subanar. 2010. *Color Pixel Classification Using Genetic Fuzzy System: Case Study on Earth Surface Classification*. International Conference on Distributed Framework for Multimedia Applications (DFmA). Universitas Gadjah Mada.

Sevani, M., Marimin., Sukoco, H. 2009. *Sistem Pakar Penentuan Kesesuaian Lahan Berdasarkan Faktor Penghambat Terbesar (Maximum Limitation Factor) Untuk Tanaman Pangan*. Jurnal Informatika, Vol. 10 No. 1. Universitas Krida Wacana.

Widodo, A. W., Mahmudy, W. F. 2010. *Penerapan Algoritma Genetika Pada Sistem Rekomendasi Wisata Kuliner*. Jurnal Ilmiah KURSOR, Vol. 5 No. 4. Universitas Brawijaya.

Zukhri, Z. 2014. *Algoritma Genetika Metode Komputasi Evolusioner Untuk Menyelesaikan Masalah Optimasi*. Yogyakarta: Penerbit Andi.

Zulfa, I., Mahmudy, W. F. 2016. *Optimasi Model Fuzzy Mamdani Dalam Penentuan Kualitas Air Sungai Dengan Menggunakan Algoritma Evolution Strategies*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol. 7 No. 25.