

**OPTIMASI FUNGSI KEANGGOTAAN *FUZZY* TSUKAMOTO
DUA TAHAP MENGGUNAKAN ALGORITMA GENETIKA
PADA PEMILIHAN CALON PENERIMA BEASISWA-PPA
DAN BBP-PPA**

(Studi Kasus: PTIIK Universitas Brawijaya Malang)

SKRIPSI

**Diajukan untuk Memenuhi Sebagian Persyaratan Memperoleh Gelar
Sarjana Komputer**



Disusun Oleh:

BUNGA AMELIA RESTUPUTRI

NIM. 115060800111114

**PROGRAM STUDI INFORMATIKA/ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2015

LEMBAR PERSETUJUAN

**OPTIMASI FUNGSI KEANGGOTAAN *FUZZY* TSUKAMOTO DUA
TAHAP MENGGUNAKAN ALGORITMA GENETIKA PADA
PEMILIHAN CALON PENERIMA BEASISWA-PPA DAN BBP-PPA
(STUDI KASUS: PTIIK UNIVERSITAS BRAWIJAYA MALANG)**

SKRIPSI

Diajukan untuk Memenuhi Sebagian Persyaratan Memperoleh Gelar Sarjana
Komputer



Disusun Oleh:

BUNGA AMELIA RESTUPUTRI

NIM. 115060800111114

Skripsi ini telah disetujui oleh dosen pembimbing pada tanggal 22 Mei 2015

Pembimbing I

Pembimbing II

Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D.

NIP. 19720919 199702 1 001

Imam Cholissodin, S.Si., M.Kom.

NIK. 850719 16 1 1 0422

LEMBAR PENGESAHAN

**OPTIMASI FUNGSI KEANGGOTAAN *FUZZY* TSUKAMOTO DUA
TAHAP MENGGUNAKAN ALGORITMA GENETIKA PADA
PEMILIHAN CALON PENERIMA BEASISWA-PPA DAN BBP-PPA
(STUDI KASUS: PTIIK UNIVERSITAS BRAWIJAYA MALANG)**

SKRIPSI

LABORATORIUM KOMPUTASI DAN SISTEM CERDAS

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun Oleh :

**BUNGA AMELIA RESTUPUTRI
NIM. 115060800111114**

Skripsi ini telah diuji dan dinyatakan lulus tanggal 11 Juni 2015

Penguji I, Penguji II,

**Suprpto, S.T., M.T.
NIP. 19710727 199603 1 001**

**Edy Santoso, S.Si., M.Kom.
NIP. 19740414 200312 2 004**

Penguji III,

**Barlian Henryranu Prasetio, S.T., M.T.
NIK. 821024 06 1 1 0254**

Mengetahui,
Ketua Program Studi Informatika/Illmu Komputer

**Drs. Marji, M.T.
NIP. 19670801 199203 1 001**

PERNYATAAN ORIGINALITAS SKRIPSI

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 dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata di dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIAT, 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, 17 Juni 2015

Mahasiswa,

Bunga Amelia Restuputri
NIM. 115060800111114

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT karena hanya dengan rahmat dan hidayah-Nya, penulis dapat menyelesaikan penyusunan skripsi dengan judul Optimasi Fungsi Keanggotaan *Fuzzy* Tsukamoto Dua Tahap Menggunakan Algoritma Genetika pada Pemilihan Calon Penerima Beasiswa-PPA dan BBP-PPA (Studi Kasus PTIIK Universitas Brawijaya Malang) dengan baik.

Dalam penyusunan skripsi penulis banyak terlibat dengan berbagai pihak. Maka dari itu, penulis mengucapkan terima kasih kepada:

1. Wayan Firdaus Mahmudy, S.Si., M.T.,Ph.D., dan Imam Cholissodin, S.Si., M.Kom., selaku dosen pembimbing, yang telah memberikan waktu, tenaga, dan pikiran untuk membimbing dalam menyelesaikan penyusunan skripsi.
2. Bapak, Ibu dan seluruh keluarga atas segenap dukungan dan kasih sayang yang telah diberikan.
3. Drs. Marji, M.T. selaku ketua Program Studi Informatika.

Dengan segala kerendahan hati, dalam penyusunan skripsi ini penulis menyadari masih terdapat banyak kekurangan dan kesalahan, maka dari itu penulis mengharapkan saran dan kritik yang bersifat membangun. Akhirnya, semoga skripsi ini dapat bermanfaat dan memberikan informasi bagi kita semua.

Malang, Juni 2015

Penulis

ABSTRAK

Bunga Amelia Restuputri. 2015. Optimasi Fungsi Keanggotaan Fuzzy Tsukamoto Dua Tahap Menggunakan Algoritma Genetika Pada Pemilihan Calon Penerima Beasiswa-PPA dan BBP-PPA (Studi Kasus: PTIIK Universitas Brawijaya Malang). Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Malang. Dosen Pembimbing : Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., dan Imam Cholissodin, S.Si., M.Kom.

Beasiswa merupakan suatu bentuk bantuan keuangan yang diberikan kepada seseorang yang sedang menempuh bidang pendidikan dengan tujuan dapat meningkatkan prestasi dan meningkatkan pemerataan kesempatan belajar di perguruan tinggi. Bagi mahasiswa, beasiswa menjadi salah satu alternatif untuk menunjang bidang pendidikan. Adanya sistem pendukung keputusan sebagai penentuan dalam penerimaan beasiswa agar beasiswa yang diberikan tepat sasaran. Kenyataannya, dalam sistem pendukung keputusan dengan menggunakan konsep logika *fuzzy* terkadang masih menghasilkan hasil akhir yang belum optimal. Maka dari itu, dalam proses dari logika *fuzzy* dapat dilakukan optimasi, salah satunya optimasi fungsi keanggotaan. Metode optimasi yang digunakan adalah Algoritma Genetika. Dalam proses algoritma genetika, untuk menghasilkan solusi terbaik atau optimal terdapat proses reproduksi dan seleksi untuk setiap kromosom. Berdasarkan hasil uji coba pada penelitian, didapatkan tingkat akurasi untuk penentuan Beasiswa-PPA sebesar 98.9% dan BBP-PPA sebesar 98.7%.

Kata kunci: algoritma genetika, FIS Tsukamoto, beasiswa

ABSTRACT

Bunga Amelia Restuputri. 2015. *Optimization Membership Function of Two-Stage Representation of Fuzzy Tsukamoto Using Genetic Algorithm for the Selection of Beasiswa-PPA and BBP-PPA (Case Study: PTIIK University of Brawijaya, Malang)*. Program of Information Technology and Computer Science, University of Brawijaya, Malang. Supervisor : Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., dan Imam Cholissodin, S.Si., M.Kom.

Scholarship is a form of financial help for students who are currently going through the field of education with the purpose of improving the achievements and to evenly distribute the learning opportunities at the college. For students, scholarship is one of the alternatives to support their education. A decision support system of scholarship admission is made to ensure the scholarship goes to the right targets, but in fact, a decision support system with the concept of fuzzy logic sometimes produces a nonoptimal results. Therefore, optimizations can be implemented in a fuzzy logic, one of them is a membership function optimization. The optimization method that is used is the genetic algorithm. In the process of genetic algorithm, reproduction and selection processes must be applied to produce the best or optimal result. Based on the results test on this research, the accuracy level of the scholarship decision support system reaches 98.9% for the Beasiswa-PPA and 98.7% for the BBP-PPA.

Keywords: *genetic algorithm, FIS Tsukamoto, scholarship*

DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN	ii
PERNYATAAN ORIGINALITAS SKRIPSI	iii
KATA PENGANTAR	iv
ABSTRAK.....	v
<i>ABSTRACT</i>	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR	x
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	4
1.6 Sistematika Penulisan.....	5
BAB II KAJIAN PUSTAKA DAN DASAR TEORI	6
2.1 Beasiswa.....	6
2.1.1 Dasar-dasar Beasiswa.....	6
2.1.2 Penetapan Beasiswa PPA dan BBP-PA	7
2.2 Logika <i>Fuzzy</i>	8
2.2.1 Sejarah Logika <i>Fuzzy</i>	8
2.2.2 Definisi Logika <i>Fuzzy</i>	9
2.2.3 Himpunan <i>Fuzzy</i>	10
2.2.4 Fungsi Keanggotaan Himpunan <i>Fuzzy</i>	11
2.3 Sistem Inferensi <i>Fuzzy</i> Metode Tsukamoto	13
2.3.1 Sistem Inferensi Fuzzy Dua Tahap	14
2.4 Algoritma Genetika	15



2.4.1	Siklus Algoritma Genetika.....	16
2.4.2	Komponen-komponen Algoritma Genetika.....	17
2.5	Penelitian Sebelumnya	22
2.6	Akurasi Sistem	24
BAB III METODOLOGI PENELITIAN.....		25
3.1	Tahapan Penelitian	25
3.2	Kebutuhan Sistem.....	26
3.3	Formulasi Permasalahan.....	26
3.4	Siklus Penyelesaian Masalah Menggunakan Inferensi Fuzzy Metode Tsukamoto.....	28
3.4.1	Himpunan Fuzzy	30
3.4.2	Fuzzyfikasi	31
3.4.3	Sistem Inferensi Fuzzy.....	38
3.4.4	Defuzzifikasi	51
3.4.5	Akurasi Sistem	52
3.5	Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika.....	54
3.5.1	Representasi Kromosom dan Perhitungan Fitness.....	56
3.5.2	Inisialisasi Populasi Awal	58
3.5.3	Reproduksi	60
3.5.4	Evaluasi	63
3.5.5	Seleksi	63
BAB IV PERANCANGAN		66
4.1	Perancangan Basis Data	66
4.2	Perancangan Antarmuka.....	67
4.3	Perancangan Skenario Pengujian	69
BAB V IMPLEMENTASI SISTEM.....		72
5.1	Implementasi Basis Data	72
5.2	Implementasi Kode Program.....	74
5.2.1	Inisialisasi Populasi Awal	74
5.2.2	Reproduksi <i>Crossover</i>	75
5.2.3	Reproduksi Mutasi	76
5.2.4	Fuzzyfikasi	77



5.2.5	Sistem Inferensi <i>Fuzzy</i>	79
5.2.6	Defuzzifikasi	81
5.2.7	Akurasi Sistem	82
5.2.8	Perhitungan Fitness	83
5.2.9	Seleksi	83
5.3	Implementasi Antarmuka	84
5.3.1	Menu Algoritma Genetika.....	85
5.3.2	Menu Individu Terbaik	86
5.3.3	Menu FIS Tsukamoto.....	86
BAB VI PENGUJIAN DAN ANALISIS		88
6.1	Hasil dan Analisis Skenario Pengujian Ukuran Populasi.....	88
6.1.1	Pengujian Ukuran Populasi Beasiswa-PPA	88
6.1.2	Pengujian Ukuran Populasi BBP-PPA.....	90
6.2	Hasil dan Analisis Skenario Pengujian Kombinasi Nilai <i>Cr</i> dan <i>Mr</i>	91
6.2.1	Pengujian Kombinasi Nilai <i>Cr</i> dan <i>Mr</i> Beasiswa-PPA	92
6.2.2	Pengujian Kombinasi Nilai <i>Cr</i> dan <i>Mr</i> BBP-PPA	93
6.3	Hasil dan Analisis Skenario Pengujian Jumlah Generasi.....	95
6.3.1	Pengujian Jumlah Generasi Beasiswa-PPA	95
6.3.2	Pengujian Jumlah Generasi BBP-PPA.....	96
6.4	Hasil dan Analisis Akurasi Sistem.....	98
6.4.1	Akurasi Sistem Beasiswa-PPA	98
6.4.2	Akurasi Sistem BBP-PPA.....	100
BAB VII KESIMPULAN DAN SARAN.....		103
7.1	Kesimpulan.....	103
7.2	Saran	104
DAFTAR PUSTAKA		DP-1

DAFTAR GAMBAR

Gambar 2.1 Grafik fungsi keanggotaan himpunan representasi Linear Naik	12
Gambar 2.2 Grafik fungsi keanggotaan himpunan representasi Linear Turun ...	12
Gambar 2.3 Grafik fungsi keanggotaan himpunan representasi Keanggotaan Segitiga.....	12
Gambar 2.4 Grafik fungsi keanggotaan himpunan fuzzy representasi Keanggotaan Trapesium.....	12
Gambar 2.5 Proses Sistem Inferensi Fuzzy	13
Gambar 2.6 Siklus Algoritma Genetika oleh David Goldberg.....	16
Gambar 2.7 Perbaikan Siklus Algoritma Genetika oleh Zbigniew Michalewicz	17
Gambar 2.8 Metode one-cut-point crossover	20
Gambar 2.9 Metode Reciprocal Exchange Mutation	21
Gambar 2.10 Metode Insertion Mutation	21
Gambar 2.11 Metode Cut Point (Titik Potong)	21
Gambar 2.12 Diagram alir Algoritma Genetika	23
Gambar 3.1 Tahapan Penelitian Optimasi Fungsi Keanggotaan Fuzzy Tsukamoto menggunakan Algoritma Genetika	25
Gambar 3.2 Siklus formulasi permasalahan	28
Gambar 3.3 Diagram alir proses Sistem Inferensi Fuzzy Tsukamoto	29
Gambar 3.4 Diagram alir fuzzyfikasi	32
Gambar 3.5 Diagram alir proses sistem inferensi fuzzy Tsukamoto.....	39
Gambar 3.6 Diagram alir proses defuzzifikasi.....	52
Gambar 3.7 Diagram alir Algoritma Genetika dan Sistem Inferensi Fuzzy metode Tsukamoto.....	55
Gambar 3.8 Grafik fungsi keanggotaan kriteria input fuzzy	56
Gambar 3.9 Grafik fungsi keanggotaan output fuzzy.....	57
Gambar 3.10 Representasi kromosom real-coded optimasi fungsi keanggotaan fuzzy.....	57
Gambar 3.11 Contoh pembangkitan populasi awal.....	58
Gambar 3.12 Contoh pembangkitan populasi awal setelah diurutkan	58



Gambar 3.13 Hasil perhitungan nilai fitness pada kromsوم P ₁	58
Gambar 3.14 Diagram alir proses inialisasi populasi awal.....	59
Gambar 3.15 Diagram alir proses one-cut-point crossover.....	60
Gambar 3.16 Diagram alir proses random mutation.....	62
Gambar 3.17 Diagram alir proses elitism selection.....	64
Gambar 3.18 Fungsi keanggotaan kriteria tanggungan orang tua.....	65
Gambar 3.19 Fungsi keanggotaan penghasilan orang tua.....	65
Gambar 4.1 Perancangan basis data algenetika.....	66
Gambar 4.2 Perancangan interface input data parameter Algoritma Genetika ...	67
Gambar 4.3 Perancangan <i>interface</i> untuk halaman individu terbaik.....	68
Gambar 4.4 Perancangan halaman perhitungan fuzzy Tsukamoto.....	69
Gambar 5.1 Kode program proses inialisasi populasi awal.....	75
Gambar 5.2 Kode program proses reproduksi one-cut point crossover.....	76
Gambar 5.3 Kode program proses reproduksi random mutation.....	77
Gambar 5.4 Data mahasiswa yang merupakan nilai crisp.....	77
Gambar 5.5 Kode program proses fuzzyfikasi.....	78
Gambar 5.6 Inferensi fuzzy untuk kriteria positif.....	79
Gambar 5.7 Inferensi fuzzy untuk kriteria negatif.....	80
Gambar 5.8 Kode program inferensi fuzzy Tsukamoto tahap kedua.....	81
Gambar 5.9 Kode program proses defuzzifikasi.....	82
Gambar 5.10 Kode program proses perhitungan akurasi sistem.....	83
Gambar 5.11 Kode program proses perhitungan fitness.....	83
Gambar 5.12 Kode program proses elitism selection.....	84
Gambar 5.13 Halaman utama menu Algoritma Genetika.....	85
Gambar 5.14 Proses perhitungan Algoritma Genetika.....	86
Gambar 5.15 Halaman menu Individu Terbaik.....	86
Gambar 5.16 Implementasi antarmuka menu FIS Tsukamoto.....	87

Gambar 6.1 Grafik hasil rata-rata nilai fitness ukuran populasi..... 90

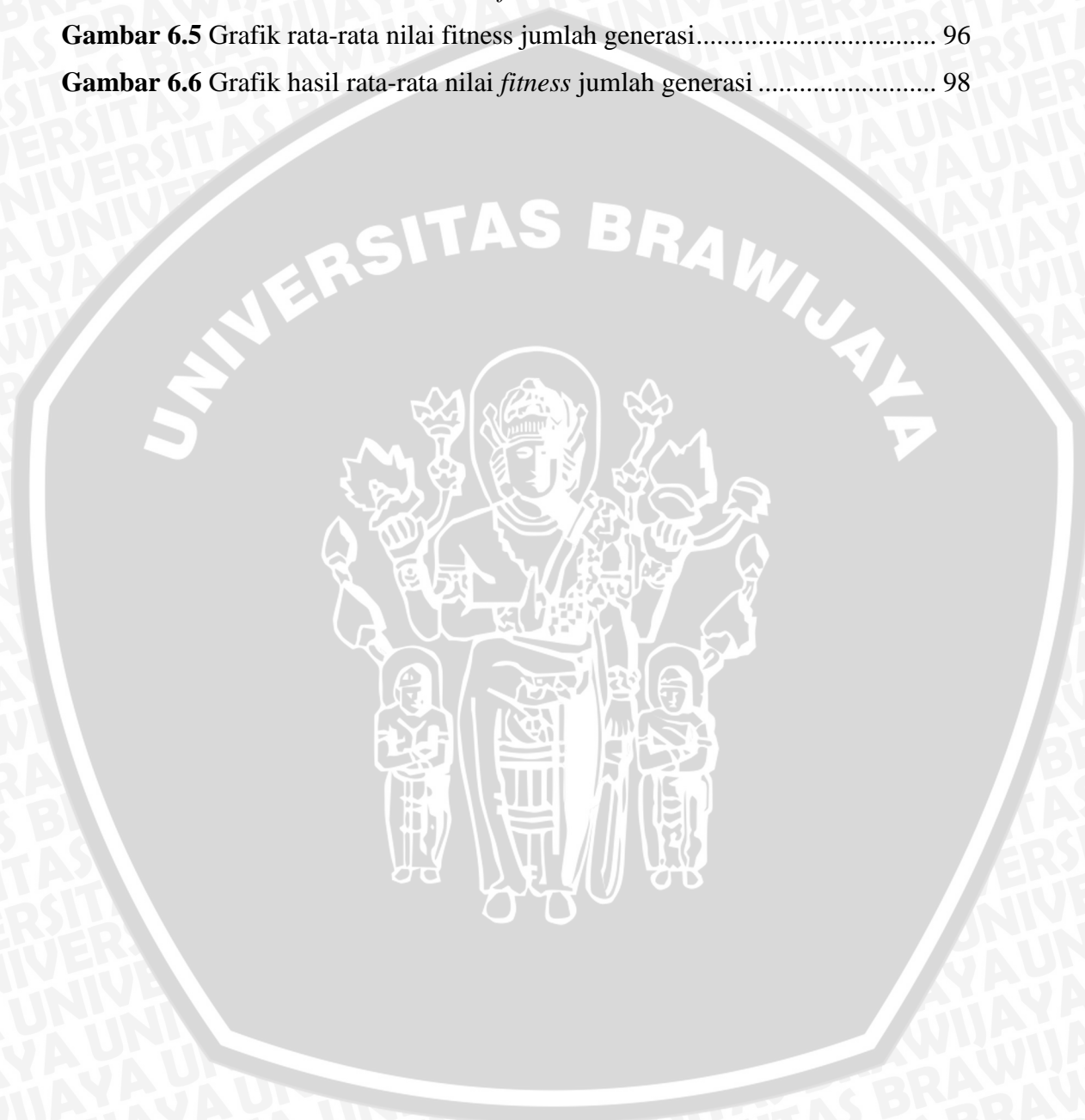
Gambar 6.2 Grafik hasil rata-rata nilai *fitness* ukuran populasi..... 91

Gambar 6.3 Grafik hasil rata-rata fitness nilai *cr* dan *mr* 93

Gambar 6.4 Grafik hasil rata-rata nilai *fitness* nilai *cr* dan *mr* 94

Gambar 6.5 Grafik rata-rata nilai fitness jumlah generasi..... 96

Gambar 6.6 Grafik hasil rata-rata nilai *fitness* jumlah generasi 98



DAFTAR TABEL

Tabel 3.1 Rentang nilai kriteria penentuan beasiswa 27

Tabel 3.2 Contoh permasalahan pada inferensi fuzzy Tsukamoto 30

Tabel 3.3 Himpunan fuzzy variabel input 31

Tabel 3.4 Kriteria positif dan negatif untuk penentuan Beasiswa 39

Tabel 3.5 Basis aturan inferensi fuzzy Tsukamoto kriteria positif 40

Tabel 3.6 Perhitungan inferensi fuzzy Tsukamoto kriteria positif 41

Tabel 3.7 Basis aturan inferensi fuzzy Tsukamoto kriteria negatif 43

Tabel 3.8 Perhitungan inferensi fuzzy Tsukamoto kriteria negatif 46

Tabel 3.9 Basis aturan inferensi fuzzy tahap kedua 50

Tabel 3.10 Proses perhitungan inferensi fuzzy Tsukamoto tahap 2 50

Tabel 3.11 Contoh kasus data 5 mahasiswa calon penerima beasiswa 53

Tabel 3.12 Hasil perhitungan nilai Z menggunakan FIS Tsukamoto 53

Tabel 3.13 Contoh permasalahan perhitungan Algoritma Genetika 55

Tabel 3.14 Pembangkitan populasi awal 60

Tabel 3.15 Hasil perhitungan nilai fitness pada tahap evaluasi 63

Tabel 4.1 Rancangan skenario pengujian ukuran populasi 70

Tabel 4.2 Rancangan skenario pengujian kombinasi nilai cr dan mr 70

Tabel 4.3 Rancangan skenario jumlah generasi 71

Tabel 5.1 Struktur Tabel data_mhs 72

Tabel 5.2 Struktur tabel ind_baik dan history 72

Tabel 6.1 Hasil skenario pengujian ukuran populasi penentuan Beasiswa-PPA . 88

Tabel 6.2 Hasil skenario pengujian ukuran populasi penentuan BBP-PPA 90

Tabel 6.3 Hasil pengujian kombinasi nilai cr dan mr penentuan Beasiswa-PPA 92

Tabel 6.4 Hasil pengujian nilai cr dan mr penentuan BBP-PPA 93

Tabel 6.5 Hasil pengujian jumlah generasi penentuan Beasiswa-PPA 95

Tabel 6.6 Hasil pengujian jumlah generasi penentuan BBP-PPA 97

Tabel 6.7 Hasil akurasi sistem untuk Beasiswa-PPA 98

Tabel 6.8 Hasil ranking penentuan Beasiswa-PPA	99
Tabel 6.9 Hasil akurasi sistem untuk BBP-PPA	100
Tabel 6.10 Hasil ranking penentuan BBP-PPA.....	101



BAB I PENDAHULUAN

1.1 Latar Belakang

Beasiswa merupakan suatu bentuk bantuan keuangan yang diberikan kepada seseorang yang sedang menempuh bidang pendidikan dengan tujuan dapat meningkatkan prestasi dan meningkatkan pemerataan kesempatan belajar di perguruan tinggi. Bagi mahasiswa, beasiswa menjadi salah satu alternatif untuk menunjang bidang pendidikan dimana zaman sekarang biaya pendidikan di universitas negeri semakin tinggi, yaitu dengan adanya Uang Kuliah Tunggal (UKT). Beasiswa juga banyak dicari oleh mahasiswa kurang mampu dan memiliki prestasi yang baik dalam pendidikannya. Salah satu beasiswa yang ditawarkan oleh Direktorat Jenderal Pendidikan Tinggi (DIKTI) yaitu Beasiswa Peningkatan Prestasi Akademik (Beasiswa-PPA) dan Bantuan Biaya Pendidikan Peningkatan Prestasi Akademik (BBP-PPA).

Beasiswa-PPA adalah bantuan biaya pendidikan yang diberikan oleh pemerintah untuk mahasiswa yang memiliki prestasi baik dalam akademik (Sailah, 2014). Sedangkan yang dimaksud dengan BBP-PPA adalah bantuan biaya pendidikan yang diberikan oleh pemerintah kepada mahasiswa yang memiliki keterbatasan ekonomi dalam membiayai pendidikan di perguruan tinggi (Sailah, 2014). Akan tetapi, dalam pemberian bantuan biaya pendidikan ini terkadang terdapat masalah salah satunya yaitu ketidaktepatan sasaran beasiswa yang diberikan kepada mahasiswa yang seharusnya layak mendapatkan. Banyak faktor yang dapat mengakibatkan masalah ini sering muncul di banyak universitas negeri, salah satunya penentuan beasiswa yang masih berjalan manual sehingga beasiswa yang diberikan tidak tepat sasaran karena adanya unsur subyektifitas dan ketidakkonsistenan dari tim penilai. Selain itu, banyaknya data dan proses untuk penentuan beasiswa dapat menyebabkan lamanya hasil keputusan penentuan beasiswa.

Solusi yang ditawarkan untuk permasalahan ini adalah suatu sistem pendukung keputusan dari pemilihan calon penerima beasiswa. Sistem ini akan

melakukan perhitungan dengan kriteria yang biasanya digunakan untuk seleksi calon penerima beasiswa, seperti Indeks Prestasi Kumulatif (IPK), penghasilan orang tua / wali, dan jumlah tanggungan orang tua / wali. Kriteria yang digunakan bisa saja bertambah agar perhitungan yang dilakukan lebih tepat sasaran. Dengan menggunakan sistem, pemberian beasiswa kepada mahasiswa yang berhak mendapatkannya dapat diberikan tepat sasaran.

Banyak penelitian yang berkaitan dengan pendukung keputusan pemilihan beasiswa, salah satunya adalah Iskandar (2012) melakukan penelitian mengenai sistem pendukung keputusan untuk menyeleksi calon penerima Beasiswa PPA dan BBM dengan menggunakan metode *Fuzzy* AHP. Pada penelitian ini kriteria yang digunakan untuk pendukung keputusan antara lain, nilai Indeks Prestasi Kumulatif (IPK), penghasilan orang tua, piagam penghargaan, tagihan listrik, tagihan telepon, tagihan PDAM, pembayaran PBB, dan tanggungan orang tua. Hasil akhir dari penelitian ini memiliki keakuratan dengan hasil kemahasiswaan yaitu Beasiswa PPA sebesar 80% dan Beasiswa BBM sebesar 33,33%.

Konsep logika *fuzzy* merupakan metode yang sesuai untuk menyelesaikan permasalahan, dikarenakan kriteria atau parameter penentuan pendukung keputusan tersebut bersifat tidak pasti (*fuzzy*). Logika *fuzzy* yang akan digunakan menggunakan inferensi *fuzzy* model Tsukamoto. Tahapan dalam metode *fuzzy* Tsukamoto adalah yang pertama menentukan fungsi keanggotaan, kemudian menentukan *rules* dari kriteria atau parameter yang digunakan, dan yang terakhir dilakukan perhitungan berupa hasil keputusan pemilihan calon penerima beasiswa. Inferensi *fuzzy* model Tsukamoto ini nantinya akan dilakukan perhitungan secara dua tahap (Fattouh, 2012). Alasan dari penggunaan inferensi *fuzzy* dua tahap adalah dikarenakan jumlah kriteria penentuan yang akan digunakan banyak sehingga akan menghasilkan *rules* yang banyak juga.

Salah satu tahapan metode *fuzzy* Tsukamoto yaitu menentukan fungsi keanggotaan, dalam tahapan ini dapat dikembangkan dengan menggunakan metode optimasi, yaitu Algoritma Genetika. Algoritma Genetika adalah algoritma yang menerapkan mekanisme dari *natural selection* dan *natural genetics* digunakan untuk memecahkan suatu masalah optimasi kombinasi (Soetanto, 1999). Kelebihan metode Algoritma Genetika dibandingkan dengan metode

optimasi lainnya adalah algoritma genetika dapat memecahkan suatu masalah yang kompleks dan memiliki ruang pencarian (*search space*) yang luas (Gen & Cheng dalam Mahmudy, 2013). Selain itu, algoritma genetika dapat memecahkan masalah optimasi dalam bidang *computer science* dengan tingkat kesuksesan yang tinggi (Anggariawan, 2014).

Pada penelitian ini, algoritma genetika digunakan untuk mendapatkan fungsi keanggotaan yang lebih sesuai dan menghasilkan keputusan yang lebih akurat. Salah satu penelitian yang terkait dengan metode algoritma genetika adalah mengenai optimasi fungsi keanggotaan yang dilakukan oleh Anggariawan (2014). Algoritma genetika diterapkan pada penelitian perhitungan Angka Metabolisme Basal (AMB) dengan mengoptimasi fungsi keanggotaan *Fuzzy Inference System* (FIS) model Sugeno. Dalam penelitian tersebut hasil yang didapat untuk perhitungan Angka Metabolisme Basal (AMB) terhadap data uji sebanyak 100 dengan menghasilkan nilai akurasi sebesar 94% lebih baik dari penelitian sebelumnya yang hanya menggunakan metode FIS model Sugeno tanpa optimasi fungsi keanggotaan memiliki nilai akurasi sebesar 75%.

Diharapkan dengan adanya penelitian ini, penentuan pemilihan calon penerima beasiswa menggunakan metode *fuzzy* Tsukamoto dengan melakukan optimasi pada fungsi keanggotaan menggunakan metode Algoritma Genetika dapat menyelesaikan masalah, yaitu hasil pemilihan calon penerima beasiswa lebih akurat dari sebelumnya. Oleh karena itu, penulis akan melakukan penelitian dengan judul “Optimasi Fungsi Keanggotaan *Fuzzy* Tsukamoto Dua Tahap Menggunakan Algoritma Genetika pada Pemilihan Calon Penerima Beasiswa-PPA dan BBP-PPA (Studi Kasus PTIIK Universitas Brawijaya Malang)”.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang terdapat dalam latar belakang, maka terdapat beberapa rumusan masalah, yaitu:

1. Bagaimana mengimplementasi Algoritma Genetika untuk optimasi fungsi keanggotaan *Fuzzy* Tsukamoto pada Pemilihan Calon Penerima Beasiswa-PPA dan BBP-PPA.

2. Bagaimana menentukan parameter Algoritma Genetika yang tepat pada optimasi fungsi keanggotaan *fuzzy* Tsukamoto.
3. Bagaimana tingkat akurasi sistem inferensi *fuzzy* Tsukamoto yang telah dioptimasi menggunakan Algoritma Genetika.

1.3 Batasan Masalah

Batasan masalah yang terdapat dalam penelitian ini, adalah:

1. Data mahasiswa calon penerima Beasiswa dan BBP PPA sebanyak 30 mahasiswa PTIIK UB.
2. Data parameter penentuan pemilihan calon penerima beasiswa, seperti IPK, penghasilan orang tua, piagam penghargaan, tagihan rekening listrik, tagihan telepon, tagihan PDAM, pembayaran PBB, dan tanggungan orang tua/wali.
3. Pengolahan data menggunakan model Logika *Fuzzy* Tsukamoto dua tahap dan metode optimasi keanggotaan menggunakan Algoritma Genetika.

1.4 Tujuan

Adapun tujuan yang diharapkan oleh penulis dalam penelitian, antara lain:

1. Mengimplementasikan Algoritma Genetika untuk optimasi fungsi keanggotaan *Fuzzy* Tsukamoto pada Pemilihan Calon Penerima Beasiswa-PPA dan BBP-PPA.
2. Menentukan parameter Algoritma Genetika yang tepat pada optimasi fungsi keanggotaan *fuzzy* Tsukamoto
3. Mengetahui tingkat akurasi sistem inferensi *fuzzy* Tsukamoto yang telah dioptimasi menggunakan Algoritma Genetika.

1.5 Manfaat

1. Mempermudah dalam pengambilan keputusan pemilihan calon penerima Beasiswa-PPA dan BBP-PPA pada PTIIK Universitas Brawijaya Malang.
2. Mendapatkan hasil keputusan pemilihan calon penerima Beasiswa-PPA dan BBP-PPA lebih optimal dari penelitian sebelumnya.

3. Mendapatkan hasil keputusan penentuan beasiswa Beasiswa-PPA dan BBP-PPA lebih tepat sasaran.
4. Menjadi lebih efektif dan efisien dalam perhitungan pengambilan keputusan pemilihan calon penerima Beasiswa-PPA dan BBP-PPA pada PTIIK Universitas Brawijaya Malang.

1.6 Sistematika Penulisan

Sistematika penulisan pada penyusunan skripsi ini terdapat 7 bab dan dalam tiap bab akan diberikan penjelasan mengenai penelitian. Berikut sistematika penulisan, antara lain:

a. BAB I : PENDAHULUAN

Bab ini membahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, dan manfaat dari penelitian

b. BAB II : KAJIAN PUSTAKA DAN DASAR TEORI

Bab ini membahas mengenai dasar teori dan juga referensi beasiswa, algoritma genetika, dan logika *fuzzy*

c. BAB III : METODE PENELITIAN

Bab ini menguraikan tentang metode dan langkah kerja yang digunakan dalam penelitian. Selain itu, disertai dengan perhitungan manual dari algoritma yang digunakan.

d. BAB IV : PERANCANGAN

Bab ini menguraikan analisa kebutuhan dalam penelitian dan perancangan yang akan digunakan dalam penelitian.

e. BAB V : IMPLEMENTASI

Bab ini menjelaskan tentang implementasi dari metode yang digunakan sesuai masalah dengan masalah yang dibahas.

f. BAB VI : PENGUJIAN DAN ANALISIS

Bab ini menjelaskan tentang hasil pengujian terhadap metode yang digunakan dan dilakukan analisis data sesuai data yang diperoleh.

g. BAB VII : PENUTUP

Bab ini menjelaskan kesimpulan akhir dari penelitian dan terdapat saran-saran untuk pengembangan penelitian.

BAB II

KAJIAN PUSTAKA DAN DASAR TEORI

2.1 Beasiswa

Beasiswa merupakan suatu bentuk bantuan keuangan yang diberikan kepada seseorang yang sedang menempuh bidang pendidikan. Di Indonesia beasiswa sudah sangat banyak diselenggarakan oleh Direktorat Jenderal Pendidikan Tinggi (DIKTI) maupun sebuah perusahaan-perusahaan yang peduli dengan dunia pendidikan. Dalam penyelenggaraannya, beasiswa banyak ditawarkan kepada perguruan tinggi. Salah satu program beasiswa yang ditawarkan khusus untuk mahasiswa perguruan tinggi oleh Direktorat Jenderal Pendidikan Tinggi (DIKTI), yaitu Beasiswa Peningkatan Prestasi Akademik (Beasiswa-PPA) dan Bantuan Biaya Pendidikan Peningkatan Prestasi Akademik (BBP-PPA). Dalam penjelasannya yang dimaksud dengan Beasiswa-PPA adalah bantuan biaya pendidikan yang diberikan oleh pemerintah untuk mahasiswa yang memiliki prestasi baik dalam akademik. Sedangkan yang dimaksud dengan BBP-PPA adalah bantuan biaya pendidikan yang diberikan oleh pemerintah kepada mahasiswa yang memiliki keterbatasan ekonomi dalam membiayai biaya pendidikan di perguruan tinggi (Sailah, 2014). Adapun tujuan dari beasiswa yang diberikan oleh mahasiswa, antara lain:

1. Dengan adanya beasiswa dapat meningkatkan prestasi mahasiswa penerima dari bidang akademik maupun non-akademik.
2. Mengurangi jumlah mahasiswa yang tidak melanjutkan ke jenjang perkuliahan dikarenakan ketidakmampuan ekonomi keluarga dalam membiayai pendidikan perguruan tinggi.
3. Mendapatkan kesempatan berkuliah di perguruan tinggi dan meningkatkan pemerataan kesempatan berkuliah bagi mahasiswa.

2.1.1 Dasar-dasar Beasiswa

Dasar-dasar yang mengatur adanya beasiswa salah satunya pada Undang-Undang Dasar Negara Republik Indonesia 1945 (UUD NRI 1945) Pasal 31 ayat (1) yang berbunyi “Setiap warga negara berhak mendapatkan

pengajaran”. Oleh karena itu, pemerintah dan pemerintah daerah wajib mengamalkan pasal tersebut dimana dengan memberikan layanan pendidikan yang layak dan mudah bagi setiap warga negara yang membutuhkan dalam bidang pendidikan (Sailah, 2014). Dasar-dasar beasiswa yang lain terdapat pada peraturan pemerintah dan undang-undang, antara lain:

1. Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional. Pada bab V Pasal 12 ayat (1c) dijelaskan bahwa “Setiap peserta didik pada setiap pendidikan berhak mendapatkan beasiswa bagi yang berprestasi yang orang tuanya tidak mampu membiayai pendidikannya”.
2. Undang-Undang Republik Indonesia Nomor 12 Tahun 2012 tentang Pendidikan Tinggi dimana dijelaskan pada Pasal 76 ayat (2a) bahwa “Pemerintah harus memenuhi hak mahasiswa dalam bentuk memberikan beasiswa kepada Mahasiswa berprestasi”.
3. Peraturan Pemerintah Nomor 48 Tahun 2008 tentang Pendanaan Pendidikan, Bagian Kelima, Pasal 27 ayat (1) bahwa “Pemerintah dan pemerintah daerah sesuai kewenangannya memberi bantuan pendidikan atau beasiswa kepada peserta didik yang orang tua atau walinya tidak mampu membiayai pendidikannya”. Dan juga Pasal 27 ayat (2) dijelaskan “Pemerintah dan pemerintah daerah sesuai dengan kewenangannya dapat memberikan beasiswa kepada peserta didik yang berprestasi”.
4. Peraturan Pemerintah Nomor 66 Tahun 2010 tentang perubahan atas Peraturan Pemerintah Nomor 17 Tahun 2010 tentang Pengelolaan dan Penyelenggaraan Pendidikan.
5. Peraturan Menteri Nomor 30 Tahun 2010 tentang pemberian bantuan biaya pendidikan kepada peserta didik dimana orang tua atau walinya tidak mampu membiayai pendidikan.

2.1.2 Penetapan Beasiswa PPA dan BBP-PA

Syarat penetapan Beasiswa PPA dan BPP-PA menurut Direktorat Jenderal Pendidikan Tinggi (DIKTI) diberikan kepada mahasiswa pada jenjang

S1 / Diploma IV untuk semester 2 sampai 6. Pada jenjang Diploma III untuk semester 2 sampai semester 5 (Sailah, 2014). Syarat lainnya antara lain untuk mahasiswa calon penerima:

1. Beasiswa PPA; mahasiswa yang ingin mendapatkan beasiswa ini maka harus memiliki Indeks Prestasi Kumulatif (IPK) paling rendah 3.00.
2. BBP-PPA; mahasiswa dengan Indeks Prestasi Kumulatif (IPK) minimal sebesar 2.75 dan melampirkan surat keterangan penghasilan orang tua atau surat pernyataan penghasilan orang tua bermaterai.

2.2 Logika *Fuzzy*

2.2.1 Sejarah Logika *Fuzzy*

Konsep himpunan *fuzzy* pertama kali ditemukan oleh seorang bernama Professor Zadeh (California Univ.) pada tahun 1965. Professor Zadeh menemukan ide mengenai perhitungan yang bersifat matematis yang menjabarkan tentang ketidakjelasan atau kesamaran dalam bentuk variabel linguistik yang disebut himpunan *fuzzy* (samar). Menurut Zadeh, terdapat suatu teori himpunan yang bersifat tidak matematis (kualitatif), sehingga muncullah himpunan *fuzzy* dimana menggunakan teori himpunan yang menggabungkan pendekatan kualitatif dan kuantitatif (Robandi, 2006).

Zadeh mengembangkan penemuannya himpunan *fuzzy* menjadi logika *fuzzy* (*fuzzy logic*) dimana teori ini sudah menyebar di Amerika, Eropa, dan Jepang. Pada perkembangan di Amerika, terdapat beberapa peneliti yaitu Mohamed El Hawary (Dalhouse Univ.), Malik (Calgary Univ.), dan El Sharkawi (Washington Univ.) yang mengembangkan logika *fuzzy* menjadi *fuzzy logic control* (FLC). Selanjutnya, pada perkembangan di Eropa logika *fuzzy* dikembangkan oleh Professor Mamdani dan Miranda. Perkembangan logika *fuzzy* di Jepang terjadi sekitar pertengahan tahun 1970, logika *fuzzy* dikembangkan oleh beberapa peneliti Jepang seperti Sugeno dan Mizumoto (Tokyo Univ.), Tanaka (Osaka Prefecture Univ.), Hiyama (Kumamoto Univ.), Yamakawa (Kyushu Institute of Tech.), dan Nishimori Katsumi (Tottori Univ.). Para peneliti tersebut telah mengembangkan logika *fuzzy* yang dapat diimplementasikan pada bidang *electrical engineering* (Robandi, 2006).

Dari prinsip yang digunakan oleh teori himpunan *fuzzy*, himpunan *fuzzy* memiliki anggota himpunan yang tetap yaitu “ya” (anggota himpunan) atau “tidak” (bukan anggota himpunan). Pada himpunan *fuzzy* memiliki batasan pada setiap elemen dari masing-masing fungsi keanggotaan himpunan *fuzzy*, yaitu pada nilai antara $[0, 1]$ (Zadeh, 1965).

2.2.2 Definisi Logika Fuzzy

Logika *fuzzy* adalah suatu metode yang dapat merepresentasikan suatu variabel bersifat samar (*fuzzy*) dengan menggunakan basis aturan atau persamaan *if-then*. Pada logika *fuzzy*, terdapat istilah memasukkan sebuah *input* ke *output* maksudnya adalah di dalam *input* dan *output* terdapat fungsi fungsi keanggotaan atau nama lainnya adalah *membership function*. Fungsi keanggotaan pada logika *fuzzy* terbentuk dari aturan *fuzzy* atau persamaan *if-then*. Jadi, cara kerja logika *fuzzy* adalah dengan menggabungkan himpunan *fuzzy* dan aturan *fuzzy* maka akan mendapatkan sebuah fungsi keanggotaan dimana fungsi tersebut dapat menyelesaikan suatu masalah (Sutojo, 2011). Secara umum, sistem logika *fuzzy* memiliki 4 elemen, yaitu:

1. *Rule-based* atau basis aturan yang merupakan aturan-aturan untuk pengambilan keputusan, biasanya aturan ini berasal dari pakar.
2. Suatu mekanisme pengambilan keputusan dimana pakar mengambil keputusan dengan menerapkan pengetahuan yang dimiliki.
3. Proses fuzzyfikasi (*fuzzyfication*) merupakan perhitungan nilai *crisp* dan menghasilkan suatu derajat keanggotaan.
4. Proses defuzzifikasi (*defuzzification*), merupakan proses perhitungan derajat keanggotaan mencari nilai *output fuzzy* melalui inferensi atau aturan-aturan berupa nilai *crisp*.

Perhitungan logika *fuzzy* dapat diaplikasikan pada berbagai bidang seperti teori manajemen, akuntansi, grafik, dan lain sebagainya yang berhubungan dengan ketidakpastian dapat diselesaikan dengan menggunakan logika *fuzzy*. Selain itu, pada perhitungan *Neural Network* (NN), *Genetic Algorithm* (GA), dan pemetaan *Artificial Immune System* (AIS) dapat

menggunakan prinsip logika *fuzzy*. Adapun alasan dari penggunaan logika *fuzzy* (Kusumadewi, 2010) yang telah banyak diaplikasikan dalam berbagai bidang, antara lain:

1. Konsep logika *fuzzy* yang matematis sangat sederhana dan mudah dimengerti.
2. Logika *fuzzy* dapat digabungkan dengan teknologi lain, seperti GA, NN, AIS, kontrol optimal, dan lain-lain.
3. Logika *fuzzy* dapat menjadikan fungsi kontinyu atau *nonlinear* menjadi sistem diskrit.
4. Logika *fuzzy* sangat fleksibel.
5. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat.

2.2.3 Himpunan Fuzzy

Himpunan *fuzzy* merupakan himpunan yang merepresentasikan derajat keanggotaan *fuzzy* (Zadeh, 1965). Setiap himpunan *fuzzy* memiliki fungsi keanggotaan (*membership function*) dengan nilai interval antara [0, 1]. Sebuah himpunan *fuzzy* A dapat didefinisikan sebagai berikut, misalkan terdapat himpunan semesta X maka dapat ditulis dalam fungsi matematis pada persamaan 2-1.

$$\mu_A(x) = \begin{cases} 1, & x \in X \\ 0, & x \notin X \end{cases} \quad (2-1)$$

dimana x adalah anggota himpunan X dan nilai $\mu_A(x)$ antara [0, 1] yang merupakan nilai dari derajat keanggotaan dari masing-masing fungsi keanggotaan (Zadeh, 1965). Dalam himpunan *fuzzy* memiliki operasi aljabar seperti himpunan *crisp*, operator yang digunakan adalah logika AND atau operasi irisan untuk operator minimum dan logika OR atau operasi gabungan untuk operator maksimum. Selain itu, operasi aljabar pada himpunan *fuzzy* lainnya adalah kesamaan dan komplemen. Misalkan terdapat dua buah himpunan *fuzzy* A dan B dalam himpunan semesta X dengan fungsi keanggotaannya μ_A dan μ_B , sehingga operasi himpunan *fuzzy* dapat dinyatakan dalam persamaan 2-2 sampai dengan 2-5.

1. Kesamaan

$$\mu_A(x) = \mu_B(x), \text{ untuk semua } x \in X \quad (2-2)$$

2. Irisan

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), \text{ untuk semua } x \in X \quad (2-3)$$

3. Gabungan

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), \text{ untuk semua } x \in X \quad (2-4)$$

4. Komplemen

$$\mu_A(x) = 1 - \mu_A(x) \quad (2-5)$$

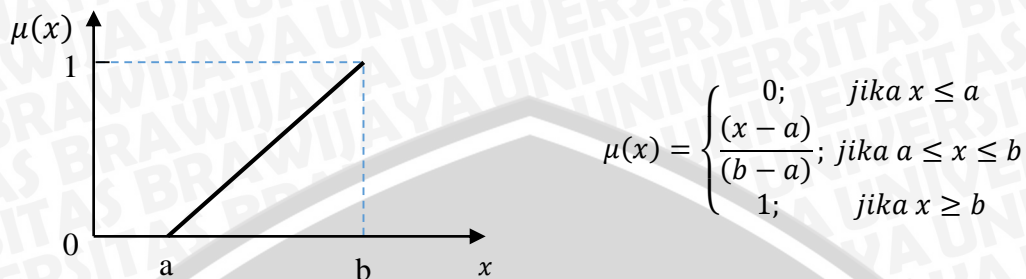
Terdapat dua nilai atau atribut yang dimiliki oleh himpunan *fuzzy* yaitu, nilai linguistik dan numerik. Nilai linguistik pada himpunan *fuzzy* adalah nilai yang merepresentasikan penamaan dari variabel *fuzzy* menggunakan sebuah kata, seperti tinggi, sedang, rendah. Sedangkan numerik merupakan nilai dari variabel *fuzzy* berupa angka, seperti 10, 20, 25, dan lain sebagainya.

2.2.4 Fungsi Keanggotaan Himpunan Fuzzy

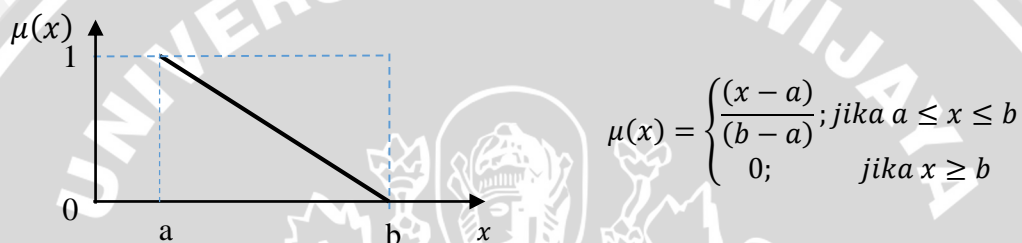
Pada perhitungan logika *fuzzy* terdapat fungsi keanggotaan (*membership function*). Fungsi keanggotaan pada logika *fuzzy* terbentuk dari aturan *fuzzy* atau persamaan *if-then*. Fungsi keanggotaan memiliki dua definisi untuk himpunan *fuzzy* yaitu, numerik dan fungsional. Fungsi keanggotaan numerik adalah fungsi keanggotaan yang memiliki derajat keanggotaan dari himpunan *fuzzy* dinyatakan dengan menggunakan vektor bilangan (Suratno, 2011). Sedangkan fungsi keanggotaan fungsional adalah fungsi keanggotaan yang bertujuan untuk mengetahui derajat keanggotaan dengan menggunakan persamaan matematis dan perhitungan (Suratno, 2011). Fungsi keanggotaan fungsional dinyatakan dalam bentuk grafik atau kurva yang terdapat titik-titik dari batas nilai *input* data dalam suatu derajat keanggotaan dengan rentang nilai antara [0, 1] (Widhiastiwi, 2007).

Fungsi keanggotaan himpunan *fuzzy* secara fungsional memiliki beberapa bentuk fungsi keanggotaan, antara lain fungsi keanggotaan linear,

fungsi keanggotaan segitiga, dan fungsi keanggotaan trapesium seperti yang ditunjukkan pada Gambar 2.1 sampai dengan 2.4.



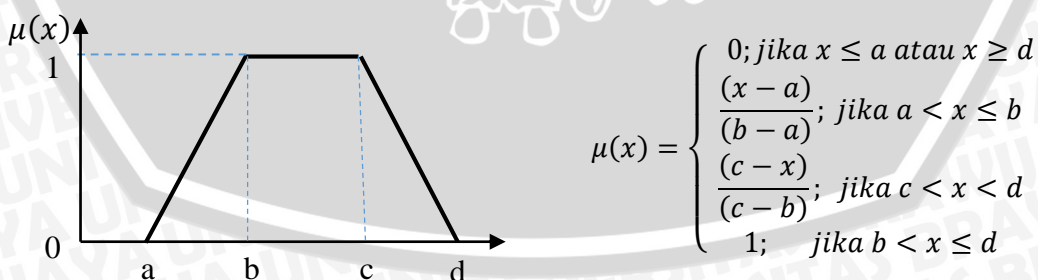
Gambar 2.1 Grafik fungsi keanggotaan himpunan representasi Linear Naik



Gambar 2.2 Grafik fungsi keanggotaan himpunan representasi Linear Turun



Gambar 2.3 Grafik fungsi keanggotaan himpunan representasi Keanggotaan Segitiga



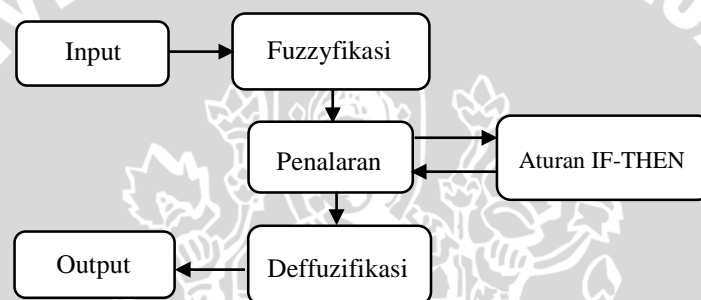
Gambar 2.4 Grafik fungsi keanggotaan himpunan fuzzy representasi Keanggotaan Trapesium

Sumber: Suratno, 2011

Variabel a , b , c , dan d merupakan batas-batas dari himpunan *fuzzy*. Batas-batas tersebut akan ditentukan dengan menggunakan metode algoritma genetika agar menghasilkan hasil akhir yang lebih optimal.

2.3 Sistem Inferensi *Fuzzy* Metode Tsukamoto

Sistem Inferensi *Fuzzy* (*Fuzzy Inference System / FIS*) merupakan suatu sistem yang melakukan perhitungan berdasarkan pada konsep teori himpunan *fuzzy*, aturan *fuzzy*, dan konsep logika *fuzzy* (Kusumadewi, 2003). Dalam pengaplikasian FIS sudah cukup banyak digunakan dalam berbagai bidang. Adapun proses sistem inferensi *fuzzy* ditunjukkan pada Gambar 2.5.



Gambar 2.5 Proses Sistem Inferensi *Fuzzy*
Sumber: Kusumadewi, 2003

Dalam sistem inferensi *fuzzy* terdapat *input fuzzy* berupa nilai *crisp*. Kemudian, nilai *crisp* akan dihitung berdasarkan aturan-aturan yang telah dibuat menghasilkan besaran *fuzzy* disebut proses fuzzyfikasi. Proses selanjutnya, defuzzifikasi yaitu proses dimana merubah besaran *fuzzy* menjadi nilai *crisp* sebagai *output* dari sistem.

Sistem Inferensi *Fuzzy* memiliki beberapa metode yang diterapkan, antara lain metode Tsukamoto, metode Mamdani, dan metode Sugeno. Metode *fuzzy* Tsukamoto menggunakan *rules based* atau basis aturan dalam bentuk “sebab-akibat” atau “*if-then*”. Cara perhitungan pada metode *fuzzy* Tsukamoto adalah langkah pertama membuat suatu aturan *fuzzy* yang dibentuk mewakili himpunan *fuzzy*. Selanjutnya, dihitung derajat keanggotaan sesuai dengan aturan yang telah dibuat. Setelah diketahui nilai derajat keanggotaan dari masing-masing aturan *fuzzy*, langkah selanjutnya mencari nilai *alpha* predikat (α) dengan cara menggunakan operasi himpunan *fuzzy*. Langkah terakhir adalah proses

defuzzifikasi dimana mencari nilai *output* berupa nilai *crisp* (z) dan dinyatakan dalam persamaan 2-6.

$$Z = \frac{\sum \alpha_{p_i} z_i}{\sum \alpha_{p_i}} \quad (2-6)$$

Dimana α_p = nilai *alpha* predikat (nilai minimal dari derajat keanggotaan), i = jumlah data pada aturan *fuzzy*, Z_i = nilai *crisp* yang didapat dari rumus derajat keanggotaan himpunan *fuzzy*, dan Z = defuzzifikasi rata-rata terpusat (*Center Average Defuzzifier*).

2.3.1 Sistem Inferensi Fuzzy Dua Tahap

Secara umum sistem inferensi *fuzzy* meliputi 3 tahap, yaitu fuzzyfikasi, inferensi, dan defuzzifikasi. Proses dalam masing-masing tahapan sistem inferensi *fuzzy* membutuhkan waktu sendiri sehingga menjadi tidak efisien. Solusi dari permasalahan tersebut adalah dengan menggunakan Sistem Inferensi *Fuzzy* Dua Tahap (Fattouh, 2012). Dalam penelitian Fattouh (2012) menyatakan bahwa penggunaan inferensi *fuzzy* dua tahap dapat mengurangi waktu operasi. Pada sistem inferensi *fuzzy* dua tahap, tahapan defuzzifikasi akan digunakan sebagai nilai input untuk tahapan fuzzyfikasi selanjutnya.

Langkah-langkah dari sistem inferensi *fuzzy* dua tahap, yaitu:

1. Membuat sebuah himpunan yang berisi kriteria atau nilai *input* dari sistem *fuzzy*. Terdapat dua buah himpunan dimana kriteria untuk masing-masing himpunan berbeda sifat. Dimisalkan pada permasalahan optimasi fungsi keanggotaan, terdapat dua himpunan kriteria, kriteria positif dan kriteria negatif. Kriteria positif meliputi IPK, tanggungan orang tua, dan piagam. Sedangkan, kriteria negatif meliputi penghasilan orang tua, tagihan telepon, listrik, PDAM, dan PBB.
2. Hasil perhitungan defuzzifikasi (nilai Z) dari kriteria positif dan negatif akan digunakan sebagai nilai *input* untuk proses fuzzyfikasi pada sistem inferensi *fuzzy* selanjutnya.
3. Hitung nilai bobot dari sistem inferensi *fuzzy* menggunakan persamaan 2-6.

2.4 Algoritma Genetika

Dalam bidang kecerdasan buatan terdapat suatu teknik pemecahan masalah dimana konsep metode yang digunakan mengadaptasi proses evolusi seleksi alam makhluk hidup dan genetik. Teknik pemecahan masalah ini biasa disebut dengan Algoritma Genetika (*Genetic Algorithms*), algoritma genetika dapat memecahkan masalah dengan cara mendapatkan solusi optimal. Sebagai contoh masalah yang memerlukan solusi optimal adalah minimalisasi biaya, maksimalisasi keuntungan, efisiensi jadwal, dan penggunaan sumber daya (Laudon, 2008). Definisi dari algoritma genetika sendiri adalah suatu teknik pemecahan masalah secara heuristik yang mengadaptasi pada proses seleksi alam makhluk hidup dan genetik.

Secara umum, proses seleksi alam atau proses evolusi merupakan suatu perubahan yang terjadi pada makhluk hidup atau pada individu secara terus-menerus dan dapat menyesuaikan diri sehingga menjadi individu baru. Dalam proses seleksi alam tersebut terdapat istilah “Hanya individu yang kuat yang mampu bertahan” (Sutojo, 2011). Makna dari istilah tersebut adalah jika solusi yang dihasilkan buruk, maka solusi tersebut tidak dapat melanjutkan proses seleksi. Sebaliknya jika solusi yang dihasilkan baik, maka solusi tersebut bertahan ke proses selanjutnya untuk mendapatkan solusi yang lebih baik lagi.

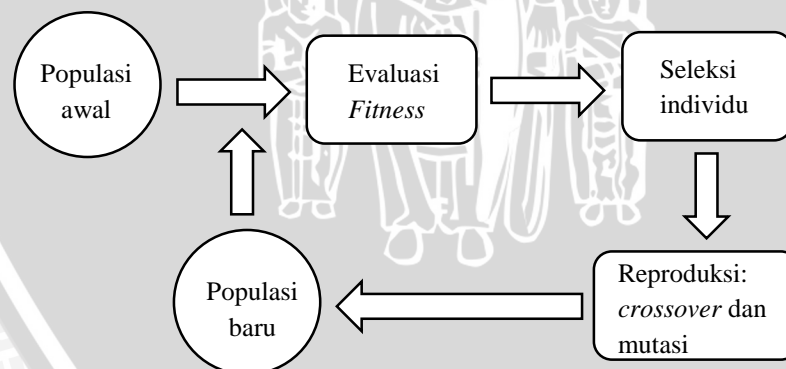
Algoritma genetika digunakan untuk memecahkan masalah menggunakan solusi optimal memiliki beberapa kelebihan (Mahmudy, 2013), antara lain:

1. Algoritma genetika merupakan suatu algoritma yang menggunakan metode optimasi dapat memecahkan suatu masalah yang kompleks dan memiliki ruang pencarian (*search space*) yang luas (Gen & Cheng dalam Mahmudy, 2013).
2. Dalam proses algoritma genetika menggunakan representasi kromosom sebagai teknik *decoding* (pengkodean) untuk mengkodekan nilai gen-gen pembentuk individu. Teknik *decoding* juga dapat melakukan pencarian dari masalah yang diselesaikan.
3. Masalah yang kompleks dapat diselesaikan menggunakan algoritma genetika dan juga menggunakan banyak variabel, seperti variabel kontinyu, diskrit, atau gabungan dari kontinyu dan diskrit.

- Solusi optimal yang dihasilkan algoritma genetika dapat menyelesaikan masalah dengan banyak objektif (Mahmudy & Rahman dalam Mahmudy, 2013).

2.4.1 Siklus Algoritma Genetika

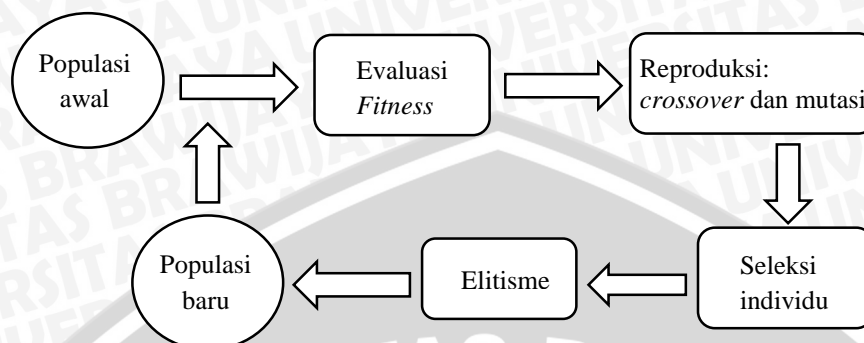
Proses-proses pada algoritma genetika membentuk sebuah siklus algoritma genetika. Siklus Algoritma Genetika diperkenalkan pertama kali oleh David Goldberg. Proses pertama pada siklus adalah membuat sebuah populasi awal secara acak. Tahapan kedua, tiap individu yang terdapat dalam populasi tersebut dihitung nilai *fitness*-nya. Pada proses ketiga, dilakukan seleksi individu, proses ini menyeleksi individu yang terbaik dimana yang memiliki nilai *fitness* terbesar. Proses keempat yaitu individu yang telah diseleksi akan dilakukan proses reproduksi dimana terdapat dua proses, antara lain proses *crossover* dan proses mutasi. Setelah dilakukan proses reproduksi maka akan terbentuk sebuah populasi baru. Proses akan berlanjut terus sampai pada generasi ke- n , populasi baru akan mengalami siklus yang sama dengan populasi sebelumnya (Sutojo, 2011). Siklus Algoritma Genetika ditunjukkan pada Gambar 2.6.



Gambar 2.6 Siklus Algoritma Genetika oleh David Goldberg
Sumber: Sutojo, 2011

Siklus algoritma genetika yang diperkenalkan oleh David Goldberg diperbaiki oleh Zbigniew Michalewicz. Perbaikan yang diberikan adalah menambahkan satu proses pada siklus tersebut, yaitu proses elitisme dan

merubah urutan proses seleksi individu dan proses reproduksi. Perbaikan siklus algoritma genetika ditampilkan pada Gambar 2.7 (Sutojo, 2011).



Gambar 2.7 Perbaikan Siklus Algoritma Genetika oleh Zbigniew Michalewicz
Sumber: Sutojo, 2011

2.4.2 Komponen-komponen Algoritma Genetika

Algoritma genetika memiliki delapan komponen utama, diantaranya adalah teknik *encoding/decoding* gen dan individu, nilai *fitness*, elitisme, seleksi, *crossover*, mutasi, dan kriteria penghentian.

2.4.2.1 Teknik *Encoding/Decoding*

Langkah awal dalam menyelesaikan suatu masalah dengan menggunakan algoritma genetika dengan menggunakan teknik *encoding/decoding* atau pengkodean. Dalam langkah ini, suatu masalah akan dirubah dalam bentuk satu atau lebih kromosom yang merepresentasikan suatu individu dengan kode tertentu. Teknik pengkodean dalam algoritma genetika terdapat 3 macam, yaitu yang pertama pengkodean bilangan real, dimana pada pengkodean ini menggunakan nilai kromosom antara [0, 1]. Pengkodean yang kedua adalah pengkodean diskrit desimal yang mempunyai nilai kromosom bilangan bulat dengan bilangan antara [0, 9]. Dan yang terakhir adalah teknik pengkodean biner, teknik ini memiliki nilai kromosom bilangan biner, yaitu 0 dan 1 (Sutojo, 2011).

2.4.2.2 Nilai *Fitness*

Nilai *fitness* dalam algoritma genetika digunakan untuk menyeleksi individu yang terbaik dengan memaksimalkan nilai *fitness*. Kualitas

kromosom dalam populasi ditunjukkan pada nilai *fitness* dari kromosom (Kusumadewi, 2003). Misalnya pada masalah optimasi akan dicari solusi yang memaksimalkan fungsi (h), maka nilai *fitness*-nya bernilai fungsi itu, yaitu $f = h$ dimana f adalah nilai *fitness*. Akan tetapi, berbeda halnya pada masalah optimasi dimana yang dicari adalah solusi minimal, maka nilai *fitness*-nya menjadi $f = \frac{1}{h}$. Fungsi (h) untuk masalah optimasi akan terdapat masalah jika bernilai 0, maka solusi dari masalah tersebut adalah dengan menggunakan persamaan 2-7.

$$f = \frac{1}{(h+a)} \quad (2-7)$$

Dimana a merupakan nilai minimum untuk menyelesaikan masalah. Sebagai contoh, terdapat fungsi h dengan nilai minimum bernilai 0, maka nilai *fitness*-nya adalah $f = \frac{1}{(h+a)}$. Nilai a dideklarasikan bernilai 0.001. Jadi nilai *fitness*-nya $f = \frac{1}{(h+a)} = \frac{1}{(0+0.001)} = 1000$ (Suyanto, 2011).

2.4.2.3 Elitisme

Proses elitisme adalah proses pada algoritma genetika dengan cara meng-copy individu yang memiliki nilai *fitness* tertinggi untuk menjaga agar individu tersebut nilai *fitness*-nya tidak turun. Nilai *fitness* suatu individu dapat menurun atau rusak biasanya terjadi pada waktu proses *crossover* atau pindah silang, dengan adanya prosedur elitisme dapat meminimalisir turunnya nilai *fitness* suatu individu. Prosedur elitisme digunakan pada algoritma genetika yang berjenis *generational replacement* (Suyanto, 2011). Algoritma genetika berjenis *generational replacement* adalah pergantian populasi dimana kromosom *parent* digantikan oleh kromosom *offspring* dari hasil *crossover* dan mutasi.

2.4.2.4 Seleksi

Seleksi merupakan proses penentuan individu dan *offspring* (kromosom anak) hasil reproduksi untuk dipilih dijadikan individu untuk generasi berikutnya. Tahap awal melakukan proses seleksi adalah melakukan pencarian

nilai *fitness* dari masing-masing individu yang kemudian nilai *fitness* tersebut digunakan untuk proses seleksi. Adapun metode seleksi pada algoritma genetika, antara lain:

1. *Roulette wheel selection*

Metode seleksi ini menggunakan nilai probabilitas seleksi setiap individu dari nilai *fitness* dan nilai probabilitas kumulatif yang didapat dari perhitungan nilai probabilitas. Selanjutnya, dengan membangkitkan bilangan random antara [0, 1]. Individu yang terpilih berasal dari perbandingan antara bilangan random dengan nilai probabilitas kumulatif tiap individu dimana bilangan random lebih kecil sama dengan nilai probabilitas kumulatif (Mahmudy, 2013).

2. *Tournament selection*

Seleksi turnamen akan menggunakan nilai *tour* yang telah ditetapkan untuk individu-individu yang dipilih secara acak. Nilai *tour* merupakan nilai acak yang digunakan untuk memilih individu. Individu yang telah terpilih secara acak tersebut akan diseleksi menjadi induk (*parent*) adalah individu yang terbaik. Nilai *tour* sendiri bernilai antara 2 sampai N, dimana N adalah jumlah individu dari suatu populasi (Kusumadewi, 2003).

3. *Elitism selection*

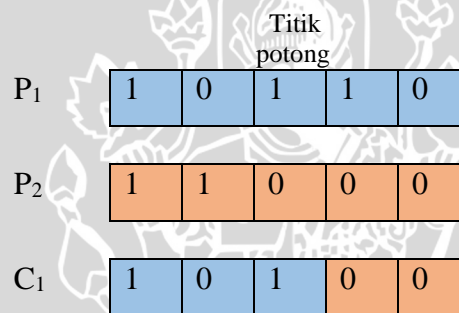
Cara kerja metode seleksi *elitism* adalah semua individu baik *parent* maupun *offspring* diurutkan dari yang terbaik berdasarkan nilai *fitness* masing-masing individu. Individu yang terbaik akan di-copy ke dalam populasi baru yang akan menjadi generasi berikutnya.

4. *Replacement selection*

Terdapat dua aturan dalam metode seleksi *replacement* (Mahmudy, 2013), yaitu: *parent* yang akan digantikan oleh *offspring* adalah *offspring* yang memiliki nilai *fitness* terbaik dimana *offspring* tersebut diperoleh dari proses mutasi. Aturan kedua adalah jika *offspring* yang dihasilkan dari proses *crossover*, maka *offspring* menggantikan *parent* pada proses *crossover* yang memiliki nilai *fitness* terlemah (Mahmudy, 2013).

2.4.2.5 Crossover (Pindah Silang)

Crossover adalah proses reproduksi dimana dilakukan pindah silang atau penyilangan (*crossover*) terhadap dua kromosom atau individu dari satu populasi yang menghasilkan kromosom anak (*offspring*). Sama halnya dengan proses reproduksi makhluk hidup, kromosom anak akan terdapat sebagian sifat dari kromosom induknya. Pada proses reproduksi *crossover* terdapat metode yang sering digunakan, yaitu *one-cut-point crossover*. Metode ini dapat digunakan pada representasi biner, permutasi, dan *real-coded*. Langkah awal *one-cut-point crossover* adalah ambil beberapa bagian dari kromosom induk pertama kemudian tukarkan dengan beberapa bagian lainnya dari kromosom induk kedua untuk menghasilkan satu *offspring*, yang dijelaskan pada Gambar 2.8 (Mahmudy, 2013).



Gambar 2.8 Metode *one-cut-point crossover*
Sumber: Mahmudy, 2013

Pada *crossover* jumlah kromosom anak (*offspring*) akan ditentukan dengan menggunakan peluang *crossover* atau *crossover rate* (*cr*). Misalkan ditentukan nilai *cr* sebesar $cr = 0.2$ dan memiliki jumlah *popSize* sebanyak 10, berarti jumlah kromosom anak yang akan dihasilkan sebanyak $cr \times \sum popSize = 0.2 \times 10 = 2$ *offspring*. Jadi, *crossover* dilakukan sebanyak dua kali untuk menghasilkan kromosom anak (*offspring*) sebanyak dua (Mahmudy, 2013).

2.4.2.6 Mutasi

Proses reproduksi yang kedua setelah *crossover* adalah mutasi. Dalam algoritma genetika, terdapat dua metode mutasi yang sering digunakan, yaitu *reciprocal exchange mutation* dan *insertion mutation*. Metode *reciprocal*

exchange mutation merupakan metode mutasi dimana cara kerjanya menukarkan dua gen kromosom (XP₁ dan XP₂) secara random (Mahmudy, 2013). Contoh *reciprocal exchange mutation* ditunjukkan pada Gambar 2.9.

	XP ₁		XP ₂		
Parent	1	3	2	4	5
Child	4	3	2	1	5

Gambar 2.9 Metode *Reciprocal Exchange Mutation*
Sumber: Mahmudy, 2013

Selanjutnya, *insertion mutation* adalah metode mutasi yang memilih salah satu gen kromosom sebagai *selected point* (SP) dan salah satu gen kromosom sebagai *insertion point* (IP) secara acak kemudian SP disisipkan pada IP (Mahmudy, 2013). Penjelasan lebih detail ditunjukkan pada gambar 2.10.

	IP	SP			
Parent	1	3	2	4	5
Child	4	1	3	2	5

Gambar 2.10 Metode *Insertion Mutation*
Sumber: Mahmudy, 2013

Kedua metode mutasi tersebut digunakan pada representasi permutasi, jika representasi biner menggunakan metode *cut point* dimana pilih salah satu gen kromosom secara acak kemudian nilai gen kromosom diganti (Mahmudy, 2013), jika nilai gen kromosom 1 menjadi 0 atau nilai gen kromosom 0 menjadi 1, seperti pada Gambar 2.11.

	Cut point				
Parent	1	0	0	1	1
Child	0	0	0	1	1

Gambar 2.11 Metode *Cut Point* (Titik Potong)
Sumber: Mahmudy, 2013

Pada mutasi jumlah kromosom anak (*offspring*) akan ditentukan dengan menggunakan peluang mutasi atau *mutation rate* (*mr*). Misalkan ditentukan

nilai mr sebesar $mr = 0.2$ dan memiliki jumlah $popSize$ sebanyak 10, berarti jumlah kromosom anak yang akan dihasilkan sebanyak $mr \times \sum popSize = 0.2 \times 10 = 2$ *offspring*. Jadi, mutasi dilakukan sebanyak dua kali untuk menghasilkan kromosom anak (*offspring*) sebanyak dua (Mahmudy, 2013).

2.4.2.7 Kriteria Penghentian

Kriteria penghentian merupakan kondisi berhenti untuk iterasi dalam proses algoritma genetika. Terdapat beberapa kriteria penghentian (Mahmudy, 2013), antara lain:

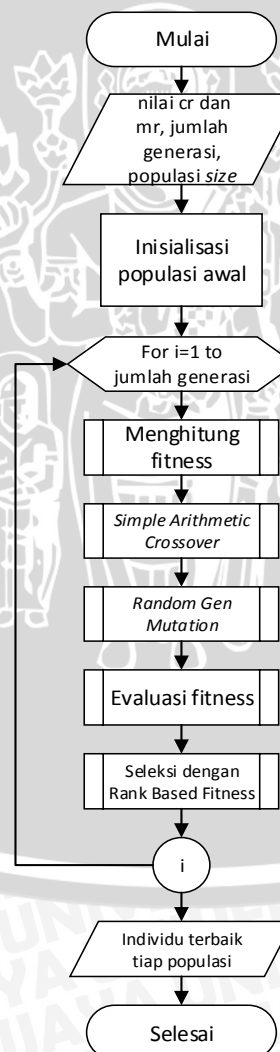
1. Iterasi algoritma genetika akan berhenti ketika iterasi tersebut dilakukan sampai generasi ke- n . Nilai n biasanya ditentukan sebelumnya berupa bilangan bulat positif. Setelah dilakukan iterasi sampai generasi ke- n mendapatkan solusi terbaik. Penentuan nilai n berdasarkan ukuran dan kompleksitas suatu masalah yang dipecahkan (Mahmudy, 2013).
2. Setelah dilakukan iterasi sampai generasi ke- n dan seterusnya tidak mendapatkan solusi yang lebih baik (Mahmudy, 2013). Jika hal ini dilakukan terus menerus, maka hal ini hanya membuang waktu dikarenakan algoritma genetika tidak menemukan solusi yang lebih baik dari sebelumnya.
3. Iterasi algoritma genetika akan berhenti dengan memberikan batasan waktu t satuan waktu (Mahmudy, 2013).

2.5 Penelitian Sebelumnya

Penelitian mengenai penerapan Algoritma Genetika dalam Optimasi Fungsi Keanggotaan (*membership function*) telah dilakukan dengan objek perhitungan Angka Metabolisme Basal (AMB). Pembentukan fungsi keanggotaan pada logika *fuzzy* biasanya dilakukan secara manual atau menurut batas-batas yang telah ditentukan sebelumnya oleh pakar. Pada penelitian ini, peneliti menerapkan metode Algoritma Genetika untuk dapat melakukan pembentukan dan optimasi fungsi keanggotaan. Dengan menggunakan metode Algoritma Genetika dapat ditemukan sebuah solusi untuk melakukan optimasi pada nilai

fungsi keanggotaan yang nantinya hasil dari pengambilan keputusan lebih maksimal. Penentuan fungsi keanggotaan dalam menggunakan metode Logika *Fuzzy* yang terletak pada fase fuzzyfikasi biasanya harus dilakukan dengan tepat agar hasil akhir pada perhitungan Angka Metabolisme Basal (AMB) sesuai dengan hasil yang sebenarnya (Anggariaawan, 2014).

Pada tahap perancangan dalam penelitian ini, siklus pada metode Algoritma Genetika terdiri dari beberapa tahap yaitu, inisialisasi populasi, menghitung nilai *fitness*, proses reproduksi, dan seleksi. Metode yang digunakan dalam proses reproduksi adalah *Simple Arithmetic Crossover* dan *Random Gen Mutation*. Sedangkan pada proses seleksi digunakan metode *Rank Based Fitness*. Siklus algoritma genetika pada penelitian ini ditampilkan pada Gambar 2.12.



Gambar 2.12 Diagram alir Algoritma Genetika

Sumber: Anggariaawan, 2014

Selanjutnya dalam tahap pengujian dan analisis sistem, didapatkan nilai *crossover rate* sebesar 0.9, *mutation rate* adalah 0.9, jumlah generasi sebanyak 75 generasi, dan *popSize* (ukuran populasi) sebanyak 50 untuk fungsi keanggotaan Indeks Massa Tubuh (IMT). Sedangkan untuk fungsi keanggotaan umur nilai *crossover rate* (*cr*) sebesar 0.9, *mutation rate* (*mr*) adalah 0.7, jumlah generasi sebanyak 100 generasi, dan *popSize* (ukuran populasi) sebanyak 50. Dari penelitian ini dapat disimpulkan bahwa nilai *crossover rate* dan *mutation rate* semakin tinggi nilai *cr* dan nilai *mr* maka nilai *fitness* semakin tinggi (baik). Hasil akurasi yang diperoleh untuk perhitungan Angka Metabolisme Basal (AMB) terhadap data uji sebanyak 100 dengan menghasilkan nilai akurasi sebesar 94% lebih baik dari penelitian sebelumnya yang memiliki nilai akurasi sebesar 75%. Jadi, metode algoritma genetika terbukti dapat menghasilkan hasil akhir yang maksimal dan sesuai dengan hasil yang sebenarnya (Anggariawan, 2014).

2.6 Akurasi Sistem

Dalam pengimplementasian sistem optimasi fungsi keanggotaan *fuzzy* Tsukamoto dibutuhkan sebuah proses akurasi sistem. Proses akurasi sistem ini menggunakan perhitungan korelasi *spearman*. Akurasi sistem yang dilakukan adalah dengan menghitung nilai bobot (*Z*) dari hasil sistem inferensi *fuzzy* Tsukamoto kemudian dilakukan *pe-ranking-an* dan dihitung dengan menggunakan persamaan korelasi *spearman*. Korelasi *spearman* merupakan ukuran antara hubungan dua variabel berdasarkan *pe-ranking-an* pada setiap nilai variabel (Supranto dalam Pradeka, 2012). Persamaan koefisien korelasi *spearman* dinyatakan dalam persamaan 2-8.

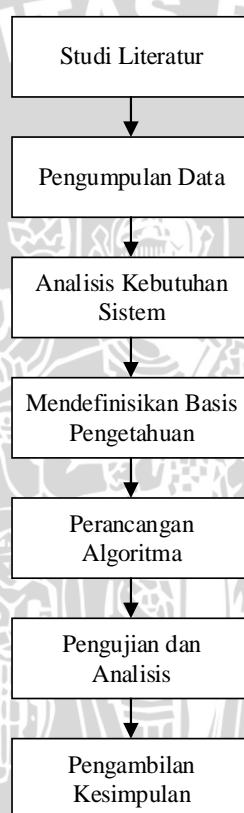
$$r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n^3 - n} \quad (2-8)$$

Dimana r_s adalah koefisien korelasi *spearman*, d_i merupakan selisih antara *ranking* yang dihasilkan antara dua variabel, dalam permasalahan optimasi ini, variabel tersebut adalah *ranking* inferensi *fuzzy* (*Z*) dengan pendapat pakar, n merupakan banyaknya data.

BAB III METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian mengenai optimasi fungsi keanggotaan *fuzzy* Tsukamoto dengan menggunakan metode algoritma genetika dalam menentukan calon penerima Beasiswa-PPA dan BBP-PA, memiliki beberapa tahapan yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian Optimasi Fungsi Keanggotaan *Fuzzy* Tsukamoto menggunakan Algoritma Genetika

Tahapan pertama pada penelitian ini, yaitu tahapan studi literatur. Tahapan studi literatur merupakan tahapan dimana peneliti mengumpulkan dan mempelajari teori yang berkenaan dengan beasiswa, logika *fuzzy*, sistem inferensi *fuzzy*, dan algoritma genetika. Selanjutnya tahapan kedua, pengumpulan data calon penerima beasiswa yang didapatkan dari penelitian sebelumnya. Tahapan analisis kebutuhan sistem merupakan perancangan kebutuhan apa saja yang

terdapat pada sistem. Tahapan definisi basis pengetahuan yaitu pembuatan *rules* dan mendefinisikan variabel apa saja yang digunakan. Tahapan perancangan algoritma merupakan tahapan dimana terdapat proses perhitungan dari algoritma genetika dan sistem inferensi *fuzzy* Tsukamoto. Kemudian, pada tahapan pengujian dan analisis, peneliti akan melakukan pengujian berdasarkan perancangan skenario yang telah dibuat yang kemudian akan dianalisis. Dan tahapan yang terakhir adalah tahapan pengambilan kesimpulan, tahapan ini akan menjelaskan hasil akhir yang didapat setelah melakukan penelitian.

3.2 Kebutuhan Sistem

Analisis kebutuhan sistem merupakan tahapan dimana menganalisis kebutuhan apa saja yang dibutuhkan oleh sistem dalam pengimplementasian optimasi fungsi keanggotaan *fuzzy* Tsukamoto menggunakan algoritma genetika. Kebutuhan umum yang biasa digunakan adalah kebutuhan *hardware* dan *software*. Kebutuhan *hardware* adalah perangkat keras yang digunakan peneliti untuk implementasi sistem. Sedangkan kebutuhan *software* merupakan perangkat lunak yang digunakan peneliti dalam implementasi sistem. Berikut adalah informasi mengenai kebutuhan sistem yang digunakan, yaitu:

1. Kebutuhan *Hardware*:
 - Laptop dengan spesifikasi Intel® Core™ i7 @2.20 GHz, RAM 4.00 GB
2. Kebutuhan *Software*:
 - Sistem operasi Microsoft Windows 8.1 64-bit
 - Browser Google Chrome
 - XAMPP 1.7.4
 - Adobe Dreamweaver CSS dan Notepad++

3.3 Formulasi Permasalahan

Optimasi fungsi keanggotaan pada logika *fuzzy* dilakukan menggunakan algoritma genetika. Pada penelitian ini, dilakukan optimasi fungsi keanggotaan dengan model *fuzzy* Tsukamoto untuk menentukan calon penerima Beasiswa-PPA dan BBP-PPA. Algoritma genetika akan membentuk batas-batas dari fungsi

keanggotaan *fuzzy* Tsukamoto sehingga hasil akhir dari penentuan calon penerima Beasiswa-PPA dan BBP-PPA akan lebih akurat.

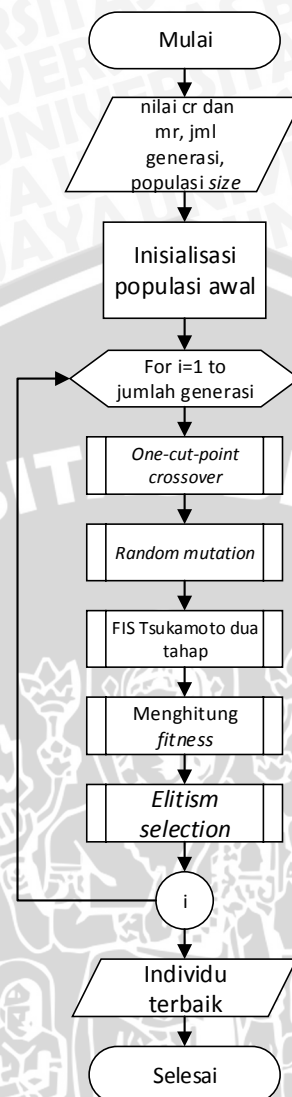
Data yang digunakan dalam penelitian ini adalah data dari penelitian Iskandar (2012) yang memuat sekitar 30 data mahasiswa PTIIK UB yang mendaftar sebagai calon penerima Beasiswa-PPA dan BBP-PPA. Dalam data mahasiswa terdapat data kriteria atau parameter untuk menentukan penerima beasiswa. Data kriteria tersebut terdiri dari 8 kriteria *input*, antara lain Indeks Prestasi Kumulatif (IPK), penghasilan orang tua, piagam penghargaan, tagihan rekening listrik, tagihan telepon, tagihan PDAM, pembayaran PBB, dan tanggungan orang tua/wali.

Pada optimasi fungsi keanggotaan *fuzzy* Tsukamoto menggunakan Algoritma Genetika diperlukan data *input*-an atau parameter algoritma genetika. Parameter algoritma genetika terdiri dari ukuran populasi (*popSize*), jumlah generasi, *crossover rate*, dan *mutation rate*. Data parameter ini nantinya dapat dilakukan pengujian untuk mendapatkan data parameter yang tepat. Siklus dari formulasi permasalahan ditunjukkan pada Gambar 3.2.

Selanjutnya, terdapat data yang berkenaan dengan tahapan inisialisasi pada algoritma genetika, yaitu data rentang nilai yang dibentuk pada masing-masing kriteria penentuan calon penerima beasiswa. Pada Tabel 3.1 ditunjukkan rentang nilai dari masing-masing kriteria.

Tabel 3.1 Rentang nilai kriteria penentuan beasiswa

Kriteria	Range nilai
IPK Beasiswa-PPA	3.0 – 4.0
IPK BBP-PPA	2.5 – 4.0
Penghasilan ortu (juta)	0 – 5.0
Tanggungan ortu (x 100rb)	0 – 5.0
Tagihan telepon (x 10rb)	0 – 15.0
Tagihan listrik (x 10rb)	0 – 20.0
Tagihan PDAM (x 10rb)	0 – 10.0
Pembayaran PBB (x 10 rb)	0 – 20.0
Keputusan Beasiswa	0 - 9

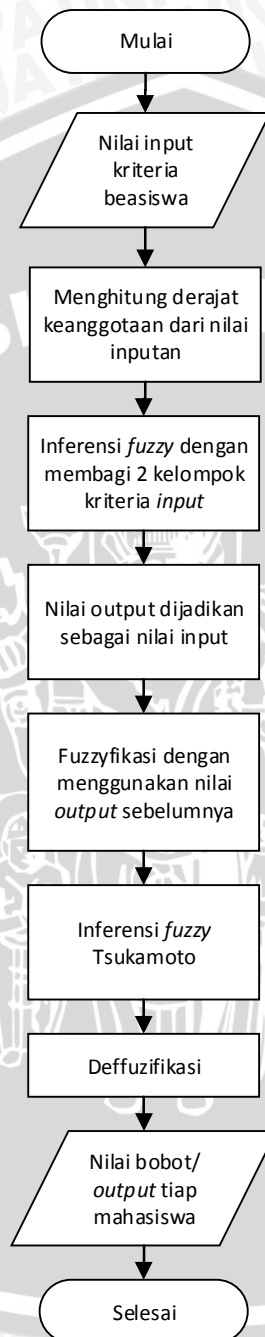


Gambar 3.2 Siklus formulasi permasalahan

3.4 Siklus Penyelesaian Masalah Menggunakan Inferensi Fuzzy Metode Tsukamoto

Pada proses inferensi *fuzzy* Tsukamoto terdapat tahapan perhitungan untuk menghasilkan keputusan berbasis seperti, fuzzyfikasi atau komposisi *fuzzy*, sistem inferensi, dan proses defuzzifikasi atau dekomposisi *fuzzy*. Dalam proses sistem inferensi *fuzzy* nantinya akan dilakukan dua tahap, dikarenakan jumlah *rules* yang sangat banyak. *Rules* tersebut berasal dari perhitungan jumlah kriteria *input* sebanyak 8 kriteria dan masing-masing kriteria memiliki 3 nilai linguistik, sehingga didapat *rules* sebanyak $3^8 = 6561$. Maka dari itu, dilakukan pemilihan beberapa *rules* yang tepat untuk sistem inferensi *fuzzy*. Pemilihan beberapa *rules*

yang kurang tepat dapat menghasilkan inferensi bernilai 0, sehingga penggunaan sistem inferensi *fuzzy* dua tahap dapat menyelesaikan masalah. Diagram alir dari sistem inferensi *fuzzy* metode Tsukamoto ditunjukkan pada Gambar 3.3.



Gambar 3.3 Diagram alir proses Sistem Inferensi *Fuzzy* Tsukamoto Dua Tahap

Proses fuzzyfikasi pada logika *fuzzy* merupakan perhitungan nilai *crisp* atau *input*-an dijadikan nilai derajat keanggotaan. Perhitungan ini berkaitan dengan fungsi keanggotaan *fuzzy* yang telah didefinisikan sebelumnya.

Selanjutnya, pada proses inferensi *fuzzy*, nilai derajat keanggotaan dihitung untuk mencari nilai *alpha* predikat sesuai dengan aturan-aturan yang telah dibuat. Proses defuzzifikasi sendiri adalah proses perhitungan untuk mencari nilai *output* berupa nilai *crisp* dengan menggunakan persamaan 2-6. Proses perhitungan sistem inferensi *fuzzy* Tsukamoto dapat dimisalkan seperti pada Tabel 3.2.

Tabel 3.2 Contoh permasalahan pada inferensi *fuzzy* Tsukamoto

Variabel input	Nilai input
IPK	3.65
Piagam	Wilayah
Tanggungan orang tua (x 100rb)	15.88
Penghasilan orang tua (juta)	6.61
Tagihan Listrik (x 10rb)	6.48
Tagihan Telepon (x 10rb)	15
Tagihan PDAM (x 10rb)	0
Tagihan PBB (x 10rb)	3.17

3.4.1 Himpunan Fuzzy

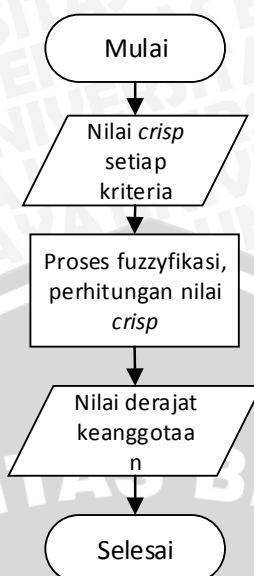
Himpunan *fuzzy* yang terdapat dalam penelitian ini mengenai optimasi fungsi keanggotaan *fuzzy* didapat dari penelitian sebelumnya, yaitu Iskandar (2012). Didalam penelitian Iskandar (2012), himpunan *fuzzy* telah didefinisikan sebelumnya. Penelitian Iskandar (2012) mengenai sistem pendukung keputusan pemilihan calon penerima beasiswa memiliki 8 jenis himpunan *fuzzy* atau kriteria variabel *input*. Sedangkan, himpunan *fuzzy* dengan variabel *output* adalah keputusan beasiswa memiliki 2 nilai linguistik, yaitu ditolak dan diterima. Iskandar (2012) telah menentukan batas-batas dari fungsi keanggotaan himpunan *fuzzy*, namun dalam penelitian ini batas-batas tersebut akan dilakukan optimasi menggunakan algoritma genetika. Himpunan *fuzzy* yang telah didefinisikan sebelumnya ditunjukkan dalam Tabel 3.3.

Tabel 3.3 Himpunan *fuzzy* variabel *input*

Himpunan <i>Fuzzy</i>	
Variabel input	Nilai Linguistik
Indeks Prestasi Kumulatif (IPK)	Rendah
	Sedang
	Tinggi
Penghasilan orang tua / wali	Rendah
	Sedang
	Tinggi
Piagam / penghargaan	Rendah
	Sedang
	Tinggi
Tagihan Listrik	Rendah
	Sedang
	Tinggi
Tagihan Telepon	Rendah
	Sedang
	Tinggi
Tagihan PDAM	Rendah
	Sedang
	Tinggi
Pembayaran PBB	Rendah
	Sedang
	Tinggi
Tanggungannya orang tua / wali	Rendah
	Sedang
	Tinggi

3.4.2 Fuzzyfikasi

Proses fuzzyfikasi merupakan perhitungan nilai *crisp* atau nilai *input* menjadi nilai derajat keanggotaan. Perhitungan dalam proses fuzzyfikasi berdasarkan batas-batas fungsi keanggotaan yang terdapat dalam grafik fungsi keanggotaan yang telah dibentuk dengan menggunakan metode algoritma genetika. Diagram alir dari proses fuzzyfikasi ditunjukkan dalam Gambar 3.4.



Gambar 3.4 Diagram alir fuzzyfikasi

Perhitungan dalam proses fuzzyfikasi melibatkan fungsi keanggotaan himpunan *fuzzy*. Fungsi keanggotaan himpunan *fuzzy* merepresentasikan suatu grafik fungsi keanggotaan yang nantinya akan menghasilkan sebuah nilai derajat keanggotaan dengan rentang nilai [0, 1]. Fungsi keanggotaan himpunan *fuzzy* dari 8 kriteria *input*, antara lain:

- a. Himpunan *fuzzy* tanggungan orang tua



- Derajat keanggotaan rendah

$$\mu_{tRendah}(x) = \begin{cases} 1 & ; x \leq 1.5 \\ \frac{3.3 - x}{3.3 - 1.5} & ; 1.5 < x \leq 3.3 \\ 0 & ; x > 3.3 \end{cases}$$

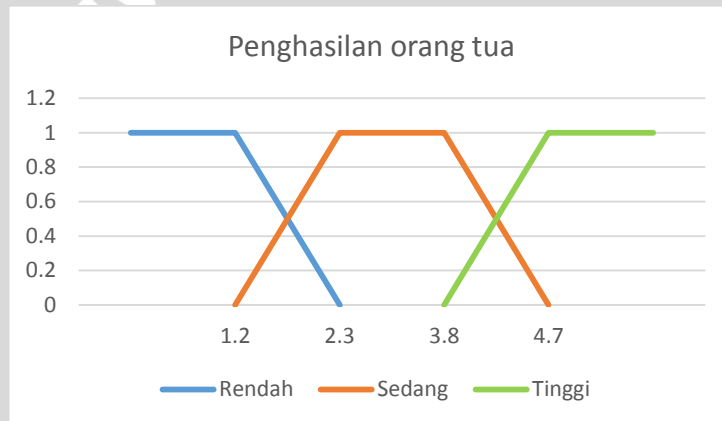
- Derajat keanggotaan sedang

$$\mu_{tSedang}(x) = \begin{cases} \frac{x - 1.5}{3 - 1.3} & ; 1.5 < x \leq 3.3 \\ 1 & ; 3.3 < x \leq 4 \\ \frac{5 - x}{5 - 4} & ; 4 < x \leq 5 \end{cases}$$

- Derajat keanggotaan tinggi

$$\mu_{tTinggi}(x) = \begin{cases} 0 & ; x \leq 4 \\ \frac{x - 4}{5 - 4} & ; 4 < x \leq 5 \\ 1 & ; x < 5 \end{cases}$$

b. Himpunan *fuzzy* penghasilan orang tua



- Derajat keanggotaan rendah

$$\mu_{pRendah}(x) = \begin{cases} 1 & ; x \leq 1.2 \\ \frac{2.3 - x}{2.3 - 1.2} & ; 1.2 < x \leq 2.3 \\ 0 & ; x > 2.3 \end{cases}$$

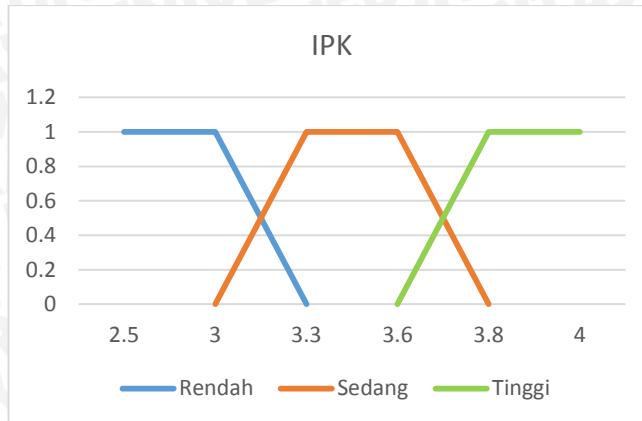
- Derajat keanggotaan sedang

$$\mu_{pSedang}(x) = \begin{cases} \frac{x - 1.2}{2.3 - 1.2} & ; 1.2 < x \leq 2.3 \\ 1 & ; 2.3 < x \leq 3.8 \\ \frac{4.7 - x}{4.7 - 3.8} & ; 3.8 < x \leq 4.7 \end{cases}$$

- Derajat keanggotaan tinggi

$$\mu_{pTinggi}(x) = \begin{cases} 0 & ; x \leq 3.8 \\ \frac{x - 3.8}{4.7 - 3.8} & ; 3.8 < x \leq 4.7 \\ 1 & ; x < 4.7 \end{cases}$$

c. Himpunan fuzzy IPK



▪ Derajat keanggotaan rendah

$$\mu_{IPKRendah}(x) = \begin{cases} 1 & ; 2.5 < x \leq 3 \\ \frac{3.3 - x}{3.3 - 3} & ; 3 < x \leq 3.3 \\ 0 & ; x > 3.3 \end{cases}$$

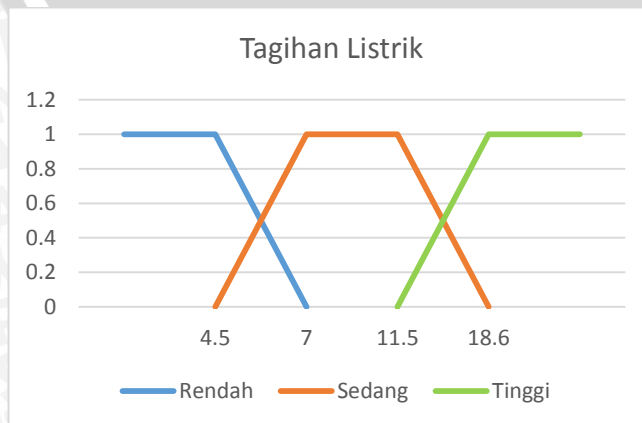
▪ Derajat keanggotaan sedang

$$\mu_{IPKSedang}(x) = \begin{cases} \frac{x - 3}{3.3 - 3} & ; 3 < x \leq 3.3 \\ 1 & ; 3.3 < x \leq 3.6 \\ \frac{3.8 - x}{3.8 - 3.6} & ; 3.6 < x \leq 3.8 \end{cases}$$

▪ Derajat keanggotaan tinggi

$$\mu_{IPKTinggi}(x) = \begin{cases} 0 & ; x \leq 3.6 \\ \frac{x - 3.6}{3.8 - 3.6} & ; 3.6 < x \leq 3.8 \\ 1 & ; 3.8 < x \leq 4 \end{cases}$$

d. Himpunan fuzzy tagihan listrik



- Derajat keanggotaan rendah

$$\mu_{ListRendah}(x) = \begin{cases} 1 & ; x \leq 4.5 \\ \frac{7-x}{7-4.5} & ; 4.5 < x \leq 7 \\ 0 & ; x > 7 \end{cases}$$

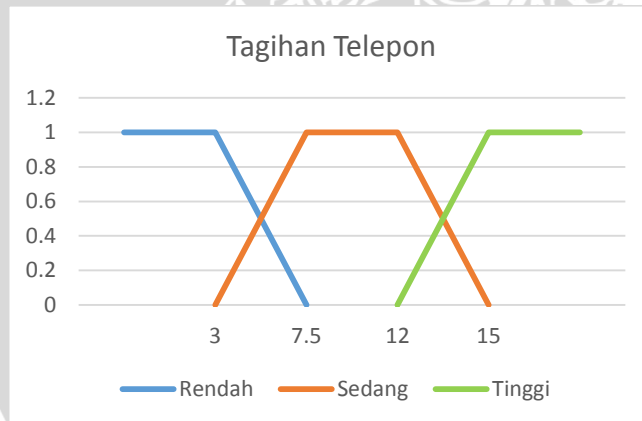
- Derajat keanggotaan sedang

$$\mu_{ListSedang}(x) = \begin{cases} \frac{x-4.5}{7-4.5} & ; 4.5 < x \leq 7 \\ 1 & ; 7 < x \leq 11.5 \\ \frac{18.6-x}{18.6-11.5} & ; 11.5 < x \leq 18.6 \end{cases}$$

- Derajat keanggotaan tinggi

$$\mu_{ListTinggi}(x) = \begin{cases} 0 & ; x \leq 11.5 \\ \frac{x-11.5}{18.6-11.5} & ; 11.5 < x \leq 18.6 \\ 1 & ; x > 18.6 \end{cases}$$

e. Himpunan *fuzzy* tagihan telepon



- Derajat keanggotaan rendah

$$\mu_{TelpRendah}(x) = \begin{cases} 1 & ; x \leq 3 \\ \frac{7.5-x}{7.5-3} & ; 3 < x \leq 7.5 \\ 0 & ; x > 7.5 \end{cases}$$

- Derajat keanggotaan sedang

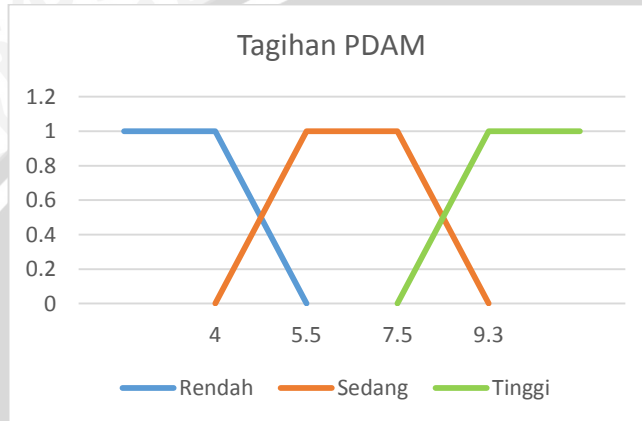
$$\mu_{TelpSedang}(x) = \begin{cases} \frac{x-3}{7.5-3} & ; 3 < x \leq 7.5 \\ 1 & ; 7.5 < x \leq 12 \\ \frac{15-x}{15-12} & ; 12 < x \leq 15 \end{cases}$$



- Derajat keanggotaan tinggi

$$\mu_{TelpTinggi}(x) = \begin{cases} 0 & ; x \leq 12 \\ \frac{x - 12}{15 - 12} & ; 12 < x \leq 15 \\ 1 & ; x < 15 \end{cases}$$

f. Himpunan *fuzzy* tagihan PDAM



- Derajat keanggotaan rendah

$$\mu_{PDAMRendah}(x) = \begin{cases} 1 & ; x \leq 4 \\ \frac{5.5 - x}{5.5 - 4} & ; 4 < x \leq 5.5 \\ 0 & ; x > 5.5 \end{cases}$$

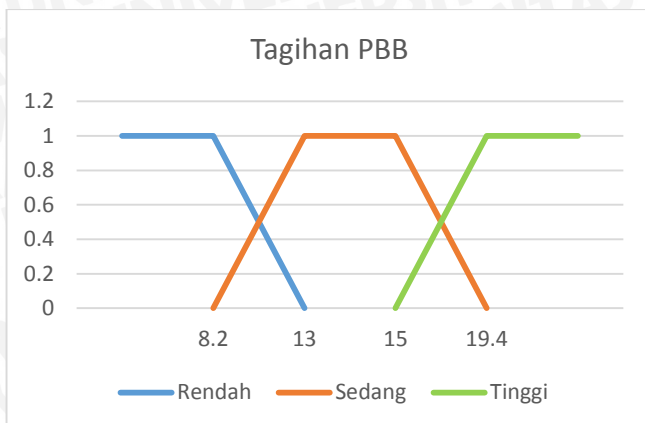
- Derajat keanggotaan sedang

$$\mu_{PDAMSedang}(x) = \begin{cases} \frac{x - 4}{5.5 - 4} & ; 4 < x \leq 5.5 \\ 1 & ; 5.5 < x \leq 7.5 \\ \frac{9.3 - x}{9.3 - 7.5} & ; 7.5 < x \leq 9.3 \end{cases}$$

- Derajat keanggotaan tinggi

$$\mu_{PDAMTinggi}(x) = \begin{cases} 0 & ; x \leq 7.5 \\ \frac{x - 7.5}{9.3 - 7.5} & ; 7.5 < x \leq 9.3 \\ 1 & ; x < 9.3 \end{cases}$$

g. Himpunan fuzzy tagihan PBB



- Derajat keanggotaan rendah

$$\mu_{PBBRendah}(x) = \begin{cases} 1 & ; x \leq 8.2 \\ \frac{13 - x}{13 - 8.2} & ; 8.2 < x \leq 13 \\ 0 & ; x > 13 \end{cases}$$

- Derajat keanggotaan sedang

$$\mu_{PBBsedang}(x) = \begin{cases} \frac{x - 8.2}{13 - 8.2} & ; 8.2 < x \leq 13 \\ 1 & ; 15 < x \leq 15 \\ \frac{19.4 - x}{19.4 - 15} & ; 15 < x \leq 19.4 \end{cases}$$

- Derajat keanggotaan tinggi

$$\mu_{PBBtinggi}(x) = \begin{cases} 0 & ; x \leq 15 \\ \frac{x - 15}{19.4 - 15} & ; 15 < x \leq 19.4 \\ 1 & ; x > 19.4 \end{cases}$$

h. Himpunan fuzzy keputusan beasiswa



- Derajat keanggotaan ditolak

$$\mu_{beasiswaTolak}(x) = \begin{cases} 1 & ; x \leq 4 \\ \frac{9-x}{9-4} & ; 4 < x \leq 9 \\ 0 & ; x > 9 \end{cases}$$

- Derajat keanggotaan diterima

$$\mu_{beasiswaTerima}(x) = \begin{cases} 0 & ; x \leq 4 \\ \frac{x-4}{9-4} & ; 4 < x \leq 9 \\ 1 & ; x > 9 \end{cases}$$

3.4.3 Sistem Inferensi Fuzzy

Tahapan sistem inferensi *fuzzy* Tsukamoto merupakan proses dimana nilai derajat keanggotaan ($\mu_A(x)$) yang diperoleh dari proses fuzzyfikasi akan dicari nilai α -predikat berdasarkan basis aturan atau *rule-based*. Basis aturan yang digunakan dalam permasalahan ini menggunakan bentuk *if-then*. Dimisalkan terdapat aturan atau *rule*, yaitu:

[R1] IF (IPK is Rendah) AND (Tanggung Ortu is Sedang) AND (Penghasilan Ortu is Sedang) THEN (Keputusan Beasiswa is Ditolak)

Proses inferensi *fuzzy* Tsukamoto selanjutnya adalah menghitung nilai α -predikat sesuai dengan basis aturan yang telah dibuat. Perhitungan α -predikat ini menggunakan operasi himpunan *fuzzy* irisan pada persamaan 2-3, yaitu;

$$\alpha_{predikat1} = \mu_{IPKRendah} \cap \mu_{tSedang} \cap \mu_{pSedang}$$

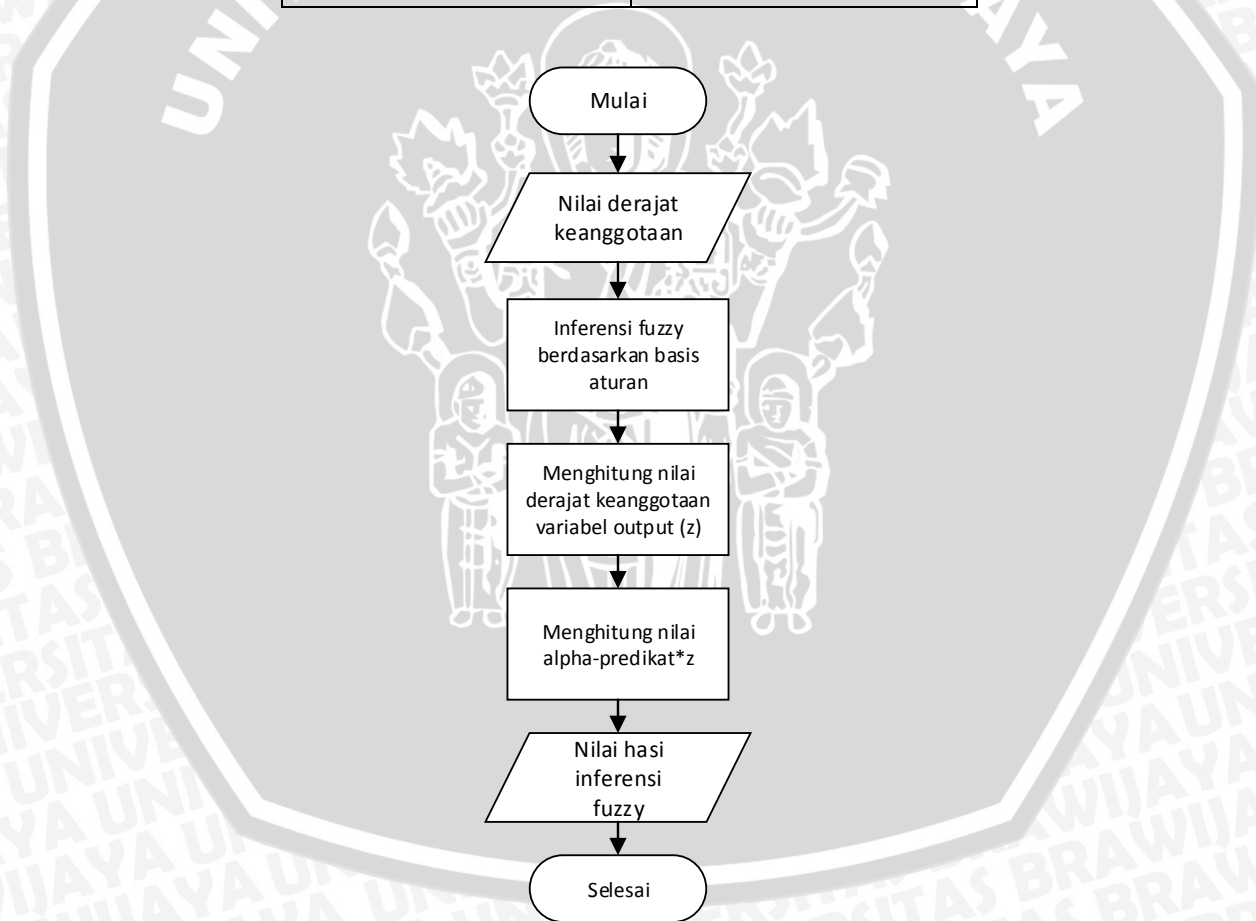
$$\alpha_{predikat1} = \min(\mu_{IPKRendah}, \mu_{tSedang}, \mu_{pSedang})$$

Sistem inferensi *fuzzy* untuk permasalahan optimasi fungsi keanggotaan terdapat dua tahap. Pada sistem inferensi *fuzzy* tahap pertama, 8 kriteria dibagi menjadi dua kelompok yaitu, kriteria positif dan kriteria negatif. Kriteria positif merupakan kriteria penentuan beasiswa dimana semakin tinggi nilai yang didapat untuk setiap kriteria, maka semakin tinggi juga nilai akhir yang diperoleh untuk penentuan beasiswa. Kriteria negatif merupakan kriteria penentuan beasiswa dimana berkebalikan dengan kriteria positif, semakin tinggi nilai setiap kriteria, maka nilai akhir penentuan beasiswa semakin rendah. Pada Tabel 3.4 ditampilkan kriteria positif dan

kriteria negatif untuk penentuan Beasiswa-PPA dan BBP-PPA. Pada Gambar 3.5 ditunjukkan alur proses dari inferensi *fuzzy*.

Tabel 3.4 Kriteria positif dan negatif untuk penentuan Beasiswa

Kriteria positif	IPK
	Tanggungannya Orang Tua
	Piagam
Kriteria negatif	Penghasilan Orang Tua
	Tagihan Listrik
	Tagihan Telepon
	Tagihan PDAM
	Tagihan PBB



Gambar 3.5 Diagram alir proses sistem inferensi *fuzzy* Tsukamoto

Basis aturan dari inferensi *fuzzy* Tsukamoto pada permasalahan ini terdapat 27 *rules* untuk kriteria positif pada himpunan *fuzzy* dan 81 *rules* untuk kriteria negatif. Pada Tabel 3.5 dan 3.6 ditunjukkan basis aturan inferensi *fuzzy* Tsukamoto untuk penentuan Beasiswa-PPA.

Tabel 3.5 Basis aturan inferensi *fuzzy* Tsukamoto kriteria positif

Rule	IPK	Tanggung ortu	Piagam	Keputusan
1	Rendah	Rendah	Rendah	Rendah
2	Rendah	Rendah	Sedang	Rendah
3	Rendah	Rendah	Tinggi	Rendah
4	Rendah	Sedang	Rendah	Rendah
5	Rendah	Sedang	Sedang	Rendah
6	Rendah	Sedang	Tinggi	Rendah
7	Rendah	Tinggi	Rendah	Rendah
8	Rendah	Tinggi	Sedang	Rendah
9	Rendah	Tinggi	Tinggi	Rendah
10	Sedang	Rendah	Rendah	Rendah
11	Sedang	Rendah	Sedang	Rendah
12	Sedang	Rendah	Tinggi	Tinggi
13	Sedang	Sedang	Rendah	Tinggi
14	Sedang	Sedang	Sedang	Tinggi
15	Sedang	Sedang	Tinggi	Tinggi
16	Sedang	Tinggi	Rendah	Rendah
17	Sedang	Tinggi	Sedang	Rendah
18	Sedang	Tinggi	Tinggi	Rendah
19	Tinggi	Rendah	Rendah	Rendah
20	Tinggi	Rendah	Sedang	Tinggi
21	Tinggi	Rendah	Tinggi	Tinggi
22	Tinggi	Sedang	Rendah	Tinggi
23	Tinggi	Sedang	Sedang	Tinggi
24	Tinggi	Sedang	Tinggi	Tinggi
25	Tinggi	Tinggi	Rendah	Tinggi

26	Tinggi	Tinggi	Sedang	Tinggi
27	Tinggi	Tinggi	Tinggi	Tinggi

Tabel 3.6 Perhitungan inferensi *fuzzy* Tsukamoto kriteria positif

Rule	IPK	Tanggungan ortu	Piagam	α_p	z	$\alpha_p * z$
1	0	0	1	0	3	0
2	0	0	0	0	3	0
3	0	0	0	0	3	0
4	0	0	1	0	3	0
5	0	0	0	0	3	0
6	0	0	0	0	3	0
7	0	1	1	0	3	0
8	0	1	0	0	3	0
9	0	1	0	0	3	0
10	0.75	0	1	0	1	0
11	0.75	0	0	0	1	0
12	0.75	0	0	0	1	0
13	0.75	0	1	0	1	0
14	0.75	0	0	0	1	0
15	0.75	0	0	0	1	0
16	0.75	1	1	0.75	1.5	1.125
17	0.75	1	0	0	3	0
18	0.75	1	0	0	3	0
19	0.25	0	1	0	1	0
20	0.25	0	0	0	1	0
21	0.25	0	0	0	1	0
22	0.25	0	1	0	1	0
23	0.25	0	0	0	1	0
24	0.25	0	0	0	1	0
25	0.25	1	1	0.25	1.5	0.375

26	0.25	1	0	0	1	0
27	0.25	1	0	0	1	0
			$\sum \alpha_{p_i} =$	1	$\sum(\alpha_{p_i} * z_i) =$	1.5

Penjelasan dari proses perhitungan inferensi *fuzzy* Tsukamoto untuk rule 16 kriteria positif, yaitu:

1. Nilai derajat keanggotaan IPK Sedang

$$\mu_{IPK_{Sedang}}(3.65) = \frac{3.8 - x}{3.8 - 3.6} = \frac{3.8 - 3.65}{3.8 - 3.6} = 0.75$$

2. Nilai derajat keanggotaan Tanggungan Orang tua Tinggi

$$\mu_{tangTinggi}(15.88) = 1$$

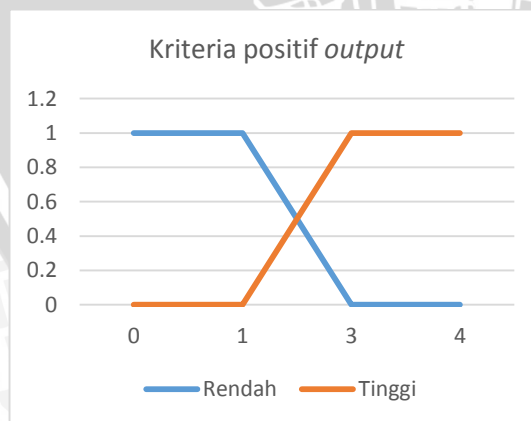
3. Nilai derajat keanggotaan Piagam Wilayah

$$\mu_{pRendah}(Wilayah) = 1$$

4. Nilai $\alpha_{predikat}$ atau nilai hasil inferensi

$$\begin{aligned} \alpha_{p_{16}} &= \mu_{IPK_{Sedang}} \cap \mu_{tangTinggi} \cap \mu_{pRendah} \\ \alpha_{p_{16}} &= \min(\mu_{IPK_{Sedang}}, \mu_{tangTinggi}, \mu_{pRendah}) \\ &= \min(0.75; 1; 1) = 0.75 \end{aligned}$$

5. Nilai *crisp* himpunan *output*, nilai ini didapat berdasarkan grafik himpunan *fuzzy output* untuk kriteria positif



$$\mu_{positifRendah}(z) = \frac{3 - z}{3 - 1}$$

$$0.75 = \frac{3 - z}{2}$$

$$1.5 = 3 - z$$

$$z = 1.5$$

6. Nilai $(\alpha_{predikat} * z)$

$$\begin{aligned} \alpha_{p_{16}} \times z_{16} &= 0.75 \times 1.5 \\ &= 1.125 \end{aligned}$$

Setelah dilakukuan proses inferensi *fuzzy* Tsukamoto untuk kriteria positif, langkah selanjutnya adalah dihitung nilai *output* kriteria positif ($Z_{positif}$) yang nantinya merupakan nilai *input* untuk perhitungan inferensi *fuzzy* Tsukamoto tahap kedua.

$$Z_{positif} = \frac{\sum(\alpha_{p_i} \times z_i)}{\sum \alpha_{p_i}} = \frac{1.5}{1} = 1.5$$

Tabel 3.7 Basis aturan inferensi *fuzzy* Tsukamoto kriteria negatif

Rule	Gaji	Listrik	Telepon	PDAM	PBB	Keputusan
1	Rendah	Rendah	Rendah	Rendah	Rendah	Rendah
2	Rendah	Rendah	Rendah	Rendah	Sedang	Rendah
3	Rendah	Rendah	Rendah	Rendah	Tinggi	Rendah
4	Rendah	Rendah	Rendah	Sedang	Rendah	Rendah
5	Rendah	Rendah	Rendah	Sedang	Sedang	Rendah
6	Rendah	Rendah	Rendah	Sedang	Tinggi	Rendah
7	Rendah	Rendah	Rendah	Tinggi	Rendah	Rendah
8	Rendah	Rendah	Rendah	Tinggi	Sedang	Rendah
9	Rendah	Rendah	Rendah	Tinggi	Tinggi	Rendah
10	Rendah	Rendah	Sedang	Rendah	Rendah	Rendah
11	Rendah	Rendah	Sedang	Rendah	Sedang	Tinggi
12	Rendah	Rendah	Sedang	Rendah	Tinggi	Tinggi
13	Rendah	Rendah	Sedang	Sedang	Rendah	Rendah
14	Rendah	Rendah	Sedang	Sedang	Sedang	Tinggi
15	Rendah	Rendah	Sedang	Sedang	Tinggi	Tinggi
16	Rendah	Rendah	Sedang	Tinggi	Rendah	Rendah
17	Rendah	Rendah	Sedang	Tinggi	Sedang	Rendah
18	Rendah	Rendah	Sedang	Tinggi	Tinggi	Rendah
19	Rendah	Rendah	Tinggi	Rendah	Rendah	Rendah
20	Rendah	Rendah	Tinggi	Rendah	Sedang	Rendah
21	Rendah	Rendah	Tinggi	Rendah	Tinggi	Rendah
22	Rendah	Rendah	Tinggi	Sedang	Rendah	Rendah
23	Rendah	Rendah	Tinggi	Sedang	Sedang	Rendah

24	Rendah	Rendah	Tinggi	Sedang	Tinggi	Rendah
25	Rendah	Rendah	Tinggi	Tinggi	Rendah	Rendah
26	Rendah	Rendah	Tinggi	Tinggi	Sedang	Rendah
27	Rendah	Rendah	Tinggi	Tinggi	Tinggi	Rendah
28	Rendah	Sedang	Rendah	Rendah	Rendah	Tinggi
29	Rendah	Sedang	Rendah	Rendah	Sedang	Tinggi
30	Rendah	Sedang	Rendah	Rendah	Tinggi	Tinggi
31	Rendah	Sedang	Rendah	Sedang	Rendah	Rendah
32	Rendah	Sedang	Rendah	Sedang	Sedang	Rendah
33	Rendah	Sedang	Rendah	Sedang	Tinggi	Tinggi
34	Rendah	Sedang	Rendah	Tinggi	Rendah	Rendah
35	Rendah	Sedang	Rendah	Tinggi	Sedang	Rendah
36	Rendah	Sedang	Rendah	Tinggi	Tinggi	Rendah
37	Rendah	Sedang	Sedang	Rendah	Rendah	Tinggi
38	Rendah	Sedang	Sedang	Rendah	Sedang	Tinggi
39	Rendah	Sedang	Sedang	Rendah	Tinggi	Tinggi
40	Rendah	Sedang	Sedang	Sedang	Rendah	Tinggi
41	Rendah	Sedang	Sedang	Sedang	Sedang	Tinggi
42	Rendah	Sedang	Sedang	Sedang	Tinggi	Tinggi
43	Rendah	Sedang	Sedang	Tinggi	Rendah	Rendah
44	Rendah	Sedang	Sedang	Tinggi	Sedang	Rendah
45	Rendah	Sedang	Sedang	Tinggi	Tinggi	Rendah
46	Rendah	Sedang	Tinggi	Rendah	Rendah	Tinggi
47	Rendah	Sedang	Tinggi	Rendah	Sedang	Tinggi
48	Rendah	Sedang	Tinggi	Rendah	Tinggi	Tinggi
49	Rendah	Sedang	Tinggi	Sedang	Rendah	Rendah
50	Rendah	Sedang	Tinggi	Sedang	Sedang	Tinggi
51	Rendah	Sedang	Tinggi	Sedang	Tinggi	Tinggi
52	Rendah	Sedang	Tinggi	Tinggi	Rendah	Rendah
53	Rendah	Sedang	Tinggi	Tinggi	Sedang	Rendah

54	Rendah	Sedang	Tinggi	Tinggi	Tinggi	Tinggi
55	Rendah	Tinggi	Rendah	Rendah	Rendah	Tinggi
56	Rendah	Tinggi	Rendah	Rendah	Sedang	Tinggi
57	Rendah	Tinggi	Rendah	Rendah	Tinggi	Tinggi
58	Rendah	Tinggi	Rendah	Sedang	Rendah	Rendah
59	Rendah	Tinggi	Rendah	Sedang	Sedang	Tinggi
60	Rendah	Tinggi	Rendah	Sedang	Tinggi	Tinggi
61	Rendah	Tinggi	Rendah	Tinggi	Rendah	Rendah
62	Rendah	Tinggi	Rendah	Tinggi	Sedang	Rendah
63	Rendah	Tinggi	Rendah	Tinggi	Tinggi	Tinggi
64	Rendah	Tinggi	Sedang	Rendah	Rendah	Tinggi
65	Rendah	Tinggi	Sedang	Rendah	Sedang	Tinggi
66	Rendah	Tinggi	Sedang	Rendah	Tinggi	Tinggi
67	Rendah	Tinggi	Sedang	Sedang	Rendah	Tinggi
68	Rendah	Tinggi	Sedang	Sedang	Sedang	Tinggi
69	Rendah	Tinggi	Sedang	Sedang	Tinggi	Tinggi
70	Rendah	Tinggi	Sedang	Tinggi	Rendah	Rendah
71	Rendah	Tinggi	Sedang	Tinggi	Sedang	Rendah
72	Rendah	Tinggi	Sedang	Tinggi	Tinggi	Tinggi
73	Rendah	Tinggi	Tinggi	Rendah	Rendah	Tinggi
74	Rendah	Tinggi	Tinggi	Rendah	Sedang	Tinggi
75	Rendah	Tinggi	Tinggi	Rendah	Tinggi	Tinggi
76	Rendah	Tinggi	Tinggi	Sedang	Rendah	Rendah
77	Rendah	Tinggi	Tinggi	Sedang	Sedang	Rendah
78	Rendah	Tinggi	Tinggi	Sedang	Tinggi	Tinggi
79	Rendah	Tinggi	Tinggi	Tinggi	Rendah	Rendah
80	Rendah	Tinggi	Tinggi	Tinggi	Sedang	Rendah
81	Rendah	Tinggi	Tinggi	Tinggi	Tinggi	Tinggi

Tabel 3.8 Perhitungan inferensi *fuzzy* Tsukamoto kriteria negatif

Rule	Gaji	Listrik	Telepon	PDAM	PBB	α_p	z	$(\alpha_p)*z$
1	0.09	0.21	0	1	1	0	3	0
2	0.09	0.21	0	1	0	0	3	0
3	0.09	0.21	0	1	0	0	3	0
4	0.09	0.21	0	0	1	0	3	0
5	0.09	0.21	0	0	0	0	3	0
6	0.09	0.21	0	0	0	0	3	0
7	0.09	0.21	0	0	1	0	3	0
8	0.09	0.21	0	0	0	0	3	0
9	0.09	0.21	0	0	0	0	3	0
10	0.09	0.21	0	1	1	0	3	0
11	0.09	0.21	0	1	0	0	1	0
12	0.09	0.21	0	1	0	0	1	0
13	0.09	0.21	0	0	1	0	3	0
14	0.09	0.21	0	0	0	0	1	0
15	0.09	0.21	0	0	0	0	1	0
16	0.09	0.21	0	0	1	0	3	0
17	0.09	0.21	0	0	0	0	3	0
18	0.09	0.21	0	0	0	0	3	0
19	0.09	0.21	1	1	1	0.09	2.82	0.2538
20	0.09	0.21	1	1	0	0	3	0
21	0.09	0.21	1	1	0	0	3	0
22	0.09	0.21	1	0	1	0	3	0
23	0.09	0.21	1	0	0	0	3	0
24	0.09	0.21	1	0	0	0	3	0
25	0.09	0.21	1	0	1	0	3	0
26	0.09	0.21	1	0	0	0	3	0
27	0.09	0.21	1	0	0	0	3	0
28	0.09	0.79	0	1	1	0	1	0
29	0.09	0.79	0	1	0	0	1	0

30	0.09	0.79	0	1	0	0	1	0
31	0.09	0.79	0	0	1	0	3	0
32	0.09	0.79	0	0	0	0	3	0
33	0.09	0.79	0	0	0	0	1	0
34	0.09	0.79	0	0	1	0	3	0
35	0.09	0.79	0	0	0	0	3	0
36	0.09	0.79	0	0	0	0	3	0
37	0.09	0.79	0	1	1	0	1	0
38	0.09	0.79	0	1	0	0	1	0
39	0.09	0.79	0	1	0	0	1	0
40	0.09	0.79	0	0	1	0	1	0
41	0.09	0.79	0	0	0	0	1	0
42	0.09	0.79	0	0	0	0	1	0
43	0.09	0.79	0	0	1	0	3	0
44	0.09	0.79	0	0	0	0	3	0
45	0.09	0.79	0	0	0	0	3	0
46	0.09	0.79	1	1	1	0.09	1.18	0.1062
47	0.09	0.79	1	1	0	0	1	0
48	0.09	0.79	1	1	0	0	1	0
49	0.09	0.79	1	0	1	0	3	0
50	0.09	0.79	1	0	0	0	1	0
51	0.09	0.79	1	0	0	0	1	0
52	0.09	0.79	1	0	1	0	3	0
53	0.09	0.79	1	0	0	0	3	0
54	0.09	0.79	1	0	0	0	1	0
55	0.09	0	0	1	1	0	1	0
56	0.09	0	0	1	0	0	1	0
57	0.09	0	0	1	0	0	1	0
58	0.09	0	0	0	1	0	3	0
59	0.09	0	0	0	0	0	1	0

60	0.09	0	0	0	0	0	1	0	
61	0.09	0	0	0	1	0	3	0	
62	0.09	0	0	0	0	0	3	0	
63	0.09	0	0	0	0	0	1	0	
64	0.09	0	0	1	1	0	1	0	
65	0.09	0	0	1	0	0	1	0	
66	0.09	0	0	1	0	0	1	0	
67	0.09	0	0	0	1	0	1	0	
68	0.09	0	0	0	0	0	1	0	
69	0.09	0	0	0	0	0	1	0	
70	0.09	0	0	0	1	0	3	0	
71	0.09	0	0	0	0	0	3	0	
72	0.09	0	0	0	0	0	1	0	
73	0.09	0	1	1	1	0	1	0	
74	0.09	0	1	1	0	0	1	0	
75	0.09	0	1	1	0	0	1	0	
76	0.09	0	1	0	1	0	3	0	
77	0.09	0	1	0	0	0	3	0	
78	0.09	0	1	0	0	0	1	0	
79	0.09	0	1	0	1	0	3	0	
80	0.09	0	1	0	0	0	3	0	
81	0.09	0	1	0	0	0	1	0	
						$\sum \alpha_{p_i} =$	0.18	$\sum (\alpha_{p_i} * z_i) =$	0.36

Penjelasan dari proses perhitungan inferensi *fuzzy* Tsukamoto untuk rule 19 kriteria negatif, yaitu:

1. Nilai derajat keanggotaan Penghasilan orang tua Rendah

$$\mu_{gajiRendah}(2.2) = \frac{2.3 - x}{2.3 - 1.2} = \frac{2.3 - 2.2}{2.3 - 1.2} = 0.09$$

2. Nilai derajat keanggotaan Tagihan Listrik Rendah

$$\mu_{listRendah}(6.48) = \frac{7 - x}{7 - 4.5} = \frac{7 - 6.48}{7 - 4.5} = 0.21$$

3. Nilai derajat keanggotaan Tagihan telepon Tinggi

$$\mu_{telpTinggi}(15) = 1$$

4. Nilai derajat keanggotaan Tagihan PDAM Rendah

$$\mu_{pdamRendah}(0) = 1$$

5. Nilai derajat keanggotaan Tagihan PBB Rendah

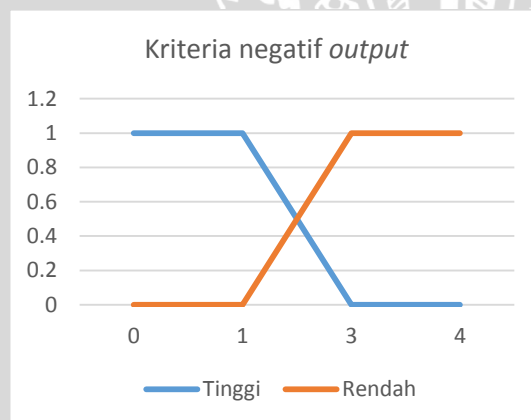
$$\mu_{pbbRendah}(3.17) = 1$$

6. Nilai $\alpha_{predikat}$ atau nilai hasil inferensi

$$\alpha_{p19} = \mu_{gajiRendah} \cap \mu_{listRendah} \cap \mu_{telpTinggi} \cap \mu_{pdamRendah} \\ \cap \mu_{pbbRendah}$$

$$\alpha_{p19} = \min(\mu_{PKSedang}, \mu_{listRendah}, \mu_{telpTinggi}, \mu_{pdamRendah}, \mu_{pbbRendah}) \\ = \min(0.09 ; 0.21 ; 1 ; 1 ; 1) = 0.09$$

7. Nilai *crisp* himpunan *output*, nilai ini didapat berdasarkan grafik himpunan *fuzzy output* untuk kriteria negatif



$$\mu_{positifRendah}(z) = \frac{3 - z}{3 - 1}$$

$$0.09 = \frac{3 - z}{2}$$

$$0.18 = 3 - z$$

$$z = 2.82$$

8. Nilai ($\alpha_{predikat} * z$)

$$\alpha_{p19} \times z_{19} = 0.09 \times 2.82 = 0.2538$$

Hasil akhir dari inferensi *fuzzy* Tsukamoto merupakan nilai *output* kriteria negatif yang dihitung menggunakan persamaan defuzzifikasi 2-6. Nilai *output* dari kriteria negatif akan digunakan sebagai nilai *input* untuk inferensi *fuzzy* tahap kedua.

$$Z_{negatif} = \frac{\sum(\alpha_{p_i} \times z_i)}{\sum \alpha_{p_i}} = \frac{0.36}{0.18} = 2$$

Selanjutnya, perhitungan terhadap α -predikat, nilai *crisp* dari kriteria himpunan *fuzzy output* (z), dan (α -predikat* z). Hasil nilai *output* pada kriteria positif sebesar 1.5 dan nilai *output* kriteria negatif sebesar 2. Kedua nilai *output* tersebut akan digunakan untuk proses inferensi *fuzzy* Tsukamoto tahap kedua. Jumlah *rules* pada inferensi *fuzzy* tahap kedua sebanyak 4 *rules*. Perhitungan pada proses inferensi *fuzzy* ditampilkan pada Tabel 3.9 dan 3.7.

Tabel 3.9 Basis aturan inferensi *fuzzy* tahap kedua

Rule	Kriteria Positif	Kriteria Negatif	Keputusan Beasiswa
1	Rendah	Rendah	Ditolak
2	Rendah	Tinggi	Ditolak
3	Tinggi	Rendah	Diterima
4	Tinggi	Tinggi	Diterima

Tabel 3.10 Proses perhitungan inferensi *fuzzy* Tsukamoto tahap 2

Rule	Positif	Negatif	α_p	z	$(\alpha_p)*z$
1	0.75	0.5	0.5	6.5	3.25
2	0.75	0.5	0.5	6.5	3.25
3	0.25	0.5	0.25	5.25	1.3125
4	0.25	0.5	0.25	5.25	1.3125

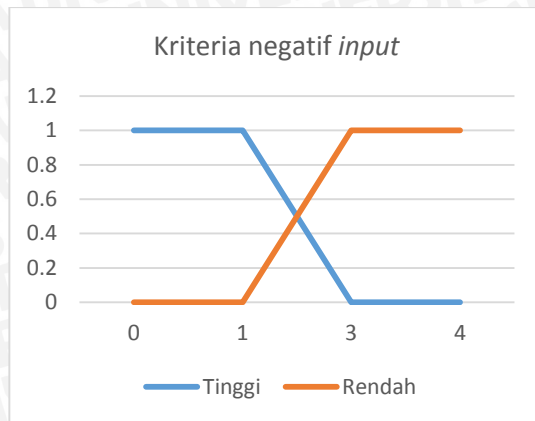
Penjelasan dari proses perhitungan inferensi *fuzzy* Tsukamoto tahap 2 untuk rule 1, yaitu:

1. Nilai derajat keanggotaan kriteria positif Rendah



$$\begin{aligned} \mu_{\text{posRendah}}(1.5) &= \frac{3 - x}{3 - 1} \\ &= \frac{3 - 1.5}{3 - 1} \\ &= 0.75 \end{aligned}$$

2. Nilai derajat keanggotaan kriteria negatif Rendah



$$\begin{aligned} \mu_{negRendah}(2) &= \frac{x - 1}{3 - 1} \\ &= \frac{2 - 1}{3 - 1} \\ &= 0.5 \end{aligned}$$

3. Nilai $\alpha_{predikat}$ atau nilai hasil inferensi

$$\begin{aligned} \alpha_{p1} &= \mu_{posRendah} \cap \mu_{negRendah} \\ \alpha_{p1} &= \min(\mu_{posRendah}, \mu_{negRendah}) \\ &= \min(0.75; 0.5) = 0.5 \end{aligned}$$

4. Nilai *crisp* himpunan *output* Keputusan Beasiswa Ditolak

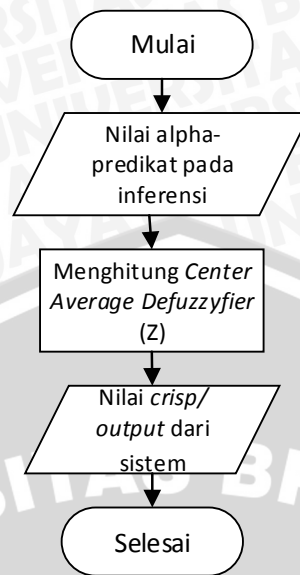
$$\begin{aligned} \mu_{beasiswaTolak}(z) &= \frac{9 - z}{9 - 4} \\ 0.5 &= \frac{9 - z}{5} \\ 2.5 &= 9 - z \\ z &= 6.5 \end{aligned}$$

5. Nilai ($\alpha_{predikat} * z$)

$$\alpha_{p1} \times z_1 = 0.5 \times 6.5 = 3.25$$

3.4.4 Defuzzifikasi

Defuzzifikasi merupakan proses perhitungan dimana mencari nilai *output* dari sistem dengan merubah nilai derajat keanggotaan menjadi nilai *crisp* (Robandi, 2006). Pada permasalahan optimasi fungsi keanggotaan *fuzzy* Tsukamoto untuk menentukan calon penerima beasiswa, proses defuzzifikasi menggunakan defuzzifikasi rata-rata terpusat (*Center Average Defuzzifier*) yang dinyatakan dalam persamaan 2-6. Diagram alir dari proses defuzzifikasi ditunjukkan pada Gambar 3.6.



Gambar 3.6 Diagram alir proses defuzzifikasi

Defuzzifikasi rata-rata terpusat merupakan pembagian antara jumlah α -predikat* z dan jumlah α -predikat, yang telah dijelaskan pada persamaan 2-6. Hasil dari perhitungan defuzzifikasi rata-rata terpusat adalah sebuah keputusan seleksi untuk calon penerima beasiswa. Contoh perhitungan defuzzifikasi rata-rata terpusat pada contoh permasalahan ini, yaitu:

- a. Jumlah dari nilai α -predikat

$$\sum \alpha_{p_i} = \alpha_{p_1} + \alpha_{p_2} + \alpha_{p_3} + \alpha_{p_4} = 1.5$$

- b. Jumlah dari α -predikat* z

$$\begin{aligned} \sum(\alpha_{p_i} * z_i) &= (\alpha_{p_1} * z_1) + (\alpha_{p_2} * z_2) + (\alpha_{p_3} * z_3) + (\alpha_{p_4} * z_4) \\ &= 9.125 \end{aligned}$$

- c. Defuzzifikasi rata-rata terpusat (*Center Average Defuzzifier*)

$$Z = \frac{\sum(\alpha_{p_i} * z_i)}{\sum \alpha_{p_i}} = \frac{9.125}{1.5} = 6.08$$

Jadi, nilai bobot untuk contoh permasalahan pada Tabel 3.1 sebesar 6.08.

3.4.5 Akurasi Sistem

Sistem inferensi fuzzy Tsukamoto dalam penerapannya tidak selalu menghasilkan hasil akhir keputusan yang akurat. Maka dari itu, nilai Z dari proses defuzzifikasi akan digunakan untuk perhitungan akurasi sistem

inferensi *fuzzy* Tsukamoto. Akurasi sistem dihitung menggunakan rumus korelasi *spearman* yang dinyatakan dalam persamaan 3-1.

$$r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n^3 - n} \quad (3-1)$$

Dimana r_s adalah koefisien kolerasi *spearman*, d_i merupakan selisih antara *ranking* yang dihasilkan dari inferensi *fuzzy* (Z) dengan pendapat pakar, n merupakan banyaknya data. Perhitungan nilai akurasi untuk sistem inferensi *fuzzy*, yaitu:

- Dimisalkan terdapat contoh kasus 5 mahasiswa calon penerima beasiswa dengan kriteria seperti pada Tabel 3.11.

Tabel 3.11 Contoh kasus data 5 mahasiswa calon penerima beasiswa

Mahasiswa	IPK	Tanggungsan Ortu	Penghasilan Ortu
A	3.25	6	4.7
B	2.75	2	1
C	3.6	3.75	1
D	3.1	5	4.5
E	3.1	4.5	3.3

- Hitung nilai defuzzifikasi rata-rata terpusat (Z) untuk masing-masing data mahasiswa dengan menggunakan batas-batas fungsi keanggotaan pada sub-bab 3.4.2. Perhitungan inferensi *fuzzy* Tsukamoto yang digunakan adalah perhitungan pada bab 3.4. Hasil nilai Z masing-masing mahasiswa ditunjukkan pada Tabel 3.12.

Tabel 3.12 Hasil perhitungan nilai Z menggunakan FIS Tsukamoto

Mahasiswa	Nilai Z (x)	Ranking (x)
A	5.41	5
B	6.02	4
C	9	1
D	6.3	2
E	6.16	3

- Hitung koefisien *spearman* rank dari nilai Z menggunakan persamaan 3-1. *Ranking* pendapat pakar didapat berdasarkan perhitungan inferensi

fuzzy Tsukamoto dengan menggunakan batas-batas fungsi keanggotaan yang berasal dari pakar.

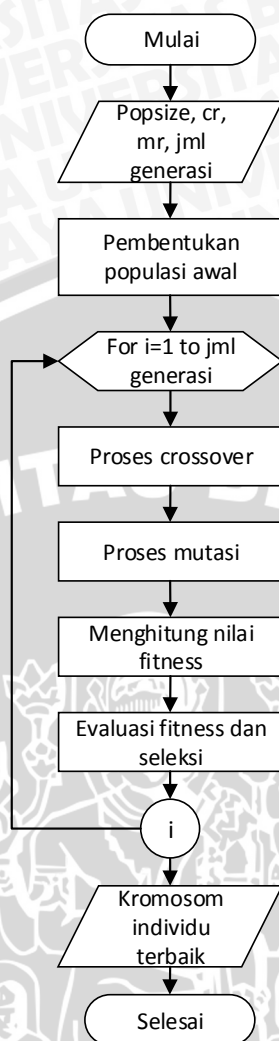
Mahasiswa	Ranking hasil fuzzy (x)	Pendapat Pakar (y)	$d_i = x_i - y_i$	d_i^2
A	5	4	1	1
B	4	5	-1	1
C	1	1	0	0
D	2	3	-1	1
E	3	2	1	1
$\sum d_i^2 = 4$				

$$r_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n^3 - n} = 1 - \frac{6 \times 4}{5^3 - 5} = 0.76$$

Hasil dari perhitungan menggunakan rumus korelasi spearman diketahui bahwa r_s sebesar 0.76, dimana korelasi atau hubungan hasil FIS Tsukamoto dengan pakar hampir sama. Jika r_s bernilai 1 maka korelasi atau hubungan sistem berbasis FIS Tsukamoto dengan pakar bersifat sama.

3.5 Siklus Penyelesaian Masalah Menggunakan Algoritma Genetika

Penggunaan sistem inferensi *fuzzy* yang telah ditunjukkan pada sub-bab 3.4 memerlukan batas-batas fungsi keanggotaan himpunan *fuzzy* yang sesuai agar didapatkan hasil yang akurat. Batas-batas tersebut bisa ditentukan secara otomatis dengan menggunakan algoritma genetika. Algoritma genetika dalam penerapannya memiliki beberapa tahap untuk menyelesaikan masalah, yaitu inisialiasi populasi, menghitung nilai *fitness*, proses reproduksi, dan seleksi. Proses reproduksi sendiri terdiri dari proses *crossover* dan mutasi. Penjelasan perancangan sistem atau algoritma genetika dijelaskan pada diagram alir Gambar 3.6.



Gambar 3.7 Diagram alir Algoritma Genetika

Langkah awal dari proses perhitungan algoritma genetika untuk optimasi fungsi keanggotaan *fuzzy* Tsukamoto adalah menentukan nilai dari parameter atau variabel algoritma genetika, seperti ukuran populasi (*popSize*), nilai *crossover rate* dan *mutation rate*. Contoh permasalahan dari proses perhitungan algoritma genetika dimisalkan seperti pada Tabel 3.13.

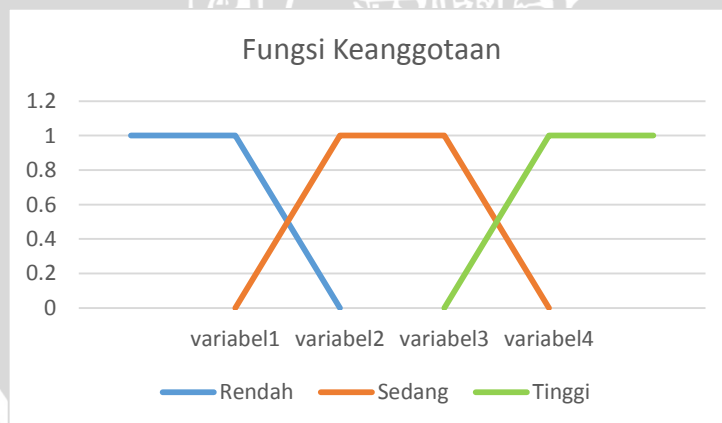
Tabel 3.13 Contoh permasalahan perhitungan Algoritma Genetika

Kriteria	Tanggungan dan penghasilan orang tua
Ukuran populasi	3
Nilai cr	0.5
Nilai mr	0.2
Generasi	1

3.5.1 Representasi Kromosom dan Perhitungan Fitness

Optimasi fungsi keanggotaan untuk logika *fuzzy* pada penelitian ini menggunakan metode algoritma genetika. Algoritma genetika akan membentuk suatu populasi. Populasi yang dibentuk terdiri dari beberapa individu atau kromosom untuk menghasilkan sebuah solusi dari permasalahan. Pada optimasi fungsi keanggotaan *fuzzy* Tsukamoto untuk menentukan calon penerima beasiswa digunakan representasi kromosom pengkodean real (*real-coded*) (Mahmudy, 2013). Alasan penggunaan representasi kromosom ini dikarenakan bilangan yang digunakan dalam batas-batas fungsi keanggotaan *fuzzy* merupakan bilangan pecahan dan bulat.

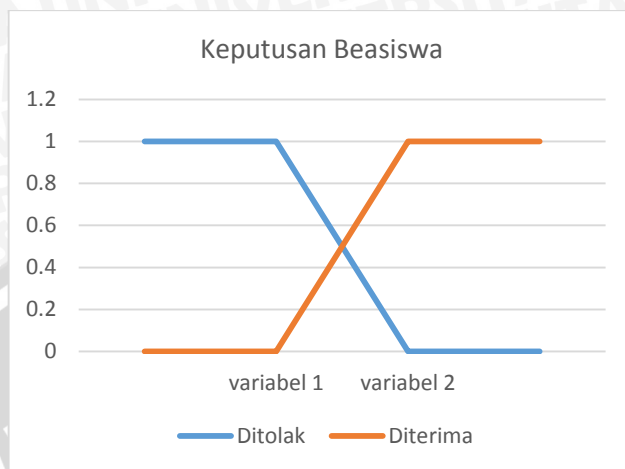
Setiap kromosom terdiri dari gen yang memuat bilangan real. Kromosom inisial dibangkitkan secara random dan bilangan random tersebut memiliki rentang nilai sesuai dengan masing-masing kriteria penentuan calon penerima beasiswa atau fungsi keanggotaan yang ditunjukkan pada Tabel 3.1. Kromosom yang dibentuk memiliki jumlah gen sebanyak 30. Kromosom dibentuk berdasarkan penggabungan antara nilai batas-batas fungsi keanggotaan dari semua kriteria *input* dan *output fuzzy*. Grafik fungsi keanggotaan kriteria *input fuzzy* pada Gambar 3.8.



Gambar 3.8 Grafik fungsi keanggotaan kriteria *input fuzzy*

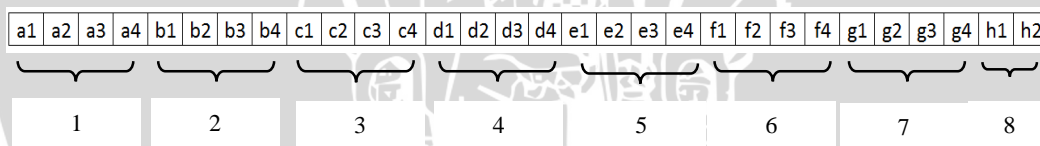
Grafik fungsi keanggotaan *fuzzy* pada Gambar 3.8 menggunakan representasi linear naik, trapesium, dan turun. Kriteria yang menggunakan grafik keanggotaan pada Gambar 3.8, antara lain IPK, penghasilan orang tua, tanggungan orang tua, tagihan telepon, tagihan listrik, tagihan PDAM, dan

pembayaran PBB. Sedangkan untuk grafik keputusan beasiswa menggunakan grafik fungsi keanggotaan seperti pada Gambar 3.9.



Gambar 3.9 Grafik fungsi keanggotaan *output fuzzy*

Pembentukan kromosom terdiri dari variabel-variabel pada fungsi keanggotaan. Variabel pada setiap kriteria berbeda-beda, dimisalkan untuk kriteria penghasilan orang tua diberikan variabel a , tanggungan orang tua diberikan variabel b , dan seterusnya. Penjelasan pembentukan kromosom ditunjukkan pada Gambar 3.10.



Gambar 3.10 Representasi kromosom *real-coded* optimasi fungsi keanggotaan *fuzzy*

Keterangan Gambar 3.10, adalah:

1. Segmen gen kriteria IPK
2. Segmen gen kriteria Penghasilan Orang Tua
3. Segmen gen kriteria Tanggungan Orang Tua
4. Segmen gen kriteria Tagihan Telepon
5. Segmen gen kriteria Tagihan Listrik
6. Segmen gen kriteria Tagihan PDAM
7. Segmen gen kriteria Pembayaran PBB
8. Segmen gen kriteria Keputusan Beasiswa

Pada contoh dimisalkan representasi kromosom hanya terdapat 2 kriteria, yaitu tanggungan dan penghasilan orang tua dimana masing-masing kriteria memiliki 4 gen. Contoh representasi kromosom dengan menggunakan pengkodean *real* ditunjukkan pada Gambar 3.11.

P ₁	2.5	4	3.5	1	3	1.5	0.8	5
----------------	-----	---	-----	---	---	-----	-----	---

Gambar 3.11 Contoh pembangkitan populasi awal

Setelah pembangkitkan kromosom secara random, nilai gen pada kromosom diurutkan secara *ascending*. Nilai gen akan diurutkan setiap segmen yang terdiri dari 4 gen. Hasil pengurutan kromosom seperti pada Gambar 3.12.

P ₁	1	2.5	3.5	4	0.8	1.5	3	5
----------------	---	-----	-----	---	-----	-----	---	---

Gambar 3.12 Contoh pembangkitan populasi awal setelah diurutkan

Selanjutnya dalam tahapan algoritma genetika terdapat perhitungan nilai *fitness* dari masing-masing kromosom atau individu yang telah dibentuk. Solusi yang dicari dalam permasalahan kasus optimasi batas-batas fungsi keanggotaan sistem inferensi *fuzzy* berupa solusi maksimal, maka perhitungan nilai *fitness* (f) untuk contoh dinyatakan dalam persamaan 3-2.

$$f = r_s \tag{3-2}$$

Dimana r_s merupakan koefisien *spearman* dan f merupakan nilai *fitness*. Perhitungan nilai r_s pada kromosom P₁ sama dengan perhitungan yang telah dilakukan sebelumnya pada sub-bab 3.4.5. Sehingga, kromosom P₁ memiliki nilai *fitness* yang ditunjukkan pada Gambar 3.13.

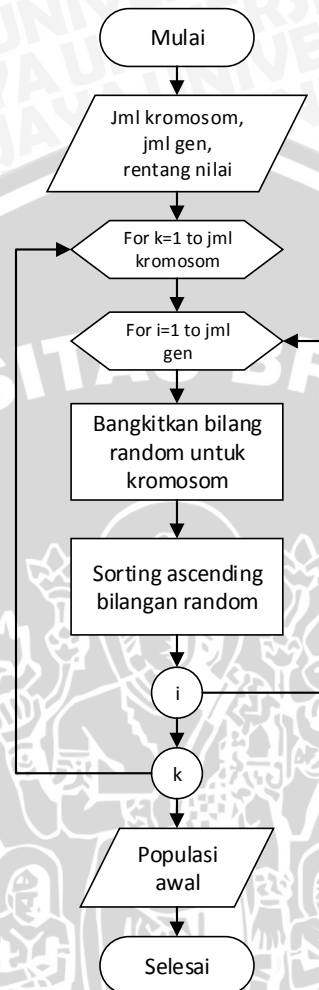
Parent	Kromosom	r_s	Fitness
P ₁	[1, 2.5, 3.5, 4, 0.8, 1.5, 3, 5]	-0.4	-0.4

Gambar 3.13 Hasil perhitungan nilai *fitness* pada kromsom P₁

3.5.2 Inisialisasi Populasi Awal

Populasi pada algoritma genetika yang dibentuk untuk permasalahan optimasi fungsi keanggotaan terdiri dari gen yang merupakan gabungan antara

nilai untuk masing-masing kriteria penentuan calon penerima beasiswa. Penjelasan proses inialisasi populasi awal ditunjukkan pada Gambar 3.14.



Gambar 3.14 Diagram alir proses inialisasi populasi awal

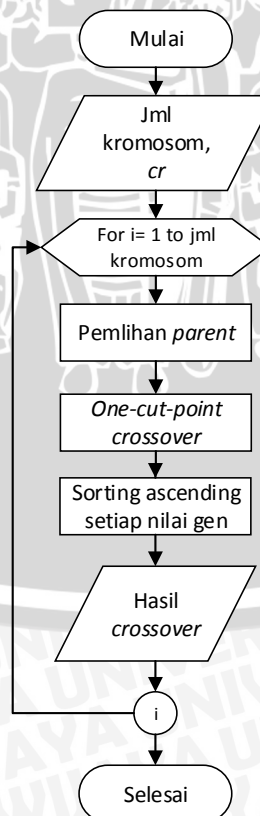
Pembangkitkan populasi awal dilakukan secara random, dari masing-masing kromosom memiliki rentang nilai gen sesuai dengan Tabel 3.1. Selanjutnya, pembentukan kromosom telah dijelaskan pada Gambar 3.10 dimana setiap individu memiliki 30 gen dari masing-masing kriteria penentuan calon penerima beasiswa. Pembangkitan populasi awal dari contoh permasalahan ini dibangkitkan sebanyak 3 individu atau *parent* dimana nilai gen pada kromosom telah diurutkan secara *ascending*. Populasi awal yang dibangkitkan telah dihitung nilai *fitness* dengan perhitungan yang sama pada sub-bab 3.5.1. Pada Tabel 3.12 ditunjukkan hasil pembangkitan populasi awal beserta nilai *fitness*-nya.

Tabel 3.14 Pembangkitan populasi awal

Individu	Kromosom	Fitness
P ₁	[1, 2.5, 3.5, 4, 0.8, 1.5, 3, 5]	-0.4
P ₂	[0.7, 2, 3.6, 4.5, 0.5, 2.3, 3, 4.2]	0.9
P ₃	[1.3, 3, 4.2, 5, 0.6, 1, 2.8, 4.5]	0.7

3.5.3 Reproduksi

Proses reproduksi dalam algoritma genetika terdiri dari proses *crossover* dan proses mutasi. Metode *crossover* yang digunakan adalah *one-cut-point crossover*. Seperti penjelasan sebelumnya pada bab 2.4, proses *crossover* akan menghasilkan *offspring* dimana jumlah *offspring* didapat berdasarkan dari rumus $offspring = cr \times popSize$. Variabel *cr* merupakan *crossover rate* yang telah ditentukan sebelumnya dengan rentang nilai [0, 1]. Sedangkan, variabel *popSize* merupakan ukuran populasi yang dibentuk. Proses dari metode *one-cut-point crossover* dijelaskan pada diagram alir Gambar 3.15.



Gambar 3.15 Diagram alir proses *one-cut-point crossover*

Dalam metode *one-cut-point crossover* dilakukan dengan memilih titik potong pada kromosom kemudian menukarkan nilai gen *parent* satu dengan *parent* lainnya (Mahmudy, 2013). Proses *one-cut-point crossover* untuk contoh permasalahan ini dihasilkan *offspring* sebanyak $offspring = cr \times popSize = 0.5 \times 3 = 1.5 = 2$ *offspring*. Langkah-langkah proses *crossover*, yaitu:

- Titik potong *crossover* ditentukan secara acak pada gen ke-1, 2, 5, dan 6 untuk P_1 dan P_2 . Gen yang terpilih dipilih secara random untuk masing-masing segmen kromosom dimana segmen pertama terdiri dari 4 gen pertama untuk fungsi keanggotaan tanggungan orang tua. Sedangkan, segmen kedua terdiri dari 4 gen selanjutnya untuk fungsi keanggotaan penghasilan orang tua.

P_1	1	2.5	3.5	4	0.8	1.5	3	5
-------	---	-----	-----	---	-----	-----	---	---

P_2	0.7	2	3.6	4.5	0.5	2.3	3	4.2
-------	-----	---	-----	-----	-----	-----	---	-----

- Tukar nilai gen P_1 dan P_2 sesuai dengan titik potong yang telah ditentukan sebagai hasil dari *crossover*, *offspring*.

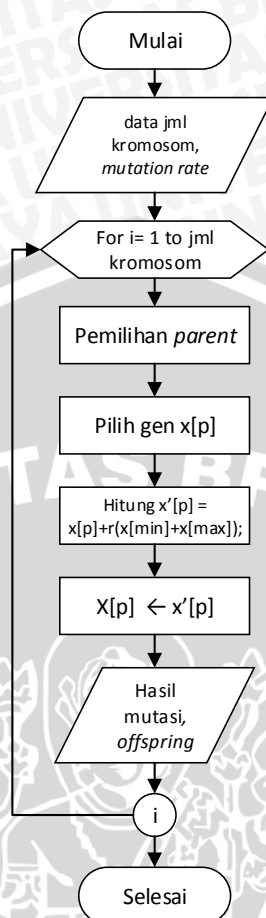
C_1	0.7	2	3.5	4	0.5	2.3	3	5
-------	-----	---	-----	---	-----	-----	---	---

C_2	1	2.5	3.6	4.5	0.8	1.5	3	4.2
-------	---	-----	-----	-----	-----	-----	---	-----

Proses reproduksi mutasi representasi kromosom *real-coded* menggunakan metode *random mutation*. Nilai gen yang terpilih pada kromosom *parent* akan dihitung dengan menggunakan persamaan 3-3 (Mahmudy, 2013).

$$x'_i = x'_i + r (max_j - min_j) \quad (3-3)$$

Dimana x'_i merupakan nilai gen dari *parent* yang terpilih, r adalah bilangan random dengan rentang nilai $[-0,1 \ 0,1]$, max_j, min_j merupakan batasan nilai dari gen yang terpilih, i dan j adalah bilangan bulat positif yang merepresentasikan jumlah gen yang terpilih (Mahmudy, 2013). Diagram alir dari proses *random mutation* ditunjukkan pada Gambar 3.16.



Gambar 3.16 Diagram alir proses *random mutation*

Jumlah anak (*offspring*) pada permasalahan ini sebanyak $offspring = mr \times popSize = 0.1 \times 3 = 0.3 = 1 \text{ offspring}$. Penjelasan dari langkah-langkah *random mutation*, yaitu:

- Parent* dipilih secara acak untuk dilakukan proses *random mutation*, pada contoh ini gen terpilih adalah ke-3 dan 7. Setiap 4 segmen yang terpilih pada kromosom mewakili fungsi keanggotaan *fuzzy* masing-masing, seperti yang telah dijelaskan pada proses *crossover*. Gen yang terpilih dipilih secara random.

P_3

1.3	3	4.2	5	0.6	1	2.8	4.5
-----	---	-----	---	-----	---	-----	-----

- Hitung nilai gen terpilih menggunakan rumus atau persamaan 3-2. Nilai max_j, min_j dapat dilihat pada Tabel 3.1 dengan kriteria tanggungan dan penghasilan orang tua. Nilai r misal ditentukan sebesar 0.035.

- Gen ke-3: $x'_3 = x'_3 + r (max_3 - min_3) = 4.2 + 0.035 (5 - 0) = 4.375$
- Gen ke-7: $x'_7 = x'_7 + r (max_7 - min_7) = 2.8 + 0.035 (5 - 0) = 2.975$

c. Hasil dari *random mutation*.

C_3	1.3	3	4.375	5	0.6	1	2.975	4.5
-------	-----	---	-------	---	-----	---	-------	-----

Jadi, total *offspring* yang dihasilkan dari proses reproduksi sebanyak 3 *offspring*.

3.5.4 Evaluasi

Setelah proses reproduksi, akan dilakukan tahap evaluasi. Pada tahap evaluasi *offspring* akan dihitung nilai *fitness*nya sesuai dengan persamaan 3-1. Kromosom atau individu yang memiliki nilai *fitness* semakin besar maka semakin besar peluang kromosom menjadi sebuah solusi (Mahmudy, 2013). Pada Tabel 3.15 ditunjukkan hasil perhitungan nilai *fitness* sesuai dengan contoh perhitungan pada sub bab 3.5.1.

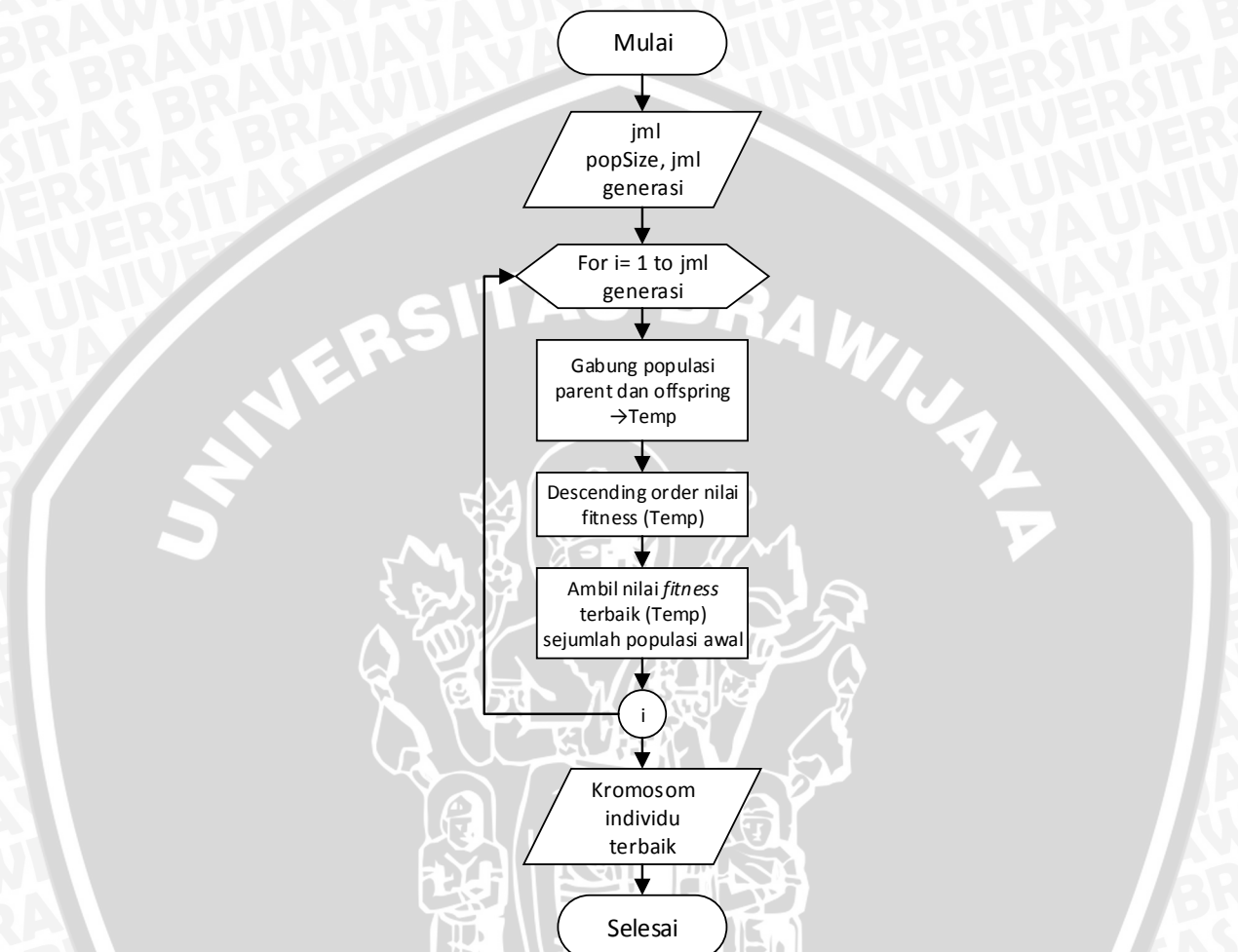
Tabel 3.15 Hasil perhitungan nilai *fitness* pada tahap evaluasi

Parent	Kromosom	Fitness
P ₁	[1, 2.5, 3.5, 4, 0.8, 1.5, 3, 5]	-0.4
P ₂	[0.7, 2, 3.6, 4.5, 0.5, 2.3, 3, 4.2]	0.9
P ₃	[1.3, 3, 4.2, 5, 0.6, 1, 2.8, 4.5]	0.7
Offspring	Kromosom	Fitness
C ₁	[0.64, 1.9, 3.5, 4, 0.44, 2.46, 3, 5]	0.5
C ₂	[1.06, 2.6, 3.6, 4.5, 0.86, 1.34, 3, 4.2]	0
C ₃	[1.3, 3, 4.375, 5, 0.6, 1, 2.975, 4.5]	0.7

3.5.5 Seleksi

Metode seleksi yang digunakan pada permasalahan optimasi fungsi keanggotan *fuzzy* adalah *elitism selection*. Pada metode *elitism selection*, populasi *parent* dan populasi *offspring* digabung menjadi satu populasi kemudian diurutkan berdasarkan nilai *fitness* tertinggi dan individu yang

terbaik dipilih sejumlah populasi awal dimana memiliki nilai *fitness* tertinggi (Mahmudy, 2013). Diagram alir dari proses *elitism selection* seperti pada Gambar 3.17.

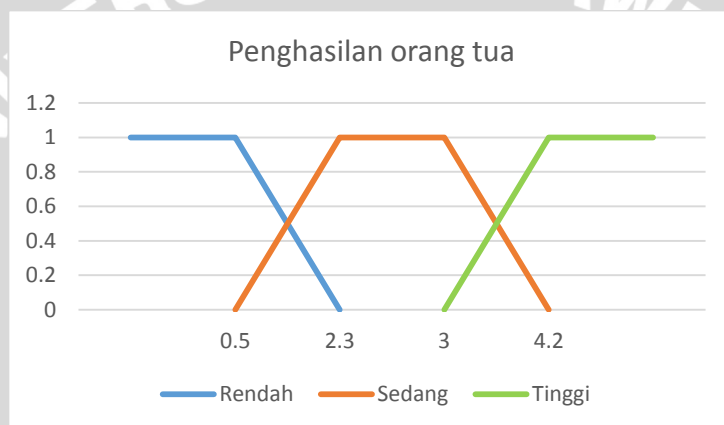


Gambar 3.17 Diagram alir proses *elitism selection*

Hasil akhir dari contoh permasalahan ini adalah individu P₂ merupakan solusi dari permasalahan optimasi fungsi keanggotaan *fuzzy* karena memiliki nilai *fitness* terbesar, sehingga fungsi keanggotaan dari kriteria tanggungan dan penghasilan orang tua dapat dilihat seperti pada Gambar 3.18 dan 3.19.



Gambar 3.18 Fungsi keanggotaan kriteria tanggungan orang tua



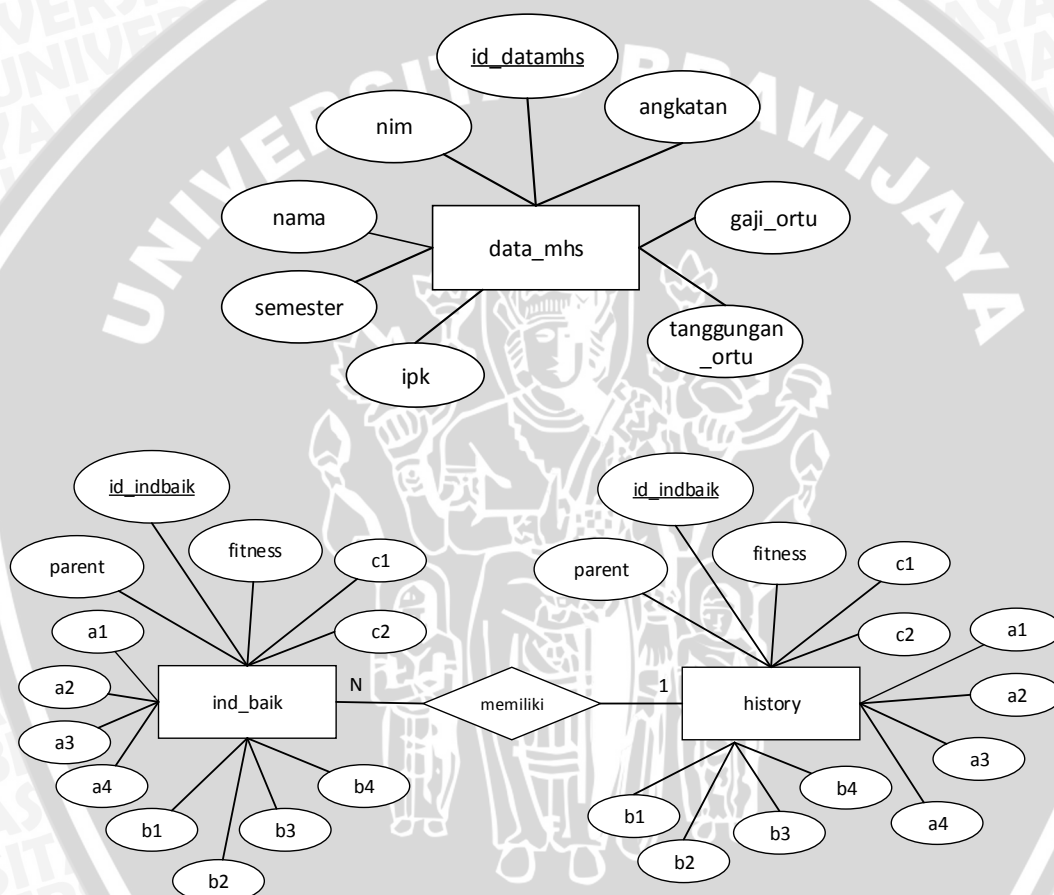
Gambar 3.19 Fungsi keanggotaan penghasilan orang tua



BAB IV PERANCANGAN

4.1 Perancangan Basis Data

Perancangan basis data yang terdapat dalam sistem optimasi fungsi keanggotaan *fuzzy* Tsukamoto memiliki 3 tabel, 2 tabel diantaranya saling berhubungan. Perancangan basi data ditunjukkan pada Gambar 4.1.



Gambar 4.1 Perancangan basis data algenetika

Tabel **data_mhs** pada Gambar 4.1 merupakan tabel yang menyimpan data mahasiswa sebanyak 30 mahasiswa. Data mahasiswa tersebut digunakan untuk menghitung pada sistem inferensi *fuzzy* Tsukamoto. Selanjutnya, pada Gambar 4.2 merupakan tabel **ind_baik** dan **history**. Tabel **ind_baik** merupakan tabel yang menyimpan semua nilai gen pada kromosom yang memiliki nilai *fitness* terbaik untuk setiap generasi. Sedangkan, pada tabel **history** memiliki struktur tabel yang

sama dengan *ind_baik*, yaitu menyimpan semua nilai gen pada kromosom yang memiliki nilai *fitness* terbaik tetapi hanya pada generasi paling terakhir.

4.2 Perancangan Antarmuka

Antarmuka atau *interface* merupakan penghubung antara sistem dan pengguna. Pada perancangan *interface* untuk penelitian ini, sistem terfokus pada pengoptimasian fungsi keanggotaan menggunakan metode algoritma genetika. Sehingga, *interface* yang akan dirancang untuk penyelesaian masalah menggunakan algoritma genetika. Rancangan *interface* untuk Sistem Optimasi Fungsi Keanggotaan Fuzzy Tsukamoto Menggunakan Algoritma Genetika pada Pemilihan Calon Penerima Beasiswa-PPA dan BBP-PA, antara lain:

1. Halaman *input* data

Perancangan untuk halaman *input* data parameter algoritma genetika, seperti ukuran populasi, jumlah generasi, nilai dari *crossover rate*-*mutation rate*.

Optimasi Fungsi Keanggotaan Fuzzy Menggunakan Metode Algoritma Genetika pada Penentuan Beasiswa

Jumlah Generasi:

Ukuran Populasi:

Crossover rate:

Mutation rate:

Hasil optimasi fungsi keanggotaan

Parent	Kromosom IPK	Penghasilan Ortu	Tanggungan Ortu	Tagihan PBB	Keputusan Beasiswa	Fitness
P1						
P2						
P3						
P4						
P5						

Gambar 4.2 Perancangan *interface input* data parameter Algoritma Genetika

Pada Gambar 4.3 perancangan *interface* halaman *input* data, *user* menginputkan semua parameter algoritma genetika. Setelah menginputkan parameter algoritma genetika, sistem akan melakukan proses algoritma sesuai dengan metode yang telah ditentukan dimana *user* menekan tombol

proses. Hasil dari optimasi fungsi keanggotaan akan ditampilkan oleh sistem berupa tabel disertai dengan hasil nilai *fitness*.

2. Halaman tampil individu terbaik

Perancangan antarmuka untuk halaman tampil individu terbaik ditunjukkan pada Gambar 4.4. Pada halaman ini akan ditampilkan satu individu terbaik dimana memiliki nilai *fitness* terbesar dari setiap generasi. Perubahan nilai *fitness* untuk setiap generasi dapat dilihat pada halaman individu terbaik.

Optimasi Fungsi Keanggotaan Fuzzy Menggunakan Metode Algoritma Genetika pada Penentuan Beasiswa									
Individu Terbaik dari proses Algoritma Genetika									
Generasi	Parent	IPK	Tanggungan Ortu	Piagam	Penghasilan Ortu	Tagihan PDAM	Tagihan PBB	Beasiswa	Nilai Fitness
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									

Gambar 4.3 Perancangan *interface* untuk halaman individu terbaik

3. Halaman perhitungan Fuzzy Tsukamoto

Perancangan *interface* pada Gambar 4.5 merupakan proses perhitungan menggunakan sistem inferensi metode *fuzzy* Tsukamoto dengan menggunakan batasan-batasan fungsi keanggotaan hasil dari proses Algoritma Genetika.

Optimasi Fungsi Keanggotaan Fuzzy Menggunakan Metode Algoritma Genetika pada Penentuan Beasiswa									
Penentuan Calon Penerima Beasiswa Menggunakan Hasil Optimasi Fungsi Keanggotaan									
Mahasiswa	IPK	Tanggung Ortu	Piagam	Penghasilan Ortu	Tagihan Listrik	Tagihan Telepon	Tagihan PDAM	Tagihan PBB	Nilai Bobot
Mahasiswa 1									
Mahasiswa 2									
Mahasiswa 3									
Mahasiswa 4									
Mahasiswa 5									
Mahasiswa 6									
Mahasiswa 7									
Mahasiswa 8									
Mahasiswa 9									
Mahasiswa 10									
Mahasiswa 11									

Gambar 4.4 Perancangan halaman perhitungan *fuzzy* Tsukamoto

Pada halaman ini *user* nantinya akan meng-*input*-kan nilai dari kriteria dalam penentuan calon penerima beasiswa. Proses perhitungan *fuzzy* Tsukamoto menggunakan batas-batas fungsi keanggotaan yang telah dioptimasi sebelumnya menggunakan algoritma genetika.

4.3 Perancangan Skenario Pengujian

Skenario pengujian digunakan untuk menguji parameter dari algoritma genetika sehingga dapat menghasilkan solusi yang terbaik. Parameter algoritma genetika yang diujikan antara lain *popSize*, *crossover rate* dan *mutation rate*, dan jumlah generasi. Selain pengujian terhadap parameter algoritma genetika, akan dilakukan suatu pengujian terhadap metode algoritma genetika. Pengujian ini berupa perbandingan sistem pendukung keputusan dengan menggunakan optimasi fungsi keanggotaan *fuzzy* dan sistem pendukung keputusan tanpa optimasi fungsi keanggotaan *fuzzy*. Penjelasan dari 3 buah skenario pengujian, yaitu:

1. Skenario Pengujian Ukuran Populasi

Pada skenario pengujian ini akan dilakukan pengujian terhadap ukuran populasi atau *popSize*. Ukuran populasi yang digunakan untuk pengujian adalah 20, 40, 60, 80, 100, 120, 140, 160, 180, dan 200. Pengujian ukuran populasi akan dilakukan untuk mengetahui ukuran populasi yang digunakan untuk pengujian berikutnya. Pada saat melakukan pengujian ukuran populasi, digunakan generasi sebanyak 100 generasi dan nilai pada

crossover rate-*mutation rate* sebesar 0.4 dan 0.2. Tabel perancangan skenario pengujian ukuran populasi ditunjukkan pada Tabel 4.1.

Tabel 4.1 Rancangan skenario pengujian ukuran populasi

Ukuran Populasi	cr = 0.4 ; mr = 0.2 ; generasi = 100					Rata-Rata Nilai <i>Fitness</i>
	Percobaan Ke - i					
	1	2	3	4	5	
20						
40						
60						
80						
100						
120						
140						
160						
180						
200						

2. Skenario Pengujian Kombinasi Nilai *cr* dan *mr*

Skenario pengujian kedua mengujikan *crossover rate* (*cr*) dan *mutation rate* (*mr*). Nilai variabel yang digunakan untuk pengujian *crossover rate* dan *mutation rate* adalah sebagai berikut 1.0 dan 0.0, 0.9 dan 0.1, 0.8 dan 0.2, 0.7 dan 0.3, 0.6 dan 0.4, 0.5 dan 0.5, 0.4 dan 0.6, 0.3 dan 0.7, 0.2 dan 0.8, 0.1 dan 0.9, serta 0.0 dan 1.0. Pada pengujian ini, ukuran populasi yang digunakan berdasarkan hasil dari pengujian sebelumnya. Sedangkan, banyaknya generasi adalah 100 generasi. Rancangan tabel skenario pengujian kombinasi nilai *cr* dan *mr* ditunjukkan pada Tabel 4.2.

Tabel 4.2 Rancangan skenario pengujian kombinasi nilai *cr* dan *mr*

Kombinasi nilai <i>cr</i> dan <i>mr</i>	popSize = ... ; generasi = 100					Rata-Rata Nilai <i>Fitness</i>
	Percobaan Ke - i					
	1	2	3	4	5	
1.0 ; 0						
0.9 ; 0.1						
0.8 ; 0.2						
0.7 ; 0.3						
0.6 ; 0.4						
0.5 ; 0.5						

0.4 ; 0.6						
0.3 ; 0.7						
0.2 ; 0.8						
0.1 ; 0.9						
0.0 ; 1.0						

3. Skenario Pengujian Jumlah Generasi

Skenario pengujian ketiga akan dilakukan pengujian terhadap jumlah generasi pada algoritma genetika. Pengujian ini akan mengujikan jumlah generasi, antara lain 20, 30, 40, 50, 60, 70, 80, 90, 100, dan 150 dengan menggunakan ukuran populasi dan *crossover rate-mutation rate* pada hasil pengujian sebelumnya. Tabel 4.3 ditampilkan sebagai rancangan tabel untuk skenario pengujian jumlah generasi.

Tabel 4.3 Rancangan skenario jumlah generasi

Jumlah generasi	<i>popSize = ... ; cr = ... ; mr = ...</i>					Rata-Rata Nilai <i>Fitness</i>
	Percobaan Ke - i					
	1	2	3	4	5	
20						
30						
40						
50						
60						
70						
80						
90						
100						
150						

BAB V IMPLEMENTASI SISTEM

5.1 Implementasi Basis Data

Basis data atau *database* yang terdapat dalam implementasi algoritma genetika untuk optimasi fungsi keanggotaan *fuzzy* Tsukamoto pada penerima calon beasiswa memiliki tiga tabel. Tabel dalam *database* berfungsi untuk menyimpan data mahasiswa calon penerima beasiswa. Data mahasiswa yang berjumlah sebanyak 30 mahasiswa digunakan untuk perhitungan sistem inferensi *fuzzy* Tsukamoto. Struktur *database* dalam implementasi ditunjukkan pada Tabel 5.1 dan 5.2.

Tabel 5.1 Struktur Tabel data_mhs

Nama database		algenetika		
Nama tabel		data_mhs		
Primary key		id_datamhs		
No	Nama Field	Tipe Data	Lebar	Keterangan
1	id_datamhs	int	15	Primary key
2	nim	Varchar	30	NIM mahasiswa
3	nama	Varchar	100	Nama mahasiswa
4	angkatan	int	11	Tahun angkatan mahasiswa
5	ipk	double	-	IPK mahasiswa
6	semester	double	-	Semester yang ditempuh mahasiswa
7	gaji_ortu	double	-	Gaji dari orang tua mahasiswa
8	tanggungan_ortu	double	-	Tanggungan dari orang tua mahasiswa

Tabel 5.2 Struktur tabel ind_baik dan history

Nama database		algenetika		
Nama tabel		ind_baik dan history		
Primary key		id_indbaik dan id_history		
No	Nama Field	Tipe Data	Lebar	Keterangan
1	id_indbaik	int	10	Primary key



2	parent	Varchar	15	Nama kromosom
3	a1	double	-	Index nilai gen kriteria IPK
4	a2	double	-	Index nilai gen kriteria IPK
5	a3	double	-	Index nilai gen kriteria IPK
6	a4	double	-	Index nilai gen kriteria IPK
7	b1	double	-	Index nilai gen kriteria penghasilan orang tua
8	b2	double	-	Index nilai gen kriteria penghasilan orang tua
9	b3	double	-	Index nilai gen kriteria penghasilan orang tua
10	b4	double	-	Index nilai gen kriteria penghasilan orang tua
11	c1	double	-	Index nilai gen kriteria tanggungan orang tua
12	c2	double	-	Index nilai gen kriteria tanggungan orang tua
13	c3	double	-	Index nilai gen kriteria tanggungan orang tua
14	c4	double	-	Index nilai gen kriteria tanggungan orang tua
15	d1	double	-	Index nilai gen kriteria tagihan telepon
16	d2	double	-	Index nilai gen kriteria tagihan telepon
17	d3	double	-	Index nilai gen kriteria tagihan telepon
18	d4	double	-	Index nilai gen kriteria tagihan telepon
19	e1	double	-	Index nilai gen kriteria tagihan listrik
20	e2	double	-	Index nilai gen kriteria tagihan listrik
21	e3	double	-	Index nilai gen kriteria tagihan listrik
22	e4	double	-	Index nilai gen kriteria tagihan listrik

23	f1	double	-	Index nilai gen kriteria tagihan PDAM
24	f2	double	-	Index nilai gen kriteria tagihan PDAM
25	f3	double	-	Index nilai gen kriteria tagihan PDAM
26	f4	double	-	Index nilai gen kriteria tagihan PDAM
27	g1	double	-	Index nilai gen kriteria tagihan PBB
28	g2	double	-	Index nilai gen kriteria tagihan PBB
29	g3	double	-	Index nilai gen kriteria tagihan PBB
30	g4	double	-	Index nilai gen kriteria tagihan PBB
31	h1	double	-	Index nilai gen kriteria keputusan beasiswa
32	h2	double	-	Index nilai gen kriteria keputusan beasiswa
33	fitness	double	-	Nilai <i>fitness</i> terbaik dari satu individu

5.2 Implementasi Kode Program

Sistem Optimasi Fungsi Keanggotaan *Fuzzy* Tsukamoto Dua Tahap Menggunakan Algoritma Genetika pada Pemilihan Calon Penerima Beasiswa-PPA dan BBP-PA diimplementasikan dalam program dengan menggunakan bahasa pemrograman PHP. Implementasi kode program sesuai dengan siklus penyelesaian masalah atau perhitungan manual pada sub-bab 3.4 dan 3.5. Implementasi dari bahasa pemrograman PHP pada sistem optimasi fungsi keanggotaan *fuzzy* Tsukamoto menggunakan algoritma genetika dijelaskan pada sub-bab 5.2.1 sampai dengan 5.2.9.

5.2.1 Inisialisasi Populasi Awal

Proses awal dari implementasi algoritma kedalam sebuah kode program yaitu dengan membangkitkan sejumlah kromosom atau individu yang menjadi suatu populasi sebanyak *input*-an dari *user*. Representasi kromosom

yang digunakan adalah *real-coded* atau menggunakan bilangan pecahan. Kromosom yang dibangkitkan secara *random* terdiri dari 4 gen untuk masing-masing kriteria *input* fungsi keanggotaan dan 2 gen untuk kriteria *output* fungsi keanggotaan. Kemudian, pada tiap gen akan diurutkan secara *ascending*. Pada Gambar 5.1 ditunjukkan kode program untuk proses inisialisasi populasi awal.

```

1  for($i=1; $i<=$popsize; $i++){ //jml popsize
2      for($j=1; $j<=4; $j++){ //jml kromosom
3          $ipk = mt_rand(250,389) / 100;
4          $parent_a['P'].$i]['a'].$j]= $ipk;
5          $temp = $parent_a['P'].$i];
6          $index = $parent_a['P'].$i];
7          sort($temp);
8          $index_key=array_keys($index);
9          $sortipk = array_combine($index_key,$temp);
10         }
11         $parent_a['P'].$i]=$sortipk;
12     }

```

Gambar 5.1 Kode program proses inisialisasi populasi awal

5.2.2 Reproduksi *Crossover*

Setelah dibangkitkan sejumlah kromosom secara acak, proses selanjutnya adalah proses reproduksi *crossover*. Pada proses reproduksi *crossover* metode yang digunakan adalah *one-cut-point crossover*. Proses *crossover* ini akan menghasilkan dua anak atau *offspring* dalam satu kali proses. Anak atau *offspring* yang dihasilkan adalah perkalian antara *crossover rate* (*cr*) dengan ukuran populasi yang di-*input*-kan oleh *user*. Implementasi kode program proses reproduksi *crossover* ditunjukkan pada Gambar 5.2.

```

1  $jml_ank=$scr*$popsize;
2  $jmlank= $jml_ank/2;
3
4  for($ank=1; $ank<=$jmlank; $ank++){
5      $pr1=mt_rand(1,$popsize);
6      $pr2=mt_rand(1,$popsize);
7      while($pr1 == $pr2) $pr2=mt_rand(1,$popsize);
8      for($j=1; $j<=4; $j++){
9          $crossover_a[$ank.'C1']['a'].$j]=$parent_a['P'].$pr1]['a'].$
10         j];
11         $crossover_a[$ank.'C2']['a'].$j]=$parent_a['P'].$pr2]['a'].$
12         j];
13     }
14     $a_rand=mt_rand(1,3);
15     for($j=1; $j<=$a_rand; $j++){

```

```

16         $ctemp_a=$crossover_a[$ank.'C1']['a'.$j];
17         $crossover_a[$ank.'C1']['a'.$j]=$crossover_a[$ank.'
18 C2']['a'.$j];
19         $crossover_a[$ank.'C2']['a'.$j]=$ctemp_a;
20     }
21     $a_tempc1=$crossover_a[$ank.'C1'];
22     $index=$crossover_a[$ank.'C1'];
23     sort($a_tempc1); $index_key=array_keys($index);
24     $a_sortc1 = array_combine($index_key,$a_tempc1);
25     $a_tempc2=$crossover_a[$ank.'C2'];
26     sort($a_tempc2);
27     $a_sortc2 = array_combine($index_key,$a_tempc2);
28     $crossover_a[$ank.'C1']=$a_sortc1;
29     $crossover_a[$ank.'C2']=$a_sortc2;
30     $parent_a['P'.(count($parent_a)+1)] =
31 $crossover_a[$ank.'C1'];
32     $parent_a['P'.(count($parent_a)+1)] =
33 $crossover_a[$ank.'C2'];

```

Gambar 5.2 Kode program proses reproduksi *one-cut point crossover*

Pada baris 1 dan 2 kode program merupakan jumlah anak yang akan dihasilkan dari proses *crossover*. Pemilihan induk secara acak ditunjukkan pada kode program baris 5 dan 6. Selanjutnya proses *one-cut-point crossover* ditunjukkan pada kode program baris 14-20. Proses *one-cut-point crossover* memilih beberapa titik potong gen secara acak (baris 14) pada 2 induk yang telah terpilih, kemudian menukarkan nilai gen induk pertama dengan induk kedua (baris 16-19). Setelah ditukar nilai gennya, proses selanjutnya mengurutkan nilai gen pada masing-masing kromosom yang ditunjukkan pada kode program baris 21-29.

5.2.3 Reproduksi Mutasi

Pada proses reproduksi mutasi menggunakan metode *random mutation*. Proses mutasi memilih satu induk secara acak. Selanjutnya, kode program baris 6-7 memilih gen dari induk yang telah terpilih secara acak dan dihitung dengan menggunakan persamaan 3-3. Kode program proses reproduksi mutasi ditunjukkan pada Gambar 5.3.

```

1  $p_mutasi = rand(1,$popsize);
2  $jml_index = rand(1,4);
3  $r = mt_rand(-100, 100)/1000;
4  for($index=1; $index<=$jml_index; $index++){
5      $index_mutasi = rand(1,4);
6      $mutasi_a['a'.$index_mutasi] =
7      round($parent_a['P'.$p_mutasi]['a'.$index_mutasi] + $r *
8      (4 - 2.5),2);

```

```

9   }
10  for($j=1; $j<=4; $j++){
11      $temp_a['a'.$j]=$parent_a['P'.$p_mutasi]['a'.$j];
12      $pmutasi_a[$ank.'C']= array_replace($temp_a,
13  $mutasi_a);
14  }
15  $a_mtemp = $pmutasi_a[$ank.'C'];
16  $index = $pmutasi_a[$ank.'C'];
17  sort($a_mtemp);
18  $index_key=array_keys($index);
19  $a_msort = array_combine($index_key,$a_mtemp);
20  $pmutasi_a[$ank.'C'] = $a_msort;
21  $parent_a['P'.(count($parent_a)+1)] =
22  $pmutasi_a[$ank.'C'];

```

Gambar 5.3 Kode program proses reproduksi *random mutation*

Hasil dari perhitungan dengan menggunakan persamaan 3-3 merupakan hasil dari reproduksi mutasi yang disebut *offspring* ditunjukkan pada kode program baris 11-12. Setelah dilakukan proses *random mutation*, nilai gen pada masing-masing diurutkan secara *ascending* seperti pada kode program baris 17-20.

5.2.4 Fuzzyfikasi

Pada sistem inferensi *fuzzy* Tsukamoto, fuzzyfikasi merupakan proses pertama untuk merubah nilai *crisp* atau nilai *input*-an dari masing-masing kriteria menjadi nilai derajat keanggotaan. Nilai *input*-an yang digunakan berdasarkan data mahasiswa yang disimpan kedalam *database*. Sedangkan, nilai batas-batas keanggotaan yang digunakan merupakan hasil dari pembangkitkan kromosom secara acak. Gambar 5.5 ditunjukkan kode program dari proses fuzzyfikasi.

```

1   require "config.php";
2   $data_mhs = mysql_query("SELECT * FROM data_mhs");
3   while($mhs=mysql_fetch_array($data_mhs)){
4       $jmhs=$mhs['id_datamhs'];
5       $datamhs['M'.$jmhs]['ipk']=$mhs['ipk'];
6       $datamhs['M'.$jmhs]['semester']=$mhs['semester'];
7       $datamhs['M'.$jmhs]['gaji']=$mhs['gaji_ortu'];
8       $datamhs['M'.$jmhs]['tang']=$mhs['tanggung_ortu'];
9   }

```

Gambar 5.4 Data mahasiswa yang merupakan nilai *crisp*

```

1   if($datamhs['M'.$jmhs]['ipk'] > 2.5 &&
2   $datamhs['M'.$jmhs]['ipk'] <= $parent_a['P'.$i]['a1']){
3       $ipk_rendah['M'.$jmhs]=1;
4       $ipk_sedang['M'.$jmhs]=0;
5       $ipk_tinggi['M'.$jmhs]=0;

```

```

6  }
7  elseif($datamhs['M'].$jmhs]['ipk'] >
8  $parent_a['P'].$i]['a1'] && $datamhs['M'].$jmhs]['ipk'] <=
9  $parent_a['P'].$i]['a2']){
10
11     $ipk_rendah['M'].$jmhs]=round(($parent_a['P'].$i]['a2'
12 ]-$datamhs['M'].$jmhs]['ipk'])/($parent_a['P'].$i]['a2']-
13 $parent_a['P'].$i]['a1']), 2);
14
15     $ipk_sedang['M'].$jmhs]=round(($datamhs['M'].$jmhs]['i
16 pk']-$parent_a['P'].$i]['a1'])/($parent_a['P'].$i]['a2']-
17 $parent_a['P'].$i]['a1']), 2);
18     $ipk_tinggi['M'].$jmhs]=0;
19 }
20 else{
21     $ipk_rendah['M'].$jmhs]=0;
22 }
23 if($datamhs['M'].$jmhs]['ipk'] > $parent_a['P'].$i]['a3'] &&
24 $datamhs['M'].$jmhs]['ipk'] <= $parent_a['P'].$i]['a4']){
25
26     $ipk_sedang['M'].$jmhs]=round(($parent_a['P'].$i]['a4'
27 ]-$datamhs['M'].$jmhs]['ipk'])/($parent_a['P'].$i]['a4']-
28 $parent_a['P'].$i]['a3']), 2);
29
30     $ipk_tinggi['M'].$jmhs]=round(($datamhs['M'].$jmhs]['i
31 pk']-$parent_a['P'].$i]['a3'])/($parent_a['P'].$i]['a4']-
32 $parent_a['P'].$i]['a3']), 2);
33 }
34 elseif($datamhs['M'].$jmhs]['ipk'] >
35 $parent_a['P'].$i]['a2'] && $datamhs['M'].$jmhs]['ipk'] <=
36 $parent_a['P'].$i]['a3']){
37     $ipk_sedang['M'].$jmhs]=1;
38     $ipk_tinggi['M'].$jmhs]=0;
39 }
40 elseif($datamhs['M'].$jmhs]['ipk'] >
41 $parent_a['P'].$i]['a4'] && $datamhs['M'].$jmhs]['ipk']
42 <=4){
43     $ipk_sedang['M'].$jmhs]=0;
44     $ipk_tinggi['M'].$jmhs]=1;
45 }

```

Gambar 5.5 Kode program proses fuzzyfikasi

Kode program Gambar 5.4 baris 1-9 merupakan nilai *input* atau *crisp* data mahasiswa yang berasal dari *database*. Pada contoh kode program Gambar 5.4, data yang digunakan sebanyak 4 kriteria, yaitu IPK mahasiswa, semester, penghasilan orang tua, dan tanggungan orang tua. Sedangkan kode program Gambar 5.5 baris 1-45 adalah perhitungan nilai *input* atau *crisp* dengan batas-batas fungsi keanggotaan yang menghasilkan nilai derajat keanggotaan.

5.2.5 Sistem Inferensi Fuzzy

Kode program sistem inferensi fuzzy Tsukamoto ditunjukkan pada Gambar 5.6 dan 5.7. Pada baris 1-30 merupakan aturan atau *rules based* kriteria positif (IPK, tanggungan orang tua, dan piagam) untuk sistem inferensi tahap pertama. Jumlah aturan yang digunakan sebanyak 27 aturan. Perhitungan alpha predikat dalam kode program terlihat pada baris 1-3, 6-8, 11-13, 16-18, 21-23, dan 27-29.

```

1  $apositif1['a_p'].$jmhs] =
2  min($ipk_rendah['M'].$jmhs], $t_rendah['M'].$jmhs], $p_rendah
3  ['M'].$jmhs]);
4  $zpos1['zpos1'].$jmhs] = 3-($apositif1['a_p'].$jmhs]*2);
5
6  $apositif2['a_p'].$jmhs] =
7  min($ipk_rendah['M'].$jmhs], $t_rendah['M'].$jmhs], $p_sedang
8  ['M'].$jmhs]);
9  $zpos2['zpos2'].$jmhs] = 3-($apositif2['a_p'].$jmhs]*2);
10
11 $apositif3['a_p'].$jmhs] =
12 min($ipk_rendah['M'].$jmhs], $t_rendah['M'].$jmhs], $p_tinggi
13 ['M'].$jmhs]);
14 $zpos3['zpos3'].$jmhs] = 3-($apositif3['a_p'].$jmhs]*2);
15
16 $apositif4['a_p'].$jmhs] =
17 min($ipk_rendah['M'].$jmhs], $t_sedang['M'].$jmhs], $p_rendah
18 ['M'].$jmhs]);
19 $zpos4['zpos4'].$jmhs] = 3-($apositif4['a_p'].$jmhs]*2);
20
21 $apositif5['a_p'].$jmhs] =
22 min($ipk_rendah['M'].$jmhs], $t_sedang['M'].$jmhs], $p_sedang
23 ['M'].$jmhs]);
24 $zpos5['zpos5'].$jmhs] = 3-($apositif5['a_p'].$jmhs]*2);
25 ...
26 ...
27 $apositif27['a_p'].$jmhs] =
28 min($ipk_tinggi['M'].$jmhs], $t_tinggi['M'].$jmhs], $p_tinggi
29 ['M'].$jmhs]);
30 $zpos27['zpos27'].$jmhs] = ($apositif27['a_p'].$jmhs]*2)+1;

```

Gambar 5.6 Inferensi fuzzy untuk kriteria positif

Selanjutnya, terdapat *rules based* yang berbeda untuk 5 kriteria penentuan beasiswa lainnya yang dikelompokkan menjadi kriteria negatif. Pada kode program baris 1-30 ditunjukkan *rules based* kriteria negatif. Jumlah aturan yang digunakan pada kriteria negatif sebanyak 81 aturan. Perhitungan alpha predikat dalam kode program terlihat pada baris 1-3, 6-8, 11-13, 16-18, 21-23, dan 27-29.

```

1  $anegatif1['a_n'].$jmhs] = min($g_rendah,
2  $list_rendah['M'].$jmhs], $telp_rendah['M'].$jmhs],
3  $pdam_rendah['M'].$jmhs], $pbb_rendah['M'].$jmhs]);
4  $zneg1['zneg1'].$jmhs] = 3-($anegatif1['a_n'].$jmhs]*2);
5
6  $anegatif2['a_n'].$jmhs] = min($g_rendah,
7  $list_rendah['M'].$jmhs], $telp_rendah['M'].$jmhs],
8  $pdam_rendah['M'].$jmhs], $pbb_sedang['M'].$jmhs]);
9  $zneg2['zneg2'].$jmhs] = 3-($anegatif2['a_n'].$jmhs]*2);;
10
11 $anegatif3['a_n'].$jmhs] = min($g_rendah,
12 $list_rendah['M'].$jmhs], $telp_rendah['M'].$jmhs],
13 $pdam_rendah['M'].$jmhs], $pbb_tinggi['M'].$jmhs]);
14 $zneg3['zneg3'].$jmhs] = 3-($anegatif3['a_n'].$jmhs]*2);
15
16 $anegatif4['a_n'].$jmhs] = min($g_rendah,
17 $list_rendah['M'].$jmhs], $telp_rendah['M'].$jmhs],
18 $pdam_sedang['M'].$jmhs], $pbb_rendah['M'].$jmhs]);
19 $zneg4['zneg4'].$jmhs] = 3 -($anegatif4['a_n'].$jmhs]*2);
20
21 $anegatif5['a_n'].$jmhs] = min($g_rendah,
22 $list_rendah['M'].$jmhs], $telp_rendah['M'].$jmhs],
23 $pdam_sedang['M'].$jmhs], $pbb_sedang['M'].$jmhs]);
24 $zneg5['zneg5'].$jmhs] = 3-($anegatif5['a_n'].$jmhs]*2);
25 ...
26 ...
27 $anegatif81['a_n'].$jmhs] = min($g_rendah,
28 $list_tinggi['M'].$jmhs], $telp_tinggi['M'].$jmhs],
29 $pdam_tinggi['M'].$jmhs], $pbb_tinggi['M'].$jmhs]);
30 $zneg81['zneg81'].$jmhs] = ($anegatif81['a_n'].$jmhs]*1)+2;

```

Gambar 5.7 Inferensi *fuzzy* untuk kriteria negatif

Pada inferensi *fuzzy* Tsukamoto pada tahap kedua, nilai input yang digunakan berasal dari nilai output inferensi *fuzzy* Tsukamoto tahap pertama. Implementasi kode program inferensi *fuzzy* Tsukamoto tahap kedua ditunjukkan pada Gambar 5.8. Pada inferensi *fuzzy* tahap kedua digunakan *rules based* sebanyak 4 aturan pada kode program baris 35-57.

```

1  if($hasilpos['Z'].$jmhs] <= 1){
2      $pos_rendah['Z'].$jmhs]=1;
3      $pos_tinggi['Z'].$jmhs]=0;
4  }
5  elseif($hasilpos['Z'].$jmhs] > 1 && $hasilpos['Z'].$jmhs]
6  <= 3){
7      $pos_rendah['Z'].$jmhs]=round((3-
8  $hasilpos['Z'].$jmhs)/2, 2);
9      $pos_tinggi['Z'].$jmhs]=round(($hasilpos['Z'].$jmhs]-
10  1)/2, 2);
11 }
12 elseif($hasilpos['Z'].$jmhs] > 3 && $hasilpos['Z'].$jmhs]
13 <= 4){
14     $pos_rendah['Z'].$jmhs]=0;
15     $pos_tinggi['Z'].$jmhs]=1;

```

```

16 }
17
18 if($hasilneg['Z'].$jmhs] <= 1){
19     $neg_rendah['Z'].$jmhs]=1;
20     $neg_tinggi['Z'].$jmhs]=0;
21 }
22 elseif($hasilneg['Z'].$jmhs] > 1 && $hasilneg['Z'].$jmhs]
23 <= 3){
24     $neg_rendah['Z'].$jmhs]=round((3-
25 $hasilneg['Z'].$jmhs)/2, 2);
26     $neg_tinggi['Z'].$jmhs]=round(($hasilneg['Z'].$jmhs]-
27 1)/2, 2);
28 }
29 elseif($hasilneg['Z'].$jmhs] > 3 && $hasilneg['Z'].$jmhs]
30 <= 3){
31     $neg_rendah['Z'].$jmhs]=0;
32     $neg_tinggi['Z'].$jmhs]=1;
33 }
34
35 $akeputusan1['a_k'].$jmhs] =
36 min($pos_rendah['Z'].$jmhs],$neg_rendah['Z'].$jmhs]);
37 $zbea1['zbea1'].$jmhs] = $parent_h['P'].$i]['h2']-
38 ($akeputusan1['a_k'].$jmhs)*($parent_h['P'].$i]['h2']-
39 $parent_h['P'].$i]['h1']);
40
41 $akeputusan2['a_k'].$jmhs] =
42 min($pos_rendah['Z'].$jmhs],$neg_tinggi['Z'].$jmhs]);
43 $zbea2['zbea2'].$jmhs] = $parent_h['P'].$i]['h2']-
44 ($akeputusan2['a_k'].$jmhs)*($parent_h['P'].$i]['h2']-
45 $parent_h['P'].$i]['h1']);
46
47 $akeputusan3['a_k'].$jmhs] =
48 min($pos_tinggi['Z'].$jmhs],$neg_rendah['Z'].$jmhs]);
49 $zbea3['zbea3'].$jmhs] =
50 ($akeputusan3['a_k'].$jmhs)*($parent_h['P'].$i]['h2']-
51 $parent_h['P'].$i]['h1'])+$parent_h['P'].$i]['h1'];
52
53 $akeputusan4['a_k'].$jmhs] =
54 min($pos_tinggi['Z'].$jmhs],$neg_tinggi['Z'].$jmhs]);
55 $zbea4['zbea4'].$jmhs] =
56 ($akeputusan4['a_k'].$jmhs)*($parent_h['P'].$i]['h2']-
57 $parent_h['P'].$i]['h1'])+$parent_h['P'].$i]['h1'];

```

Gambar 5.8 Kode program inferensi *fuzzy* Tsukamoto tahap kedua

5.2.6 Defuzzifikasi

Proses defuzzifikasi merupakan proses terakhir dalam FIS Tsukamoto dengan perhitungan defuzzifikasi rata-rata terpusat (*Center Average Defuzzifier*) pada persamaan 2-6. Kode program dari proses defuzzifikasi ditunjukkan pada Gambar 5.9.

```

1 $crisbea['crisbea'].$jmhs] =
2 ($akeputusan1['a_k'].$jmhs]*$zbea1['zbea1'].$jmhs]) +
3 ($akeputusan2['a_k'].$jmhs]*$zbea2['zbea2'].$jmhs]) +

```



```

4 ($akeputusan3['a_k'].$jmhs]*$zbea3['zbea3'].$jmhs]) +
5 ($akeputusan4['a_k'].$jmhs]*$zbea4['zbea4'].$jmhs]);
6
7 $crisbeal['crisbeal'].$jmhs] =
8 $akeputusan1['a_k'].$jmhs] + $akeputusan2['a_k'].$jmhs] +
9 $akeputusan3['a_k'].$jmhs] + $akeputusan4['a_k'].$jmhs];
10
11 $hasilbea['P'.$i]['Z'].$jmhs]=
12 round($crisbea['crisbea'].$jmhs]/$crisbeal['crisbeal'].$jmhs], 2);
13

```

Gambar 5.9 Kode program proses defuzzifikasi

Pada baris 1-7 menunjukkan kode program perhitungan jumlah dari nilai α -predikat. Selanjutnya, kode program baris 9-13 menunjukkan perhitungan Jumlah dari (α -predikat*z). Sedangkan, perhitungan defuzzifikasi rata-rata terpusat (*Center Average Defuzzifier*) ditunjukkan pada kode program baris 15-16.

5.2.7 Akurasi Sistem

Implementasi kode program proses akurasi sistem menggunakan perhitungan korelasi *spearman* seperti yang telah dijelaskan pada sub-bab 3.4.5. Pada proses akurasi sistem semua data mahasiswa akan dilakukan perankingan berdasarkan hasil dari perhitungan FIS Tsukamoto. Implementasi kode program ditunjukkan pada Gambar 5.10.

```

1 for($i=1; $i<=count($parent_a); $i++){
2     for($jmhs=1; $jmhs<=30; $jmhs++){
3         $temp = $hasil['P'.$i];
4         arsort($temp);
5         $ranks =
6     array_fill(1, count($temp), 'ranking');
7         $b=array_keys($temp);
8         $c=array_keys($ranks);
9         $rank = array_combine($b,$c);
10        $a = $hasil['P'.$i];
11    }
12    $ranked['P'.$i]=array_replace($a,$rank);
13 }
14 require "pakar.php";
15
16 for($i=1; $i<=count($parent_a); $i++){
17     for($jmhs=1; $jmhs<=30; $jmhs++){
18         $d_spearman['P'.$i]['Z'].$jmhs] =
19     pow($ranked['P'.$i]['Z'].$jmhs]-$ranked_p['Z'].$jmhs, 2);
20         $jmlh = $d_spearman['P'.$i];
21     }
22     $total['P'.$i]= array_sum($jmlh);
23     $rs['P'.$i]= round(1-

```

24	((6*\$total['P'.'. \$i]) / (pow(30,3)-30)), 3);
25	}

Gambar 5.10 Kode program proses perhitungan akurasi sistem

Kode program baris 3-12 merupakan proses perangkaian dari hasil perhitungan FIS Tsukamoto untuk setiap data mahasiswa yang berjumlah 30 mahasiswa. Selanjutnya pada kode program baris 14, proses pengambilan rangking menurut pendapat pakar yang sebelumnya telah dihitung menggunakan FIS Tsukamoto. Proses perhitungan korelasi *spearman* pada implementasi kode program ditunjukkan pada baris 18-24. Hasil dari perhitungan korelasi *spearman* akan dijadikan sebagai nilai *fitness* dari setiap individu.

5.2.8 Perhitungan Fitness

Perhitungan nilai *fitness* dalam implementasi kode program dinyatakan dalam Gambar 5.11. Nilai *fitness* yang didapat nantinya akan digunakan untuk mencari individu terbaik dari populasi yang telah terbentuk. Individu yang terbaik itulah sebagai solusi dari permasalahan. Solusi permasalahan yang digunakan dalam kasus optimasi merupakan solusi maksimal seperti pada yang telah dijelaskan sebelumnya pada sub-bab 3.5.1. Baris 1-2 dari kode program merupakan perhitungan korelasi *spearman*. Hasil dari korelasi *spearman* akan digunakan sebagai nilai *fitness* dari masing-masing individu yang ditunjukkan pada kode program baris 4.

1	\$rs['P'.'. \$i]= round(1-((6*\$total['P'.'. \$i]) / (pow(30,3)-30)),
2	3);
3	
4	\$fitness = \$rs;

Gambar 5.11 Kode program proses perhitungan *fitness*

5.2.9 Seleksi

Proses seleksi yang digunakan adalah *elitism selection* dimana menyeleksi semua individu, yaitu *parent* dan *offspring* dengan mengurutkan nilai *fitness* yang dimiliki setiap individu secara *descending*. Setelah diurutkan, akan diambil individu dengan nilai *fitness* terbaik sejumlah ukuran populasi awal untuk dilakukan generasi selanjutnya. Pada Gambar 5.12 merupakan implementasi kode program proses seleksi.

```

1  $fitness = $rs;
2  arsort($fitness);
3  $b=array_keys($fitness);
4
5  $pnew_a = array(); $pnew_b = array();
6  $pnew_c = array(); $pnew_d = array();
7  $pnew_e = array(); $pnew_f = array();
8  $pnew_g = array(); $pnew_h = array();
9
10 for($i=0; $i<$popsize; $i++){
11     for($j=1; $j<=4; $j++){
12         $pnew_a['P'.($i+1)]['a'.'. $j]=$parent_a[$b[$i]]['a'.'. $j];
13         $pnew_b['P'.($i+1)]['b'.'. $j]=$parent_b[$b[$i]]['b'.'. $j];
14         $pnew_c['P'.($i+1)]['c'.'. $j]=$parent_c[$b[$i]]['c'.'. $j];
15         $pnew_d['P'.($i+1)]['d'.'. $j]=$parent_d[$b[$i]]['d'.'. $j];
16         $pnew_e['P'.($i+1)]['e'.'. $j]=$parent_e[$b[$i]]['e'.'. $j];
17         $pnew_f['P'.($i+1)]['f'.'. $j]=$parent_f[$b[$i]]['f'.'. $j];
18         $pnew_g['P'.($i+1)]['g'.'. $j]=$parent_g[$b[$i]]['g'.'. $j];
19     }
20     for($j=1; $j<=2; $j++){
21         $pnew_h['P'.($i+1)]['h'.'. $j]=$parent_h[$b[$i]]['h'.'. $j];
22     }
23 }
24 $hasil = array(
25     'parent_a' => $pnew_a,
26     'parent_b' => $pnew_b,
27     'parent_c' => $pnew_c,
28     'parent_d' => $pnew_d,
29     'parent_e' => $pnew_e,
30     'parent_f' => $pnew_f,
31     'parent_g' => $pnew_g,
32     'parent_h' => $pnew_h
33 );

```

Gambar 5.12 Kode program proses *elitsm selection*

Baris 1 dari kode program proses seleksi ditunjukkan pengurutan secara *descending* berdasarkan nilai *fitness* dari masing-masing individu. Selanjutnya, pada baris 24-33 ditunjukkan proses meng-*copy* individu baru ke dalam individu lama sejumlah populasi awal.

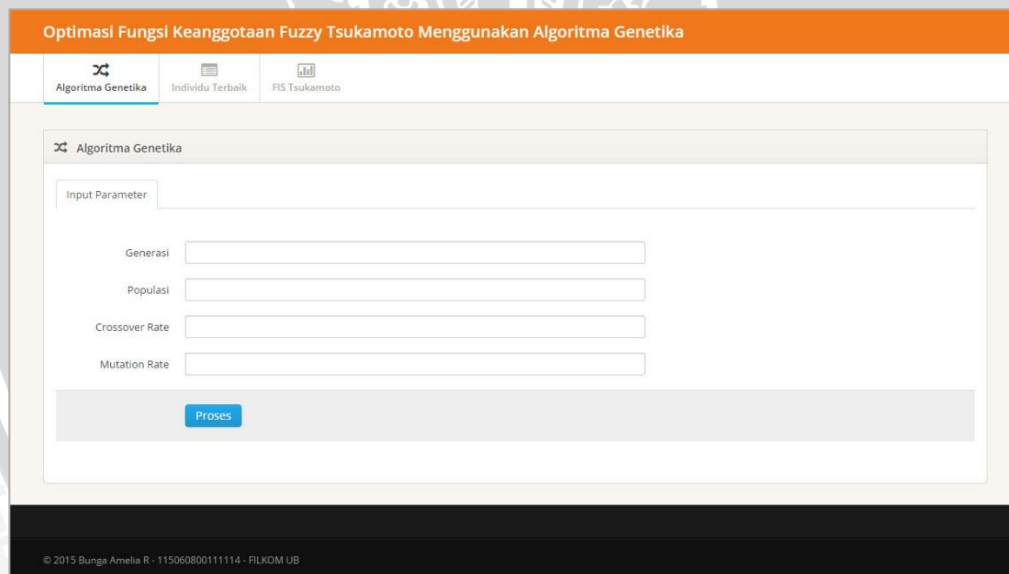
5.3 Implementasi Antarmuka

Implementasi antarmuka dalam Sistem Optimasi Fungsi Keanggotaan *Fuzzy Tsukamoto* yang digunakan berbasis *website*. Antarmuka sistem optimasi ini digunakan oleh pengguna untuk melakukan optimasi batasan fungsi keanggotaan kedalam sistem. Pengguna hanya meng-*input*-kan nilai parameter algoritma genetika untuk mendapatkan hasil optimasi. Dalam implementasi

antarmuka terdapat 3 halaman utama yang dapat diakses oleh pengguna, antara lain menu Algoritma Genetika, Individu Terbaik, dan FIS Tsukamoto.

5.3.1 Menu Algoritma Genetika

Pada saat akses pertama kali dalam sistem optimasi ini, pengguna akan langsung dihubungkan pada halaman menu Algoritma Genetika. Menu ini menampilkan *form input* untuk parameter algoritma genetika, yaitu jumlah generasi, ukuran populasi awal, *crossover rate*, dan *mutation rate*. Pengguna meng-*input*-kan masing-masing nilai dari parameter untuk mendapatkan hasil optimasi batasan fungsi keanggotaan *fuzzy* Tsukamoto. Setelah di-*input*-kan nilai parameter algoritma genetika, sistem menampilkan individu baru beserta nilai *fitness* dari masing-masing individu hasil dari proses algoritma genetika sejumlah banyaknya generasi yang di-*input*-kan oleh prngguna. Halaman menu Algoritma Genetika ditampilkan pada Gambar 5.13 dan 5.14.



The screenshot shows a web application interface titled "Optimasi Fungsi Keanggotaan Fuzzy Tsukamoto Menggunakan Algoritma Genetika". It features three tabs: "Algoritma Genetika", "Individu Terbaik", and "FIS Tsukamoto". The "Algoritma Genetika" tab is active, displaying an "Input Parameter" section with four input fields: "Generasi", "Populasi", "Crossover Rate", and "Mutation Rate". A blue "Proses" button is located below the input fields. At the bottom of the page, there is a copyright notice: "© 2015 Bunga Amelia R - 115060800111114 - FLKOM UB".

Gambar 5.13 Halaman utama menu Algoritma Genetika



This screenshot shows the same "Algoritma Genetika" input form as in Gambar 5.13, but with numerical values entered into the input fields. The values are: Generasi: 5, Populasi: 10, Crossover Rate: 0.4, and Mutation Rate: 0.2. The "Proses" button remains visible at the bottom.

Parent	Kromosom IPK	Penghasilan Ortu (It)	Tanggungan Ortu	Tagihan Telepon	Tagihan Listrik	Tagihan PDAM	Tagihan PBB
P15	2.99 3.47 3.53 3.64	1.21 3.32 3.6 4.33	3.32 3.71 4.97 4.97	5.48 6.91 14.95 14.96	9.63 14.34 17.86 19.84	2.68 6.07 7.19 7.44	5.01 16.16 18.0
P6	2.99 3.47 3.64 3.64	1.21 3.32 3.96 4.33	3.32 4.07 4.97 4.97	5.48 8 14.95 14.96	9.63 15.8 17.86 19.84	2.68 6.8 7.44 7.92	6.47 17.62 18.0
P10	2.65 2.87 3.62 3.84	3.43 4.34 4.64 4.82	3.77 3.94 4.32 4.37	5.93 12.6 12.69 13.21	5.19 8.76 15.21 18.01	0.27 1.52 6.35 8.56	1.07 1.76 3.81
P16	2.7 2.84 3.53 3.81	1.36 2.35 3.15 3.56	2.33 3.54 3.73 3.93	6.91 8.71 14.47 14.83	4.61 14.34 20.58 20.84	1.05 1.74 7.19 8.48	5.01 12.32 15.7
P13	2.52 2.83 3.68 3.76	3.59 4.83 4.92 5	2.18 3.21 4.7 5	6.54 9.58 9.59 13.66	7.56 7.63 9.78 12.12	3.05 4 5.65 9.67	9.44 17.5 18.32
P9	2.72 3.13 3.68 3.76	2.22 4.83 4.92 5	3.86 4.66 4.7 5	4.68 6.54 9.59 13.66	6.2 7.56 9.78 12.12	3.97 7.65 7.82 9.67	4.41 16.82 18.3
P2	2.85 3.22 3.3 3.83	2.98 4.93 4.96 4.98	3.76 4.67 4.91 4.97	8.52 8.8 12.93 13.51	9.57 17.7 19.11 19.6	2.28 3.06 3.66 4.47	7.89 9.21 16.5
P1	2.91 2.96 2.99 3.45	1.69 2.58 2.78 4.46	3.08 3.74 4.16 4.66	5.25 5.52 7.77 12.8	5.75 6.21 10.27 12.08	0.48 3.38 9.52 9.71	5.49 6.32 6.5 11
P4	2.7 2.71 2.85 3.81	1.36 1.93 2.73 3.56	1.91 3.51 3.54 3.73	7.45 14.33 14.47 14.83	4.61 18.47 18.9 19.16	0.21 0.9 8.48 9.06	9.25 10.64 14.0
P14	2.72 3.03 3.1 3.13	2.22 3.85 4.84 4.89	3.86 4.66 4.8 4.85	4.68 11.16 14.16 14.34	7.56 7.63 9.78 12.12	3.97 7.47 7.65 7.82	4.41 16.82 19.3

Gambar 5.14 Proses perhitungan Algoritma Genetika

5.3.2 Menu Individu Terbaik

Menu Individu Terbaik merupakan tampilan antarmuka yang terdapat kromosom dari masing-masing kriteria beserta nilai *fitness* tertinggi untuk setiap generasi. Kromosom atau individu yang ditampilkan berdasarkan hasil dari perhitungan algoritma genetika pada saat pengguna meng-*input*-kan nilai parameter algoritma genetika. Pada Gambar 5.15 ditampilkan implementasi antarmuka untuk menu individu terbaik.

Optimasi Fungsi Keanggotaan Fuzzy Tsukamoto Menggunakan Algoritma Genetika							
Individu Terbaik		FIS Tsukamoto					
Penghasilan Ortu	Tanggungan Ortu	Tagihan Telepon	Tagihan Listrik	Tagihan PDAM	Tagihan PBB	Keputusan Beasiswa	Fitness
1.21 3.32 3.6 4.33	3.32 3.71 4.97 4.97	5.48 6.91 14.95 14.96	9.63 14.34 17.86 19.84	2.68 6.07 7.19 7.44	5.01 16.16 18.04 18.51	5.24 5.8	0.751
1.21 3.32 3.6 4.33	3.32 3.71 4.97 4.97	5.48 6.91 14.95 14.96	9.63 14.34 17.86 19.84	2.68 6.07 7.19 7.44	5.01 16.16 18.04 18.51	5.24 5.8	0.751
1.21 3.32 4.57 4.82	3.32 3.71 4.3 4.97	5.48 12.48 12.6 13.21	4.91 8.48 17.86 19.84	2.68 6.07 7.19 8.56	1.76 3.53 5.01 11.19	5.24 7.62	0.854
1.21 3.32 4.57 4.82	3.32 3.71 4.3 4.97	5.48 12.48 12.6 13.21	4.91 8.48 17.86 19.84	2.68 6.07 7.19 8.56	1.76 3.53 5.01 11.19	5.24 7.62	0.854
1.21 3.32 4.49 4.82	3.32 4.07 4.17 4.4	5.48 12.23 12.35 13.51	8.14 9.63 15.61 17.39	1.21 2.68 6.41 8.56	1.42 3.93 6.47 10.85	5.24 7.47	0.891

Gambar 5.15 Halaman menu Individu Terbaik

5.3.3 Menu FIS Tsukamoto

Menu ketiga dari implementasi antarmuka sistem optimasi yaitu FIS Tsukamoto. Menu ini menampilkan hasil perhitungan FIS Tsukamoto dengan

menggunakan hasil optimasi batasan-batasan fungsi keanggotaan. Setelah dilakukan perhitungan FIS Tsukamoto dengan hasil optimasi batasan-batasan fungsi keanggotaan, pada sistem ditampilkan data-data mahasiswa yang telah diranking berdasarkan nilai bobot. Tampilan menu FIS Tsukamoto ditunjukkan pada Gambar 5.16.

Optimasi Fungsi Keanggotaan Fuzzy Tsukamoto Menggunakan Algoritma Genetika

Algoritma Genetika Individu Terbaik FIS Tsukamoto

FIS Tsukamoto

Mahasiswa	IPK	Tanggungan Orang Tua	Plagam	Penghasilan Orang Tua	Tagihan Listrik	Tagihan Telepon	Tagihan PDAM	Tagihan PBB	Nilai Bobot
Mahasiswa 17	3.87	225000	-	1000000	81300	-	-	204100	7.47
Mahasiswa 30	3.87	1508000	Propinsi	13300000	500000	150000	100000	597000	7.47
Mahasiswa 9	3.87	1510000	Propinsi	13300000	500000	150000	100000	597000	7.47
Mahasiswa 13	3.87	398000	Propinsi	2000000	230000	30000	62500	-	7
Mahasiswa 4	3.91	182000	-	1500000	169300	50000	-	327600	6.97
Mahasiswa 12	3.93	1000000	-	9000000	134800	350000	11000	55100	6.95
Mahasiswa 22	3.33	315000	-	3300000	100000	-	40000	-	6.82
Mahasiswa 16	3.76	240000	-	3310000	89800	100000	-	195700	6.63
Mahasiswa 2	3.87	143000	-	4680000	93300	75000	86100	103900	6.59
Mahasiswa 19	3.64	400000	Nasional	5000000	50000	-	-	20000	6.56
Mahasiswa 8	3.29	176000	Propinsi	3600000	29300	100000	22800	-	6.55
Mahasiswa 15	3.76	191000	Propinsi	5950000	45800	35000	56400	34500	6.53
Mahasiswa 3	3.32	-	-	2570000	200000	-	18000	-	6.52
Mahasiswa 10	3.3	199000	-	2440000	65000	50000	18500	44900	6.5

Gambar 5.16 Implementasi antarmuka menu FIS Tsukamoto



BAB VI

PENGUJIAN DAN ANALISIS

Proses pengujian terhadap sistem optimasi fungsi keanggotaan *fuzzy* Tsukamoto akan dilakukan sesuai dengan skenario pengujian yang telah dijelaskan pada bab 4.3. Skenario pengujian yang dilakukan sebanyak tiga skenario, antara lain pengujian pertama adalah pengujian terhadap ukuran populasi, pengujian kedua akan dilakukan pengujian terhadap nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*), dan pengujian ketiga merupakan pengujian terhadap jumlah generasi yang dilakukan dalam proses algoritma genetika. Pengujian yang dilakukan ini bertujuan untuk mendapatkan nilai *fitness* terbaik dari satu kromosom. Kromosom yang memiliki nilai *fitness* terbaik itulah yang merupakan hasil optimasi dari permasalahan.

6.1 Hasil dan Analisis Skenario Pengujian Ukuran Populasi

Skenario pengujian pertama akan dilakukan pengujian terhadap ukuran populasi. Ukuran populasi (*popSize*) yang digunakan dalam pengujian, antara lain 20, 40, 60, 80, 100, 120, 140, 160, 180, dan 200. Pengujian ukuran populasi ini bertujuan mengetahui ukuran populasi yang tepat untuk menghasilkan nilai *fitness* terbaik. Ukuran populasi dari hasil pengujian ini akan digunakan untuk skenario pengujian kombinasi nilai *cr* dan *mr* dan jumlah generasi. Pada saat melakukan pengujian ukuran populasi, digunakan generasi sebanyak 100 generasi dan nilai *crossover rate*-*mutation rate* sebesar 0.4 dan 0.2. Pengujian dilakukan sebanyak 5 kali pada masing-masing ukuran populasi kemudian dihitung rata-rata nilai *fitness*-nya. Pengujian ini dilakukan terhadap optimasi fungsi keanggotaan *fuzzy* Tsukamoto pada penentuan Beasiswa-PPA dan BBP-PPA.

6.1.1 Pengujian Ukuran Populasi Beasiswa-PPA

Pengujian Beasiswa-PPA merupakan skenario pengujian yang menguji ukuran populasi untuk penentuan Beasiswa-PPA. Hasil skenario pengujian ukuran populasi penentuan Beasiswa-PPA ditunjukkan pada Tabel 6.1.

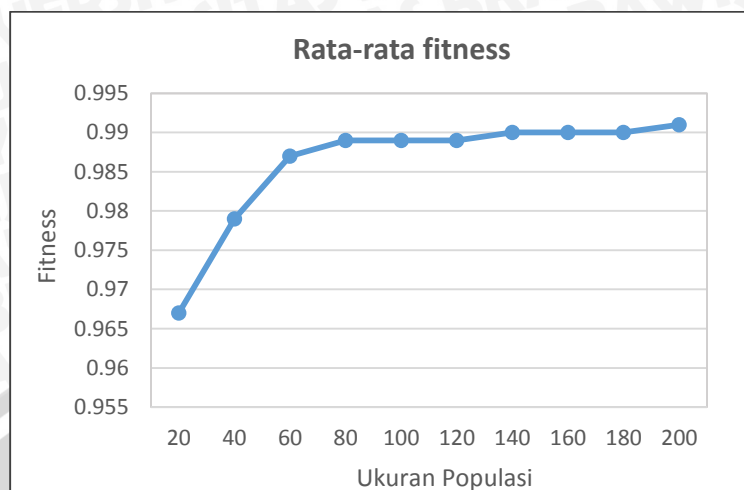
Tabel 6.1 Hasil skenario pengujian ukuran populasi penentuan Beasiswa-PPA

Ukuran Populasi	<i>cr</i> = 0.4 ; <i>mr</i> = 0.2 ; generasi = 100	Rata-rata nilai <i>fitness</i>
	Percobaan Ke - i	

	1	2	3	4	5	
20	0.974	0.976	0.956	0.972	0.96	0.967
40	0.985	0.988	0.976	0.969	0.979	0.979
60	0.992	0.986	0.991	0.977	0.989	0.987
80	0.984	0.993	0.991	0.993	0.987	0.989
100	0.987	0.988	0.995	0.989	0.985	0.989
120	0.988	0.988	0.99	0.986	0.979	0.989
140	0.989	0.99	0.988	0.994	0.991	0.99
160	0.988	0.99	0.989	0.995	0.992	0.99
180	0.996	0.984	0.989	0.993	0.99	0.99
200	0.995	0.991	0.992	0.99	0.988	0.991

Setelah dilakukan pengujian pada masing-masing ukuran populasi, rata-rata nilai *fitness* setiap ukuran populasi tentunya berbeda-beda. Ukuran populasi yang memiliki rata-rata nilai *fitness* terbesar merupakan ukuran populasi yang akan digunakan pada skenario pengujian kombinasi nilai *cr* dan *mr*.

Berdasarkan hasil rata-rata nilai *fitness* pengujian ukuran populasi untuk penentuan Beasiswa-PPA dapat diketahui rata-rata nilai *fitness* terbesar adalah 0.989 dengan ukuran populasi sebesar 80. Jumlah populasi 80 inilah akan digunakan dalam skenario pengujian kombinasi nilai *cr* dan *mr* untuk penentuan Beasiswa-PPA. Sedangkan, rata-rata nilai *fitness* terendah terletak pada ukuran populasi sebanyak 20. Pada Gambar 6.1 ditunjukkan grafik yang merepresentasikan rata-rata nilai *fitness*.



Gambar 6.1 Grafik hasil rata-rata nilai *fitness* ukuran populasi

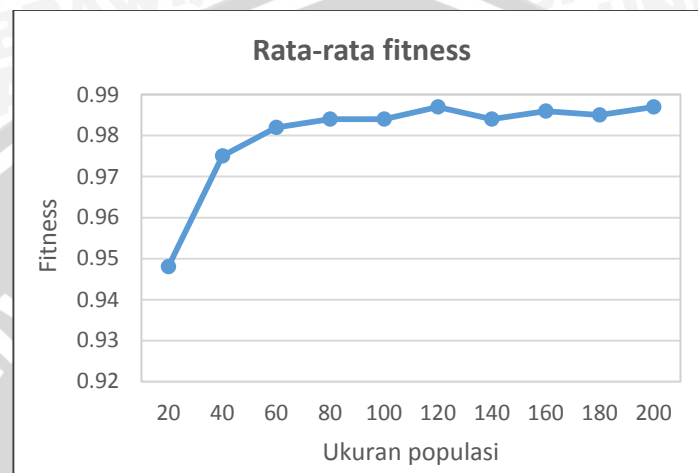
6.1.2 Pengujian Ukuran Populasi BBP-PPA

Pengujian ukuran populasi selanjutnya dilakukan pengujian terhadap ukuran populasi untuk penentuan BBP-PPA Hasil skenario pengujian ukuran populasi penentuan Beasiswa-PPA ditunjukkan pada Tabel 6.2.

Tabel 6.2 Hasil skenario pengujian ukuran populasi penentuan BBP-PPA

Ukuran Populasi	$cr = 0.4 ; mr = 0.2 ; generasi = 100$					Rata-rata nilai <i>fitness</i>
	Percobaan Ke - i					
	1	2	3	4	5	
20	0.943	0.954	0.951	0.945	0.948	0.948
40	0.977	0.968	0.974	0.976	0.98	0.975
60	0.979	0.987	0.98	0.98	0.985	0.982
80	0.984	0.993	0.971	0.991	0.985	0.984
100	0.98	0.976	0.99	0.987	0.987	0.984
120	0.99	0.991	0.987	0.986	0.985	0.987
140	0.982	0.982	0.984	0.988	0.988	0.984
160	0.995	0.988	0.986	0.98	0.984	0.986
180	0.987	0.983	0.978	0.991	0.99	0.985
200	0.992	0.986	0.985	0.985	0.989	0.987

Rata-rata nilai *fitness* terbesar pada pengujian ini untuk penentuan BBP-PPA terdapat pada ukuran populasi sebesar 80 dengan rata-rata nilai sebesar 0.984. Ukuran populasi 80 akan digunakan pada pengujian kombinasi nilai *cr* dan *mr* untuk penentuan BBP-PPA. Pada Gambar 6.2 ditunjukkan grafik dari rata-rata nilai *fitness* ukuran populasi.



Gambar 6.2 Grafik hasil rata-rata nilai *fitness* ukuran populasi

Gambar 6.1 dan 6.2 menunjukkan bahwa semakin besar ukuran populasi maka nilai *fitness* yang didapatkan juga semakin besar. Tetapi pada titik tertentu, pada kasus ini ukuran populasi sebesar 80 untuk penentuan Beasiswa-PPA dan BBP-PPA, penambahan ukuran populasi tidak memberikan kenaikan *fitness* yang signifikan bahwa mungkin didapatkan penurunan. Pola seperti ini juga didapatkan pada penelitian Pratiwi (2014) yang menerapkan algoritma genetika untuk optimasi biaya pemenuhan kebutuhan gizi dan Utomo (2012) yang melakukan penelitian mengenai penjadwalan *flexible job shop problem* dengan menggunakan *real coded genetic algorithm*.

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

Skenario pengujian kedua merupakan pengujian terhadap nilai *crossover rate* (*cr*) dan *mutation rate* (*mr*). Pada pengujian ini, nilai yang digunakan untuk menguji nilai *crossover rate* dan *mutation rate* adalah 1.0 dan 0.0, 0.9 dan 0.1, 0.8 dan 0.2, 0.7 dan 0.3, 0.6 dan 0.4, 0.5 dan 0.5, 0.4 dan 0.6, 0.3 dan 0.7, 0.2 dan 0.8, 0.1 dan 0.9, serta 0.0 dan 1.0. Pengujian ini bertujuan untuk mendapatkan nilai *cr*

dan mr yang tepat sehingga pada proses algoritma genetika menghasilkan hasil akhir yang optimal. Pengujian terhadap nilai *crossover rate* dan *mutation rate* untuk masing-masing penentuan Beasiswa-PPA dan BBP-PPA ditunjukkan pada sub-bab 6.2.1 dan 6.2.2.

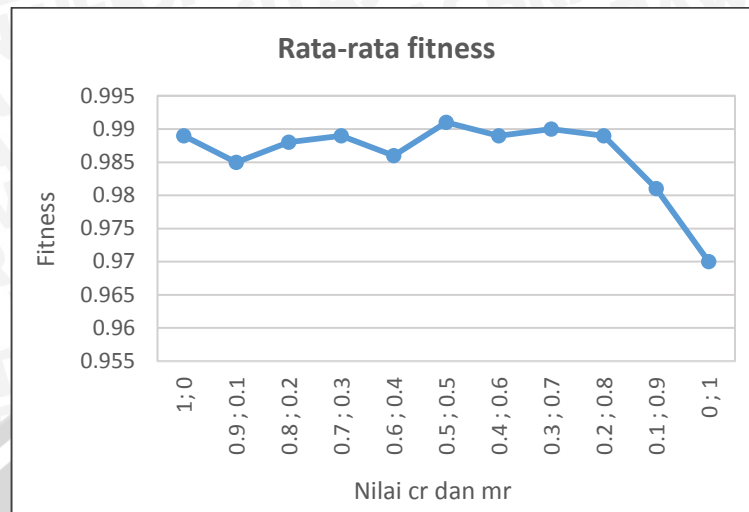
6.2.1 Pengujian Kombinasi Nilai Cr dan Mr Beasiswa-PPA

Pada pengujian kombinasi nilai cr dan mr untuk penentuan Beasiswa-PPA digunakan nilai parameter algoritma genetika ukuran populasi sebesar 80, yang didapat dari pengujian ukuran populasi dan jumlah generasi sebanyak 100 generasi. Tabel 6.3 merupakan hasil dari skenario pengujian yang masing-masing kombinasi nilai cr dan mr dilakukan sebanyak 5 kali.

Tabel 6.3 Hasil pengujian kombinasi nilai cr dan mr penentuan Beasiswa-PPA

Kombinasi nilai cr dan mr	<i>popSize</i> = 80 ; generasi = 100					Rata-rata nilai <i>fitness</i>
	Percobaan Ke - i					
	1	2	3	4	5	
1.0 ; 0	0.988	0.984	0.991	0.993	0.988	0.989
0.9 ; 0.1	0.98	0.993	0.988	0.988	0.978	0.985
0.8 ; 0.2	0.984	0.99	0.989	0.991	0.988	0.988
0.7 ; 0.3	0.985	0.86	0.993	0.987	0.995	0.989
0.6 ; 0.4	0.986	0.988	0.992	0.981	0.986	0.986
0.5 ; 0.5	0.989	0.988	0.994	0.991	0.989	0.991
0.4 ; 0.6	0.991	0.988	0.991	0.984	0.992	0.989
0.3 ; 0.7	0.992	0.988	0.993	0.986	0.992	0.99
0.2 ; 0.8	0.994	0.984	0.985	0.993	0.991	0.989
0.1 ; 0.9	0.974	0.984	0.984	0.982	0.985	0.981
0.0 ; 1.0	0.976	0.965	0.978	0.963	0.97	0.97

Berdasarkan hasil rata-rata nilai *fitness* pengujian dapat diketahui bahwa rata-rata nilai *fitness* terbesar terdapat pada kombinasi nilai $cr = 0.5$ dan $mr = 0.5$. Rata-rata nilai *fitness* yang didapat untuk nilai $cr = 0.5$ dan $mr = 0.5$ adalah sebesar 0.991. Pada grafik Gambar 6.3 ditunjukkan hasil rata-rata nilai *fitness* pengujian kombinasi nilai cr dan mr untuk penentuan Beasiswa-PPA.



Gambar 6.3 Grafik hasil rata-rata *fitness* nilai *cr* dan *mr*

6.2.2 Pengujian Kombinasi Nilai *Cr* dan *Mr* BBP-PPA

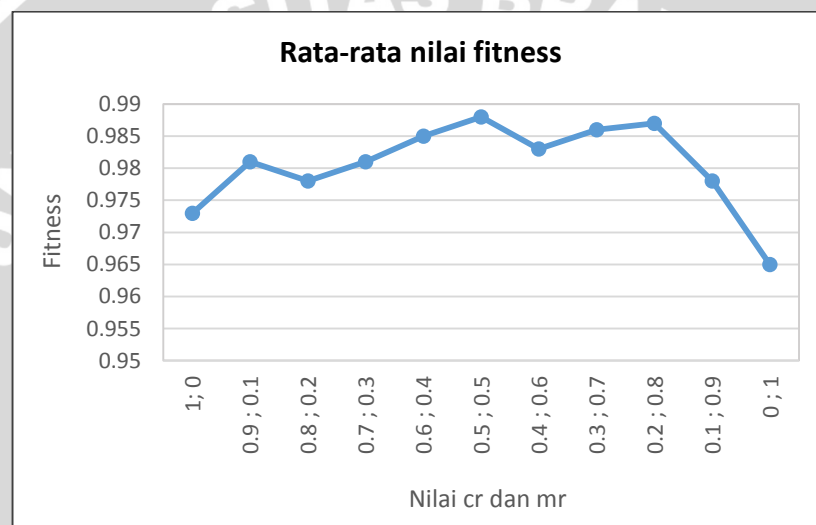
Pengujian kombinasi nilai *cr* dan *mr* untuk penentuan BBP-PPA digunakan nilai paramater algoritma genetika sama dengan pengujian kombinasi nilai *cr* dan *mr* untuk Beasiswa-PPA, dengan ukuran populasi sebesar 80 dan jumlah generasi sebanyak 100 generasi. Pengujian terhadap nilai *cr* dan *mr* dilakukan masing-masing sebanyak 5 kali untuk setiap nilai kombinasi *cr* dan *mr*. Pada Tabel 6.4 ditunjukkan hasil pengujian terhadap nilai *cr* dan *mr* untuk penentuan BBP-PPA.

Tabel 6.4 Hasil pengujian nilai *cr* dan *mr* penentuan BBP-PPA

Kombinasi nilai <i>cr</i> dan <i>mr</i>	<i>popSize</i> = 80 ; generasi = 100					Rata-rata nilai <i>fitness</i>
	Percobaan Ke - i					
	1	2	3	4	5	
1.0 ; 0	0.973	0.957	0.979	0.984	0.974	0.973
0.9 ; 0.1	0.984	0.981	0.984	0.98	0.976	0.981
0.8 ; 0.2	0.981	0.975	0.972	0.978	0.987	0.978
0.7 ; 0.3	0.987	0.979	0.983	0.973	0.984	0.981
0.6 ; 0.4	0.984	0.985	0.98	0.988	0.988	0.985
0.5 ; 0.5	0.989	0.988	0.988	0.99	0.986	0.988
0.4 ; 0.6	0.984	0.988	0.984	0.976	0.987	0.983
0.3 ; 0.7	0.988	0.986	0.99	0.988	0.982	0.986
0.2 ; 0.8	0.991	0.981	0.988	0.988	0.987	0.987

0.1 ; 0.9	0.98	0.974	0.981	0.978	0.978	0.978
0.0 ; 1.0	0.966	0.956	0.96	0.974	0.97	0.965

Pada hasil rata-rata nilai *fitness* pengujian terhadap nilai *cr* dan *mr* didapatkan rata-rata nilai *fitness* terbesar adalah 0.988 pada kombinasi nilai *cr* = 0.5 dan *mr* = 0.5. Kombinasi nilai tersebut merupakan nilai *cr* dan *mr* terbaik yang akan digunakan pada pengujian jumlah generasi untuk penentuan BBP-PPA. Pada grafik Gambar 6.4 ditunjukkan hasil rata-rata nilai *fitness* pengujian untuk penentuan BBP-PPA.



Gambar 6.4 Grafik hasil rata-rata nilai *fitness* nilai *cr* dan *mr*

Pada pengujian ini didapatkan hasil kombinasi nilai *cr* dan *mr* terbaik adalah sebesar 0.5 dan 0.5 untuk penentuan Beasiswa-PPA dan BBP-PPA. Maka, dapat diketahui analisis dari pengujian terhadap kombinasi nilai *cr* dan *mr* bahwa penggunaan kombinasi nilai *cr* tinggi dan *mr* rendah menyebabkan algoritma genetika lebih cepat mencapai konvergensi (*early convergence*) dan tidak mampu mencari solusi alternatif yang lebih baik. Sebaliknya, jika menggunakan nilai *cr* yang rendah dan nilai *mr* tinggi maka algoritma genetika akan bekerja seperti *random search* dan tidak mampu untuk mengeksplorasi daerah pencarian secara efektif (Mahmudy, 2014).

6.3 Hasil dan Analisis Skenario Pengujian Jumlah Generasi

Pada skenario pengujian ini akan dilakukan pengujian terhadap jumlah generasi yang dilakukan pada proses algoritma genetika sendiri. Jumlah generasi yang akan diujikan sebanyak 10 nilai jumlah generasi, antara lain 20, 30, 40, 50, 60, 70, 80, 90, 100, dan 150. Dalam pengujian jumlah generasi bertujuan untuk mendapatkan jumlah generasi terbaik dimana dapat menghasilkan hasil akhir yang optimal. Pengujian jumlah generasi untuk masing-masing penentuan Beasiswa-PPA dan BBP-PPA ditunjukkan pada sub-bab 6.3.1 dan 6.3.2.

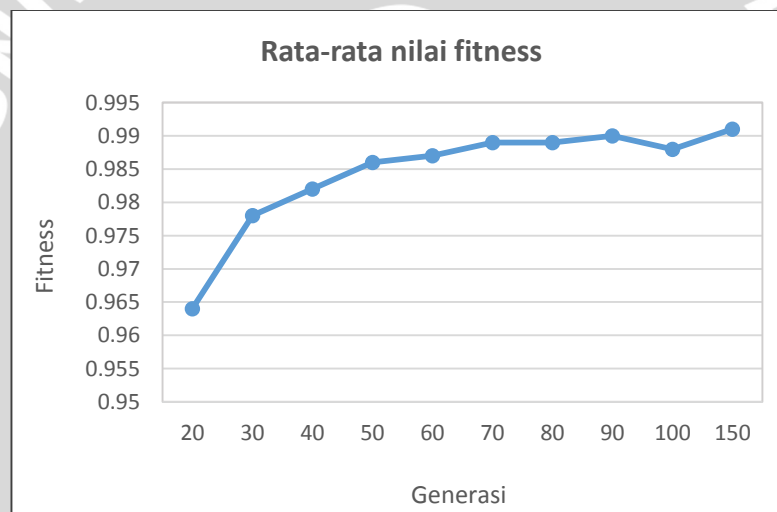
6.3.1 Pengujian Jumlah Generasi Beasiswa-PPA

Pengujian jumlah generasi untuk penentuan Beasiswa-PPA akan digunakan nilai parameter algoritma genetika, antara lain ukuran populasi sebanyak 80 populasi dan nilai $cr = 0.5$ dan $mr = 0.5$ yang masing-masing didapat dari pengujian sebelumnya. Hasil pengujian terhadap jumlah generasi ditunjukkan pada Tabel 6.5.

Tabel 6.5 Hasil pengujian jumlah generasi penentuan Beasiswa-PPA

Jumlah generasi	<i>popSize=80 ; cr = 0.5 ; mr = 0.5</i>					Rata-rata nilai <i>fitness</i>
	Percobaan Ke - i					
	1	2	3	4	5	
20	0.95	0.977	0.949	0.969	0.976	0.964
30	0.98	0.972	0.975	0.979	0.984	0.978
40	0.982	0.978	0.987	0.979	0.984	0.982
50	0.984	0.994	0.981	0.984	0.991	0.986
60	0.988	0.988	0.995	0.982	0.986	0.987
70	0.992	0.989	0.993	0.987	0.988	0.989
80	0.984	0.992	0.992	0.992	0.987	0.989
90	0.993	0.981	0.997	0.992	0.99	0.99
100	0.992	0.987	0.994	0.988	0.98	0.988
150	0.989	0.996	0.988	0.994	0.99	0.991

Berdasarkan dari Tabel 6.5 dapat diketahui bahwa jumlah generasi sebanyak 150 generasi merupakan jumlah generasi yang terbaik dengan rata-rata nilai *fitness* sebesar 0.991. Semakin besar jumlah generasi yang digunakan, maka semakin besar mendapatkan hasil akhir yang optimal. Tetapi, jika jumlah generasi yang terlalu besar dapat mengakibatkan waktu eksekusi algoritma genetika lebih lama dan belum pasti mendapatkan hasil akhir yang lebih optimal. Pada generasi lebih dari 150 tidak menunjukkan perubahan rata-rata nilai *fitness*. Sebaliknya, perubahan rata-rata nilai *fitness* cenderung bernilai sama (Pratiwi, 2014). Sebuah grafik pada Gambar 6.5 merepresentasikan hasil rata-rata nilai *fitness* jumlah generasi untuk penentuan Beasiswa-PPA.



Gambar 6.5 Grafik rata-rata nilai *fitness* jumlah generasi

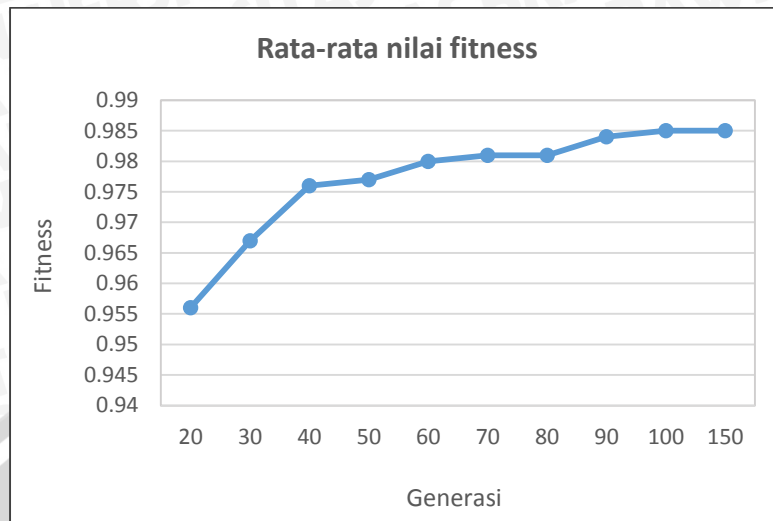
6.3.2 Pengujian Jumlah Generasi BBP-PPA

Pada pengujian jumlah generasi untuk penentuan BBP-PPA digunakan nilai parameter algoritma genetika sama dengan pengujian jumlah generasi untuk penentuan Beasiswa-PPA. Pada saat melakukan pengujian masing-masing jumlah generasi dilakukan pengujian sebanyak 5 kali. Tabel 6.6 merupakan hasil pengujian terhadap jumlah generasi untuk penentuan BBP-PPA.

Tabel 6.6 Hasil pengujian jumlah generasi penentuan BBP-PPA

Jumlah generasi	<i>popSize=80 ; cr = 0.5 ; mr = 0.5</i>					Rata-rata nilai <i>fitness</i>
	Percobaan Ke - i					
	1	2	3	4	5	
20	0.966	0.943	0.958	0.964	0.948	0.956
30	0.963	0.965	0.974	0.97	0.964	0.967
40	0.979	0.975	0.969	0.978	0.981	0.976
50	0.984	0.978	0.979	0.976	0.97	0.977
60	0.981	0.98	0.974	0.989	0.98	0.980
70	0.976	0.976	0.982	0.99	0.983	0.981
80	0.983	0.984	0.981	0.977	0.98	0.981
90	0.986	0.983	0.99	0.981	0.982	0.984
100	0.984	0.982	0.986	0.985	0.988	0.985
150	0.98	0.983	0.987	0.985	0.99	0.985

Setelah dihitung rata-rata nilai *fitness* untuk setiap jumlah generasi, dapat diketahui bahwa jumlah generasi sebanyak 100 generasi merupakan jumlah generasi terbaik dengan rata-rata nilai *fitness* sebesar 0.985. Sedangkan, pada jumlah generasi 20 memiliki rata-rata nilai *fitness* terendah dengan nilai sebesar 0.956. Maka dapat disimpulkan bahwa, semakin besar jumlah generasi yang digunakan, maka semakin besar mendapatkan hasil akhir yang optimal. Tetapi, jika jumlah generasi yang terlalu besar dapat mengakibatkan waktu eksekusi algoritma genetika lebih lama dan belum pasti mendapatkan hasil akhir yang lebih baik dari jumlah generasi sebelumnya. Hal itu dapat dilihat pada generasi 150 yang tidak menunjukkan perubahan rata-rata nilai *fitness*. Sebaliknya, perubahan rata-rata nilai *fitness* cenderung bernilai sama (Pratiwi, 2014). Pada Gambar 6.6 ditunjukkan grafik yang merepresentasikan rata-rata nilai *fitness* jumlah generasi untuk penentuan BBP-PPA.



Gambar 6.6 Grafik hasil rata-rata nilai *fitness* jumlah generasi

6.4 Hasil dan Analisis Akurasi Sistem

Setelah didapatkan nilai dari masing-masing parameter algoritma genetika, akan dilakukan pengujian terhadap akurasi sistem. Akurasi sistem merupakan pe-ranking-an data mahasiswa dengan menggunakan hasil optimasi dibandingkan dengan pe-ranking-an data mahasiswa menurut pakar. Pengujian akurasi sistem dilakukan sebanyak 5 kali untuk masing-masing penentuan Beasiswa-PPA dan BBP-PPA. Nilai parameter algoritma genetika yang digunakan adalah hasil dari pengujian skenario ukuran populasi, kombinasi nilai *cr* dan *mr*, dan jumlah generasi.

6.4.1 Akurasi Sistem Beasiswa-PPA

Pengujian akurasi sistem untuk penentuan Beasiswa-PPA digunakan ukuran populasi sebesar 80, kombinasi nilai *cr* dan *mr* yaitu 0.5 dan 0.5, dan jumlah generasi sebesar 150. Selanjutnya, data mahasiswa yang digunakan pada pengujian ini sebanyak 30 data mahasiswa. Pada Tabel 6.7 ditunjukkan hasil pengujian akurasi sistem untuk penentuan Beasiswa-PPA.

Tabel 6.7 Hasil akurasi sistem untuk Beasiswa-PPA

No.	Nilai akurasi sistem	Persentase akurasi
1.	0.982	98.2%
2.	0.985	98.5%
3.	0.988	98.8%

4.	0.989	98.9%
5.	0.987	98.7%

Nilai akurasi sistem didapatkan dengan menggunakan persamaan korelasi *spearman*, persamaan 2-8. Jika nilai akurasi sistem bernilai 1 maka hasil optimasi fungsi keanggotaan *fuzzy* Tsukamoto sama dengan hasil penentuan fungsi keanggotaan menurut pakar. Sebaliknya, jika bernilai -1 hasilnya bertolak belakang dengan pendapat pakar. Berdasarkan hasil pengujian akurasi sistem pada percobaan ke-4 memiliki persentase akurasi tertinggi sebesar 98.9%. Hasil *pe-ranking-an* data mahasiswa dengan persentasi akurasi 98.9% ditunjukkan pada Tabel 6.8.

Tabel 6.8 Hasil ranking penentuan Beasiswa-PPA

Nama	Ranking hasil optimasi	Ranking pendapat pakar
Mahasiswa 1	29	28
Mahasiswa 2	18	17
Mahasiswa 3	8	9
Mahasiswa 4	11	6
Mahasiswa 5	23	24
Mahasiswa 6	20	29
Mahasiswa 7	19	25
Mahasiswa 8	21	18
Mahasiswa 9	3	1
Mahasiswa 10	17	19
Mahasiswa 11	24	22
Mahasiswa 12	7	4
Mahasiswa 13	1	3
Mahasiswa 14	12	12
Mahasiswa 15	9	11
Mahasiswa 16	5	7
Mahasiswa 17	6	8
Mahasiswa 18	22	20

Mahasiswa 19	4	5
Mahasiswa 20	27	26
Mahasiswa 21	25	21
Mahasiswa 22	10	10
Mahasiswa 23	13	15
Mahasiswa 24	15	16
Mahasiswa 25	14	14
Mahasiswa 26	26	23
Mahasiswa 27	28	27
Mahasiswa 28	30	30
Mahasiswa 29	16	13
Mahasiswa 30	2	2

6.4.2 Akurasi Sistem BBP-PPA

Nilai parameter yang digunakan untuk pengujian akurasi sistem pada penentuan BBP-PPA berdasarkan hasil pengujian, antara lain ukuran populasi sebesar 80, kombinasi nilai $cr = 0.5$ dan $mr = 0.5$, dan jumlah generasi sebanyak 100 generasi. Pada Tabel 6.9 ditunjukkan hasil pengujian akurasi sistem untuk penentuan BBP-PPA.

Tabel 6.9 Hasil akurasi sistem untuk BBP-PPA

No.	Nilai akurasi sistem	Persentase akurasi
1.	0.985	98.5%
2.	0.986	98.6%
3.	0.987	98.7%
4.	0.971	97.1%
5.	0.984	98.4%

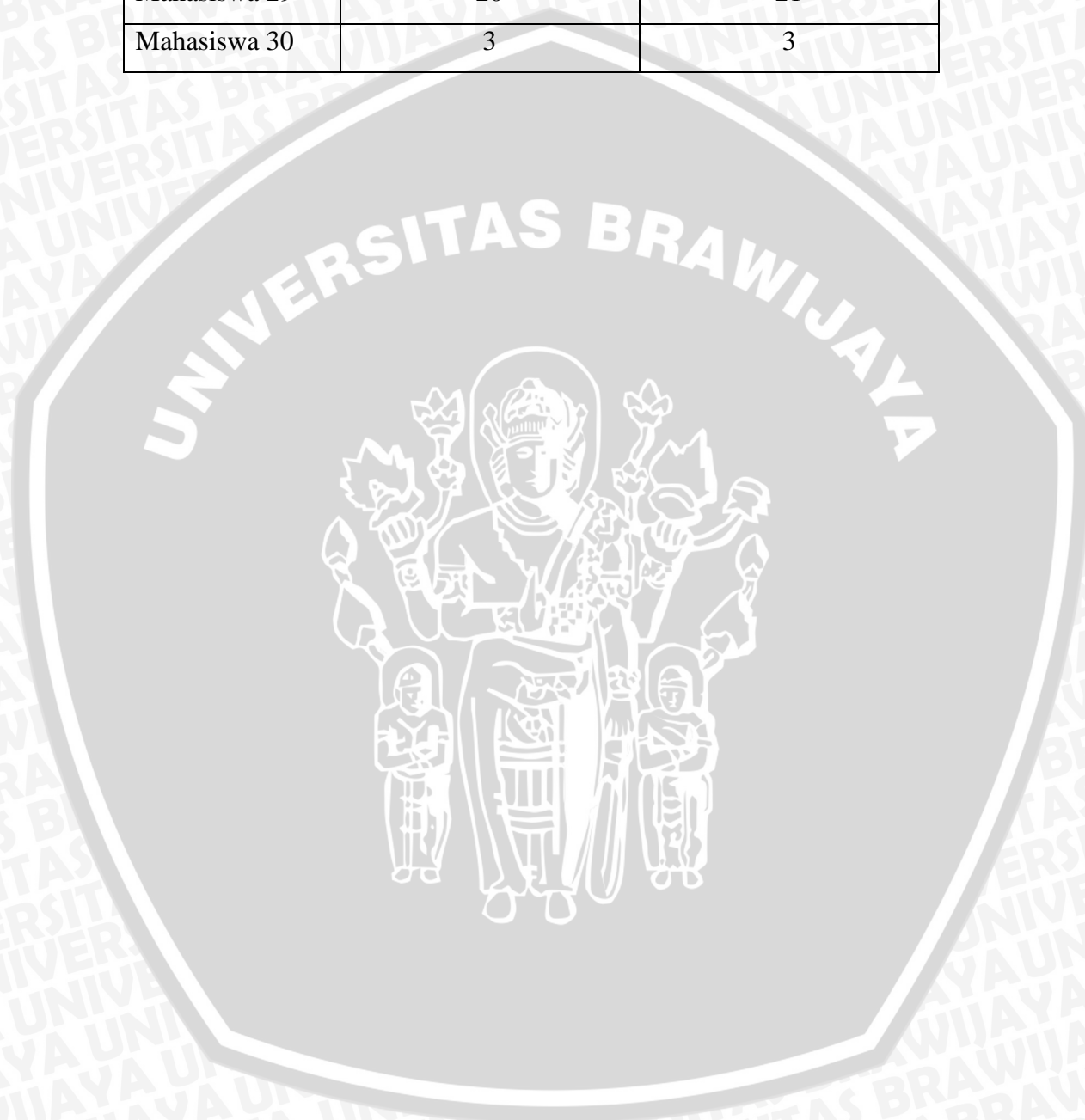
Berdasarkan hasil pengujian akurasi sistem pada percobaan ke-3 memiliki persentase akurasi tertinggi sebesar 98.7%. Sama halnya dengan penentuan Beasiswa-PPA, nilai akurasi sistem dihitung dengan menggunakan persamaan korelasi *spearman*. Jika nilai akurasi sistem bernilai 1 maka hasil optimasi fungsi keanggotaan *fuzzy* Tsukamoto sama dengan hasil penentuan

fungsi keanggotaan menurut pakar. Sebaliknya, jika bernilai -1 hasilnya bertolak belakang dengan pendapat pakar. Hasil perankingan data mahasiswa menggunakan optimasi fungsi keanggotaan dan menurut pendapat pakar ditunjukkan pada Tabel 6.10.

Tabel 6.10 Hasil ranking penentuan BBP-PPA

Nama	Ranking hasil optimasi	Ranking pendapat pakar
Mahasiswa 1	29	29
Mahasiswa 2	11	23
Mahasiswa 3	6	5
Mahasiswa 4	22	12
Mahasiswa 5	16	11
Mahasiswa 6	27	26
Mahasiswa 7	25	28
Mahasiswa 8	8	9
Mahasiswa 9	4	2
Mahasiswa 10	14	19
Mahasiswa 11	26	25
Mahasiswa 12	9	7
Mahasiswa 13	1	4
Mahasiswa 14	23	20
Mahasiswa 15	5	6
Mahasiswa 16	18	16
Mahasiswa 17	10	17
Mahasiswa 18	12	18
Mahasiswa 19	2	1
Mahasiswa 20	24	15
Mahasiswa 21	21	13
Mahasiswa 22	13	10
Mahasiswa 23	19	22
Mahasiswa 24	17	24
Mahasiswa 25	15	14

Mahasiswa 26	28	27
Mahasiswa 27	7	8
Mahasiswa 28	30	30
Mahasiswa 29	20	21
Mahasiswa 30	3	3



BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan perancangan, implementasi, dan pengujian yang dilakukan terhadap sistem optimasi fungsi keanggotaan *fuzzy* Tsukamoto menggunakan metode algoritma genetika dalam menentukan calon penerima Beasiswa-PPA dan BBP-PA diperoleh beberapa kesimpulan, antara lain:

1. Optimasi fungsi keanggotaan *fuzzy* Tsukamoto menggunakan metode algoritma genetika dapat digunakan dengan menggunakan representasi kromosom *real-coded*. Representasi kromosom ini mewakili nilai batas-batas fungsi keanggotaan *fuzzy* Tsukamoto. Selain itu, dengan proses *crossover* yang digunakan adalah *one-cut-point crossover* dan *random mutation* untuk proses mutasi, permasalahan optimasi fungsi keanggotaan *fuzzy* Tsukamoto mampu diselesaikan.
2. Hasil nilai parameter algoritma genetika yang didapat dari sistem optimasi fungsi keanggotaan *fuzzy* Tsukamoto menggunakan metode algoritma genetika adalah ukuran populasi sebesar 80 dengan rata-rata nilai *fitness* 0.989. Kombinasi nilai *cr* dan *mr* adalah 0.5 dan 0.5, dan jumlah generasi sebanyak 150 generasi untuk penentuan Beasiswa-PPA. Sedangkan untuk penentuan BBP-PPA, ukuran populasi terbaik adalah 80, kombinasi nilai *cr* dan *mr* adalah 0.5 dan 0.5, dan jumlah generasi sebanyak 100 generasi.
3. Hasil akurasi sistem dengan menggunakan perhitungan korelasi *spearman* didapatkan untuk penentuan Beasiswa-PPA sebesar 98.9% dan penentuan BBP-PPA sebesar 98.7%. Hasil akurasi ini menunjukkan peningkatan dari penelitian Iskandar (2012) yang membuktikan bahwa algoritma genetika mampu menghasilkan hasil akhir yang optimal.

7.2 Saran

Sistem optimasi fungsi keanggotaan *fuzzy* Tsukamoto menggunakan algoritma genetika pada pemilihan calon penerima Beasiswa-PPA dan BBP-PPA masih dapat dikembangkan lebih baik lagi. Saran yang diberikan mengenai sistem ini, antara lain:

1. Pengembangan dalam sistem ini yaitu dapat mengoptimasi *rules based* atau basis aturan yang terdapat dalam inferensi *fuzzy* Tsukamoto sehingga hasil akhir penentuan beasiswa menjadi lebih optimal dibandingkan dengan hanya optimasi fungsi keanggotaan.
2. Penggunaan metode *Hybrid Genetic Algorithms* (HGAs) merupakan metode yang dapat dikembangkan untuk sistem optimasi ini. Metode HGAs dapat memberikan hasil atau solusi yang optimal dan lebih efisien dibanding dengan metode algoritma genetika pada umumnya (Mahmudy, 2013).
3. Metode dalam proses reproduksi dan seleksi dapat diganti dengan metode lainnya sehingga hasil akhir yang didapat lebih beragam dan dapat diketahui metode apa yang tepat untuk menghasilkan hasil akhir yang optimal.

DAFTAR PUSTAKA

- Anggariawan, H., Dewi, C., & Setiawan, B.D., 2014. *Penerapan Algoritma Genetika Untuk Optimasi Fungsi Keanggotaan Fuzzy Inference System Model Sugeno Pada Perhitungan Angka Metabolisme Basal (AMB)*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol. 4, No. 4.
- Fattouh, A., & FadiFouz., 2012. *A Two-Stage Representation of Fuzzy Systems*. International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 3, pp. 2660-2665.
- Iskandar, F.M., Soebroto, A.A., & Regasari, R., 2012. *Sistem Pendukung Keputusan Pemilihan Calon Penerima Beasiswa PPA Dan BBM Menggunakan Metode Fuzzy AHP*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol. 1, No. 8.
- Kusumadewi, S., 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Penerbit Graha Ilmu.
- Laudon, K.C., & Laudon, J.P., 2008. *Sistem Informasi Manajemen, Edisi 10 Buku 2*. Jakarta: Penerbit Salemba Empat.
- Mahmudy, W.F., 2013. *Algoritma Evolusi*. Malang: Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
- Mahmudy, W.F., Marian, R.M., & Luong, L.H.S., 2014, *Hybrid Genetic Algorithms for Part Type Selection and Machine Loading Problems with Alternative Production Plans in Flexible Manufacturing System*. ECTI Transactions on Computer and Information Technology (ECTI-CIT), Vol. 8, No. 1, pp. 80-93.
- Mahmudy, W.F., Marian, R.M., & Luong, L.H.S., 2013, *Modeling and Optimization of Part Type Selection and Loading Problems in Flexible*

Manufacturing System Using Real Coded Genetic Algorithms, International Journal of Electrical, Electronic Science and Engineering, Vol. 7, No. 4, pp. 181-190.

Pradeka, R., Setiawan, A., & Linawati, L., 2012. *Uji Koefisien Korelasi Spearman dan Kendall Menggunakan Metode Bootstrap (Studi Kasus: Beberapa Kurs Mata Uang Asing Terhadap Rupiah)*. Salatiga: Seminar Nasional Matematika 2012. Fakultas Sains dan Matematika, Universitas Kristen Satya Wacana.

Pratiwi, M.I., Mahmudy, W.F., & Dewi, C. 2014. *Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya. Vol. 4, No. 6.

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

Sailah, I., 2014. *Pedoman Beasiswa dan Bantuan Biaya Pendidikan PPA 2014*. Jakarta: Direktorat Jenderal Pendidikan Tinggi.

Soetanto, T.V., & Soetanto, D.P., 1999. *Penjadualan Flowshop dengan Algoritma Genetika*. Jurnal Teknik Industri Universitas Kristen Petra. Vol. 1, No. 1, hal 4.

Suratno. 2011. *Pengaruh Perbedaan Tipe Fungsi Keanggotaan Pada Pengendali Logika Fuzzy Terhadap Tanggapan Waktu Sistem Orde Dua Secara Umum*. Jurusan Teknik Elektro, Fakultas Teknik, Universitas Diponegoro. Semarang.

Sutojo, T., Mulyanto, E., & Suhartono, V., 2011. *Kecerdasan Buatan*. Yogyakarta: Andi Publisher.

Suyanto., 2011. *Artificial Intelligence (Edisi Revisi)*. Bandung: Informatika.

repository.ub.ac.id

Utomo, M.C.C., Mahmudy, W.F., & Marji, 2014, *Penyelesaian Penjadwalan Flexible Job Shop Problem dengan Menggunakan Real Coded Genetic Algorithm*. DORO: Repository Jurnal Mahasiswa PTIIK Universitas Brawijaya, Vol. 3, No. 13.

Zadeh, L.A., 1965. "Fuzzy Sets". *Information and Control*. Vol. 8, Issue 3. Elsevier Inc. hal 338-353.

