

**PERANGKINAN PESERTA SERTIFIKASI GURU
MENGUNAKAN MODEL MULTIPLE ATTRIBUTE DECISION
MAKING (MADM) DENGAN MEMANFAATKAN ALGORITMA
GENETIKA DALAM PENCARIAN BOBOT ATRIBUTNYA**

SKRIPSI

Oleh :
ACHMAD DANI ZULFIKAR
0510963002 - 96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

UNIVERSITAS BRAWIJAYA



**PERANGKINAN PESERTA SERTIFIKASI GURU
MENGUNAKAN MODEL MULTIPLE ATTRIBUTE DECISION
MAKING (MADM) DENGAN MEMANFAATKAN ALGORITMA
GENETIKA DALAM PENCARIAN BOBOT ATRIBUTNYA**

SKRIPSI

Sebagai salah satu syarat untuk memperoleh
gelar Sarjana Komputer dalam bidang Ilmu Komputer

Oleh :

ACHMAD DANI ZULFIKAR

0510963002 - 96



**PROGRAM STUDI ILMU KOMPUTER
JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS BRAWIJAYA
MALANG
2011**

UNIVERSITAS BRAWIJAYA



LEMBAR PENGESAHAN SKRIPSI

PERANGKINAN PESERTA SERTIFIKASI GURU MENGUNAKAN MODEL MULTIPLE ATTRIBUTE DECISION MAKING (MADM) DENGAN MEMANFAATKAN ALGORITMA GENETIKA DALAM PENCARIAN BOBOT ATRIBUTNYA

Oleh:

ACHMAD DANI ZULFIKAR

0510963002-96

Setelah dipertahankan di depan Majelis Penguji
Pada tanggal 9 Agustus 2011
dan dinyatakan memenuhi syarat untuk memperoleh
gelar Sarjana Komputer dalam bidang Ilmu Komputer

Pembimbing I

Dian Eka R, SSi., Mkom.

NIP. 197306192002122001

Mengetahui,
Ketua Jurusan Matematika
Fakultas MIPA Universitas Brawijaya

Dr. Abdul Rouf Alghofari, MSc.

NIP. 196709071992031001

UNIVERSITAS BRAWIJAYA



LEMBAR PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Achmad Dani Zulfikar
NIM : 0510963002-96
Jurusan : Matematika
Program Studi : Ilmu Komputer
Penulis Skripsi berjudul : PERANGKINAN PESERTA
SERTIFIKASI GURU
MENGUNAKAN MODEL
MULTIPLE ATTRIBUTE
DECISION MAKING (MADM)
DENGAN MEMANFAATKAN
ALGORITMA GENETIKA
DALAM PENCARIAN BOBOT
ATRIBUTNYA

Dengan ini menyatakan bahwa :

1. Isi dari Skripsi yang saya buat adalah benar-benar karya sendiri dan tidak menjiplak karya orang lain, selain nama-nama yang termaktub di isi dan tertulis di daftar pustaka dalam Skripsi ini.
2. Apabila dikemudian hari ternyata Skripsi yang saya tulis terbukti hasil jiplakan, maka saya akan bersedia menanggung segala resiko yang akan saya terima.

Demikian pernyataan ini dibuat dengan segala kesadaran.

Malang, 9 Agustus 2011
Yang menyatakan,

(Achmad Dani Zulfikar)
NIM. 0510963002-96

UNIVERSITAS BRAWIJAYA



**PERANGKINAN PESERTA SERTIFIKASI GURU
MENGUNAKAN MODEL MULTIPLE ATTRIBUTE DECISION
MAKING (MADM) DENGAN MEMANFAATKAN ALGORITMA
GENETIKA DALAM
PENCARIAN BOBOT ATRIBUTNYA**

ABSTRAK

Salah satu tahapan pelaksanaan sertifikasi guru adalah penetapan peserta oleh dinas pendidikan provinsi dan kabupaten/kota. Tahapan ini didasarkan pada urutan prioritas : masa kerja sebagai guru, usia, pangkat atau golongan, beban kerja, tugas tambahan, dan prestasi kerja. Guna membantu tugas dinas pendidikan provinsi dan kabupaten/kota dalam pernagkingan peserta sertifikasi guru, digunakan metode *Multiple Attribute Decision Making* (MADM) yang di dalamnya memanfaatkan algoritma genetika sebagai metode untuk mencari bobot atribut setiap alternatifnya.

Algoritma genetika yang digunakan dalam skripsi ini menggunakan pengkodean kromosom biner, seleksi menggunakan roda roulette, *crossover* menggunakan *crossover* satu titik dan mutasi menggunakan mutasi biner. Data yang digunakan adalah data peserta sertifikasi guru dari Dinas Pendidikan Kabupaten Malang.

Setelah dilakukan ujicoba, Dari hasil ujicoba diperoleh hasil pada peluang *crossover* 0.9 dan peluang mutasi 0.1 diperoleh nilai fitness yang optimal.

UNIVERSITAS BRAWIJAYA



TEACHER CERTIFICATION PARTICIPANTS RANKING PROCESS USING MULTIPLE ATTRIBUTE DECISION MAKING (MADM) MODEL AND USING GENETIC ALGORITHM FOR WEIGHT ATTRIBUTE SEARCHING

ABSTRACT

Part of certification stage is determination of the participants by provincial education department and distric/city. This stage is based on priority order: duration as teacher, age, rank or class, workload and work achievement. In order to help the duty of the province education department and city/district determining teacher certification participants, Multiple Attribute Decision Making (MADM) which exploits genetic algorithm is used to calculate attribute weight of each alternative.

Genetic algorithm in this study are using binary chromosome encoding, roulette wheel selection, single point crossover and binary mutation. Teacher certification participants data taken from Education department of Malang district.

The experiment obtain crossover rate of 0.9 and mutation rate of 0.1 which is the optimal fitness value



UNIVERSITAS BRAWIJAYA



KATA PENGANTAR

Puji syukur Penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan segala Rahmat, Karunia dan Hidayah-Nya sehingga Penulis dapat menyelesaikan skripsi dengan judul ” **PERANGKINAN PESERTA SERTIFIKASI GURU MENGGUNAKAN MODEL MULTIPLE ATTRIBUTE DECISION MAKING (MADM) DENGAN MEMANFAATKAN ALGORITMA GENETIKA DALAM PENCARIAN BOBOT ATRIBUTNYA**”.

Dalam penyusunan skripsi ini penulis banyak menerima bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Dian Eka R, S.Si., MKom., selaku dosen pembimbing utama atas arahan serta bimbingannya dalam penyusunan skripsi ini
2. M. Tanzil Furqon, S.Kom., selaku dosen pembimbing pendamping atas arahan serta bimbingannya dalam penyusunan skripsi.
3. Drs. Marji, MT, selaku Ketua Program Studi Ilmu Komputer Universitas Brawijaya Malang
4. Dr. Abdul Rouf Alghofari, MSc., selaku Ketua Jurusan Matematika FMIPA Universitas Brawijaya.
5. Reza Andria S. ST., selaku dosen pembimbing akademik atas bimbingan serta nasehat selama penulis kuliah di program studi ilmu komputer jurusan matematika FMIPA Universitas Brawijaya Malang
6. Segenap bapak dan ibu dosen yang telah mendidik dan mengajarkan ilmunya kepada Penulis selama menempuh pendidikan di Program Studi Ilmu Komputer Jurusan Matematika FMIPA Universitas Brawijaya.
7. Segenap staf dan karyawan Jurusan Matematika FMIPA Universitas Brawijaya yang telah banyak membantu Penulis dalam pelaksanaan penyusunan skripsi ini.
8. Orang tua, dan kakak, Penulis atas segala doa restu, dukungan moril dan materiil kepada Penulis.
9. Rekan-rekan Ilmu Komputer Universitas Brawijaya yang telah memberikan dukungannya dalam penyusunan skripsi ini.

10. Semua pihak yang telah membantu terselesainya laporan ini yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa laporan skripsi ini tentunya tidak terlepas dari berbagai kekurangan dan kesalahan. Oleh karena itu, segala kritik dan saran yang bersifat membangun sangat Penulis harapkan dari berbagai pihak guna peningkatan kualitas penelitian serupa di masa mendatang.

Malang, Agustus 2011

Penulis



DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	iii
HALAMAN PERNYATAAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR GRAFIK	xix
DAFTAR LAMPIRAN	xxi
 BAB I PENDAHULUAN	 1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
 BAB II TINJAUAN PUSTAKA	 5
2.1 Algoritma Genetika	5
2.2 Komponen – Komponen Utama Algoritma Genetika	5
2.2.1 Teknik Penyandian	5
2.2.2 Prosedur Inisialisasi	5
2.2.3 Fungsi Evaluasi	6
2.2.4 Seleksi	6
2.2.5 Operator Genetika	8
2.2.6 Penentuan Parameter	10
2.3 <i>Multi Criteria Decision Making</i>	10
2.3.1 Konsep Dasar <i>Multi Attribute Decision Making</i> ...	11
2.3.2 Pencarian Bobot	14
2.3.3 Mencari Bobot Pendekatan Subjektif	16
2.3.4 Mencari Bobot dengan Algoritma Genetika	18
2.3.5 Proses Perangkingan	21
2.4 Kriteria Penetapan Peserta Sertifikasi Guru	21

BAB III METODE PENELITIAN.....	26
3.1 Tahap Rancangan Global Sistem.....	26
3.2 Sumber Data	26
3.3 Block Diagram dan Flowchart.....	26
3.4 Perancangan Sistem	37
3.5 Perancangan Interface.....	49
3.6 Perancangan Ujicoba	51
BAB IV IMPLEMENTASI DAN UJI COBA	53
4.1 Lingkungan Implementasi	53
4.1.1 Lingkungan Perangkat Keras	53
4.1.2 Lingkungan Perangkat Lunak	53
4.2 Implementasi	54
4.2.1 Input Data.....	54
4.2.1.1 Data Alternatif	55
4.2.2 Deskripsi Program.....	56
4.2.2.1 Proses Normalisasi	56
4.2.2.2 Pembentukan Matriks Saaty	57
4.2.2.3 Pembentukan Populasi Awal	58
4.2.2.4 Seleksi	60
4.2.2.5 <i>Crossover</i>	62
4.2.2.6 Mutasi.....	65
4.2.2.7 Perangkingan	68
4.3 Penerapan Aplikasi	70
4.4 Ujicoba dan Analisa Hasil.....	75
4.4.1 Pengaruh peluang <i>crossover</i> dan peluang mutasi terhadap nilai fitness.....	75
4.4.2 Pengaruh peluang <i>crossover</i> dan peluang mutasi terhadap nilai konvergensi.....	79
4.4.3 Membandingkan Hasil Perangkingan dengan Aplikasi dan Perangkingan Manual	82
BAB V KESIMPULAN DAN SARAN	86
5.1 Kesimpulan	86
5.2 Saran	87
DAFTAR PUSTAKA	88

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Matrik Saaty.....	14
Gambar 3.1 Blok Diagram Sistem.....	26
Gambar 3.2 Flowchart Sistem	27
Gambar 3.3 Flowchart Normalisasi	28
Gambar 3.4 Proses Algoritma Genetika	29
Gambar 3.5 Flowchart pembangkitan populasi awal dan evaluasi fungsi/fitness	30
Gambar 3.6 Flowchart seleksi	32
Gambar 3.7 Flowchart <i>Crossover</i>	33
Gambar 3.8 Flowchart mutasi	34
Gambar 3.9 Flowchart Perangkingan	36
Gambar 3.10 Rancangan Interface proses algoritma genetika	49
Gambar 3.11 Rancangan Interface proses input data alternatif.....	50
Gambar 4.1 Tampilan Utama Aplikasi.....	53
Gambar 4.2 Tampilan Data Alternatif	54
Gambar 4.3 Prosedur Normalisasi	56
Gambar 4.4 Pembentukan Matrik Saaty	56
Gambar 4.5 Pembentukan Populasi Awal	59
Gambar 4.6 Seleksi.....	60
Gambar 4.7 <i>Crossover</i>	63
Gambar 4.8 Mutasi	67
Gambar 4.9 Perangkingan	69
Gambar 4.10 Tampilan Matrik Alternatif.....	71
Gambar 4.11 Tampilan Matrik Normalisasi	71
Gambar 4.12 Tampilan Matrik Saaty	71
Gambar 4.13 Tampilan Input Parameter Genetika.....	72
Gambar 4.14 Tampilan Proses Algoritma Genetika.....	72
Gambar 4.15 Tampilan Hasil Proses Perangkingan	73
Gambar 4.16 Tampilan Proses Tambah Data.....	73
Gambar 4.17 Tampilan Proses Edit Data	74
Gambar 4.18 Tampilan Proses Hapus Data.....	75

UNIVERSITAS BRAWIJAYA



DAFTAR TABEL

Halaman

Tabel 2.1 Matrik Saaty	16
Tabel 2.2 Skala Perbandingan Saaty	17
Tabel 2.3 Skor Masa Kerja	22
Tabel 2.4 Skor Usia	22
Tabel 2.5 Skor Pangkat	23
Tabel 2.6 Skor Tugas Tambahan.....	23
Tabel 2.7 Skor Lomba dan Karya Akademik	24
Tabel 2.8 Skor Pembimbingan kepada Sejawat/Siswa	25
Tabel 3.1 Alternatif	37
Tabel 3.2 Populasi Awal	42
Tabel 3.3 Fitness Relatif dan Fitnes Kumulatif.....	43
Tabel 3.4 Bilangan Acak untuk Seleksi	43
Tabel 3.5 Kromosom Baru Hasil Seleksi	44
Tabel 3.6 Bilangan Acak untuk <i>Crossover</i>	44
Tabel 3.7 Kromosom yang Terpilih untuk <i>Crossover</i>	45
Tabel 3.8 Kromosom setelah <i>Crossover</i>	46
Tabel 3.9 Bilangan Acak untuk Mutasi	47
Tabel 3.10 Kromosom dan Posisinya yang Terkena Mutasi	47
Tabel 3.11 Kromosom setelah Mutasi.....	48
Tabel 3.12 Pengaruh peluang <i>cross over</i> dan peluang mutasi terhadap nilai fitness.....	51
Tabel 3.13 Pengaruh peluang <i>crossover</i> dan peluang mutasi terhadap konvergensi	51
Tabel 3.14 Perbandingan Hasil Perangkingan Manual dengan Aplikasi	52
Tabel 4.1 Kombinasi default parameter genetika	70
Tabel 4.2 Nilai <i>Fitness</i> pada peluang <i>crossover</i> 0.1 dan peluang mutasi yang berbeda	76
Tabel 4.3 Nilai <i>Fitness</i> pada peluang <i>crossover</i> 0.3 dan peluang mutasi yang berbeda	76
Tabel 4.4 Nilai <i>Fitness</i> pada peluang <i>crossover</i> 0.5 dan peluang mutasi yang berbeda	76
Tabel 4.5 Nilai <i>Fitness</i> pada peluang <i>crossover</i> 0.7 dan peluang mutasi yang berbeda	77

Tabel 4.6	Nilai <i>Fitness</i> pada peluang <i>crossover</i> 0.9 dan peluang mutasi yang berbeda	77
Tabel 4.7	Nilai konvergensi pada peluang <i>crossover</i> 0.1 dan peluang mutasi yang berbeda	79
Tabel 4.8	Nilai konvergensi pada peluang <i>crossover</i> 0.3 dan peluang mutasi yang berbeda	79
Tabel 4.9	Nilai konvergensi pada peluang <i>crossover</i> 0.5 dan peluang mutasi yang berbeda	80
Tabel 4.10	Nilai konvergensi pada peluang <i>crossover</i> 0.7 dan peluang mutasi yang berbeda	80
Tabel 4.11	Nilai konvergensi pada peluang <i>crossover</i> 0.9 dan peluang mutasi yang berbeda	80
Tabel 4.12	Perbandingan Hasil Perangkingan Manual dengan Aplikasi	82



DAFTAR GRAFIK

Halaman

Grafik 4.1 Perbandingan fitness dengan peluang mutasi dan peluang <i>crossover</i>	78
Grafik 4.2 Perbandingan konvergensi dengan peluang mutasi dan peluang <i>crossover</i>	81



UNIVERSITAS BRAWIJAYA



DAFTAR LAMPIRAN

Halaman

Lampiran 1 Data Peserta Sertifikasi Guru Kabupaten Malang tahun 2010	89
Lampiran 2 Data Urutan Peserta Sertifikasi Guru Kabupaten Malang tahun 2010 (Manual)	95
Lampiran 3 Data Urutan Peserta Sertifikasi Guru Kabupaten Malang tahun 2010 (Aplikasi)	101



UNIVERSITAS BRAWIJAYA



BAB I

PENDAHULUAN

1.1 Latar Belakang

Undang – undang nomor 14 tahun 2005 tentang guru dan dosen menyatakan bahwa guru adalah pendidik profesional dengan tugas utama mendidik, mengajar, membimbing, mengarahkan, melatih, menilai, dan mengevaluasi peserta didik pada pendidikan anak usia dini jalur pendidikan formal, pendidikan dasar, dan pendidikan menengah. Guru profesional harus memiliki kualifikasi akademik minimum sarjana (S-1) atau diploma empat (D-IV), menguasai kompetensi (*pedagogic, professional, social* dan kepribadian), memiliki sertifikat pendidik, sehat jasmani dan rohani, serta memiliki kemampuan untuk mewujudkan tujuan pendidikan nasional. Diharapkan agar guru sebagai tenaga profesional dapat berfungsi untuk meningkatkan martabat dan peran guru sebagai agen pembelajaran dan berfungsi untuk meningkatkan mutu pendidikan nasional. Dengan terlaksananya sertifikasi guru, diharapkan akan berdampak pada meningkatnya mutu pembelajaran dan mutu pendidikan secara berkelanjutan. Pelaksanaan sertifikasi guru dimulai pada tahun 2007 setelah diterbitkannya Peraturan Mendiknas Nomor 18 Tahun 2007 tentang sertifikasi bagi guru dalam jabatan. Tahapan pelaksanaan sertifikasi guru dimulai dengan pembentukan panitia pelaksanaan sertifikasi guru di tingkat provinsi dan kabupaten/kota, pemberian kuota kepada dinas pendidikan provinsi dan kabupaten/kota, dan penetapan peserta oleh dinas pendidikan provinsi dan kabupaten/kota.

Penetapan peserta oleh dinas pendidikan provinsi dan kabupaten/kota didasarkan pada urutan prioritas peserta sertifikasi guru. Kriteria urutan prioritasnya yaitu : masa kerja sebagai guru, usia, pangkat atau golongan, beban kerja, tugas tambahan, dan prestasi kerja.(Kemendiknas,2007)

Dalam penelitian ini, akan dibuat suatu sistem yang membantu tugas dinas pendidikan provinsi dan kabupaten/kota dalam merangking peserta sertifikasi guru sesuai dengan kriteria menggunakan metode *Multiple Attribute Decision Making* (MADM) yang di dalamnya memanfaatkan algoritma genetika sebagai metode untuk menghitung bobot atribut setiap alternatifnya.

Multiple Attribute Decision Making (MADM) sendiri merupakan bagian dari *Multiple Criteria Decission Making* (MCDM) . MCDM

adalah suatu metode pengambilan keputusan untuk menentukan alternatif terbaik dari sejumlah alternatif berdasarkan kriteria tertentu (Zimmerman, 1991). MCDM menghasilkan solusi yang terdekat dengan solusi ideal berdasarkan parameter – parameter yang merefleksikan perilaku dari decision maker (publish.uwo.ca, 2006). MCDM dibagi menjadi dua : *Multiple Attribute Decision Making* (MADM) dan *Multiple Objective Decision Making* (MODM) (Zimmerman, 1991). MADM adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu, sedangkan MODM digunakan untuk merancang alternatif terbaik (Hartati, 2006). Dengan demikian, model MADM dapat digunakan untuk menyelesaikan permasalahan sertifikasi guru, sebab alternatif – alternatif peserta sertifikasi yang akan disusun sudah diketahui sebelumnya untuk kemudian dilakukan penyusunan dan penetapan peserta sertifikasi guru.

Dalam MADM terjadi evaluasi terhadap m alternatif A_i ($i=1,2,3,4,\dots,m$) terhadap sekumpulan atribut C_j ($j=1,2,3,4,\dots,n$). (Hartati,2006). Salah satu tahap yang dilakukan pada MADM adalah menghitung bobot dari sekumpulan atribut C_j . Pada dasarnya, ada 2 pendekatan untuk mencari nilai bobot atribut, yaitu pendekatan subyektif dan pendekatan obyektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subyektif, nilai bobot ditentukan berdasarkan subyektifitas dari para pengambil keputusan. Sedangkan pada pendekatan obyektif, nilai bobot dihitung secara matematis sehingga mengabaikan subyektifitas dari pengambil keputusan (Ustinovichius, 2007).

Beberapa metode yang digunakan untuk proses penghitungan bobot antara lain : *Weigthed Least Square* dan *Eigenvector*. Kedua metode tersebut mampu memberikan nilai bobot yang lebih besar dari nol untuk semua atribut, namun *Weigthed Least Square* secara konsep lebih mudah dipahami dan lebih optimal hasilnya (Chu,1979). Metode *Weigthed Least Square* juga digunakan untuk menghitung bobot atribut pada MADM terutama pada pendekatan subjektif (Fan, 2005).

Optimasi adalah proses penyesuaian masukan atau karakteristik dari suatu perangkat, proses matematis, atau percobaan untuk menemukan output minimum atau maksimum. (Haupt,2004). Beberapa contoh algoritma optimasi : *exhaustive search*, *Nelder-Mead Downhill Simplex Method* dan algoritma genetika. Algoritma genetika sendiri adalah algoritma pencarian yang didasarkan atas mekanisme seleksi alami dan

evolusi biologis. Pada setiap generasi, himpunan baru dari deretan individu dibuat berdasarkan kecocokan pada generasi sebelumnya (Goldberg,1989). Di dalam algoritma genetika terdapat beberapa tahap yaitu, inisialisasi, seleksi, *crossover* dan mutasi.

Kelebihan algoritma genetika yaitu mampu menyelesaikan permasalahan yang memiliki banyak variabel, menghindari minimum lokal, mengoptimasi baik variabel diskrit maupun kontinu, sesuai digunakan untuk komputer paralel, dan tidak memerlukan fungsi turunan (Haupt,2004). Algoritma genetika dapat digunakan untuk menyelesaikan penghitungan bobot atribut pada permasalahan MADM dengan pendekatan subjektif (Kusumadewi, 2005).

Dalam penelitian ini, akan dilakukan pembuatan program implementasi algoritma genetika dalam menghitung bobot atribut pada MADM pendekatan subjektif dengan menggunakan studi kasus perbandingan peserta sertifikasi guru sesuai dengan kriteria guna membantu tugas dinas pendidikan kabupaten/kota. Pemilihan model MADM dipilih sebab didalam penetapan dan penyusunan peserta sertifikasi guru, alternatif – alternatif yang disediakan memiliki atribut dengan nilai bobot yang berbeda - beda. Pendekatan subjektif dipilih karena di dalam pendekatan subjektif, subjektifitas pengambil keputusan tidak diabaikan dari proses pencarian nilai bobot atribut MADM. Metode algoritma genetika dipilih karena permasalahan penghitungan bobot pendekatan subjektif adalah permasalahan optimasi.

1.2 Rumusan Masalah

- 1.2.1 Bagaimana implementasi algoritma genetika dalam menghitung bobot atribut pada MADM menggunakan studi kasus perbandingan peserta sertifikasi guru?
- 1.2.2 Bagaimana pengaruh peluang *crossover* dan peluang mutasi terhadap nilai *fitness*?
- 1.2.3 Bagaimana pengaruh peluang *crossover* dan peluang mutasi terhadap konvergensi?
- 1.2.4 Bagaimana perbandingan hasil perbandingan dari aplikasi dengan manual?

1.3 Batasan Masalah

- 1.3.1 Menggunakan pendekatan subjektif dalam menghitung bobot atribut pada MADM
- 1.3.2 Penelitian ini lebih memfokuskan pada algoritma genetika
- 1.3.3 Dalam algoritma genetika, pengujian dilakukan dengan membandingkan peluang *crossover* dan peluang mutasi yang bervariasi dengan fitness yang dihasilkan serta konvergensi.
- 1.3.4 Menggunakan data perangkingan secara manual sebagai sarana untuk menguji keberhasilan aplikasi untuk diimplementasikan

1.4 Tujuan

Penelitian ini bertujuan untuk :

- 1.4.1 Mengimplementasikan metode algoritma genetika dalam penghitungan bobot atribut MADM pendekatan subjektif pada studi kasus perangkingan peserta sertifikasi guru.
- 1.4.2 Melakukan pengujian terhadap implementasi metode algoritma genetika dalam penghitungan bobot atribut MADM pendekatan subjektif pada studi kasus perangkingan peserta sertifikasi guru

1.5 Manfaat

Penelitian ini bermanfaat untuk :

- 1.5.1 Dapat mengetahui implementasi metode algoritma genetika dalam menghitung bobot atribut MADM pendekatan subjektif.
- 1.5.2 Dapat membantu dinas pendidikan kabupaten/propinsi dalam penyusunan dan penetapan peserta sertifikasi guru

BAB II

TINJAUAN PUSTAKA

2.1 Algoritma Genetika

Algoritma Genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Algoritma Genetika pertama kali dikembangkan oleh John Holland dari Universitas Michigan. (Kusumadewi, 2005). Kelebihan algoritma genetika yaitu mampu menyelesaikan permasalahan yang memiliki banyak variabel, menghindari minimum lokal, mengoptimasi baik variabel diskrit maupun kontinu, sesuai digunakan untuk komputer paralel, dan tidak memerlukan fungsi turunan (Haupt, 2004).

2.2 Komponen – Komponen Utama Algoritma Genetika

2.2.1 Teknik Penyandian

Teknik penyandian di sini meliputi penyandian gen dan kromosom. Gen merupakan bagian dari kromosom. Satu gen biasanya akan mewakili variabel. Gen dapat direpresentasikan dalam bentuk : String bit, pohon, array bilangan real, daftar aturan, elemen permutasi, atau representasi lainnya. Demikian juga, kromosom dapat direpresentasikan dengan menggunakan :

String bit	:	10011, 01101, 11101 dst
Bilangan real	:	65.65, -67.98, 562.88, dst
Elemen permutasi	:	E2, E10, E5, dst
Daftar aturan	:	R1, R2, R3, dst

(Kusumadewi, 2005)

2.2.2 Prosedur Inisialisasi

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada. (Kusumadewi, 2005)

2.2.3 Fungsi Evaluasi

Ada 2 hal yang harus dilakukan dalam melakukan evaluasi kromosom, yaitu: evaluasi fungsi objektif (fungsi tujuan) dan konversi fungsi objektif ke dalam fungsi *fitness*. Secara umum, fungsi *fitness* diturunkan dari fungsi objektif dengan nilai yang tidak negatif. Apabila ternyata fungsi objektif memiliki nilai negatif, maka perlu ditambahkan suatu konstanta C agar nilai *fitness* yang terbentuk menjadi tidak negatif. (Kusumadewi, 2005)

2.2.4 Seleksi

Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit.

Ada beberapa metode seleksi dari induk, antara lain:

a. *Rank-based fitness assignment*

Pada *rank based fitness*, populasi diurutkan menurut nilai objektifnya. Nilai *fitness* dari tiap-tiap individu hanya tergantung pada posisi individu tersebut dalam urutan, dan tidak dipengaruhi oleh nilai objektifnya.

b. *Roulette wheel selection*

Metode roda roulette ini merupakan metode yang paling sederhana dan sering juga dikenal dengan nama *stochastic sampling with replacement*. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan *fitness*-nya. Sebuah bilangan random dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan random tersebut akan terseleksi. Proses ini diulang hingga diperoleh sejumlah individu yang diharapkan.

Algoritma seleksi roda roulette :

1. Hitung total *fitness* (F) :
 - $TotFitness = \sum F_k ; k = 1, 2, \dots, popsize$
2. Hitung *fitness* relative tiap individu
 - $P_k = F_k / TotFitness$
3. Hitung *fitness* kumulatif
 - $q_1 = p_1$
 - $q_k = q_{k-1} + p_k ; k=2, 3, \dots, popsize$
4. Pilih induk yang akan menjadi kandidat untuk di-*crossover* dengan cara :

- Bangkitkan bilangan random r
 - Jika $q_k \leq r$ dan $q_{k+1} > r$, maka pilih kromosom ke $(k+1)$ sebagai kandidat induk.
- c. *Stochastic universal sampling*
Stochastic universal sampling memiliki nilai bias nol dan penyebaran yang minimum. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran fitnessnya seperti halnya pada seleksi roda roulette. Kemudian diberikan sejumlah *pointer* sebanyak individu yang ingin diseleksi di pada garis tersebut. Andaikan N adalah jumlah individu yang akan diseleksi, maka jarak antar *pointer* adalah $1/N$, dan posisi *pointer* pertama diberikan secara acak pada range $[1, 1/N]$.
- d. *Local selection*
 Pada seleksi lokal, setiap individu yang berada di dalam konstrain tertentu disebut dengan nama lingkungan lokal. Interaksi antar individu hanya dilakukan di dalam wilayah tersebut. Lingkungan tersebut ditetapkan sebagai struktur dimana populasi tersebut terdistribusi. Lingkungan tersebut juga dapat dipandang sebagai kelompok pasangan-pasangan yang potensial.
- e. *Truncation selection*
 Pada metode-metode seleksi yang telah dijelaskan terdahulu, seleksi dilakukan secara alami, Pada seleksi dengan pemotongan ini, lebih berkesan sebagai seleksi buatan. Seleksi ini biasanya digunakan oleh populasi yang jumlahnya sangat besar. Pada metode ini, individu – individu diurutkan berdasarkan *fitness*-nya. Hanya individu – individu yang terbaik saja yang akan diseleksi sebagai induk. Parameter yang digunakan dalam metode ini adalah suatu nilai ambang *trunc* yang mengindikasikan ukuran populasi yang akan diseleksi sebagai induk yang berkisar antara 50%-10%. Individu – individu yang ada di bawah nilai ambang ini tidak akan menghasilkan keturunan
- f. *Tournament selection*
 Pada metode seleksi dengan *tournament* ini, akan ditetapkan suatu nilai *tour* untuk individu – individu yang dipilih secara random dari suatu populasi. Individu – individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk.

Parameter yang digunakan pada metode ini adalah ukuran tour yang bernilai antara 2 sampai N (jumlah individu dalam suatu populasi). (Kusumadewi, 2005)

2.2.5 Operator Genetika

Ada 2 operator genetika, yakni sebagai berikut:

a. Operator untuk melakukan *crossover*, yang terdiri dari :

i. *Crossover* bernilai real

a. *Crossover* diskret

Crossover diskret akan menukar nilai variabel antar kromosom induk.

b. *Crossover* intermediate

Crossover menengah merupakan metode *crossover* yang hanya dapat digunakan untuk variabel real (dan variabel yang bukan biner). Nilai variabel anak dipilih di sekitar dan antara nilai – nilai variabel induk.

c. *Crossover* garis

Pada dasarnya *crossover* garis ini sama dengan *crossover* menengah, hanya saja nilai alpha untuk semua variabel sama.

ii. *Crossover* bernilai biner

a. *Crossover* satu titik

Pada penyilangan satu titik, posisi penyilangan k ($k=1,2,\dots,N-1$) dengan N =panjang kromosom diseleksi secara random. Variabel-variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan anak.

b. *Crossover* banyak titik

Pada penyilangan banyak titik, m posisi penyilangan k_i ($k=1,2,\dots,N-1$) I ($I=1,2,\dots,m$) dengan N =panjang kromosom diseleksi secara random dan tidak diperbolehkan ada posisi yang sama, serta diurutkan naik. Variabel-variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan anak

c. *Crossover* seragam

Pada penyilangan seragam, setiap lokasi memiliki potensi sebagai tempat penyilangan. Sebuah mask penyilangan dibuat sepanjang panjang kromosom secara random yang menunjukkan bit-bit dalam mask yang mana induk akan mensuplai anak dengan bit-bit yang ada.

iii. *Crossover* dengan permutasi

Pada penyilangan dengan permutasi ini, kromosom-kromosom anak diperoleh dengan cara memilih sub-barisan suatu tour dari satu induk dengan tetap menjaga urutan dan posisi sejumlah kota yang mungkin terhadap induk yang lainnya.

Pada *crossover* ada satu parameter yang sangat penting yaitu peluang *crossover* (p_c). Peluang *crossover* menunjukkan rasio dari anak yang dihasilkan dalam setiap generasi dengan ukuran populasi. (Kusumadewi, 2005)

b. Mutasi

Setelah mengalami proses *crossover*, pada *offspring* dapat dilakukan mutasi. Variabel *offspring* dimutasi dengan menambahkan nilai random yang sangat kecil (ukuran langkah mutasi), dengan probabilitas yang rendah. Peluang mutasi (p_m) didefinisikan sebagai persentasi dari jumlah total gen pada populasi yang mengalami mutasi. Peluang mutasi mengendalikan banyaknya gen baru yang akan dimunculkan untuk dievaluasi. Jika peluang mutasi terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Tetapi bila peluang mutasi ini terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya, dan juga algoritma akan kehilangan kemampuan untuk belajar dari histori pencarian. Ada beberapa pendapat mengenai laju mutasi ini. Ada yang berpendapat bahwa, laju mutasi sebesar $1/n$ akan memberikan hasil yang cukup baik. Ada juga yang beranggapan bahwa laju mutasi tidak tergantung pada ukuran populasinya. Kromosom hasil mutasi harus diperiksa, apakah masih berada pada domain solusi, dan bila perlu bisa dilakukan perbaikan

i. Mutasi bernilai real

Pada mutasi bilangan real, ukuran langkah mutasi biasanya sangat sulit ditentukan. Ukuran yang kecil biasanya sering mengalami kesuksesan, namun adakalanya ukuran yang lebih besar akan berjalan lebih cepat

ii. Mutasi bernilai biner

Cara sederhana untuk mendapatkan mutasi biner adalah dengan mengganti satu atau beberapa nilai gen dari kromosom.

(Kusumadewi, 2005)

2.2.7 Penentuan Parameter

Yang disebut dengan parameter di sini adalah parameter kontrol algoritma genetika, yaitu: ukuran populasi (*popsize*), peluang *crossover* (p_c), dan peluang mutasi (p_m). Nilai parameter ini ditentukan juga berdasarkan permasalahan yang akan dipecahkan. (Kusumadewi, 2005)

2.3 Multi Criteria Decision Making

Multiple Criteria Decision Making (MCDM) adalah suatu metode pengambilan keputusan untuk menetapkan alternatif terbaik dari sejumlah alternatif berdasarkan beberapa kriteria tertentu. Kriteria biasanya berupa ukuran-ukuran, aturan-aturan atau standar yang digunakan dalam pengambilan keputusan. Berdasarkan tujuannya, MCDM dapat dibagi menjadi 2 model (Zimmerman, 1991): *Multi Attribute Decision Making* (MADM); dan *Multi Objective Decision Making* (MODM). Seringkali MADM dan MODM digunakan untuk menerangkan kelas atau kategori yang sama. MADM digunakan untuk menyelesaikan masalah – masalah dalam ruang diskret. Oleh karena itu, pada MADM biasanya digunakan untuk melakukan penilaian atau seleksi terhadap beberapa alternatif dalam jumlah yang terbatas. Sedangkan MODM digunakan untuk menyelesaikan masalah – masalah pada ruang kontinyu (seperti permasalahan pemrograman matematis). Secara umum dapat dikatakan bahwa, MADM menyeleksi alternatif terbaik dari sejumlah alternatif; sedangkan MODM merancang alternatif terbaik (Yoon, 1981).

Ada beberapa fitur umum yang akan digunakan dalam MCDM (Janko, 2005), yaitu:

- Alternatif, alternatif adalah objek – objek yang berbeda dan memiliki kesempatan yang sama untuk dipilih oleh pengambil keputusan.
- Atribut, atribut sering juga disebut sebagai karakteristik, komponen, atau kriteria keputusan. Meskipun pada kebanyakan kriteria bersifat satu level, namun tidak menutup kemungkinan adanya sub kriteria yang berhubungan dengan kriteria yang telah diberikan
- Konflik antar kriteria, beberapa kriteria biasanya mempunyai konflik antara satu dengan lainnya, misalnya kriteria keuntungan akan mengalami konflik dengan kriteria biaya.
- Bobot keputusan, bobot keputusan menunjukkan kepentingan relatif dari setiap kriteria, $W = (w_1, w_2, \dots, w_n)$. Pada MCDM akan dicari bobot kepentingan dari setiap kriteria.
- Matriks keputusan, suatu matriks keputusan X yang berukuran $m \times n$, berisi elemen-elemen x_{ij} , yang merepresentasikan rating dari alternatif A_i ($i=1,2,\dots,m$) terhadap kriteria C_j ($j=1,2,\dots,n$).

2.3.1 Konsep Dasar *Multi Attribute Decision Making* (MADM)

Pada dasarnya, proses MADM dilakukan melalui 3 tahap, yaitu penyusunan komponen-komponen situasi, analisis, dan sintesis informasi (Rudolphi, 2000). Pada tahap penyusunan komponen-komponen situasi, akan dibentuk tabel taksiran yang berisi identifikasi alternatif dan spesifikasi tujuan, kriteria dan atribut. Salah satu cara untuk menspesifikasikan tujuan situasi $|O_i, = i=1,\dots,t|$ adalah dengan cara mendaftar konsekuensi-konsekuensi yang mungkin dari alternatif yang telah teridentifikasi $|A_i, = i=1,\dots,n|$. Selain itu juga disusun atribut-atribut yang akan digunakan $|A_k, = k=1,\dots,m|$. (Hartati dkk, 2006)

Tahap analisis dilakukan melalui dua langkah. Pertama, mendaftarkan taksiran dari besaran yang potensial, kemungkinan, dan ketidakpastian yang berhubungan dengan dampak-dampak yang mungkin pada setiap alternatif. Kedua, meliputi pemilihan dari preferensi pengambil keputusan untuk setiap nilai, dan ketidakpedulian terhadap resiko yang timbul. Pada langkah pertama, beberapa metode menggunakan fungsi distribusi $|p_j(x)|$ yang menyatakan probabilitas kumpulan atribut $|a_k|$ terhadap setiap alternatif $|A_i|$. Konsekuensi juga dapat ditentukan secara langsung dari agregasi sederhana yang dilakukan pada informasi terbaik yang tersedia. Demikian pula, ada beberapa cara untuk

menentukan preferensi pengambil keputusan pada setiap konsekuensi yang dapat dilakukan pada langkah kedua. Metode yang paling sederhana adalah untuk menurunkan bobot atribut dan kriteria adalah dengan fungsi utilitas atau penjumlahan terbobot. (Hartati dkk, 2006)

Secara umum, model *multi-attribute decision making* dapat didefinisikan sebagai berikut (Zimmermann, 1991) : Definisi 3.1: Misalkan $A = \{a_i \mid i = 1, \dots, n\}$ adalah himpunan alternatif-alternatif keputusan dan $C = \{c_j \mid j = 1, \dots, m\}$ adalah himpunan tujuan yang diharapkan, maka akan ditentukan alternatif x^0 yang memiliki derajat harapan tertinggi terhadap tujuan – tujuan yang relevan c_j .

Sebagian besar pendekatan MADM dilakukan melalui 2 langkah, yaitu: pertama, melakukan agregasi terhadap keputusan-keputusan yang tanggap terhadap semua tujuan pada setiap alternatif; kedua, melakukan perbandingan alternatif-alternatif keputusan tersebut berdasarkan hasil agregasi keputusan. (Hartati, 2006)

Dengan demikian, bisa dikatakan bahwa, masalah *multi-attribute decision making* (MADM) adalah mengevaluasi m alternatif A_i ($i=1,2,\dots,m$) terhadap sekumpulan atribut atau kriteria C_j ($j=1,2,\dots,n$), dimana setiap atribut saling tidak bergantung satu dengan yang lainnya. Matriks keputusan setiap alternatif terhadap setiap atribut, X , diberikan sebagai:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \quad (2.1)$$

Dimana x_{ij} merupakan rating kinerja alternatif ke- i terhadap atribut ke- j . Nilai bobot yang menunjukkan tingkat kepentingan relatif setiap atribut, diberikan sebagai, W :

$$W = \{w_1, w_2, \dots, w_n\} \quad (2.2)$$

Rating kinerja (X), dan nilai bobot (W) merupakan nilai utama yang merepresentasikan preferensi absolut dari pengambil keputusan. Masalah MADM diakhiri dengan proses perbandingan untuk mendapatkan alternatif terbaik yang diperoleh berdasarkan nilai keseluruhan preferensi yang diberikan (Yeh, 2002).

Ada beberapa metode yang dapat digunakan untuk menyelesaikan masalah MADM, antara lain:

- Simple Additive Weighting Method* (SAW)
- Weighted Product (WP)* (Hartati dkk, 2006)

Sebagaimana konsep tentang permasalahan MADM, akan diseleksi sekumpulan alternatif yang didasarkan atas beberapa atribut. Misalkan $S = \{S_1, S_2, \dots, S_m\}$ adalah himpunan alternatif; $C = \{C_1, C_2, \dots, C_n\}$ adalah himpunan atribut (kriteria), dan $A = \{a_{ij} \mid i = 1, 2, \dots, m; j = 1, 2, \dots, n\}$ adalah matriks keputusan dengan a_{ij} adalah nilai numeris alternatif ke- i pada atribut ke- j .

Sebelumnya pada matriks A dilakukan normalisasi terlebih dahulu, sehingga nilai a_{ij} terletak pada range $[0,1]$. Misalnya matriks B adalah matriks yang elemen-elemennya adalah elemen-elemen matriks A yang sudah dinormalisasikan dengan rumus (Fan, 1999):

$$b_{ij} = \frac{a_j^{\text{Max}} - a_{ij}}{a_j^{\text{Max}} - a_j^{\text{Min}}} \quad (2.3)$$

untuk C_{ij} adalah atribut biaya

$$b_{ij} = \frac{a_{ij} - a_j^{\text{Min}}}{a_j^{\text{Max}} - a_j^{\text{Min}}} \quad (2.4)$$

untuk C_{ij} adalah atribut keuntungan dengan

$$a_j^{\text{Max}} = \max \{a_{1j}, a_{2j}, \dots, a_{mj}\} \quad (2.5)$$

$$a_j^{\text{Min}} = \min \{a_{1j}, a_{2j}, \dots, a_{mj}\} \quad (2.6)$$

$$i = 1, 2, 3, \dots, m; j = 1, 2, 3, \dots, n$$

kriteria biaya adalah kriteria yang diminimumkan nilainya, sedangkan kriteria keuntungan adalah kriteria yang dimaksimumkan nilainya.

untuk menyelesaikan permasalahan ini, maka yang perlu ditetapkan sebelumnya adalah nilai alternatif dari setiap atribut dan matriks perbandingan berpasangan antar atribut (terutama untuk pendekatan

subjektif). Apabila kedua hal tersebut telah diketahui, maka pertama-tama harus dicari terlebih dahulu bobot setiap atribut. (Hartati dkk, 2006)

2.3.2 Pencarian Bobot

Permasalahan dasar dalam teori himpunan fuzzy (Bellman, 1966) adalah pencarian derajat keanggotaan dari setiap anggota kepada himpunan tersebut. Saaty (Saaty, 1977) menyederhanakan permasalahan ini menjadi sebuah masalah matriks eigenvalue. Metode *imbedding* (Kalaba, 1979) digunakan untuk mendapatkan *eigenvalue* dan *eigenvector* dari matriks tersebut. Misalkan $w_i > 0$, $i = 1, 2, \dots, n$, merupakan derajat keanggotaan dari n anggota. Saaty merancang matriks bobot relatif yang semua nilai didalamnya adalah w_i/w_j . Kemudian menyatakan bahwa vektor $(w_1, w_2, \dots, w_n)^T$ adalah sebuah *eigenvector* yang anggotanya adalah *eigenvalue* terbesar (Akar Perron-Frobenius) dari matriks tersebut. Sedangkan semua *eigenvalue* yang lain adalah nol. Saaty memisalkan objek tersebut dengan $A_1, \dots, A_n$ dan bobotnya dengan $w_1, \dots, w_n$ dimana n adalah banyaknya kriteria yang akan dibandingkan. Perbandingan berpasangannya dibuat dalam bentuk matriks

$$A = \begin{matrix} & \begin{matrix} \textcircled{A_1} & \textcircled{A_2} & \dots & \textcircled{A_n} \end{matrix} \\ \begin{bmatrix} w_1/w_1 & w_1/w_2 & \dots & w_1/w_n \\ w_2/w_1 & w_2/w_2 & \dots & w_2/w_n \\ \vdots & \vdots & \ddots & \vdots \\ w_n/w_1 & w_n/w_2 & \dots & w_n/w_n \end{bmatrix} & \begin{matrix} \textcircled{A_1} \\ \textcircled{A_2} \\ \vdots \\ \textcircled{A_n} \end{matrix} \end{matrix}$$

Gambar 2.1 Matrik Saaty

Matriks ini disebut matriks resiprokal yang bernilai positif untuk setiap nilainya dan memenuhi persyaratan resiprokal

$$a_{ji} = 1/a_{ij} \quad (2.7)$$

dengan mengalikan matriks A dengan vektor $(w_1, w_2, \dots, w_n)^T$ didapatkan

$$Aw = nw, \quad (2.8)$$

atau

$$(A - nI)w = 0. \quad (2.9)$$

Ini merupakan persamaan linier homogen yang memiliki solusi non-trivial, jika determinan dari $(A - \lambda I)$ tidak ada, dengan kata lain, λ adalah *eigenvalue* dari A . Matriks A juga konsisten, dengan kata lain

$$a_{jk} = a_{ik} / a_{ij} \quad (2.10)$$

Dalam kasus umum, nilai-nilai yang tepat dari w_i/w_j tidak diketahui dan harus diperkirakan. Karena *eigenvalue* dipengaruhi koefisien maka persamaan 2.8 menjadi

$$A'w' = \lambda_{\max} w', \quad (2.11)$$

dimana, $\lambda_{\max}$ adalah *eigenvalue* terbesar dari A' . Bentuk sederhana dari persamaan 2.11 adalah

$$Aw = \lambda_{\max} w, \quad (2.12)$$

dimana matriks A adalah matriks perbandingan berpasangan Saaty. *Eigenvektor* yang anggotanya adalah *eigenvalue* terbesar adalah vektor bobot yang diinginkan. Kalaba dan Spingran (Kalaba,) menggunakan metode numerik untuk mendapatkan vektor bobot tersebut.

Metode lain adalah metode *Weigthed Least Square*. Berdasarkan elemen a_{ij} dari matriks Saaty A pada persamaan 2.12 untuk menghitung bobot w_{ij} , diberikan

$$a_{ij} \approx w_i / w_j \quad (2.13)$$

Bobot dapat diperoleh dengan menyelesaikan persamaan optimasi yang memiliki *constrain*, yaitu

$$S = \sum_{i=1}^n \sum_{j=1}^n (a_{ij}w_j - w_i)^2, \quad (2.14)$$

$$\sum_{i=1}^n w_i = 1, \quad (2.15)$$

minimize S .

Kedua metode tersebut mampu memberikan nilai bobot yang lebih besar dari nol untuk semua atribut, namun *Weigthed Least Square* secara konsep lebih mudah dipahami dan lebih optimal hasilnya. (Chu, 1979).

Pada dasarnya, ada 2 pendekatan yang umum digunakan untuk mencari nilai bobot atribut, yaitu pendekatan subjektif dan pendekatan objektif. Masing-masing pendekatan memiliki kelebihan dan kelemahan. Pada pendekatan subjektif, nilai bobot ditentukan berdasarkan subjektifitas dari para pengambil keputusan, sehingga beberapa faktor dalam proses perangkingan alternatif bisa ditentukan secara bebas.

Sedangkan pada pendekatan objektif, nilai bobot dihitung secara matematis sehingga mengabaikan subjektifitas dari pengambil keputusan. (Ustinovichius, 2007)

2.3.3 Mencari Bobot Pendekatan Subjektif

Misalkan $S = \{S_1, S_2, \dots, S_m\}$ adalah himpunan alternatif; $C = \{C_1, C_2, \dots, C_n\}$ adalah himpunan atribut (kriteria), dan $A = \{a_{ij} \mid i = 1, 2, \dots, m; j = 1, 2, \dots, n\}$ adalah matriks keputusan dengan a_{ij} adalah nilai numeris alternatif ke- i pada atribut ke- j .

Misalkan pengambil keputusan memberikan matriks keputusan $D = \{d_{kj} \mid k, j = 1, 2, \dots, n\}$ yang didasarkan pada matriks Saaty (Sawaragi, 1987), dengan elemen – elemen D mengikuti batasan :

$$d_{ij} > 0; d_{jk} = 1/d_{kj}, \text{ dan } d_{kk} = 1; k, j = 1, 2, \dots, n. \quad (2.16)$$

d_{kj} menunjukkan bobot relatif atribut C_k terhadap atribut C_j

Tabel 2.1 Matriks Saaty

attribute	1	2		p		m-1	m
1	1	$\frac{w(1)}{w(2)}$		$\frac{w(1)}{w(p)}$		$\frac{w(1)}{w(m-1)}$	$\frac{w(1)}{w(m)}$
2	$\frac{w(2)}{w(1)}$	1		$\frac{w(2)}{w(p)}$		$\frac{w(2)}{w(m-1)}$	$\frac{w(2)}{w(m)}$
.....							
p	$\frac{w(p)}{w(1)}$	$\frac{w(p)}{w(2)}$		1		$\frac{w(p)}{w(m-1)}$	$\frac{w(p)}{w(m)}$
.....							
m-1	$\frac{w(m-1)}{w(1)}$	$\frac{w(m-1)}{w(2)}$		$\frac{w(m-1)}{w(p)}$		1	$\frac{w(m-1)}{w(m)}$
m	$\frac{w(m)}{w(1)}$	$\frac{w(m)}{w(2)}$		$\frac{w(m)}{w(p)}$		$\frac{w(m)}{w(m-1)}$	1

Tabel 2.1 merupakan bentuk dari matriks keputusan (D) yang didasarkan pada matriks Saaty (Sawaragi, 1987) . Nilai tiap elemen matriks tersebut berdasarkan bobot preferensi yang diberikan oleh pengambil keputusan. Bobot preferensi tersebut menunjukkan tingkat kepentingan satu kriteria terhadap kriteria yang lain. Skala yang digunakan adalah skala perbandingan Saaty (Saaty, 2008) yaitu :

Tabel 2.2 Skala Perbandingan Saaty

<i>Intensity of Importance</i>	<i>Definition</i>	<i>Explanation</i>
1	Equal Importance	Two activities contribute equally to the objective
2	Weak or slight	
3	Moderate importance	Experience and judgement slightly favour one activity over another
4	Moderate plus	
5	Strong importance	Experience and judgement strongly favour one activity over another
6	Strong plus	
7	Very strong or demonstrated importance	An activity is favoured very strongly over another; its dominance demonstrated in practice
8	Very, very strong	
9	Extreme importance	The evidence favouring one activity over another is of the highest possible order of affirmation
Reciprocals of above	If activity <i>i</i> has one of the above non-zero numbers assigned to it when compared with activity <i>j</i> , then <i>j</i> has the reciprocal value when compared with <i>i</i>	A reasonable assumption

Sebagai contoh, misalkan akan dilakukan perbandingan berpasangan atas empat kriteria A_1 , A_2 , A_3 dan A_4 . Pengambil keputusan memberikan bobot preferensinya, sebagai berikut:

- A_1 banding A_2 adalah 2
- A_1 banding A_3 adalah 4
- A_1 banding A_4 adalah 9
- A_2 banding A_3 adalah 3
- A_2 banding A_4 adalah 5
- A_3 banding A_4 adalah 7

Dengan demikian matriks perbandingan berpasangannya dapat dinyatakan dengan

$$\begin{bmatrix} 1 & 2 & 4 & 9 \\ 1/2 & 1 & 3 & 5 \\ 1/4 & 1/3 & 1 & 7 \\ 1/9 & 1/5 & 1/7 & 1 \end{bmatrix}$$

Misalkan w_j ($j = 1, 2, \dots, n$) adalah bobot yang menunjukkan kepentingan relatif dari atribut C_j , dengan

$$W_j \in G = \{w_j \geq 0, j = 1, 2, \dots, n; \sum_{j=1}^n w_j = 1\} \quad (2.17)$$

maka langkah selanjutnya adalah bagaimana mencari nilai bobot w_j ini. (Kusumadewi, 2004)

Dalam metode ini prinsip penghitungan bobot metode *Weighed Least Square* dipakai untuk menghitung bobot pada pendekatan subjektif (Fan, 1999). Pada pendekatan subjektif, bobot – bobot w_j $\{j = 1, 2, \dots, n\}$ dapat diselesaikan menggunakan metode *Weighted Least Square*, yaitu (Chu, 1979) .

Minimumkan :

$$z_1 = \sum_{k=1}^n \sum_{j=1}^n [(d_{kj}w_j - w_k)]^2 \quad (2.18)$$

Dengan batasan :

$$\sum_{j=1}^n w_j = 1 \quad (2.19)$$

$$w_j \geq 0$$

(Hartati dkk, 2006)

2.3.4 Mencari Bobot dengan Algoritma Genetika

Optimasi adalah proses penyesuaian masukan atau karakteristik dari suatu perangkat, proses matematis, atau percobaan untuk menemukan output minimum atau maksimum. Inputnya terdiri dari variabel proses atau fungsi dikenal sebagai fungsi biaya, fungsi tujuan atau fungsi fitness dan outputnya adalah biaya atau fitness. Jika prosesnya merupakan sebuah percobaan, maka variabelnya adalah input fisik ke dalam percobaan tersebut. Apabila biaya adalah sesuatu yang harus diminimalkan, maka optimasi bisa juga disebut minimasi (Haupt, 2004). Beberapa contoh algoritma minimasi yaitu :

1. *Exhaustive Search*,

Pelacakan *exhaustive* membutuhkan jumlah yang sangat besar dari evaluasi fungsi biaya untuk menemukan hasil yang optimal. Dengan sampling yang cukup baik, pencarian *exhaustive* tidak terjebak di lokal minima dan bekerja baik untuk variabel kontinu atau diskontinu. Namun, metode ini membutuhkan waktu yang sangat lama untuk menemukan minimum global. Kekurangan lain dari pendekatan ini adalah minimum global mungkin saja terlewatkan karena *undersampling*. *Undersampling* sangat mudah terjadi saat waktu yang dibutuhkan untuk menghitung fungsi biaya sangat lama. Oleh karena itu pencarian *exhaustive* hanya

praktis untuk sejumlah variabel yang sedikit dan ruang pencarian terbatas (Haupt,2004)

2. *Nelder-Mead Downhill Simplex Method*,

Nelder dan Mead memperkenalkan metode ini dengan tidak membutuhkan penghitungan turunan. Tujuan dari metode ini adalah memindahkan simplex sampai di sekitar minimum, kemudian menetapkan simplex di sekitar minimum sampai batas kesalahan tertentu. Kelemahan dari metode ini adalah dapat terjebak pada lokal minima (Haupt,2004).

3. Algoritma Genetika

Algoritma genetika sendiri adalah algoritma pencarian yang didasarkan atas mekanisme seleksi alami dan evolusi biologis. Pada setiap generasi, himpunan baru dari deretan individu dibuat berdasarkan kecocokan pada generasi sebelumnya (Goldberg,1989). Di dalam algoritma genetika terdapat beberapa tahap yaitu, inisialisasi, seleksi, *crossover* dan mutasi. (Hartati dkk, 2006). Kelebihan algoritma genetika yaitu mampu menyelesaikan permasalahan yang memiliki banyak variabel, menghindari minimum lokal, mengoptimasi baik variabel diskrit maupun kontinu, sesuai digunakan untuk komputer paralel, dan tidak memerlukan fungsi turunan (Haupt,2004)

Pada awalnya algoritma genetika dimulai dengan mendefinisikan kromosom (v) untuk dioptimalkan (Haupt,2004). Tahap ini disebut juga tahap penyandian. Dalam permasalahan penghitungan bobot variabel inputnya adalah w (bobot kepentingan relatif). Sehingga

$$\text{kromosom} = [w] \quad (2.20)$$

Setiap kromosom memiliki *fitness* atau biaya yang dihitung dari fungsi biaya atau *fitness*, f pada $v_1, v_2, \dots, v_l$ dimana l adalah banyaknya kromosom, sehingga

$$\text{Biaya} = f(\text{kromosom}) = f(v_1, v_2, \dots, v_l) \quad (2.21)$$

Untuk mencari bobot kepentingan relatif (w), sebelumnya digunakan variabel temporer, yaitu variabel x ($x_1, x_2, \dots, x_n$) dengan n adalah jumlah atribut. Kromosom (v) merupakan representasi dari variabel x yang

berbentuk string biner. Kromosom terbagi atas n gen ($v_1, v_2, \dots, v_n$). Sedangkan panjang setiap gen adalah sama. Range yang diijinkan untuk setiap x_i adalah $[a, b]$, dengan a dan b sembarang bilangan real, dan ketepatan (presisi) misal 2 angka di belakang koma, maka panjang gen ke- i (L_i) dapat dirumuskan (Kusumadewi, 2003) :

$$L_i = \lceil 2 \log[(b - a)10^2 + 1] \rceil \quad (2.22)$$

Sedangkan nilai x_i dapat dirumuskan sebagai :

$$x_i = a + [(b - a)/(2^{L_i} - 1)] * v_i \quad (2.23)$$

Karena adanya batasan :

$$\sum_{i=1}^n w_i = 1$$

Maka nilai x_i perlu dimodifikasi dengan langkah – langkah sebagai berikut :

1. Hitung jumlah total x_i (TotX)
2. Hitung

$$w_i = \frac{x_i}{\text{TotX}} \text{ dengan } i = 1, 2, \dots, n \quad (2.24)$$

Karena permasalahan yang diangkat adalah minimasi, maka fungsi fitness atau biaya yang digunakan, pada pendekatan subyektif Kusumadewi menggunakan persamaan 2.18. Sehingga rumusannya adalah :

$$\text{Fitness} = \frac{1}{\sum_{k=1}^n \sum_{j=1}^n (d_{kj} w_j - w_k)^2} \quad (2.25)$$

Proses seleksi kromosom baru dilakukan dengan menggunakan metode Roda Roulette, metode crossover dilakukan dengan metode penyilangan satu titik, dan mutasi dilakukan dengan metode mutasi biner. (Kusumadewi, 2004).

2.3.5 Proses Perangkingan

Proses perangkingan bertujuan untuk memilih alternatif terbaik yang akan terpilih sebagai solusi. Untuk mendapatkan urutan rangking, maka sebelumnya perlu dihitung terlebih dahulu nilai alternatif ke-i, g_i , ($i = 1, 2, \dots, m$) dimana m adalah banyaknya alternatif, sedangkan n adalah banyaknya atribut. Rumusnya adalah sebagai berikut (Fan,1999) :

$$g_i = \sum_{j=1}^n w_j b_{ij} \quad (2.26)$$

Nilai g_i terbesar menunjukkan alternatif ke-i menduduki rangking terbaik. (Kusumadewi,2004)

2.4 Kriteria Penetapan Peserta Sertifikasi Guru

Penetapan peserta merupakan kegiatan terpenting dalam pelaksanaan sertifikasi guru. Apabila terjadi kesalahan atau ketidakadilan dalam penetapan peserta oleh Dinas Pendidikan Kabupaten/Kota, maka tidak menutup kemungkinan akan terjadi aksi ketidakpuasan dari para guru. Untuk mengantisipasi terjadinya peristiwa tersebut, perlu dibuat kriteria untuk menyusun prioritas peserta yang akan mengikuti sertifikasi guru. Penentuan guru calon peserta sertifikasi guru dalam jabatan menggunakan sistem ranking bukan berdasarkan seleksi melalui tes. Kriteria penyusunan ranking (setelah memenuhi syarat kualifikasi akademik S1/D-IV) adalah: masa kerja/pengalaman mengajar, usia, pangkat/golongan (bagi PNS), beban mengajar, jabatan/tugas tambahan, dan prestasi kerja. Kriteria penyusunan ranking yang menjadi dasar urutan prioritas dijelaskan sebagai berikut.

a. Masa kerja/pengalaman mengajar

Masa kerja dihitung sejak yang bersangkutan bekerja sebagai guru baik sebagai PNS maupun non PNS. Contoh perhitungan masa kerja:

Guru “G” adalah seorang guru PNS yang telah diangkat selama 10 tahun 5 bulan, namun guru “G” tersebut sebelum diangkat PNS telah mengajar sebagai tenaga honorer di sebuah SD selama 5 tahun 2 bulan. Masa kerja guru “G” dihitung komulatif semenjak yang bersangkutan bertugas sebagai guru

yaitu 15 tahun 7 bulan. Guru “R” adalah guru non PNS yang sudah bekerja di beberapa SMP swasta sejak bulan Januari 1990 sehingga jika dihitung secara kumulatif masa kerja guru “R” sampai bulan Juni 2007 adalah 17 tahun 6 bulan. Namun, guru “R” tersebut pada tahun 2005-2007 tidak mengajar selama 24 bulan karena alasan keluarga. Masa kerja guru “R” sesungguhnya adalah 15 tahun 6 bulan setelah dikurangi 24 bulan tidak mengajar.

Tabel 2.3 Skor Masa Kerja

Masa Kerja (tahun)	Skor
> 25	190
23 - 25	175
20 - 22	160
17 - 19	145
14 - 16	130
11 - 13	115
8 - 10	100
5 - 7	85

b. Usia

Usia dihitung berdasarkan tahun kelahiran yang tercantum dalam akta kelahiran atau bukti lain yang sah.

Tabel 2.4 Skor Usia

Usia	Skor
56 - 58	190
53 - 55	175
50 - 52	160
47 - 49	145
44 - 46	130
41 - 43	115
38 - 40	100
35 - 37	85
32 - 34	70
29 - 31	55
26 - 28	40

c. Pangkat/Golongan

Kriteria ini khusus untuk guru PNS saja. Pangkat/golongan adalah pangkat/golongan yang sedang diduduki guru saat ini.

Tabel 2.5 Skor Pangkat

Pangkat	Skor
4A	5
3D	4
3C	3
3B	2
3A	1

d. Beban Kerja

Beban kerja adalah jumlah jam mengajar tatap muka per minggu yang diemban oleh guru saat didaftarkan sebagai peserta sertifikasi guru.

e. Tugas Tambahan

Tugas tambahan adalah jabatan atau tugas yang diemban oleh guru pada saat guru yang bersangkutan diusulkan sebagai calon peserta sertifikasi guru. Tugas tambahan yang dimaksud misalnya Kepala Sekolah, Wakil Kepala Sekolah, Ketua Program/Jurusan, Kepala Laboratorium, dan Kepala Bengkel Pembina kegiatan ekstra kurikuler

Tabel 2.6 Skor Tugas Tambahan

Tugas Tambahan	Skor per tahun
Kepala sekolah	4
Wakil kepala sekolah/ketua jurusan/kepala lab/ kepala bengkel	2
Pembina kegiatan ekstra kurikuler (pramuka, drum band, mading, KIR, dsb.)	1

f. Prestasi Kerja

Prestasi kerja yang dimaksudkan disini adalah prestasi yang pernah diraih guru seperti meraih predikat sebagai guru teladan, guru berprestasi, guru berdedikasi, disiplin, dedikasi, dan loyalitas, pembimbingan teman sejawat, pembimbingan siswa sampai mendapatkan penghargaan baik tingkat kecamatan, kabupaten/kota, provinsi, nasional, maupun internasional, dsb.

Tabel 2.7 Skor Lomba dan karya akademik

Prestasi	Tingkat	Skor
Bukti juara lomba akademik	Internasional	60
	Nasional	40
	Provinsi	30
	Kabupaten/Kota	20
	Kecamatan	10
Sertifikat Keahlian/keterampilan (Guru OR)	Internasional	30
	Nasional	20
	Regional	10
Bukti menemukan karya monumental	Pendidikan	60
	Nonpendidikan	40

Tabel 2.8 Skor Pembimbingan kepada sejawat/siswa

Jenis pembimbingan kpd teman sejawat/siswa	Skor	
Instruktur	Nasional	40
	Propinsi	30
	Kab/Kota	20
Guru Inti/Tutor/Pemandu		20
Pamong PPL Calon Guru	1 -4 org/smt	10
	5 -8 org/smt	15
	Lebih dari 8 org/sm	20
Pembimbingan siswa dalam berbagai lomba/karya sampai meraih juara, tingkat	Internasional	40
	Nasional	25
	Provinisi	20
	Kabupaten/Kota	15
	Kecamatan	10
Pembimbingan siswa dalam berbagai lomba/karya tidak mencapai juara		5

(Kementrian Pendidikan Nasional, 2010)

BAB III

METODE PENELITIAN

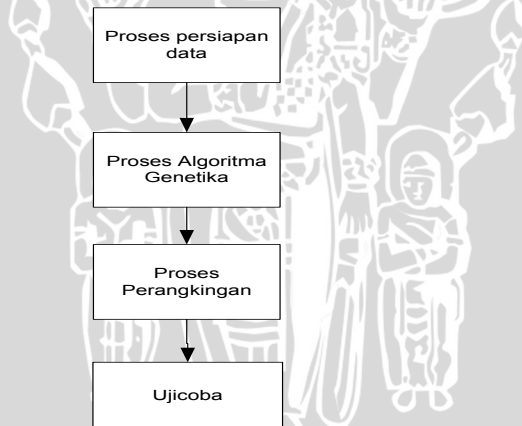
3.1 Tahap Rancangan Global Sistem

Sistem yang akan dibuat adalah sebuah sistem yang membantu tugas dinas pendidikan provinsi dan kabupaten/kota dalam perangkingan peserta sertifikasi guru sesuai dengan kriteria menggunakan metode *Multiple Attribute Decision Making* (MADM) yang di dalamnya memanfaatkan algoritma genetika sebagai metode untuk menghitung bobot atribut setiap alternatifnya.

3.2 Sumber Data

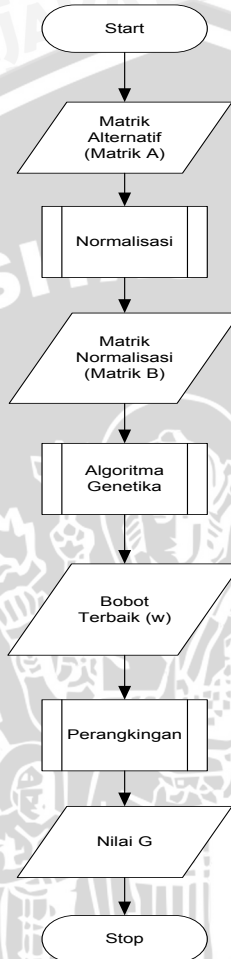
Sumber data yang digunakan dalam penelitian ini adalah data peserta sertifikasi guru yang diambil dari dinas pendidikan kabupaten Malang.

3.3 Blok Diagram dan Flowchart



Gambar 3.1
Blok Diagram Sistem

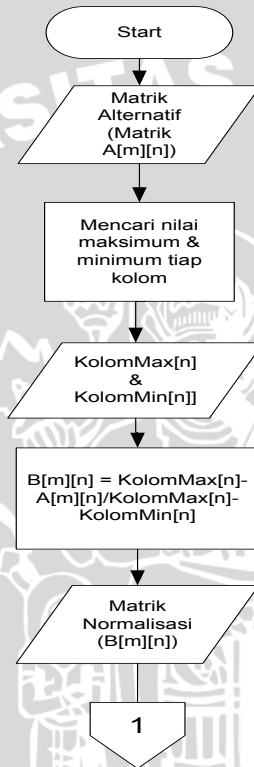
Blok diagram pada gambar 3.1 terdiri dari tiga proses yaitu proses persiapan data, proses algoritma genetika, dan proses perangkingan.



Gambar 3.2
Flowchart Sistem

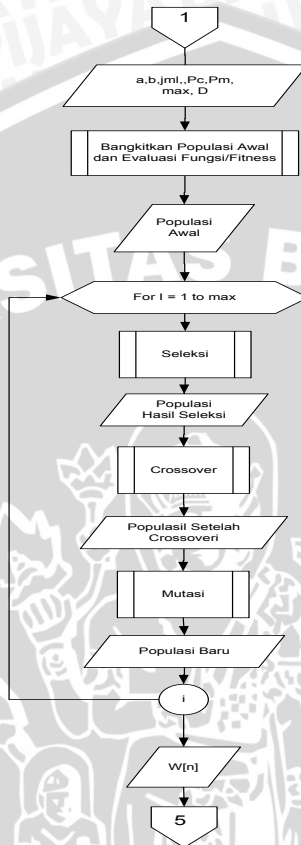
Gambar 3.2 merupakan flowchart sistem, dimulai dari input matrik alternatif (matrik A) yang kemudian dilakukan normalisasi menjadi matrik normalisasi (matrik B). Setelah didapatkan matrik normalisasi (matrik B) dilakukan proses algoritma genetika untuk

menghasilkan kromosom dengan *fitness* terbaik guna mencari nilai bobot(w). Setelah itu nilai bobot(w) tersebut digunakan pada prose perangkian dengan menghitung nilai rangking(g). Kemudian didapatkan urutan dari alternatif yang terdapat di matrik A berdasarkan nilai g dari yang paling tinggi.



Gambar 3.3
Flowchart Normalisasi

Gambar 3.3 merupakan flowchart proses normalisasi. Pertama adalah mencari nilai maksimum dan minimum dari tiap kolom pada matriks A. Kemudian dihasilkan array kolomMax dan kolomMin. Selanjutnya nilai tersebut digunakan untuk menghitung normalisasi matriks A sehingga dihasilkan matriks B.

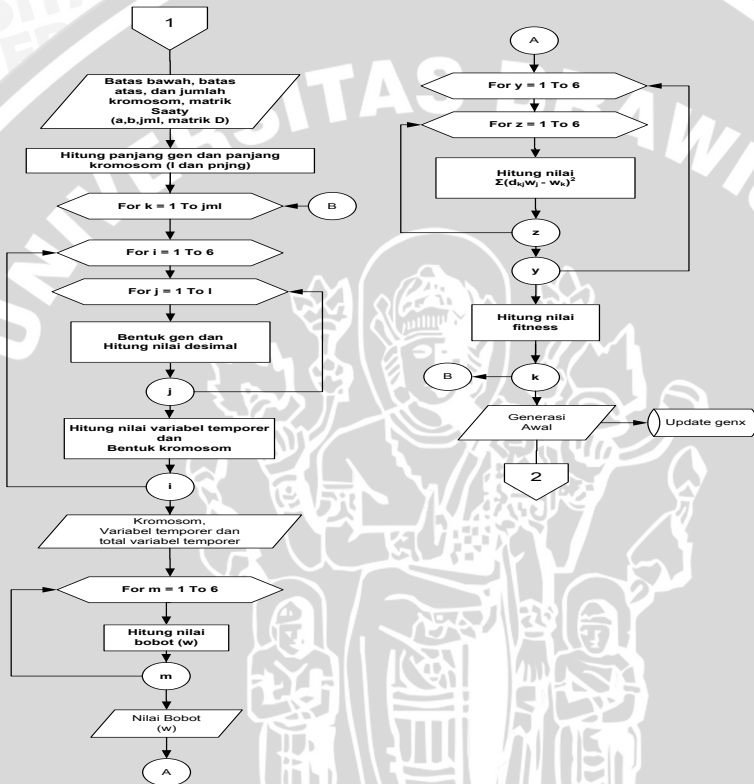


Gambar 3.4
Proses Algoritma Genetika

Gambar 3.4 Proses algoritma genetika, dimulai dari input batas bawah (a), batas bawah (b), jumlah kromosom(jml), peluang *crossover*

(Pc), peluang mutasi (Pm), maksimum generasi (max), dan matrik Saaty (D), lalu dilakukan proses pembangkitan populasi awal dan evaluasi fungsi/*fitness* yang menghasilkan populasi awal. Kemudian, proses *looping* sebanyak maksimum generasi, yang didalamnya dilakukan

proses-proses secara berurutan , yaitu : proses seleksi, yang menghasilkan generasi setelah seleksi, proses *crossover*, yang menghasilkan generasi setelah *crossover* dan yang terakhir adalah proses mutasi, yang menghasilkan generasi baru. Setelah maksimum generasi terpenuhi, maka dihasilkan nilai bobot terbaik dari kromosom dengan *fitness* terbaik.



Gambar 3.5

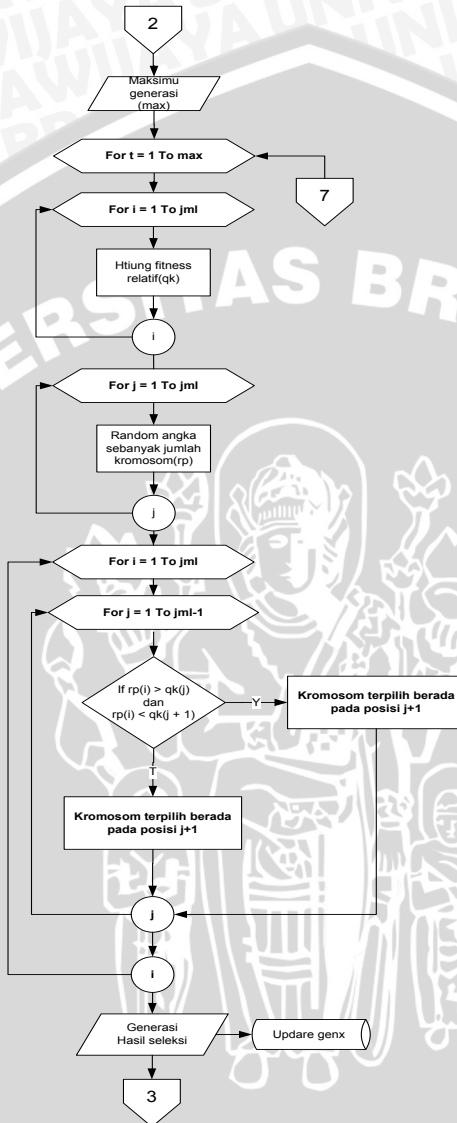
Flowchart pembangkitan populasi awal dan evaluasi fungsi/*fitness*

Pada gambar 3.5, pada proses pembangkitan populasi awal dan evaluasi fungsi/*fitness*, tahap pertama adalah menghitung panjang gen (L_i) dengan *range* tertentu $[a,b]$. Selanjutnya menghitung panjang kromosom(pnjng) berdasarkan panjang gen yang telah diketahui ($L_i * n$). Kemudian dibangkitkan populasi awal berdasarkan input jumlah kromosom(jml), peluang *crossover* (P_c), peluang mutasi (P_m), dan maksimum generasi. Dari populasi awal tersebut dapat dihitung nilai

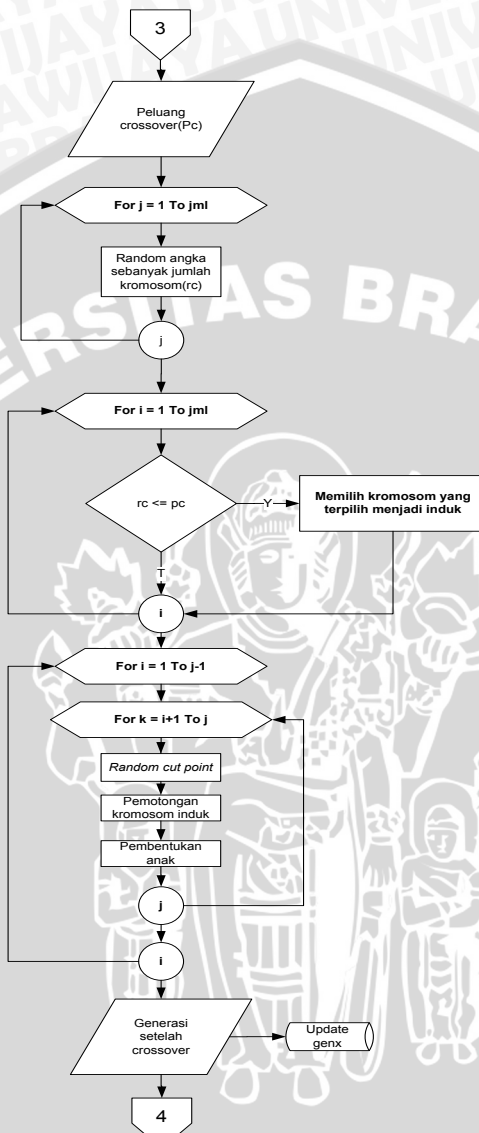
fitness-nya (matrik Saaty atau matrik D diperlukan di tahap ini). Dari sini didapatkan generasi awal.

Pada gambar 3.6, merupakan proses seleksi. Tahap pertama yang dilakukan adalah menghitung nilai *fitness* relatif. Kemudian membangkitkan random angka sebanyak jumlah kromosom. Kemudian dilakukan pemilihan kromosom yang lolos seleksi. Pada tahap ini dihasilkan generasi hasil seleksi.

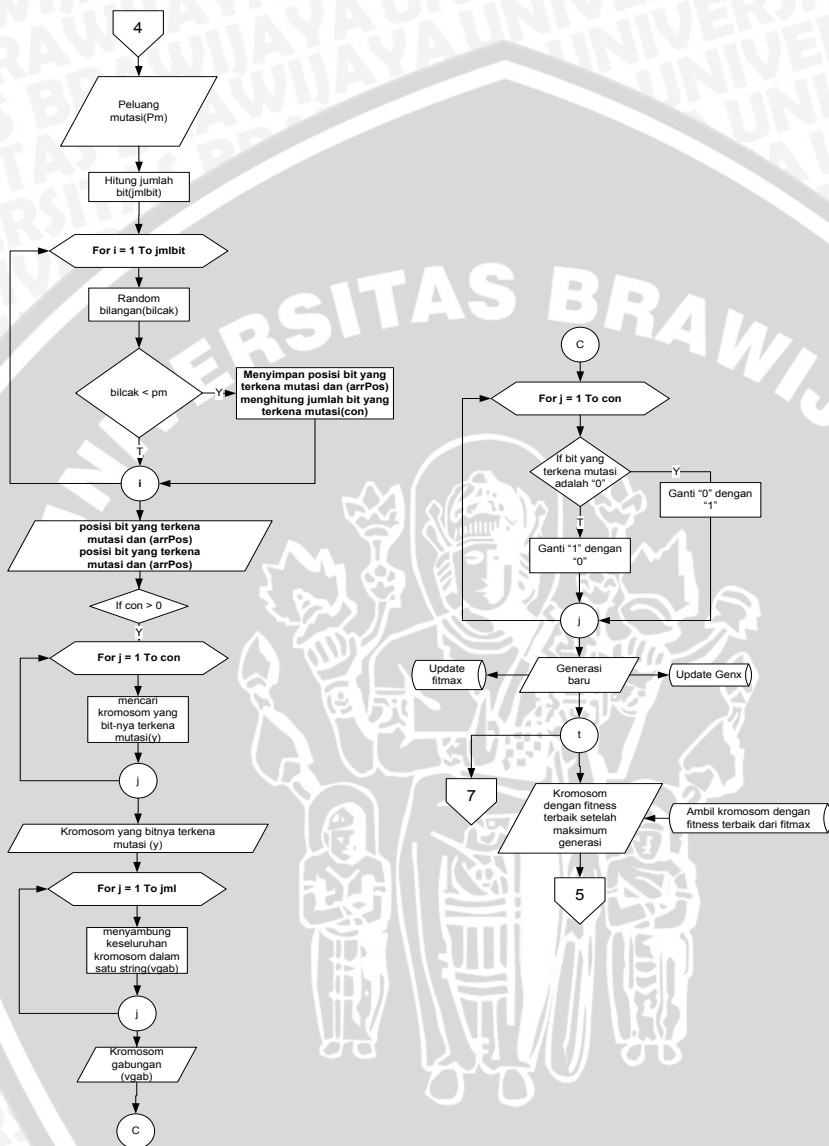




Gambar 3.6
Flowchart seleksi



Gambar 3.7
Flowchart *crossover*

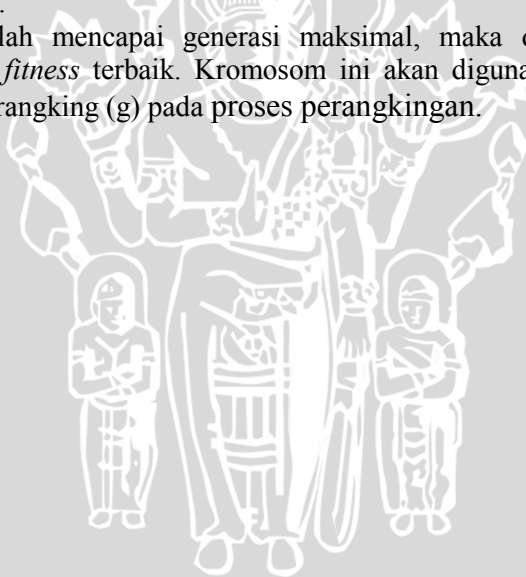


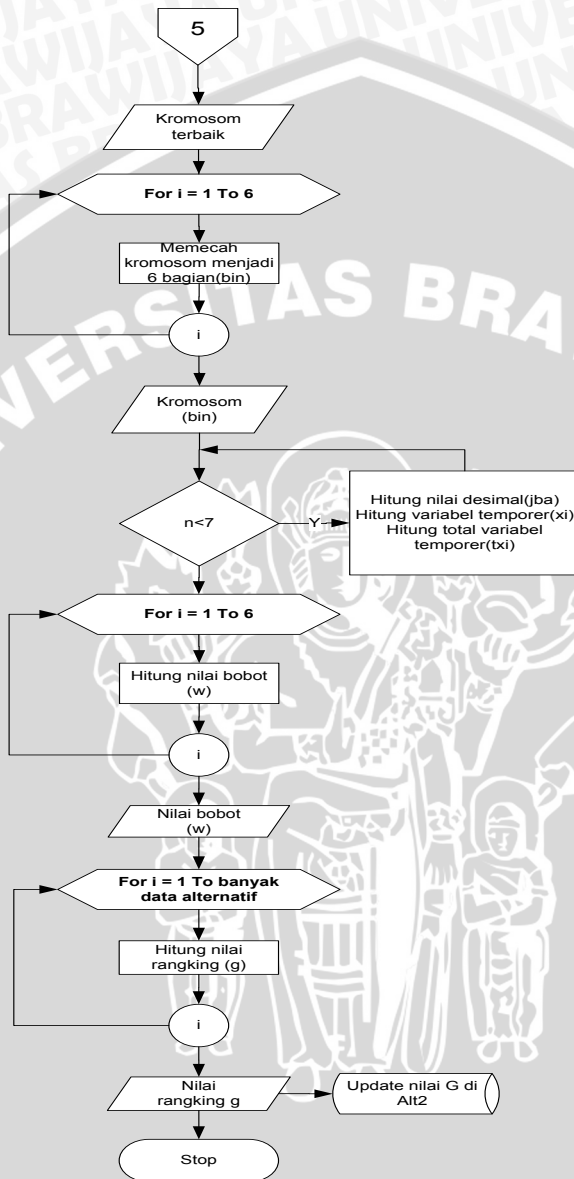
Gambar 3.8
Flowchart mutasi

Pada gambar 3.7, merupakan proses crossover. Tahap pertama yang dilakukan adalah membangkitkan random angka sebanyak jumlah kromosom. Kemudian dilakukan pemilihan kromosom yang menjadi induk. Tahap selanjutnya adalah memilih satu *random cut point* (titik potong) dari panjang kromosom, pemotongan kromosom induk, dan pembentukan anak. Setelah itu, didapatkan generasi setelah *crossover*. Metode *crossover* yang digunakan adalah *one cut point crossover*.

Pada gambar 3.8, merupakan proses mutasi. Tahap pertama yang dilakukan adalah menghitung jumlah bit dalam satu generasi (jumlah kromosom * panjang kromosom). Kemudian dilakukan pencarian posisi bit yang terkena mutasi dan menghitung jumlahnya. Setelah itu mencari kromosom yang bit-nya terkena mutasi. Pada tahap ini proses mutasi dilakukan, yaitu dengan mengganti bit dengan 0 atau 1. Kemudian didapatkan generasi baru. Generasi baru ini menjadi generasi awal untuk *looping* algoritma genetika selanjutnya. Metode mutasi yang digunakan adalah mutasi biner.

Apabila telah mencapai generasi maksimal, maka dihasilkan kromosom dengan *fitness* terbaik. Kromosom ini akan digunakan pada penghitungan nilai rangking (g) pada proses perangkingan.





Gambar 3.9
Flowchart Perangkingan

Gambar 3.9 adalah flowchart proses perangkingan. Prosesnya adalah data w (bobot) dan matriks B digunakan dalam persamaan untuk menghitung nilai g. Kemudian data alternatif bisa diurutkan berdasarkan nilai g mulai dari yang tertinggi sampai terendah.

3.4 Perancangan Sistem

Permasalahan utama yang akan diselesaikan dalam penelitian ini adalah masalah menyusun dan menetapkan peserta sertifikasi guru sesuai dengan kriteria. Untuk menyelesaikan permasalahan ini digunakan model MADM pendekatan subjektif dengan penghitungan bobot yang menggunakan algoritma genetika sebagai pendekatannya. Sebagai gambarannya akan dijelaskan sebagai berikut :

Terdapat lima data peserta serifikasi guru (data peserta pada lampiran 1, diambil lima data pertama). Dari lima peserta tersebut akan disusun daftar peserta sertifikasi guru sesuai dengan kriteria yang ditentukan yaitu:

- C1 = Masa kerja
- C2 = Usia
- C3 = Pangkat
- C4 = Beban kerja
- C5 = Tugas tambahan
- C6 = Prestasi kerja

Tabel 3.1 Alternatif

Alternatif	Atribut (Kriteria)					
	C1	C2	C3	C4	C5	C6
S1	190	190	5	24	23	60
S2	175	175	5	37	4	15
S3	145	145	4	38	10	50
S4	85	40	1	36	0	0
S5	190	175	5	27	16	90

Dengan demikian diperoleh matriks A

$$A = \begin{bmatrix} 190 & 190 & 5 & 24 & 23 & 60 \\ 175 & 175 & 5 & 37 & 4 & 15 \\ 145 & 145 & 4 & 38 & 10 & 50 \\ 85 & 40 & 1 & 36 & 0 & 0 \\ 190 & 175 & 5 & 27 & 16 & 90 \end{bmatrix}$$

Matrik perbandingan berpasangan yang diberikan oleh pengambil keputusan yang didasarkan pada matriks Saaty seperti yang dijelaskan di bab sebelumnya, adalah sebagai berikut :

- g. C₁ banding C₂ adalah 2
- h. C₁ banding C₃ adalah 2
- i. C₁ banding C₄ adalah 3
- j. C₁ banding C₅ adalah 5
- k. C₁ banding C₆ adalah 6
- l. C₂ banding C₃ adalah 2
- m. C₂ banding C₄ adalah 3
- n. C₂ banding C₅ adalah 4
- o. C₂ banding C₆ adalah 5
- p. C₃ banding C₄ adalah 2
- q. C₃ banding C₅ adalah 4
- r. C₃ banding C₆ adalah 5
- s. C₄ banding C₅ adalah 2
- t. C₄ banding C₆ adalah 3
- u. C₅ banding C₆ adalah 2

$$D = \begin{bmatrix} 1 & 2 & 2 & 3 & 5 & 6 \\ 1/2 & 1 & 2 & 3 & 4 & 5 \\ 1/2 & 1/2 & 1 & 2 & 4 & 5 \\ 1/3 & 1/3 & 1/2 & 1 & 2 & 3 \\ 1/5 & 1/4 & 1/4 & 1/2 & 1 & 2 \\ 1/6 & 1/5 & 1/5 & 1/3 & 1/2 & 1 \end{bmatrix}$$

Dari informasi yang ada. Langkah selanjutnya adalah membuat matriks normalisasi (matrik B) sebagai hasil dari normalisasi dari matriks A. Semua kriteria adalah kriteria keuntungan, yaitu kriteria yang dimaksimumkan nilainya, sehingga digunakan persamaan 2.4 :

$$A = \begin{bmatrix} 190 & 190 & 5 & 24 & 23 & 60 \\ 175 & 175 & 5 & 37 & 4 & 15 \\ 145 & 145 & 4 & 38 & 10 & 50 \\ 85 & 40 & 1 & 36 & 0 & 0 \\ 190 & 175 & 5 & 27 & 16 & 90 \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{190-85}{190-85} & \frac{190-40}{190-40} & \frac{5-1}{5-1} & \frac{24-24}{38-24} & \frac{23-0}{4-0} & \frac{60-0}{60-15} \\ \frac{190-85}{175-85} & \frac{190-40}{175-40} & \frac{5-1}{5-1} & \frac{38-24}{37-24} & \frac{23-0}{4-0} & \frac{90-0}{60-15} \\ \frac{175-85}{175-85} & \frac{175-40}{175-40} & \frac{5-1}{5-1} & \frac{37-24}{37-24} & \frac{4-0}{4-0} & \frac{15-0}{60-15} \\ \frac{190-85}{145-85} & \frac{190-40}{145-40} & \frac{5-1}{4-1} & \frac{24-24}{38-24} & \frac{23-0}{10-0} & \frac{60-0}{50-0} \\ \frac{145-85}{145-85} & \frac{145-40}{145-40} & \frac{4-1}{4-1} & \frac{38-24}{38-24} & \frac{10-0}{10-0} & \frac{50-0}{50-0} \\ \frac{85-85}{190-85} & \frac{40-40}{190-40} & \frac{1-1}{5-1} & \frac{36-24}{38-24} & \frac{0-0}{23-0} & \frac{0-0}{90-0} \\ \frac{85-85}{85-85} & \frac{40-40}{40-40} & \frac{1-1}{1-1} & \frac{36-24}{36-24} & \frac{0-0}{0-0} & \frac{0-0}{0-0} \\ \frac{190-85}{190-85} & \frac{190-40}{190-40} & \frac{5-1}{5-1} & \frac{38-24}{38-24} & \frac{23-0}{23-0} & \frac{90-0}{90-0} \\ \frac{190-85}{190-85} & \frac{175-40}{175-40} & \frac{5-1}{5-1} & \frac{27-24}{27-24} & \frac{16-0}{16-0} & \frac{90-90}{90-90} \\ \frac{190-85}{190-85} & \frac{175-40}{175-40} & \frac{5-1}{5-1} & \frac{38-27}{38-27} & \frac{23-0}{23-0} & \frac{90-0}{90-0} \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0.667 \\ 0.857 & 0.9 & 1 & 0.929 & 0.174 & 0.167 \\ 0.571 & 0.7 & 0.75 & 1 & 0.435 & 0.556 \\ 0 & 0 & 0 & 0.857 & 0 & 0 \\ 1 & 0.9 & 1 & 0.214 & 0.696 & 1 \end{bmatrix}$$

a. Representasi Solusi

Untuk mencari nilai bobot kepentingan relatif (w), sebelumnya digunakan variabel temporer, yaitu variabel x ($x_1, x_2, \dots, x_n$) dengan n adalah jumlah atribut. Kromosom (v) merupakan representasi dari variabel x yang berbentuk string biner. Kromosom terbagi atas n gen ($v_1, v_2, \dots, v_n$). Sedangkan panjang setiap gen adalah sama. *Range* yang diijinkan untuk setiap x_i adalah $[a, b]$, dengan a dan b sembarang bilangan *real*, dan ketepatan (presisi) misal 2 angka di belakang koma, maka panjang gen ke- i (L_i) dapat dihitung dengan persamaan 2.22. Sedangkan nilai variabel temporer (x_i) dapat dihitung dengan persamaan 2.23. Karena adanya batasan :

$$\sum_{i=1}^n w_i = 1$$

Maka nilai variabel temporer (x_i) perlu dimodifikasi dengan langkah – langkah sebagai berikut :

3. Hitung jumlah total variabel temporer atau x_i (TotX)
4. Hitung bobot kepentingan relatif (W_i) sesuai dengan persamaan 2.24

Contoh penghitungan :

Panjang gen dengan *range* [0,1] adalah

$$L_i = \lceil 2 \log[(1 - 0)10^2 + 1] \rceil \\ = 2 \log[101] = 7$$

Sehingga apabila ada 6 atribut (6 gen), maka panjang kromosom adalah

$$6 \times 7 = 42$$

Apabila suatu vektor, dengan 6 gen :

$$V = 1100011 \text{ } 1011011 \text{ } 0000001 \text{ } 1111100 \text{ } 0011100 \text{ } 0000010$$

Maka :

$$V_1 = 1100011 = 99$$

$$X_1 = 0 + [(1-0)/(2^7 - 1)] * 99 = 0.779527559055118$$

$$V_2 = 1011011 = 91$$

$$X_2 = 0 + [(1-0)/(2^7 - 1)] * 91 = 0.716535433070866$$

$$V_3 = 0000001 = 1$$

$$X_3 = 0 + [(1-0)/(2^7 - 1)] * 1 = 0.0078740157480315$$

$$V_4 = 1111100 = 124$$

$$X_4 = 0 + [(1-0)/(2^7 - 1)] * 124 = 0.976377952755906$$

$$V_5 = 0011100 = 28$$

$$X_5 = 0 + [(1-0)/(2^7 - 1)] * 28 = 0.220472440944882$$

$$V_6 = 0000010 = 2$$

$$X_6 = 0 + [(1-0)/(2^7 - 1)] * 2 = 0.015748031496063$$

Karena adanya batasan :

$$\sum_{i=1}^n w_i = 1$$

Maka nilai x_i perlu dimodifikasi dengan langkah – langkah sebagai berikut :

i. Hitung jumlah total x_i (TotX)

$$\begin{aligned} \text{TotX} &= 0.779527559055118 + 0.716535433070866 + \\ &0.0078740157480315 + 0.976377952755906 + \\ &0.220472440944882 + 0.015748031496063 \end{aligned}$$

$$\text{TotX} = 2.7165354330708700$$

Hitung,

$$w_i = \frac{x_i}{\text{TotX}} \text{ dengan } i = 1, 2, \dots, n$$

$$\bullet \quad w_1 = \frac{0.779527559055118}{2.7165354330708700} = 0.28695652173913$$

$$\bullet \quad w_2 = \frac{0.716535433070866}{2.7165354330708700} = 0.263768115942029$$

$$\bullet \quad w_3 = \frac{0.0078740157480315}{2.7165354330708700} = 0.00289855072463768$$

$$\bullet \quad w_4 = \frac{0.976377952755906}{2.7165354330708700} = 0.359420289855072$$

$$\bullet \quad w_5 = \frac{0.220472440944882}{2.7165354330708700} = 0.0811594202898551$$

$$\bullet \quad w_6 = \frac{0.015748031496063}{2.7165354330708700} = 0.00579710144927536$$

Karena permasalahan yang diangkat adalah minimasi, maka fungsi *fitness* yang digunakan, pada pendekatan subyektif digunakan persamaan 2.25

Contoh penghitungan nilai *fitness* yaitu:

$$\begin{aligned}
 \text{Fitness} &= 1/((1 \times 0.28695652173913) - 0.28695652173913)^2 + \\
 &((2 \times 0.263768115942029) - 0.28695652173913)^2 + \\
 &((2 \times 0.00289855072463768) - 0.28695652173913)^2 + \\
 &((3 \times 0.359420289855072) - 0.28695652173913)^2 + \\
 &((5 \times 0.0811594202898551) - 0.28695652173913)^2 + \\
 &((6 \times 0.00579710144927536) - 0.28695652173913)^2 + \text{dst} \\
 &= 0.361728253564561
 \end{aligned}$$

b. Penentuan Parameter

Misalkan parameter – parameter yang digunakan adalah

- Jumlah kromosom = 5
- $P_c = 0,4$
- $P_m = 0,01$
- Maksimum generasi 1

c. Inisialisasi Populasi Awal

Misalkan populasi awal yang dihasilkan secara random adalah

Tabel 3.2 Populasi Awal

Kromosom	Bentuk biner	<i>Fitness</i>
V_1	110001110110110000001111110000111000000010	0.361728253564561
V_2	011011010110101001010011010111000111100000	0.184979265384935
V_3	110111000101101101100101100000001010101000	0.546931862598435
V_4	001100011100001110000011001100010111010000	0.252776957726331
V_5	011110010101101110010010100001010100111000	0.54739462671866
Total <i>Fitness</i> = 1.893810965992920		

d. Seleksi

Proses seleksi kromosom baru dilakukan dengan menggunakan metode Roda Roulette, langkah – langkah yang harus dikerjakan :

- Total *fitness* adalah 1.893810965992920. Sehingga *fitness* relatif (p_k) tiap – tiap kromosom dapat dicari sebagai berikut:

- $P_1 = F_1 / \text{Total Fitness}$
 $= 0.361728253564561 / 1.893810965992920$
 $= 0.191005469954552$

Dan seterusnya. Hasil selengkapnya pada tabel 3.3

- *Fitness* kumulatif (q_k) dapat dicari sebagai berikut :
 - $q_1 = p_1 = 0.191005469954552$
 - $q_2 = q_1 + p_2$
 $= 0.191005469954552 + 0.0976756755064782$
 $= 0.28868114546103$

Dan seterusnya. Hasil selengkapnya pada tabel 3.3

- Bangkitkan bilangan acak sebanyak jumlah kromosom=5.
 Misalkan bilangan acak yang terbentuk seperti pada tabel 3.4

Tabel 3.3 *Fitness* Relatif dan *Fitness* Kumulatif

Kromosom	<i>Fitness</i> relatif (p_k)	<i>Fitness</i> kumulatif (q_k)
V_1	0.191005469954552	0.191005469954552
V_2	0.0976756755064782	0.28868114546103
V_3	0.288799606940537	0.577480752401567
V_4	0.133475284632645	0.710956037034213
V_5	0.289043962965787	1

Tabel 3.4 Bilangan Acak untuk Seleksi

No.	Bilangan Acak (r_i)
1	0.977713227272034
2	0.0738452076911926
3	0.457553148269653
4	0.322740972042084
5	0.46159827709198

- Bilangan random pertama $r_1 = 0.977713227272034$, bisa dilihat bahwa $q_4 \leq r_1$ dan $q_5 > r_1$, dengan demikian, maka v_5 akan terpilih menjadi kromosom baru yang pertama. Dan seterusnya. Hasil selengkapnya pada tabel 3.5

Tabel 3.5 Kromosom Baru Hasil Seleksi

Kromosom	Bentuk biner	<i>Fitness</i>	Asal
V_1'	01111001010110110010010100001010100111000	0.54739462671866	V_5
V_2'	110001110110110000001111110000111000000010	0.361728253564561	V_1
V_3'	110111000101101101100101100000001010101000	0.546931862598435	V_3
V_4'	110111000101101101100101100000001010101000	0.546931862598435	V_3
V_5'	110111000101101101100101100000001010101000	0.546931862598435	V_3

e. *Crossover*

Metode *crossover* dilakukan dengan metode penyilangan satu titik. Langkah – langkah yang harus dilakukan adalah :

- Karena peluang *crossover* (p_c) adalah 0,4, maka diharapkan 40% dari populasi akan mengalami *crossover* (2 dari 5 kromosom). Untuk memilih kromosom – kromosom mana saja yang akan , bangkitkan bilangan acak antara [0,1] sebanyak 5 buah. Misalkan bilangan – bilangan itu seperti terlihat pada tabel 3.6.

Tabel 3.6 Bilangan Acak untuk *Crossover*

No.	Bilangan Acak (r_1)
1	0.123654901981354
2	0.638949871063232
3	0.358644187450409
4	0.668402791023254
5	0.739126980304718

- Pilih bilangan-bilangan acak yang kurang dari p_c (misalkan r_s), maka kromosom V_s' berhak untuk melakukan *crossover*. Bilangan acak pertama $r_1 = 0.123654901981354$; bilangan ini lebih kecil jika dibandingkan dengan p_c (0,4). Dengan demikian kromosom V_1' akan mengalami *crossover*. Bilangan acak yang kurang dari p_c adalah bilangan ke-1 dan 3. Tabel 3.7 menunjukkan kromosom yang akan di *crossover*

Tabel 3.7 Kromosom yang Terpilih untuk *Crossover*

Kromosom	Bentuk biner	<i>Fitness</i>
V_1'	011110010101101110010010100001010100111000	0.54739462671866
V_3'	110111000101101101100101100000001010101000	0.546931862598435

- Silangkan V_1' dengan V_3' . Pilih bilangan acak antara 1 sampai 41 (panjang kromosom(L)-1). Bilangan ini akan menentukan posisi *crossover* satu titik. Misalkan bilangan itu adalah 21
 $V_2' =$ 011110010101101110010010100001010100111000
 $V_3' =$ 110111000101101101100101100000001010101000
 Anak sebagai hasil *crossover* (V_2'' dan V_3'') adalah
 $V_2'' =$ 011110010101101110010101100000001010101000
 $V_3'' =$ 110111000101101101100010100001010100111000
- Kromosom hasil *crossover* dan kromosom hasil seleksi digabung, lalu dipilih lima kromosom terbaik
- Hasil setelah dilakukan *crossover* terlihat pada tabel 3.8

Tabel 3.8 Kromosom setelah *Crossover*

Kromosom	Bentuk biner	<i>Fitness</i>
V_2''	011110010101101110010101100000001010101000	0.622811746583834
V_1'	011110010101101110010010100001010100111000	0.54739462671866
V_3'	110111000101101101100101100000001010101000	0.546931862598435
V_4'	110111000101101101100101100000001010101000	0.546931862598435
V_5'	110111000101101101100101100000001010101000	0.546931862598435

f. Mutasi

Mutasi dilakukan dengan metode mutasi biner. Langkah-langkah yang harus dilakukan :

- Hitung jumlah bit yang ada pada populasi, yaitu : jumlah kromosom * $L = 5 * 42 = 210$
- Karena peluang mutasi (p_m) adalah 0,01; maka diharapkan 1% dari total bit akan mengalami mutasi (2,1 bit = 2 bit). Untuk memilih bit-bit mana saja yang akan dilakukan mutasi, bangkitkan bilangan acak antara [0,1] sebanyak 210 buah. Misalkan bilangan – bilangan acak pada kromosom peratama terlihat pada tabel 3.8.
- Pada kromosom pertama, bit pertama, bilangan acak yang terbentuk adalah 0.2985582. Bilangan ini lebih besar jika dibandingkan dengan $p_m = 0,01$; ini berarti bahwa kromosom pertama, bit pertama, tidak akan terkena mutasi. Kemudian dicari lagi nilai bilangan acak yang lebih kecil dari p_m . Dan seterusnya hasilnya pada tabel 3.10

Tabel 3.9 Bilangan Acak untuk Mutasi

No	Bil. Acak	No.	Bil. Acak	No.	Bil. Acak
Kromosom ke-1					
1	0.2985582	15	0.06383115	29	0.5124471
2	0.7792181	16	0.800843	30	0.4549953
3	0.3372384	17	0.3242282	31	0.8929299
4	0.9287567	18	0.7540725	32	0.3135469
5	0.6230432	19	0.995698	33	0.7861165
6	0.3695446	20	0.2708216	34	0.8995661
7	0.1492818	21	0.9142925	35	0.8247967
8	0.5444911	22	0.2494589	36	0.2895832
9	0.3588633	23	0.9783805	37	0.3510801
10	0.1813258	24	0.05719495	38	0.6354309
11	0.4483591	25	0.5551724	39	0.04795772
12	0.6472592	26	0.3268193	40	0.07595658
13	0.316138	27	0.2774578	41	0.2575392
14	0.3095018	28	0.1599631	42	0.4472121

Tabel 3.10 Kromosom dan Posisinya yang Terkena Mutasi

Kromosom	Bit
4	151

- Kromosom hasil mutasi dan generasi hasil crossover digabung, lalu dipilih lima kromosom terbaik
- Hasil setelah dilakukan mutasi terlihat pada tabel 3.11

Tabel 3.11 Kromosom setelah Mutasi

Kromosom	Bentuk biner	<i>Fitness</i>
V_4''	011110010101101110010101100000001010101000	0.622811746583834
$V_4'^*$	110111000101101101100101000000001010101000	0.575981418487085
V_1'	011110010101101110010010100001010100111000	0.54739462671866
V_3''	110111000101101101100101100000001010101000	0.546931862598435
V_4'	110111000101101101100101100000001010101000	0.546931862598435

Populasi akhir setelah dilakukan mutasi ini menjadi populasi awal untuk generasi selanjutnya. Dengan cara yang sama dilakukan pada generasi selanjutnya. Nilai akhir yang diperoleh :

- Nilai Bobot
 - $w_1 = 0.152671755725191$
 - $w_2 = 0.21882951653944$
 - $w_3 = 0.290076335877863$
 - $w_4 = 0.223918575063613$
 - $w_5 = 0.0127226463104326$
 - $w_6 = 0.101781170483461$
- Nilai *fitness* terbaik : 0.622811746583834

g. Proses Perangkingan

Proses perangkingan bertujuan untuk memilih alternatif terbaik yang akan terpilih sebagai solusi. Untuk mendapatkan urutan rangking, maka sebelumnya perlu dihitung terlebih dahulu nilai alternatif ke- i , g_i , ($i = 1, 2, \dots, m$) dengan persamaan 2.26. Nilai g_i terbesar menunjukkan alternatif ke- i menduduki rangking terbaik.

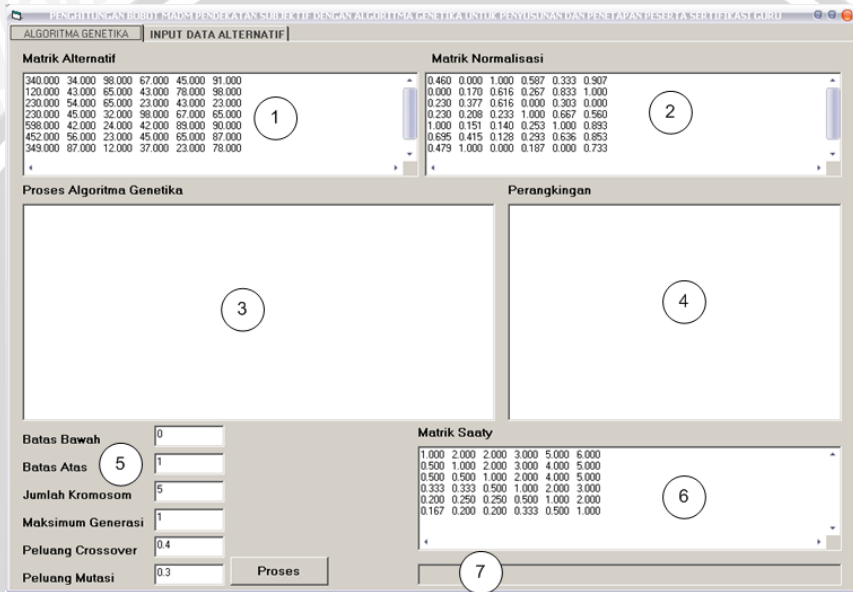
Contoh penghitungan g :

$$\begin{aligned}
 g_i &= (0.152671755725191 * 1) + (0.21882951653944 * 1) + \\
 &\quad (0.290076335877863 * 1) + (0.223918575063613 * 0) + \\
 &\quad (0.0127226463104326 * 1) + (0.101781170483461 * 0.667) \\
 &= 0.742154368108567
 \end{aligned}$$

dengan perhitungan yang sama diperoleh nilai $g_2 = 0.742154368108567$

$g_3 = 0.742154368108567$, $g_4 = 0.191930207197383$ dan $g_5 = 0.191930207197383$, sehingga urutan alternatifnya adalah S2 – S5 – S3 – S1 – S4.

3.5 Perancangan *Interface*



Gambar 3.10
Rancangan *interface* proses algoritma genetika

Pada bagian proses algoritma genetika, aplikasi yang akan dibuat akan memiliki *interface* seperti di atas. Terdiri dari :

1. Tampilan matrik alternatif (matrik A)
2. Tampilan matrik normalisasi (matrik B)
3. Tampilan proses algoritma genetika
4. Tampilan hasil perangkingan
5. Input parameter algoritma genetika
6. Tampilan matrik Saaty (matrik D)
7. *Progress bar*

ALGORITMA GENETIKA INPUT DATA ALTERNATIF

1

ID 7

Nama

Masa Kerja

Usia

Pangkat

Beban Kerja

Tugas Tambahan

Prestasi Kerja

Tambah

Update

Delete

ID 3

2

3

4

DATA ALTERNATIF

ID	Nama	Masa Kerja	Usia	Pangkat	Beban Kerja	Tugas Tambahan	Prestasi Kerja	G
1	Raden Supratman	340	34	98	67	45	91	0
2	Fernando Husein	120	43	65	43	78	98	0
3	Yuli Harun	230	54	65	23	43	23	0
4	Martin Palermo	230	45	32	98	67	65	0
5	George Hananto	598	42	24	42	89	90	0
6	Yukito Amakura	452	56	23	45	65	87	0
7	Mintazal Utuhla	349	87	12	37	23	78	0

Gambar 3.11
Rancangan *interface* proses input data alternatif

Pada bagian proses input data alternatif, aplikasi yang akan dibuat akan memiliki *interface* seperti di atas. Terdiri dari :

1. Pilihan tindakan yang ingin dilakukan (input data, *edit* data atau hapus data)
2. Isian data dan tombol tambah, simpan dan *delete*
3. Cari data
4. Tampilan data alternatif

3.6 Perancangan Ujicoba

Tabel 3.12 Pengaruh peluang *crossover* dan peluang mutasi terhadap nilai fitness

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1					
2					
3					
4					
5					
Rataan					

Dengan peluang *crossover* dan mutasi yang berbeda akan dianalisa nilai *fitness* yang dihasilkan. Sedangkan jumlah kromosom atau individu dan maksimum generasi sama. Percobaan dilakukan sebanyak 5 kali untuk tiap peluang mutasi dan dihitung nilai rata-ratanya.

Tabel 3.13 Pengaruh peluang *crossover* dan peluang mutasi terhadap konvergensi

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1					
2					
3					
4					
5					
Rataan					

Dengan Peluang *crossover* dan mutasi yang berbeda akan dianalisa konvergensi terjadi pada tiap percobaan. Sedangkan jumlah kromosom atau individu dan maksimum generasi sama. Percobaan dilakukan sebanyak 5 kali untuk tiap peluang mutasi dan dihitung nilai rata-ratanya

Ujicoba ketiga adalah membandingkan hasil perangkian dari aplikasi ini dengan hasil perangkian secara manual.

Tabel 3.14 Perbandingan Hasil Perangkingan Manual dengan Aplikasi

Manual		Aplikasi		Cek
No	Nama	No	Nama	
1		1		
2		2		
3		3		
4		4		
5		5		



BAB IV

IMPLEMENTASI DAN PEMBAHASAN

Pada bab ini akan dibahas seluruh proses yang telah dirancang pada bab sebelumnya, tampilan antarmuka dan bagian-bagian *source code* yang dibuat, serta analisa terhadap hasil uji coba data yang telah dihasilkan oleh sistem.

4.1 Lingkungan Implementasi

Implementasi merupakan proses transformasi representasi rancangan ke dalam bahasa pemrograman yang dapat dimengerti oleh komputer. Lingkungan implementasi yang akan dijelaskan meliputi lingkungan perangkat keras dan lingkungan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan aplikasi algoritma genetika dalam penghitungan bobot atribut MADM pada studi kasus penyusunan dan penetapan peserta sertifikasi guru ini adalah sebagai berikut :

1. Prosesor Intel® Core™2 Duo T7000 @ 2.00 GHz
2. RAM 0.99 MB
3. *Harddisk* dengan kapasitas 300 GB
4. Monitor 14.1"
5. Keyboard
6. Mouse

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem ini adalah :

1. Sistem Operasi Windows XP *Service Pack 2* sebagai tempat aplikasi dijalankan.
2. *Microsoft Visual Basic 6* sebagai *software development* dalam membuat algoritma genetika dalam penghitungan bobot atribut MADM pada studi kasus penyusunan dan penetapan peserta sertifikasi guru ini.
3. Bahasa pemrograman menggunakan VB

4. *Microsoft Access 2007* sebagai database yang digunakan untuk menyimpan data.

4.2 Implementasi

Implementasi perangkat lunak ini berupa aplikasi pemrograman yang menerapkan algoritma genetika dalam penghitungan bobot atribut MADM pada studi kasus perangkingan peserta sertifikasi guru. Tampilan utama dari aplikasi algoritma genetika dalam penghitungan bobot atribut MADM pada studi kasus perangkingan peserta sertifikasi guru dapat dilihat pada gambar 4.1

ALGORITMA GENETIKA | INPUT DATA ALTERNATIF

Matrik Alternatif

340.000	34.000	98.000	67.000	45.000	91.000
120.000	43.000	65.000	43.000	78.000	98.000
230.000	54.000	65.000	23.000	43.000	23.000
230.000	45.000	32.000	98.000	67.000	65.000
598.000	42.000	24.000	42.000	89.000	90.000
452.000	56.000	23.000	45.000	65.000	87.000
349.000	87.000	12.000	37.000	23.000	78.000

Matrik Normalisasi

0.460	0.000	1.000	0.587	0.333	0.907
0.000	0.170	0.616	0.267	0.833	1.000
0.230	0.377	0.616	1.000	0.303	0.000
0.220	0.208	0.233	1.000	0.687	0.560
1.000	0.151	0.140	0.253	1.000	0.893
0.695	0.415	0.128	0.293	0.636	0.853
0.479	1.000	0.000	0.187	0.000	0.733

Proses Algoritma Genetika

Generasi Awal

ID	Kromosom	Fitness
1.	1001001000010100000100010010100111000100	0.135005975594491
2.	0000111101110111110100110110010100011111	0.433077347277994
3.	01110010000010100110000110000001101100001	8.28908715759617E-02
4.	110001000000110110101000010111100111001	0.166896875220859
5.	100101000000010000001011001011000110111	0.239498252274618

Generasi ke-1

ID	Kromosom	Fitness
1.	100111110111011111010011010010100011111	1.24106234081031
2.	100011110111011111010011010010100011111	1.1091363234958
3.	00011110011011011010111100010010011111	0.5072758656108
4.	0000111101110111110100110110010100011111	0.433077347277994
5.	0000111101110111110100110110010100011111	0.433077347277994

Fitness Terbaik tiap generasi
Generasi Fitness

Perangkingan

Kromosom terbaik = 100111110111011111010011010010100011111
Fitness terbaik = 1.24106234081031

Urutan Data

No.	Nama	Score
1.	Raden Supatman	G -> 0.564031386302343
2.	George Harianto	G -> 0.443005118428398
3.	Mintarzal Uthukla	G -> 0.420744333370233
4.	Yukio Amakura	G -> 0.41866526329195
5.	Fernando Husein	G -> 0.396182148234621
6.	Yuli Harun	G -> 0.363331051268515
7.	Martin Palermo	G -> 0.333098772464116

Matrik Saaty

1.000	2.000	2.000	3.000	5.000	6.000
0.500	1.000	2.000	3.000	4.000	5.000
0.500	0.500	1.000	2.000	4.000	5.000
0.333	0.333	0.500	1.000	2.000	3.000
0.200	0.250	0.250	0.500	1.000	2.000
0.167	0.200	0.200	0.333	0.500	1.000

Batas Bawah: 0
Batas Atas: 1
Jumlah Kromosom: 5
Maksimum Generasi: 1
Peluang Crossover: 0.4
Peluang Mutasi: 0.3

Proses

Gambar 4.1. Tampilan utama aplikasi

4.2.1 Input Data

Input data pada aplikasi ini meliputi data alternatif dan variabel algoritma genetika. Komponen yang merupakan algoritma genetika yaitu batas bawah, batas atas, jumlah kromosom, maksimum generasi, peluang crossover dan peluang mutasi. Batas atas dan batas bawah menentukan panjang satu gen dalam satu kromosom sesuai dengan rumus 2.22 :

$$L_i = \lceil 2 \log[(b - a)10^2 + 1] \rceil \quad (2.22)$$

Sedangkan jumlah kromosom menentukan banyaknya kromosom dalam satu populasi atau generasi. Maksimum generasi menentukan seberapa banyak generasi yang diinginkan dalam satu proses algoritma genetika. Peluang crossover dan peluang mutasi menentukan berapa banyak kromosom yang terkena crossover dan mutasi.

4.2.1.1 Data Alternatif

Data alternatif digunakan untuk menyimpan, mengedit atau menghapus data alternatif yaitu, ID, nama, masa kerja, usia, pangkat, beban kerja, tugas tambahan dan prestasi kerja.

The screenshot shows a software interface titled 'ALGORITMA GENETIKA' with a sub-tab 'INPUT DATA ALTERNATIF'. The interface includes a form for entering data for 7 individuals, a table of existing data, and buttons for adding, updating, deleting, and canceling.

Form Fields:

- ID: 7
- Nama: [Text Field]
- Masa Kerja: [Text Field]
- Usia: [Text Field]
- Pangkat: [Text Field]
- Beban Kerja: [Text Field]
- Tugas Tambahan: [Text Field]
- Prestasi Kerja: [Text Field]

Buttons: Tambah, Update, Delete, ID [Text Field], Can

DATA ALTERNATIF Table:

ID	Nama	Masa Kerja	Usia	Pangkat	Beban Kerja	Tugas Tambahan	Prestasi Kerja	G
1	Raden Supratman	340	34	98	67	45	91	0.564031386
2	Fernando Husein	120	43	65	43	78	98	0.3961821482
3	Yuli Haran	230	54	65	23	43	23	0.363331051
4	Martin Palermo	230	45	32	38	67	65	0.3330897723
5	George Hananto	598	42	24	42	89	90	0.4430051184
6	Yukito Amakura	452	56	23	45	65	87	0.4186625243
7	Mintarizal Utuhila	349	87	12	37	23	78	0.4207443333

Gambar 4.2. Tampilan data alternatif

4.2.2 Deskripsi Program

Berdasarkan analisa dan perancangan proses yang terdapat pada Bab III, maka pada sub bahasan ini akan dijelaskan implementasi proses-proses tersebut

4.2.2.1 Proses Normalisasi

Tahap persiapan data dimulai dengan normalisasi matriks alternatif menjadi matriks normalisasi. Prosedur normalisasi matriks alternatif dapat dilihat pada gambar 4.3.

```
1 Private Sub normalisasi()  
2 Dim i, j, n, max(10), min(10) As Double  
3 Dim s1, s2, r(10), t(10) As String  
4  
5 'banyaknya kriteria  
6 n = 6  
7  
8 'bentuk string sql untuk mencari maksimum tiap kolom  
9 SQL = "SELECT MAX(Masa_Kerja), MAX(Usia), MAX(Pangkat),  
10 MAX(Beban_Kerja), MAX(Tugas_Tambahan), MAX(Prestasi_Kerja)  
11 FROM alt2"  
12 rs.Open SQL, koneksi, adOpenDynamic, adLockPessimistic  
13  
14 'masukkan maksimum tiap kolom dalam array max  
15 For i = 1 To 6  
16     max(i) = rs.Fields(i - 1)  
17 Next i  
18 rs.Close  
19  
20 'bentuk string sql untuk mencari minimum tiap kolom  
21 SQL = "SELECT MIN(Masa_Kerja), MIN(Usia), MIN(Pangkat),  
22 MIN(Beban_Kerja), MIN(Tugas_Tambahan), MIN(Prestasi_Kerja)  
23 FROM alt2"  
24 ar.Open SQL, koneksi, adOpenDynamic, adLockOptimistic  
25  
26 'masukkan minimum tiap kolom dalam array min  
27 For i = 1 To 6  
28     min(i) = ar.Fields(i - 1)  
29 Next i  
30 ar.Close  
31  
32 'bentuk matrik B atau matrik normalisasi  
33 For i = 1 To cdat  
34     For j = 1 To 6  
35         matB(i, j) = (matA(i, j) - min(j)) / (max(j) - min(j))  
36         Text6.Text = Text6.Text & Format(Round(matB(i, j),  
37             3), "#0.000") & Space(3)  
38     Next j  
39     Text6.Text = Text6.Text & vbCrLf  
40 Next i
```

41	
42	End Sub

Gambar 4.3 Prosedur Normalisasi

4.2.2.2 Pembentukan Matriks Saaty

Tahap persiapan data selanjutnya adalah pembentukan matrik Saaty . Matrik Saaty adalah matrik yang memiliki nilai tiap elemen didasarkan pada bobot preferensi yang diberikan oleh pengambil keputusan. Bobot preferensi tersebut menunjukkan tingkat kepentingan satu kriteria terhadap kriteria yang lain. Skala yang digunakan adalah skala perbandingan Saaty. Prosedurnya dapat dilihat pada gambar 4.4

```

1 Private Sub resiprok()
2 Dim dty() As String
3 Dim n, i, j As Double
4
5 dty = Split("2/2/3/5/6/2/3/4/5/2/4/5/2/3/2", "/", 15)
6 n = 0
7 For i = 1 To 6
8     For j = i To 6
9         If j = i Then
10             matD(i, j) = 1
11         Else
12             matD(i, j) = dty(n)
13             n = n + 1
14         End If
15     Next j
16 Next i
17
18 For i = 2 To 6
19     For j = 1 To i - 1
20         matD(i, j) = 1 / matD(j, i)
21     Next j
22 Next i
23
24
25 For i = 1 To 6
26     For j = 1 To 6
27         Text7.Text = Text7.Text & Format(Round(matD(i, j),
28 3), "#0.000") & Space(3)
29     Next j
30     Text7.Text = Text7.Text & vbCrLf
31 Next i
32
33 End Sub

```

Gambar 4.4 Pembentukan Matriks Saaty

4.2.2.3 Pembentukan Populasi Awal

Tahap selanjutnya adalah tahap pembentukan populasi awal. Pada tahap pembentukan populasi awal yang dilakukan adalah menghitung panjang gen (L_i) dengan range tertentu $[a,b]$, dapat dilihat pada gambar 4.5 baris ke-11 dan ke-12. Selanjutnya menghitung panjang kromosom berdasarkan panjang gen yang telah diketahui ($L_i * n$), dapat dilihat pada gambar 4.5 baris ke-18. Kemudian dibentuk populasi awal, Pembentukan kromosom dapat dilihat pada gambar 4.5 baris ke-31 sampai ke-44. Kromosom yang digunakan adalah kromosom biner. Dalam membentuk satu kromosom akan dirandom angka 1 dan 0 sebanyak panjang kromosom. Kemudian dihitung nilai fitnessnya. Pada gambar 4.5 baris ke-59 dilakukan penghitungan nilai fitness. Setelah terbentuk populasi awal maka populasi tersebut disimpan dalam database pada tabel genx. Prosedur pembentukan populasi awal dapat dilihat pada gambar 4.5

```
1 Private Sub Bangun()  
2 Dim i, j, k, m, y, z As Double  
3 Dim xi(10), yi(10), wi(10), txi, jbi, ri, bi, t, h, p,  
4 tot, fitn As Double  
5  
6 List1.Clear  
7 List4.Clear  
8 koneksi.Execute "DELETE FROM fitmax"  
9  
10 'panjang gen  
11 c = ((b - a) * 10 ^ 2) + 1  
12 l = Round(Log(c) / Log(2))  
13  
14 'Jumlah Kromosom  
15 jml = Val(Trim(Text3.Text))  
16  
17 'panjang kromosom  
18 pnjng = l * 6  
19  
20 s = "Generasi Awal"  
21 List1.AddItem s  
22  
23 s = "ID" & Space(8) & "Kromosom" & Space(73) &  
24 "Fitness"  
25 List1.AddItem s  
26  
27 For k = 1 To jml  
28 v = ""  
29 txi = 0  
30 tot = 0  
31  
32 For i = 1 To 6  
33 vi = ""
```

```

34      jbi = 0
35      For j = 1 To 1
36          Randomize
37          ri = Round(Rnd())
38          bi = ri * (2 ^ (1 - j))
39          vi = vi & ri
40          jbi = jbi + bi
41      Next j
42      xi(i) = a + ((b - a) / ((2 ^ 1) - 1)) * jbi
43      v = v & vi
44      txi = txi + xi(i)
45  Next i
46
47  For m = 1 To 6
48      wi(m) = xi(m) / txi
49  Next m
50
51  For y = 1 To 6
52      For z = 1 To 6
53          t = matD(y, z) * wi(z)
54          h = t - wi(y)
55          p = h ^ 2
56          tot = tot + p
57      Next z
58  Next y
59
60  fitn = 1 / tot
61
62  If k < 10 Then
63      List1.AddItem k & ". " & Space(8) & v & Space(5)
64  &
65      fitn
66  ElseIf k < 100 Then
67      List1.AddItem k & ". " & Space(6) & v & Space(5)
68  &
69      fitn
70  Else
71      List1.AddItem k & ". " & Space(4) & v & Space(5)
72  &
73      fitn
74  End If
75
76  's = k & Space(10) & v & Space(5) & fitn
77  'List1.AddItem s
78
79  SQL = "SELECT * FROM genx"
80  rs.Open SQL, koneksi, adOpenDynamic,
81  adLockOptimistic
82  rs.Find "ID =" & k
83  If rs.EOF = True Then
84      rs.AddNew
85  End If
86  rs!ID = k
87  rs!kromosom = v

```



```

88      rs!fitness = fitn
89      rs.Update
90      rs.Close
91      Next k
92
93      List1.AddItem ""
94
95      Adodc1.CommandType = adCmdTable
96      Adodc1.RecordSource = "alt2"
97      Adodc1.Refresh
98
99      For i = 1 To cdat
100         Adodc1.Recordset!g = 0
101         Adodc1.Recordset.Update
102
103         If i < cdat Then
104             Adodc1.Recordset.MoveNext
105         End If
106     Next i
107 End Sub

```

Gambar 4.5 Pembentukan populasi awal

4.2.2.4 Seleksi

Proses selanjutnya adalah seleksi. Metode yang digunakan adalah roda roulette. Langkah pertama adalah menghitung nilai fitness relatif. (gambar 4.6 baris ke-13 sampai ke-21). Kemudian dibangkitkan bilangan random sebanyak jumlah kromosom (gambar 4.6 baris ke-24 sampai ke -27). Selanjutnya adalah proses seleksi roda roulette (gambar 4.6 baris ke-30 sampai ke-54). Populasi hasil seleksi disimpan menggantikan populasi awal dalam database pada tabel genx. Prosedur seleksi dapat dilihat pada gambar 4.6

```

1 Private Sub Seleksi()
2     Dim tf, f, p, fit(100), qk(100), rp(100) As Double
3     Dim i, j As Double
4     Dim vbaru(100), z As String
5
6     SQL = "SELECT SUM(fitness) FROM genx"
7     rs.Open SQL, koneksi, adOpenDynamic, adLockOptimistic
8     tf = rs.Fields(0)
9     rs.Close
10
11     'List1.AddItem "Nilai fitnes relatif"
12     qk(0) = 0
13     For i = 1 To jml
14         SQL = "SELECT fitness FROM genx WHERE ID=" & i
15         rs.Open SQL, koneksi, adOpenDynamic, adLockOptimistic
16         f = rs.Fields(0)
17         rs.Close

```

```

18     p = f / tf
19     qk(i) = qk(i - 1) + p
20     'List1.AddItem qk(i)
21 Next i
22 'List1.AddItem ""
23
24 For i = 1 To jml
25     Randomize
26     rp(i) = Rnd()
27 Next i
28
29 'List1.AddItem "Hasil Seleksi"
30 For i = 1 To jml
31     For j = 1 To jml - 1
32         If rp(i) > qk(j) And rp(i) < qk(j + 1) Then
33             SQL = "SELECT * FROM genx WHERE ID=" & j + 1
34             rs.Open SQL, koneksi, adOpenDynamic,
35                 adLockOptimistic
36             vbaru(i) = rs.Fields(1)
37             fit(i) = rs.Fields(2)
38             rs.Close
39             z = "asal:" & j + 1
40         Else
41             If qk(1) > rp(i) Then
42                 SQL = "SELECT * FROM genx WHERE ID=" & 1
43                 rs.Open SQL, koneksi, adOpenDynamic,
44                     adLockOptimistic
45                 vbaru(i) = rs.Fields(1)
46                 fit(i) = rs.Fields(2)
47                 rs.Close
48                 z = "asal:" & 1
49             End If
50         End If
51     Next j
52     'List1.AddItem vbaru(i) & Space(3) & fit(i) & z & " ("
53     & rp(i) & ")"
54 Next i
55
56 List1.AddItem ""
57
58 For i = 1 To jml
59     SQL = "SELECT * FROM genx"
60     rs.Open SQL, koneksi, adOpenDynamic, adLockOptimistic
61     rs.Find "id =" & i
62     rs!kromosom = vbaru(i)
63     rs!fitness = fit(i)
64     rs.Update
65     rs.Close
66 Next i
67 End Sub

```

Gambar 4.6 Seleksi

4.2.2.5 Crossover

Tahap selanjutnya adalah *crossover*. Metode yang digunakan adalah *crossover* biner dengan *cut point* satu titik. Langkah pertama adalah membentuk bilangan random sebanyak jumlah kromosom (gambar 4.7 baris ke 16 sampai ke-20). Kemudian mencari kromosom induk yang terkena *crossover* (gambar 4.7 baris ke-25 sampai ke-37). Tahap selanjutnya adalah proses *crossover* berdasarkan peluang *crossover* yang telah diinputkan. Proses *crossover* dimulai dengan menentukan titik potong secara *random* dengan *range* maksimum panjang kromosom, kemudian dilakukan penukaran bagian kanan titik potong dari kedua induk agar menghasilkan kromosom anak. Setiap anak yang dihasilkan akan dihitung nilai fitnessnya (gambar 4.7 baris ke-41 sampai ke-85). Anak hasil *crossover* tersebut akan diseleksi, apabila nilai fitnessnya lebih baik daripada induknya maka akan menggantikan induk tersebut (gambar 4.7 baris ke-89 sampai ke-129). Prosedur *crossover* dapat dilihat pada gambar 4.7

```
1 Private Sub cross()  
2 Dim rc(100) As Double  
3 Dim no(100) As Double  
4 Dim fitn, rn, sisa, fitsmt As Double  
5 Dim fitank1, fitank2 As Double  
6 Dim i, j, k As Double  
7 Dim vbaru(100) As String  
8 Dim a1, a2, b1, b2, anak1, anak2, vsmt, vanak As String  
9  
10  
11 'peluang crossover  
12 pcros = Val(Trim(Text5.Text))  
13  
14 'Random bilangan sebanyak jumlah kromosom  
15 'List1.AddItem "Random bilangan sebanyak jumlah  
16 kromosom"  
17 For i = 1 To jml  
18 Randomize  
19 rc(i) = Rnd()  
20 'List1.AddItem rc(i)  
21 Next i  
22  
23 'List1.AddItem ""  
24 'mencari kromosom induk yang terkena crossover  
25 'List1.AddItem "kromosom induk yang terkena crossover"  
26 j = 0  
27 For i = 1 To jml  
28 SQL = "SELECT * FROM genx WHERE id=" & i  
29 rs.Open SQL, koneksi, adOpenDynamic,  
30 adLockOptimistic  
31 If rc(i) <= pcros Then
```

```

32      vbaru(i) = rs.Fields(1)
33      j = j + 1
34      no(j) = i
35      s = "v ke-" & i & ": " & vbaru(i)
36      'List1.AddItem s
37      End If
38      rs.Close
39      Next i
40
41      'List1.AddItem ""
42
43      'proses crossover
44      'List1.AddItem "Anak hasil crossover"
45      For i = 1 To j - 1
46          For k = i + 1 To j
47              Randomize
48              rn = Int((pnjng - 1) * Rnd() + 1)
49              sisa = pnjng - rn
50
51              'pemotongan kromosom induk
52              a1 = Mid(vbaru(no(i)), 1, rn)
53              a2 = Mid(vbaru(no(i)), rn + 1, sisa)
54
55              b1 = Mid(vbaru(no(k)), 1, rn)
56              b2 = Mid(vbaru(no(k)), rn + 1, sisa)
57
58              'pembentukan kromosom anak pertama
59              anak1 = a1 & b2
60              fitank1 = hitfit(Trim(anak1))
61
62              SQL = "SELECT * FROM tempgen1"
63              rs.Open SQL, koneksi, adOpenDynamic,
64              adLockOptimistic
65
66              rs.AddNew
67              rs!kromosom = anak1
68              rs!fitness = fitank1
69              rs.Update
70
71              'pembentukan kromosom anak kedua
72              anak2 = b1 & a2
73              fitank2 = hitfit(Trim(anak2))
74
75              rs.AddNew
76              rs!kromosom = anak2
77              rs!fitness = fitank2
78              rs.Update
79
80              s = "v : " & anak1 & " = " & fitank1
81              'List1.AddItem s
82
83              s = "v : " & anak2 & " = " & fitank2
84              'List1.AddItem s
85              rs.Close

```

```

86      Next k
87      Next i
88
89      'List1.AddItem ""
90
91      'Menyimpan sementara kromosom sebelumnya plus anak dalam
92      tabel tempgen1
93      For i = 1 To jml
94          SQL = "SELECT * FROM genx WHERE id=" & i
95          rs.Open      SQL,      koneksi,      adOpenDynamic,
96          adLockOptimistic
97          vsmt = rs!kromosom
98          fitsmt = rs!fitness
99          SQL = "INSERT INTO tempgen1(kromosom,fitness)
100          VALUES('" & vsmt & "'," & fitsmt & ")"
101          koneksi.Execute SQL
102          rs.Close
103      Next i
104
105      koneksi.Close
106      koneksi.Open
107
108      'Mengambil hanya n kromosom dengan fitness terbaik
109      'List1.AddItem "Generasi setelah crossover"
110
111      SQL = "SELECT * FROM tempgen1 ORDER BY fitness DESC"
112      rs.Open SQL, koneksi, adOpenDynamic, adLockOptimistic
113      rs.MoveFirst
114
115      For i = 1 To jml
116          vanak = rs.Fields(0)
117          fitn = rs.Fields(1)
118
119          s = i & Space(10) & vanak & Space(5) & fitn
120          'List1.AddItem s
121
122          'List1.AddItem vanak & "> " & fitn
123          rs.MoveNext
124
125          SQL = "SELECT * FROM genx"
126          ar.Open      SQL,      koneksi,      adOpenDynamic,
127          adLockOptimistic
128          ar.Find "ID=" & i
129          ar!kromosom = vanak
130          ar!fitness = fitn
131          ar.Update
132          ar.Close
133      Next i
134      rs.Close
135      'List1.AddItem ""
136      koneksi.Execute "Delete from tempgen1"
137      End Sub

```

Gambar 4.7 Crossover

4.2.2.6 Mutasi

Tahap selanjutnya adalah mutasi. Metode yang digunakan adalah mutasi biner. Langkah pertama adalah menghitung jumlah gen dari semua kromosom dalam satu generasi. Kemudian mendaftar semua posisi gen yang terkena mutasi berdasarkan peluang mutasi yang telah diinputkan. Kemudian menyambung keseluruhan kromosom dalam satu string (gambar 4.8 baris ke-8 sampai ke-48). Kemudian proses seleksi dilakukan dengan mengganti nilai gen yang terkena mutasi dengan angka 0 jika gen yang terkena mutasi adalah angka 1, begitu juga sebaliknya. Setelah itu dilakukan pemotongan kembali string gabungan, menjadi kromosom hasil mutasi sebanyak jumlah kromosom dan dihitung fitnessnya (gambar 4.8 baris ke-52 sampai ke-84). Hasil mutasi tersebut akan diseleksi, apabila nilai fitnessnya lebih baik daripada induknya maka akan menggantikan induk tersebut (gambar 4.8 baris ke-87 sampai ke-139). Prosedur mutasi dapat dilihat pada gambar 4.8

```
1 Private Sub mutasi()  
2 Dim arrPos(10000) As Double  
3 Dim vmut(10000) As String  
4 Dim t, vgab, vanak As String  
5 Dim i, fitn, jmlBit, con, bilCak, x, y, z, fitmut As  
6 Double  
7  
8 'Jumlah bit dari semua kromosom dalam satu generasi  
9 jmlBit = jml * pnjng  
10  
11 'peluang mutasi  
12 pmut = Val(Trim(Text8.Text))  
13  
14 'mendaftar posisi bit yang terkena mutasi  
15 con = 0  
16 For i = 1 To jmlBit  
17 Randomize  
18 bilCak = Rnd()  
19 If bilCak < pmut Then  
20 con = con + 1  
21 arrPos(con) = i  
22 End If  
23 Next i  
24  
25 'mencari kromosom yang bit-nya terkena mutasi  
26 If con > 0 Then  
27 'List1.AddItem "mencari kromosom yang bit-nya terkena  
28 mutasi"  
29 For i = 1 To con  
30 x = arrPos(i) Mod (1 * 6)  
31 If x >= 0 Then
```



```

32         y = (arrPos(i) \ (1 * 6)) + 1
33         vmut(y) = "1"
34     End If
35     'List1.AddItem arrPos(i) & ">" & y
36 Next i
37
38 'List1.AddItem ""
39
40 'menyambung keseluruhan kromosom dalam satu string
41 vgab = ""
42 For i = 1 To jml
43     SQL = "SELECT * FROM genx WHERE ID=" & i
44     rs.Open SQL, koneksi, adOpenDynamic,
45     adLockOptimistic
46     v = rs.Fields(1)
47     vgab = vgab & v
48     rs.Close
49 Next i
50
51 'proses mutasi
52
53 For i = 1 To con
54     x = Mid(vgab, arrPos(i), 1)
55     If x = "0" Then
56         y = Mid(vgab, 1, arrPos(i) - 1)
57         z = Mid(vgab, arrPos(i) + 1, jmlBit - arrPos(i))
58         vgab = y & "1" & z
59     Else
60         y = Mid(vgab, 1, arrPos(i) - 1)
61         z = Mid(vgab, arrPos(i) + 1, jmlBit - arrPos(i))
62         vgab = y & "0" & z
63     End If
64 Next i
65
66 'menampilkan hasil mutasi
67 'List1.AddItem "Hasil Mutasi ->" & con
68 For i = 1 To jml
69     SQL = "SELECT * FROM tempgen1"
70     rs.Open SQL, koneksi, adOpenDynamic,
71     adLockOptimistic
72     If vmut(i) = "1" Then
73         t = Mid(vgab, ((i - 1) * pnjng) + 1, pnjng)
74         fitmut = hitfit(Trim(t))
75
76         'List1.AddItem vmut(i)
77         'List1.AddItem t & Space(5) & fitmut
78
79         rs.AddNew
80         rs!kromosom = t
81         rs!fitness = fitmut
82         rs.Update
83     End If
84     rs.Close
85 Next i

```

```

86 End If
87
88 'List1.AddItem ""
89 'menyimpan sementara kromosom sebelumnya plus anak hasil
90 mutasi dalam tabel tempgen1
91 For i = 1 To jml
92     SQL = "SELECT * FROM genx WHERE id=" & i
93     rs.Open SQL, koneksi, adOpenDynamic,
94     adLockOptimistic
95
96     vanak = rs!kromosom
97     fitn = rs!fitness
98
99     SQL = "INSERT INTO tempgen1(kromosom,fitness)
100     VALUES('" & vanak & "','" & fitn & ")"
101     koneksi.Execute SQL
102     rs.Close
103 Next i
104
105 'mengambil hanya n kromosom dengan fitness terbaik
106 SQL = "SELECT * FROM tempgen1 ORDER BY fitness DESC"
107 rs.Open SQL, koneksi, adOpenDynamic, adLockOptimistic
108 rs.MoveFirst
109
110 'List1.AddItem "Generasi setelah Mutasi"
111 For i = 1 To jml
112     vanak = rs.Fields(0)
113     fitn = rs.Fields(1)
114
115     If i < 10 Then
116         List1.AddItem i & ". " & Space(8) & vanak &
117         Space(5) & fitn
118     ElseIf i < 100 Then
119         List1.AddItem i & ". " & Space(6) & vanak &
120         Space(5) & fitn
121     Else
122         List1.AddItem i & ". " & Space(4) & vanak &
123         Space(5) & fitn
124     End If
125     's = i & Space(10) & vanak & Space(5) & fitn
126     'List1.AddItem s
127
128     'Text18.Text = Text18.Text & s & vbNewLine
129     'List1.AddItem vanak & "> " & fitn
130     rs.MoveNext
131
132     SQL = "SELECT * FROM genx"
133     ar.Open SQL, koneksi, adOpenDynamic,
134     adLockOptimistic
135     ar.Find "ID=" & i
136     ar!kromosom = vanak
137     ar!fitness = fitn
138     ar.Update
139     ar.Close

```

140	Next i
141	rs.Close
142	'List1.AddItem ""
143	koneksi.Execute "Delete from tempgen1"
144	End Sub

Gambar 4.8 Mutasi

4.2.2.7 Perangkingan

Pada proses perangkingan, pertama dilakukan adalah mengambil kromosom terbaik diantara semua generasi (gambar 4.9 baris ke-6 sampai ke-11) . Kemudian dilakukan penghitungan nilai g tiap alternatif. Nilai g adalah nilai yang menentukan ranking suatu alternatif terhadap alternatif yang lain. Tahapnya adalah memotong kromosom menjadi enam bagian, kemudian potongan tersebut diubah menjadi nilai desimal, lalu menghitung nilai x atau variabel temporer dan terakhir menghitung nilai w atau bobot kepentingan relatif. Dari nilai w tersebut dapat dihitung nilai g tiap alternatif. Prosedur perangkingan dapat dilihat pada gambar 4.9

1	Private Sub ranking()
2	Dim xi(10), wl(10), g, gtmp As Double
3	Dim jba, ft, i, n, m, txa, j As Double
4	Dim bin(10), kr As String
5	
6	'ambil kromosom terbaik diantara semua generasi
7	SQL = "select top 1 * from fitmax order by fitness DESC"
8	rs.Open SQL, koneksi, adOpenDynamic, adLockOptimistic
9	kr = rs.Fields(0)
10	ft = rs.Fields(1)
11	rs.Close
12	List4.AddItem "Kromosom terbaik = " & kr
13	List4.AddItem "Fitness terbaik = " & ft
14	
15	'potong kromosom menjadi 6 bagian
16	For i = 1 To 6
17	bin(i) = Mid(Trim(kr), (i - 1) * 1 + 1, 1)
18	'List4.AddItem "bin(" & i & ") = " & bin(i)
19	Next i
20	'List4.AddItem ""
21	
22	n = 1
23	Do While n < 7
24	'ubah kromosom menjadi bentuk desimal
25	jba = BinaryToDecimal(bin(n))
26	
27	'hitung nilai x
28	xi(n) = a + ((b - a) / ((2 ^ 1) - 1)) * jba)
29	'List4.AddItem "v(" & n & ") = " & bin(n) & " = " &
30	jba
31	'List4.AddItem "X(" & n & ") = " & xi(n)

```

32      txa = txa + xi(n)
33      n = n + 1
34  Loop
35  'List4.AddItem ""
36  'hitung nilai w
37  For m = 1 To 6
38      wi(m) = xi(m) / txa
39      'List4.AddItem "X(" & m & ") = " & xi(m)
40      'List4.AddItem "W(" & m & ") = " & wi(m)
41  Next m
42  'List4.AddItem ""
43  'hitung nilai g
44  'List4.AddItem "Nilai g"
45  For i = 1 To cdat
46      SQL = "SELECT * FROM alt2"
47      rs.Open SQL, koneksi, adOpenDynamic, adLockOptimistic
48      g = 0
49      For j = 1 To 6
50          gtmp = (wi(j) * matB(i, j))
51          g = g + gtmp
52          rs.Find "ID=" & i
53          rs!g = g
54          rs.Update
55      Next j
56      'List4.AddItem "g(" & i & ") = " & g
57      rs.Close
58  Next i
59  'List4.AddItem ""
60  'cetak urutan data
61  List4.AddItem "Urutan Data"
62  List4.AddItem "No. " & Space(4) & " Nama "
63  SQL = "SELECT ID,nama,g FROM alt2 ORDER BY g DESC"
64  rs.Open SQL, koneksi, adOpenDynamic, adLockOptimistic
65  For i = 1 To cdat
66
67      If i < 10 Then
68          List4.AddItem i & ". " & Space(8) & rs.Fields(1) &
69  " G -> " & rs.Fields(2)
70      ElseIf i < 100 Then
71          List4.AddItem i & ". " & Space(6) & rs.Fields(1) &
72  " G -> " & rs.Fields(2)
73      Else
74          List4.AddItem i & ". " & Space(4) & rs.Fields(1) &
75  " G -> " & rs.Fields(2)
76      End If
77
78      If i < cdat Then
79          rs.MoveNext
80      End If
81  Next i
82  rs.Close
83  End Sub

```

Gambar 4.9 Perangkingan

4.3 Penerapan Aplikasi

Aplikasi ini diterapkan dengan mengambil data dari database data.accdb atau memasukkan data alternatif pada form input data alternatif.

Langkah selanjutnya adalah memasukkan nilai dari parameter genetika. Secara *default* aplikasi memiliki kombinasi parameter seperti tabel 4.1

Tabel 4.1. Kombinasi default parameter genetika

Batas atas	1
Batas bawah	0
Jumlah kromosom	100
Maksimum Generasi	100
Peluang <i>Crossover</i>	0.4
Peluang Mutasi	0.1

Pada tabel 4.1 dapat dijelaskan bahwa pada saat proses genetika, jumlah generasi atau iterasi yang dilakukan adalah 100 generasi. Setiap generasi menghasilkan 100 kromosom atau 100 individu. Individu dipilih dengan peluang *crossover* adalah sebesar 0.4. Sedangkan peluang mutasi sebanyak 0.1.

Apabila nilai parameter sudah disimpan, maka proses selanjutnya adalah proses pencarian bobot kepentingan relatif (w) MADM dengan algoritma genetika.

ALGORITMA GENETIKA

INPUT DATA ALTERNATIF

Matrik Alternatif

340.000	34.000	98.000	67.000	45.000	91.000
120.000	43.000	65.000	43.000	78.000	98.000
230.000	54.000	65.000	23.000	43.000	23.000
230.000	45.000	32.000	98.000	67.000	65.000
598.000	42.000	24.000	42.000	89.000	90.000
452.000	56.000	23.000	45.000	65.000	87.000
349.000	87.000	12.000	37.000	23.000	78.000

Gambar 4.10 Tampilan Matrik Alternatif

Matrik Normalisasi						
0.460	0.000	1.000	0.587	0.333	0.907	
0.000	0.170	0.616	0.267	0.833	1.000	
0.230	0.377	0.616	0.000	0.303	0.000	
0.230	0.208	0.233	1.000	0.667	0.560	
1.000	0.151	0.140	0.253	1.000	0.893	
0.695	0.415	0.128	0.293	0.636	0.853	
0.479	1.000	0.000	0.187	0.000	0.733	

Gambar 4.11 Tampilan Matrik Normalisasi

Matrik Saaty						
1.000	2.000	2.000	3.000	5.000	6.000	
0.500	1.000	2.000	3.000	4.000	5.000	
0.500	0.500	1.000	2.000	4.000	5.000	
0.333	0.333	0.500	1.000	2.000	3.000	
0.200	0.250	0.250	0.500	1.000	2.000	
0.167	0.200	0.200	0.333	0.500	1.000	

Gambar 4.12 Tampilan Matrik Saaty

Gambar 4.10 adalah tampilan matrik alternatif yang merupakan matrik dari data skor peserta sertifikasi guru. Dari matrik alternatif tersebut dihasilkan matrik normalisasi, seperti terlihat pada gambar 4.11. Sedangkan matrik dibuat dari perbandingan berpasangan yang diberikan oleh pengambil keputusan yang didasarkan pada matrik, Saaty yang terdapat pada bab 3, terlihat pada gambar 4.12

Batas Bawah	0
Batas Atas	1
Jumlah Kromosom	20
Maksimum Generasi	100
Peluang Crossover	0.9
Peluang Mutasi	0.1

Gambar 4.13 Tampilan Input Parameter Genetika

Proses Algoritma Genetika

Generasi Awal		
ID	Kromosom	Fitness
1.	0010010000110110101100010010111101111010	6.38681648672873E-02
2.	11110000001011100001110010011000101100000	0.19021081181873
3.	100011101111001110011011010111100000011110	0.285941344762551
4.	001001001101001101111010101011110001110001	0.107903672907265
5.	111000111110111001110110010111111100100101	0.414365006207695
6.	01111011010000110001000100101111111011111	0.152603934622296
7.	10100010010011101011111011100110101100011	0.176788962416341
8.	101101100000001011100001110001000100010110	0.59618807269822
9.	101010000010111100111001101111011110001000	0.169529960926684
10.	000101100100110011011010101000101011110111	4.04830553650461E-02
11.	100111110110001001100101010111011011110011	0.190792998713959
12.	01010111001101000011000101011000111110101	7.53724578529285E-02
13.	011110101110111011010101011101110101000111	0.305367391832731
14.	00111011010100001110110001011010010001011	0.154549556500314
15.	00011110101001010011010011000011100000110	0.641039859642691
16.	1110000111000100000010110101010001010111	0.393411398141481
17.	0011100110110110000000010010001011001001100	0.137508960394188

Gambar 4.14 Tampilan Hasil Proses Algoritma Genetika

Setelah input parameter genetika tersebut diisi seperti pada gambar 4.13., klik tombol proses untuk menjalankan proses algoritma genetika. Gambar 4.14 menunjukkan hasil proses algoritma genetika. Setelah proses algoritma genetika untuk mencari nilai bobot kepentingan relatif (w) selesai, maka dilakukan proses perangkungan, seperti pada gambar 4.15. Dari proses perangkungan dihasilkan urutan alternatif berdasarkan nilai ranking (g)

Perangkungan

Kromosom terbaik = 111011010100101000010010001000101110001111
 Fitness terbaik = 12.837239475876
 Urutan Data

No.	Nama	
1.	Mamik Idah Wahyuni S.Pd	G -> 0.747854367267221
2.	Masrukin	G -> 0.588954635108481
3.	Nurkhalis	G -> 0.588630223830517
4.	Sri Gunawati	G -> 0.51127907112791
5.	Mustakim S.Pd	G -> 0.242603550295858

Gambar 4.15 Tampilan Hasil Proses Perangkungan

Tambah Data ▼

ID	8	
Nama	Achmad Dani Zulfikar	+
Masa Kerja	120	Tambah
Usia	45	Update
Pangkat	4	Delete
Beban Kerja	27	
Tugas Tambahan	2	
Prestasi Kerja	77	

ID

Gambar 4.16 Tampilan Proses Tambah Data

Pada gambar 4.16 terlihat tampilan proses tambah data. Setelah isian yang disediakan selesai ditulis maka disimpan di dalam database.

Edit Data

ID	3	
Nama	Yuli Harun	+
Masa Kerja	230	Tambah
Usia	54	Update
Pangkat	3	Delete
Beban Kerja	23	
Tugas Tambahan	2	
Prestasi Kerja	23	

ID

Gambar 4.17 Tampilan Proses Edit Data

Untuk mengedit data, sebelumnya dicari data yang akan diedit berdasarkan ID. Setelah data yang dicari ditemukan, maka data tersebut dapat diedit, kemudian diupdate. Prosesnya terlihat pada gambar 4.17

Hapus Data	
ID	8
Nama	Achmad Dani Zulfikar
Masa Kerja	120
Usia	45
Pangkat	4
Beban Kerja	27
Tugas Tambahan	2
Prestasi Kerja	77
Tambah Update Delete	
ID 8 Cari	

Gambar 4.18 Tampilan Proses Hapus Data

Untuk menghapus data, sebelumnya dicari data yang akan dihapus berdasarkan ID. Setelah data yang dicari ditemukan, maka data tersebut dapat dihapus. Prosesnya terlihat pada gambar 4.18

4.4 Uji Coba dan Analisa Hasil

Pada aplikasi ini terjadi perubahan nilai *fitness* dan konvergensi yang dipengaruhi oleh peluang crossover dan peluang mutasi .

4.4.1. Pengaruh peluang *crossover* dan peluang mutasi terhadap nilai *fitness*

Uji coba yang pertama dilakukan untuk mengetahui pengaruh peluang *crossover* dan peluang mutasi terhadap nilai *fitness*. Untuk setiap peluang *crossover* dan peluang mutasi dilakukan percobaan sebanyak 5 kali dan diambil nilai rata-ratanya.

Parameter genetika yang digunakan, adalah maksimum generasi sebanyak 100 generasi, batas atas 1, batas bawah 0, dan jumlah kromosom 100 kromosom.

Tabel 4.2 Nilai *Fitness* pada peluang *crossover* 0.1 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	15.04	14.94	10.34	8.84	9.94
2	14.84	14.61	9.18	12.35	12.33
3	14.12	14.34	7.05	6.91	7.38
4	15.99	13.24	12.14	8.72	6.62
5	14.80	14.61	10.09	10.79	7.79
Rataan	14.758	13.348	9.76	9.522	8.812

Tabel 4.3 Nilai *Fitness* pada peluang *crossover* 0.3 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	14.88	14.74	14.68	8.28	12.86
2	15.03	14.99	15.01	7.27	8.09
3	15.05	14.90	13.73	8.99	5.46
4	15.04	14.00	13.31	12.48	8.98
5	14.96	14.25	7.78	14.66	6.55
Rataan	14.992	14.576	12.902	10.336	8.387

Tabel 4.4 Nilai *Fitness* pada peluang *crossover* 0.5 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	14.99	14.55	13.27	12.54	14.74
2	15.12	15.03	10.59	13.39	14.96
3	15.10	13.83	12.25	12.82	6.64
4	14.59	14.33	13.23	13.88	12.01
5	15.02	14.62	14.85	14.55	12.40
Rataan	14.964	14.472	12.838	13.436	12.15

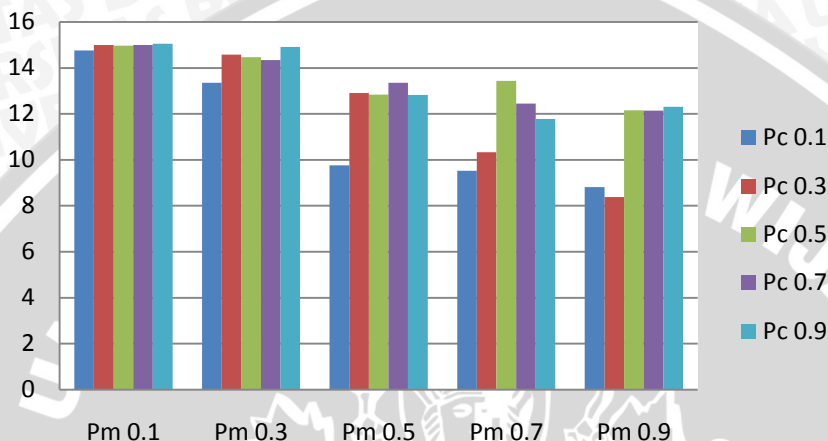
Tabel 4.5 Nilai *Fitness* pada peluang *crossover* 0.7 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	15.07	15.06	11.81	14.50	14.21
2	14.99	14.91	11.96	10.23	10.23
3	15.09	14.81	14.45	12.72	14.05
4	14.80	12.18	14.34	9.94	7.54
5	15.01	14.73	14.23	14.84	14.65
Rataan	14.992	14.338	13.358	12.446	12.136

Tabel 4.6 Nilai *Fitness* pada peluang *crossover* 0.9 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	15.10	14.84	9.20	14.27	14.98
2	14.98	14.92	11.81	11.39	8.71
3	15.02	14.99	14.41	8.51	13.79
4	15.04	14.94	14.28	11.08	10.52
5	15.12	14.88	14.14	13.64	13.55
Rataan	15.052	14.912	12.826	11.778	12.31

Perbandingan Fitness Terhadap Perubahan Peluang Crossover dan Peluang Mutasi



Grafik 4.1 Perbandingan *fitness* dengan peluang mutasi dan peluang *crossover*

Berdasarkan hasil ujicoba diketahui bahwa perubahan nilai peluang *crossover* dan mutasi memberikan perubahan terhadap nilai *fitness*-nya. Pada grafik 4.1 diketahui bahwa peningkatan nilai peluang *crossover* menyebabkan nilai *fitness* semakin besar dan peningkatan peluang mutasi membuat nilai *fitness* semakin kecil. Misalkan pada peluang *crossover* 0.1 dan peluang mutasi 0.1 nilai *fitness*-nya adalah 14.758 sedangkan pada peluang *crossover* 0.9 dan peluang mutasi 0.1 *fitness*-nya adalah 15.052. Lalu pada peluang *crossover* 0.1 dan peluang mutasi 0.1 nilai *fitness*-nya adalah 14.758 sedangkan pada peluang *crossover* 0.1 dan peluang mutasi 0.9 *fitness*-nya adalah 8.812.

Hal ini menunjukkan bahwa peningkatan nilai peluang *crossover* menyebabkan peningkatan nilai *fitness* sedangkan peningkatan nilai

peluang mutasi menyebabkan penurunan nilai fitness. Hal ini disebabkan, metode crossover satu titik tidak menyebabkan perubahan secara menyeluruh pada kromosom induk, semakin besar nilai peluang crossover semakin banyak anak yang dihasilkan. Sedangkan metode mutasi biner menyebabkan perubahan pada tingkat bit di kromosom induk sehingga sifat induk yang sudah baik akan hilang apabila nilai peluang mutasi terlalu besar.

4.4.2. Pengaruh peluang *crossover* dan peluang mutasi terhadap konvergensi

Uji coba yang pertama dilakukan untuk mengetahui pengaruh pengaruh peluang crossover dan peluang mutasi terhadap konvergensi. Untuk setiap peluang crossover dan peluang mutasi dilakukan percobaan sebanyak 5 kali dan diambil nilai rata-ratanya.

Parameter genetika yang digunakan, adalah maksimu generasi sebanyak 100 generasi, batas atas 1, batas bawah 0, dan jumlah kromsoms 100 kromosom.

Tabel 4.7 Nilai konvergensi pada peluang *crossover* 0.1 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	68	43	13	15	17
2	86	21	80	9	16
3	51	32	41	11	7
4	33	67	32	22	4
5	36	26	26	20	12
Rataan	54.8	37.8	38.4	15.4	11.2

Tabel 4.8 Nilai konvergensi pada peluang *crossover* 0.3 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	69	46	7	11	9
2	38	93	7	11	10
3	17	7	10	62	5
4	37	18	5	20	9
5	50	9	10	8	12
Rataan	38.2	34.6	7.8	22.4	9

Tabel 4.9 Nilai konvergensi pada peluang *crossover* 0.5 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	35	6	11	9	8
2	21	7	9	7	6
3	45	4	6	7	6
4	38	7	7	7	6
5	30	6	11	6	6
Rataan	33.8	6	8.8	7.2	6.4

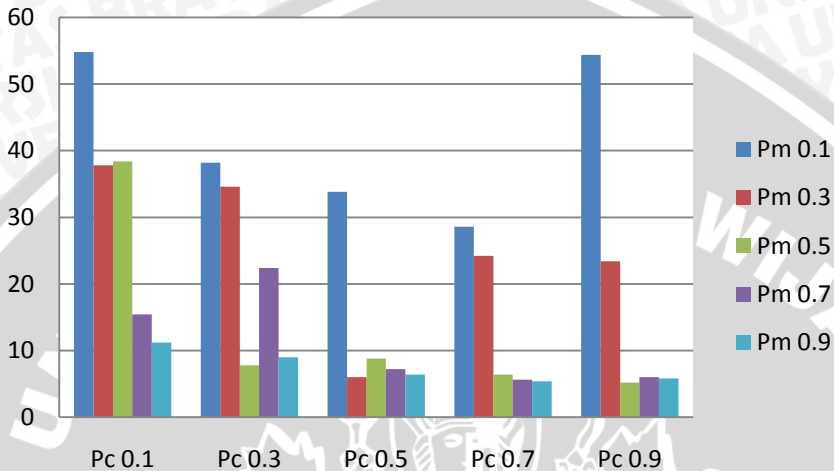
Tabel 4.10 Nilai konvergensi pada peluang *crossover* 0.7 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	26	9	5	4	7
2	36	8	6	5	6
3	41	9	8	7	5
4	11	7	7	6	4
5	19	88	6	6	5
Rataan	28.6	24.2	6.4	5.6	5.4

Tabel 4.11 Nilai konvergensi pada peluang *crossover* 0.9 dan peluang mutasi yang berbeda

	Pm 0,1	Pm 0,3	Pm 0,5	Pm 0,7	Pm 0,9
1	34	6	6	7	8
2	37	11	6	6	4
3	40	15	5	4	8
4	67	9	5	6	4
5	94	76	4	7	5
Rataan	54.4	23.38	5.2	6.0	5.8

Perbandingan Konvergensi Terhadap Perubahan Peluang Crossover dan Peluang Mutasi



Grafik 4.2 Perbandingan konvergensi dengan peluang mutasi dan peluang *crossover*

Berdasarkan hasil ujicoba dan grafik 4.2 diketahui bahwa perubahan nilai peluang *crossover* atau mutasi memberikan perubahan terhadap nilai konvergensi-nya. Misalkan pada peluang crossover 0.1 dan peluang mutasi 0.1 konvergensi pada generasi 54.8 sedangkan pada peluang crossover 0.5 dan peluang mutasi 0.5 konvergensi pada generasi 8.8.

Hal ini menunjukkan bahwa peningkatan nilai peluang crossover dan peluang mutasi menyebabkan nilai konvergensi semakin rendah atau diperoleh konvergensi pada generasi lebih awal. Hal ini disebabkan karena nilai peluang crossover atau mutasi yang lebih tinggi membuat area pencarian terhadap individu yang lebih baik semakin luas, sehingga kemungkinan ditemukan individu terbaik semakin cepat.

Peluang crossover yang disarankan untuk digunakan menyelesaikan permasalahan pencarian bobot MADM menggunakan algoritma genetika ini adalah antara 0.5 sampai 0.9 dan peluang mutasi yang disarankan adalah 0.1 sampai 0.3 karena pada rentang tersebut diperoleh hasil fitness yang optimal.

4.4.3 Membandingkan Hasil Perangkingan dengan Aplikasi dan Perangkingan Manual

Data perangkingan dengan aplikasi dapat dilihat di lampiran 3 dan data perangkingan dengan cara manual dapat dilihat pada lampiran 2. Dari tabel 4.12 dapat dilihat, dengan membandingkan kedua data tersebut didapatkan 34 kali posisi salah dan 66 posisi benar (simbol O menyatakan benar dan X menyatakan salah). Sehingga hasil perangkingan menggunakan aplikasi tidak sama dengan hasil perangkingan dengan cara manual

Tabel 4.12 Perbandingan Hasil Perangkingan Manual dengan Aplikasi

Manual		Aplikasi		Cek
No	Nama	No	Nama	
1	Mukhlis	1	Mukhlis	O
2	Ali Riwayat	2	Ali Riwayat	O
3	Sukirno	3	Sukirno	O
4	Samsudi Jaly	4	Samsudi Jaly	O
5	Misbahul Munir	5	Misbahul Munir	O
6	Sri Gunawati	6	Hasan Bisri	X
7	Hasan Bisri	7	Sri Gunawati	X
8	Afandi	8	Afandi	O
9	Badruz Zaman	9	Badruz Zaman	O
10	Masrukin	10	Mufrihul Anam	X
11	Mufrihul Anam	11	Masrukin	X
12	Mohammad	12	Rosyidi	X
13	Rosyidi	13	Mohammad	X
14	Nurkhalis	14	Nurkhalis	O
15	Abdul Ghofur	15	Abdul Ghofur	O

Manual		Aplikasi		Cek
No	Nama	No	Nama	
16	Muslichah	16	Muslichah	O
17	Imam Sofwan	17	Imam Sofwan	O
18	Kustiani	18	Kustiani	O
19	Zaenah	19	Zaenah	O
20	Ahmad Arif	20	Ahmad Arif	O
21	Suhartono	21	Suhartono	O
22	Moh. Cholil Irsyad	22	Moh. Cholil Irsyad	O
23	Ulafaur Rosidin	23	Ulafaur Rosidin	O
24	Moh. Alimin	24	Moh. Alimin	O
25	Maulu'ah	25	Maulu'ah	O
26	Moh. Kholil	26	Moh. Kholil	O
27	Syaiful Islam	27	Syaiful Islam	O
28	Muniri	28	Moh. Kholil	X
29	Moh. Kholil	29	Muniri	X
30	Badriyah Andarini	30	Mustakim	X
31	Mustakim	31	Badriyah Andarini	X
32	M. Buchri	32	M. Buchri	O
33	Moh. Ridwan	33	Moh. Ridwan	O
34	Siti Qomariyah	34	Siti Qomariyah	O
35	Jumiati	35	Abdul Mujib	X
36	Abdul Mujib	36	Jumiati	X
37	M.Syafii	37	M.Syafii	O
38	Khoiri	38	Chosniyeh	X
39	Khoiri	39	Khoiri	O
40	Chosniyeh	40	Khoiri	X
41	Dardiri	41	Dardiri	O
42	Farkhan	42	Farkhan	O
43	Moh. Murodhi	43	Moh. Murodhi	O
44	Chlilah	44	Chlilah	O
45	Wiwik Alfiah	45	Wiwik Alfiah	O
46	Umi Kulsum	46	Qosim	X
47	Qosim	47	Umi Kulsum	X
48	Muhajir	48	Chudori	X
49	Chudori	49	Muhajir	X
50	Faridah	50	Faridah	O

Manual		Aplikasi		Cek
No	Nama	No	Nama	
51	Zuhri	51	Mustofa	X
52	Mustofa	52	Zuhri	X
53	Muh. Yasin	53	Moh. Machi M.Nur	X
54	Moh. Machi M.Nur	54	Nuril Huda	X
55	Muhyidin	55	Muhyidin	O
56	Nuril Huda	56	Muh. Yasin	X
57	Sulistyowati	57	Sulistyowati	O
58	Kusen	58	Kusen	O
59	Sayuti	59	Syaifurrofi'i	X
60	Syaifurrofi'i	60	Sayuti	X
61	Moh. Junaedi	61	Moh. Junaedi	O
62	Mujiati	62	Mujiati	O
63	Badriyah	63	Badriyah	O
64	Khotijah	64	Khotijah	O
65	Kholil Multazam	65	Kholil Multazam	O
66	Muslichah	66	Muslichah	O
67	Sholihin	67	Sholihin	O
68	Mashudi	68	Mashudi	O
69	Khoirudin	69	Khoirudin	O
70	Sumiarsih	70	Istiqomah	X
71	Siti Aisyah	71	Sumiarsih	X
72	Istiqomah	72	Siti Aisyah	X
73	Khoiriyah	73	Khoiriyah	O
74	Edy Sucipto	74	Edy Sucipto	O
75	Umi Zahriyah	75	Umi Zahriyah	O
76	Mujadi	76	Mujadi	O
77	Istibsyaroh	77	Istibsyaroh	O
78	Tri Susilowati	78	Tri Susilowati	O
79	Choiriyah	79	Choiriyah	O
80	Taufiq	80	Taufiq	O
81	M. Syahid	81	M. Syahid	O
82	Sutrisno	82	Sutrisno	O
83	Hadiyati	83	Syamsul Hadi	X
84	Syamsul Hadi	84	Hadiyati	X
85	Moh. Salam	85	Moh. Salam	O

Manual		Aplikasi		Cek
No	Nama	No	Nama	
86	Siti Aminah	86	Siti Aminah	O
87	Suaeib	87	Suaeib	O
88	Fathonah	88	Fathonah	O
89	Supaat	89	Abdul Rosyid	X
90	Abdul Rosyid	90	Supaat	X
91	Ahmad Bakri	91	Ahmad Bakri	O
92	Masuda	92	Masuda	O
93	Abdul Halim	93	Syamsul Hadi	X
94	Syamsul Hadi	94	Abdul Halim	X
95	Mamik Idah Wahyuni	95	Mamik Idah Wahyuni	O
96	Ali Musyafa	96	Ali Musyafa	O
97	Mansur	97	Mansur	O
98	Arifin	98	Arifin	O
99	Mas'ud	99	Mas'ud	O
100	Khoiriyah	100	Khoiriyah	O

Hal ini disebabkan hasil perangkingan manual tidak lepas dari pengaruh faktor – faktor luar yang mempengaruhi subjektifitas pengambil keputusan dalam hal ini adalah pihak dinas pendidikan propinsi dan kabupaten/kota.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang dapat diambil pada skripsi ini adalah:

1. Algoritma Genetika dapat digunakan sebagai metode untuk mencari bobot MADM pada penentuan peserta serifikasi guru. Dalam menerapkan algoritma genetika sebagai metode untuk mencari bobot MADM pada penentuan peserta sertifikasi guru, dilakukan dalam beberapa tahapan yaitu : proses persiapan data, penghitungan bobot atribut dengan algoritma genetika, perangkingan dan ujicoba. Hasilnya adalah berupa urutan data alternatif peserta sertifikasi guru berdasarkan nilai bobot tersebut
2. Pemakaian peluang *crossover* dan peluang mutasi yang berbeda, menimbulkan perubahan terhadap nilai *fitness*. Peningkatan nilai peluang crossover menyebabkan peningkatan nilai fitness sedangkan peningkatan nilai peluang mutasi menyebabkan penurunan nilai fitness. Dari hasil ujicoba diperoleh hasil pada peluang crossover 0.9 dan peluang mutasi 0.1 diperoleh nilai fitness yang optimal.
3. Pemakaian peluang *crossover* dan peluang mutasi yang berbeda, menimbulkan perubahan terhadap generasi konvergensi. Peningkatan nilai peluang crossover dan peluang mutasi menyebabkan nilai konvergensi semakin rendah atau diperoleh konvergensi pada generasi lebih awal.
4. Hasil perangkingan menggunakan aplikasi tidak sama dengan hasil perangkingan dengan cara manual. Dengan membandingkan kedua data tersebut didapatkan 34 kali posisi salah dan 66 posisi benar

5.2 Saran

Aplikasi yang dibangun dalam skripsi ini masih belum sempurna, berikut ini saran-saran untuk meningkatkan hasil yang telah dicapai:

1. Pada tahap seleksi, bisa digunakan metode seleksi yang lain contoh : *rank-based fitness*, *Stochastic Universal Sampling*, *tournament selection* dan lain-lain untuk membandingkan nilai fitness yang dihasilkan
2. Pada tahap crossover bisa digunakan *multipoint* untuk membandingkan nilai fitness yang dihasilkan



DAFTAR PUSTAKA

- Hartati, Sri dkk. 2006. *Fuzzy Multi-Attribute Decision Making(Fuzzy MADM)*. Yogyakarta : Graha Ilmu
- Kusumadewi, Sri dan Hari Purnomo.2005. *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik*. Yogyakarta : Graha Ilmu
- Fan, Zhiping Jiang Ma dan Peng Tian(1999).*A Subjective and Objective Integrated Approach for the Determination of Attribute Weights*. Kwooloon, Hongkong
- Sawaragi, Y., Inoue, K dan Nakayama. 1987. *Towards Interactive and Intelligent Decision Support Systems*. Springer-Verlag
- Kusumadewi, Sri.2004. *Pencarian Bobot Atribut pada Multiple Attribute Decision Making (MADM) dengan Pendekatan Subjektif Menggunakan Algoritma Genetika (Studi Kasus : Penentuan Lokasi Gudang)*. Seminar Nasional Pendidikan Teknik Elektro (SNPTE 2004)
- Kusumadewi, Sri. 2003.*Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta : Graha Ilmu
- Chu, ATW, Kalaba RE, dan Spingam K.1979.*AComparison of Two Methods for Determining The Weights of Belonging to Fuzzy Sets*. Journal of Optimazion Theory and Application.
- Zimmermann.1991.*Fuzzy Sets Theory and Its Applications*.Edisi 2. Kluwer Academic Publishers : Massachusetts.
- Hartati, Sri dkk. 2006. *FuzzyMulti-Attribute Decision Making(FuzzyMADM)*. Yogyakarta : Graha Ilmu

Ustinovichius, L., E.K. Zavadskas dan V. Podvezko.2007.

Application of a Quantitative MultipleCriteria Decision Making (MCDM-1) Approach to the Analysis of Investments in Construction. Vilnius,Lithuania.

Haupt, Randy L., Sue Ellen Haupt.2004.*Practical Genetic Algorithm(Second Edition).* New Jersey : Willey-Interscience.

Goldberg,D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley

Yoon,KP., dan Hwang C-L.1981.*Multiple Attribute Decision Making,Methods and Application, A State-of-the-Art Survey.*

Janko, Wolfgang.2005.*Multi-Criteria Decision Making : An Application Study of ELECTRE & TOPSIS.*

Rudolphi,Wictoria.2000.*Multi-CriteriaDecision Analysis as A Framework for Integrated Land Use Management in Canadian National Parks.*

Simon Fraser University Simoes-Marques, M., Ribeiro, R., dan Gamiero-Marques,A._*A Fuzzy Decision Support System for Equipment Repair Under Battle Conditions.*Fuzzy Sets and Systems ,115:141-157.

Yeh, Chung-Hsing.2002. *A Problem-based Selection of Multi-Attribute Decision Making Methods.* International transactions in Operational Research Blackwell Publishing.

http://publish.uwo.ca/jmalczew/gida_1/Tkach/Tkach.htm,
tanggal akses : 21 September 2010

Sanandaji,Houman.(2006).*A Study of Different Decision-Making Models and Their Pros and Cons.*Ottawa, Kanada

Bellman, R., Kalaba, R., and Zadeh, L.1966. *Abstraction and Pattern Classification*. Journal of Mathematical Analysis and Applications, Vol. 13, pp.1-7, 1966

Chu, ATW, Kalaba RE, dan Spingam K .1979. *A Comparison of Two Methods for Determining The Weights of Belonging to Fuzzy Sets*. Journal of Optimazion Theory and Application.

Saaty, T. L.1977. *A Scaling Method for Priorities in Hierarchical Structures*, Journal of Mathematical Psychology, Vol. 15, No. 3, 1977.

Saaty, T. L.1977. *Modeling Unstructured Decision Problems. The Theory of Analytical Hierarchies*, Proceedings of the First International Conference on Mathematical Modeling, Vol. 1, University of Missouri, Rolla, Missouri, 1977

Kalaba, R., and Spingarn, K____. *Numerical Approaches to the Eigenvalues of Saaty's Matrices for Fuzzy Sets*, Computers and Mathematics with Applications (to appear).

Saaty, Thomas L.2008. *Decision making with the analytic hierarchy process (Int. J. Services Sciences, Vol. 1, No. 1, 2008)*.

Kementrian Pendidikan Nasional. *Pedoman Penetapan Peserta dna Pelaksanaan Sertifikasi Guru dalam Jabatan Cetakan Kedua*.2007. Direktorat jenderal peningkatan mutu pendidikan dan tenaga kependidikan Kementrian Pendidikan Nasional.

Kementrian Pendidikan Nasioanl. *Sertifikasi Guru dalam Jabatan Tahun 2010 Buku I Pedoman Penetapan Peserta*.2010. Direktorat jenderal peningkatan mutu pendidikan dan tenaga kependidikan. Kementrian Pendidikan Nasional.

LAMPIRAN 1

Data Peserta Sertifikasi Guru Kabupaten Malang tahun 2010

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangk	Beban kerja	Skor Tugas tamba	Skor Prestas i kerja
1	Sri Gunawati	190	190	5	24	23	60
2	Nurkhalis	175	175	5	37	4	15
3	Mustakim	145	145	4	38	10	50
4	Mamik Idah Wahyuni	40	85	1	36	0	0
5	Masrukin	175	190	5	27	16	90
6	Dardiri	145	130	3	36	2	20
7	Mufrihul Anam	190	190	5	29	10	10
8	Abdul Halim	55	85	2	29	0	0
9	Misbahul Munir	175	190	5	40	0	60
10	Afandi	175	190	5	36	4	40
11	Kusen	130	100	4	26	2	0
12	Edy Sucipto	85	85	2	40	0	0
13	Kustiani	175	160	5	37	4	90
14	Ali Musyafa	70	85	1	24	0	10
15	Farkhan	115	145	4	27	2	20
16	Hasan Bisri Mustofa	160	190	5	39	4	70
17	Badriyah	100	100	2	39	2	70
18	Moh. Kholil	160	145	5	35	3	0
19	Khotijah	115	100	2	35	1	30
20	Umi Kulsum	130	130	3	26	2	90
21	Arifin	40	85	1	29	0	20
22	Moh. Ridwan	175	145	4	33	1	60
23	Moh. Cholil Irsyad	160	175	5	26	0	70
24	M.Syafii	145	145	4	32	4	60
25	Khoiriyah	40	85	1	25	0	10
26	Sukirno	190	190	5	33	16	10
27	Sulistyowati	100	115	2	39	2	80
28	Badriyah Andarini	175	160	4	26	4	30
29	Sutrisno	85	85	1	39	0	0

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangka	Beban kerja	Skor Tugas tambah	Skor Prestas i kerja
30	Muslichah	100	100	2	38	1	10
31	Nuril Huda	130	115	2	36	2	5
32	Moh. Murodhi	160	130	3	25	4	10
33	Muhajir	175	115	3	24	1	20
34	Supaat	55	85	2	32	0	10
35	Faridah	115	130	3	26	5	70
36	Sholihin	130	100	2	30	2	0
37	Chlilah	160	115	3	32	1	0
38	M. Buchri	190	145	4	30	4	20
39	Ali Riwayat	175	190	5	34	18	60
40	Kholil Multazam	100	100	2	40	0	0
41	Maulu'ah	145	160	4	40	14	0
42	Mansur	40	85	1	30	0	0
43	Taufiq	100	85	1	31	2	20
44	Abdul Mujib	160	130	4	39	7	80
45	Zaenah	145	175	5	30	9	120
46	Muhyidin	115	130	3	24	4	0
47	Moh. Salam	55	85	2	34	2	40
48	M. Syahid	85	85	1	39	1	0
49	Abdul Rosyid	85	85	1	32	0	10
50	Moh. Junaedi	70	115	3	34	3	60
51	Muh. Yasin	85	130	4	24	2	30
52	Mas'ud	40	85	1	26	0	60
53	Rosyidi	190	160	5	40	4	90
54	Masuda	40	85	2	33	0	20
55	Qosim	145	115	3	32	3	30
56	Chosniyeh	175	130	3	32	4	0
57	Mohammad	175	190	5	31	5	10
58	Mashudi	70	115	2	33	3	15
59	Istibsyaroh	85	100	2	28	2	20
60	Sayuti	85	130	3	26	2	20
61	Hadiyati	55	100	2	28	4	0
62	Suhartono	145	160	5	38	1	90

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambahan	Skor Prestasi kerja
63	Imam Sofwan	190	175	5	30	2	30
64	Ulafaur Rosidin	160	175	5	25	4	50
65	Khoiri	115	145	4	30	1	80
66	Fathonah	55	85	2	31	2	40
67	Tri Susilowati	70	100	2	32	1	20
68	Syaifurrofi'i	130	115	2	30	3	20
69	Sumiarsih	55	100	2	38	0	90
70	Mujadi	70	100	2	30	0	90
71	Istiqomah	115	85	1	40	1	30
72	Wiwik Alfiah	145	115	3	35	2	0
73	Khoirudin	100	85	2	34	4	90
74	Jumiati	145	160	4	30	4	20
75	Mustofa	100	115	3	37	1	90
76	Choiriyah	55	100	2	34	0	30
77	Moh. Machi M.Nur	145	115	2	31	1	70
78	Umi Zahriyah	55	100	2	35	1	70
79	Siti Aminah	40	85	2	38	1	0
80	Ahmad Arif	130	175	5	33	10	40
81	Mujiati	70	115	3	33	3	80
82	Muniri	130	160	5	29	15	10
83	Suaeib	85	85	2	24	1	80
84	Syamsul Hadi	100	85	1	33	1	20
85	Mukhlis	175	190	5	40	20	70
86	M. Zaenal	100	145	3	27	3	40
87	Muh. Kholil Hasan	130	175	5	32	4	0
88	Zuhri	115	130	3	26	2	70
89	Siti Qomariyah	175	145	4	32	3	15
90	Maslichah	160	190	5	27	10	20
91	Khoiriyah	70	100	2	28	2	190
92	Syaiful Islam	145	160	4	40	4	10
93	Ahmad Bakri	55	85	1	36	2	20
94	Samsudi Jaly	190	190	5	35	4	70
95	Badruz Zaman	190	190	5	33	2	20

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambahan	Skor Prestasi kerja
96	Siti Aisyah	55	100	2	39	4	0
97	Chudori	115	115	3	38	1	40
98	Abdul Ghofur	160	190	5	29	4	50
99	Syamsul Huda	55	85	1	36	1	0
100	Moh. Alimin	190	160	5	27	4	0

LAMPIRAN 2

Data Urutan Peserta Sertifikasi Guru Kabupaten Malang tahun 2010 (Manual)

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambahan	Skor Prestasi kerja
1	Mukhlis	190	175	5	40	20	70
2	Ali Riwayat	190	175	5	34	18	60
3	Sukirno	190	190	5	33	16	10
4	Samsudi Jaly	190	190	5	35	4	70
5	Misbahul Munir	175	190	5	40	0	60
6	Sri Gunawati	190	190	5	24	23	60
7	Hasan Bisri	160	190	5	39	4	70
8	Afandi	175	190	5	36	4	40
9	Badruz Zaman	190	190	5	33	2	20
10	Masrukin	175	190	5	27	16	90
11	Mufrihul Anam	190	190	5	29	10	10
12	Mohammad	175	190	5	31	5	10
13	Rosyidi	190	160	5	40	4	90
14	Nurkhalis	175	175	5	37	4	15
15	Abdul Ghofur	160	190	5	29	4	50
16	Muslichah	100	100	2	38	1	10
17	Imam Sofwan	190	175	5	30	2	30
18	Kustiani	175	160	5	37	4	90

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambahan	Skor Prestasi kerja
19	Zaenah	145	175	5	30	9	120
20	Ahmad Arif	130	175	5	33	10	40
21	Suhartono	145	160	5	38	1	90
22	Moh. Cholil Irsyad	160	175	5	26	0	70
23	Ulafaur Rosidin	160	175	5	25	4	50
24	Moh. Alimin	190	160	5	27	4	0
25	Maulu'ah	145	160	4	40	14	0
26	Moh. Kholil	160	145	5	35	3	0
27	Syaiful Islam	145	160	4	40	4	10
28	Muniri	130	160	5	29	15	10
29	Moh. Kholil	160	145	5	35	3	0
30	Badriyah Andarini	175	160	4	26	4	30
31	Mustakim	145	145	4	38	10	50
32	M. Buchri	190	145	4	30	4	20
33	Moh. Ridwan	175	145	4	33	1	60
34	Siti Qomariyah	175	145	4	32	3	15
35	Jumiati	145	160	4	30	4	20
36	Abdul Mujib	160	130	4	39	7	80
37	M.Syafii	145	145	4	32	4	60
38	Khoiri	115	145	4	30	1	80
39	Khoiri	115	145	4	30	1	
40	Chosniyeh	175	130	3	32	4	0
41	Dardiri	145	130	3	36	2	20
42	Farkhan	115	145	4	27	2	20
43	Moh. Murodhi	160	130	3	25	4	10
44	Chlilah	160	115	3	32	1	0
45	Wiwik Alfiah	145	115	3	35	2	0
46	Umi Kulsum	130	130	3	26	2	90
47	Qosim	145	115	3	32	3	30
48	Muhajir	175	115	3	24	1	20
49	Chudori	115	115	3	38	1	40
50	Faridah	115	130	3	26	5	70
51	Zuhri	115	130	3	26	2	70

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambahan	Skor Prestasi kerja
52	Mustofa	100	115	3	37	1	90
53	Muh. Yasin	85	130	4	24	2	30
54	Moh. Machi M.Nur	145	115	2	31	1	70
55	Muhyidin	115	130	3	24	4	0
56	Nuril Huda	130	115	2	36	2	5
57	Sulistiyowati	100	115	2	39	2	80
58	Kusen	130	100	4	26	2	0
59	Sayuti	85	130	3	26	2	20
60	Syaifurrofi'i	130	115	2	30	3	20
61	Moh. Junaedi	70	115	3	34	3	50
62	Mujiati	70	115	3	33	3	80
63	Badriyah	100	100	2	39	2	70
64	Khotijah	115	100	2	35	1	30
65	Kholil Multazam	100	100	2	40	0	0
66	Muslichah	100	100	2	38	1	10
67	Sholihin	130	100	2	30	2	0
68	Mashudi	70	115	2	33	3	15
69	Khoirudin	100	85	2	34	4	90
70	Sumiarsih	55	100	2	38	0	90
71	Siti Aisyah	55	100	2	39	4	0
72	Istiqomah	115	85	1	40	1	30
73	Khoiriyah	70	100	2	28	2	190
74	Edy Sucipto	85	85	2	40	0	0
75	Umi Zahriyah	55	100	2	35	1	70
76	Mujadi	70	100	2	30	0	90
77	Istibsyaroh	85	100	2	28	2	20
78	Tri Susilowati	70	100	2	32	1	20
79	Choiriyah	55	100	2	34	0	30
80	Taufiq	100	85	1	31	2	20
81	M. Syahid	85	85	1	39	1	0
82	Sutrisno	85	85	1	39	0	0
83	Hadiyati	55	100	2	28	4	0
84	Syamsul Hadi	100	85	1	33	1	20

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambahan	Skor Prestasi kerja
85	Moh. Salam	55	85	2	34	2	40
86	Siti Aminah	40	85	2	38	1	0
87	Suaeib	85	85	2	24	1	80
88	Fathonah	55	85	2	31	2	40
89	Supaat	55	85	2	32	0	10
90	Abdul Rosyid	85	85	1	32	0	10
91	Ahmad Bakri	55	85	1	36	2	20
92	Masuda	40	85	2	33	0	20
93	Abdul Halim	55	85	2	29	0	0
94	Syamsul Hadi	100	85	1	33	1	20
95	Mamik Idah Wahyuni	40	85	1	36	0	0
96	Ali Musyafa	70	85	1	24	0	10
97	Mansur	40	85	1	30	0	0
98	Arifin	40	85	1	29	0	20
99	Mas'ud	40	85	1	26	0	60
100	Khoiriyah	70	100	2	28	2	190

LAMPIRAN 3

Data Urutan Peserta Sertifikasi Guru Kabupaten Malang tahun 2010 (Aplikasi)

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambahan	Skor Prestasi kerja
1	Mukhlis	190	175	5	40	20	70
2	Ali Riwayat	190	175	5	34	18	60
3	Sukirno	190	190	5	33	16	10
4	Samsudi Jaly	190	190	5	35	4	70
5	Misbahul Munir	175	190	5	40	0	60
6	Hasan Bisri	160	190	5	39	4	70

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambahan	Skor Prestasi kerja
7	Sri Gunawati	190	190	5	24	23	60
8	Afandi	175	190	5	36	4	40
9	Badruz Zaman	190	190	5	33	2	20
10	Mufrihul Anam	190	190	5	29	10	10
11	Masrukin	175	190	5	27	16	90
12	Rosyidi	190	160	5	40	4	90
13	Mohammad	175	190	5	31	5	10
14	Nurkhalis	175	175	5	37	4	15
15	Abdul Ghofur	160	190	5	29	4	50
16	Muslichah	100	100	2	38	1	10
17	Imam Sofwan	190	175	5	30	2	30
18	Kustiani	175	160	5	37	4	90
19	Zaenah	145	175	5	30	9	120
20	Ahmad Arif	130	175	5	33	10	40
21	Suhartono	145	160	5	38	1	90
22	Moh. Cholil Irsyad	160	175	5	26	0	70
23	Ula faur Rosidin	160	175	5	25	4	50
24	Moh. Alimin	190	160	5	27	4	0
25	Maulu'ah	145	160	4	40	14	0
26	Moh. Kholil	160	145	5	35	3	0
27	Syaiful Islam	145	160	4	40	4	10
28	Moh. Kholil	160	145	5	35	3	0
29	Muniri	130	160	5	29	15	10
30	Mustakim	145	145	4	38	10	50
31	Badriyah Andarini	175	160	4	26	4	30
32	M. Buchri	190	145	4	30	4	20
33	Moh. Ridwan	175	145	4	33	1	60
34	Siti Qomariyah	175	145	4	32	3	15
35	Abdul Mujib	160	130	4	39	7	80
36	Jumiati	145	160	4	30	4	20
37	M.Syafii	145	145	4	32	4	60
38	Chosniyeh	175	130	3	32	4	0
39	Khoiri	115	145	4	30	1	80

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambahan	Skor Prestasi kerja
40	Khoiri	115	145	4	30	1	80
41	Dardiri	145	130	3	36	2	20
42	Farkhan	115	145	4	27	2	20
43	Moh. Murodhi	160	130	3	25	4	10
44	Chlilah	160	115	3	32	1	0
45	Wiwik Alfiah	145	115	3	35	2	0
46	Qosim	145	115	3	32	3	30
47	Umi Kulsum	130	130	3	26	2	90
48	Chudori	115	115	3	38	1	40
49	Muhajir	175	115	3	24	1	20
50	Faridah	115	130	3	26	5	70
51	Mustofa	100	115	3	37	1	90
52	Zuhri	115	130	3	26	2	70
53	Moh. Machi M.Nur	145	115	2	31	1	70
54	Nuril Huda	130	115	2	36	2	5
55	Muhyidin	115	130	3	24	4	0
56	Muh. Yasin	85	130	4	24	2	30
57	Sulistiyowati	100	115	2	39	2	80
58	Kusen	130	100	4	26	2	0
59	Syaifurrofi'i	130	115	2	30	3	20
60	Sayuti	85	130	3	26	2	20
61	Moh. Junaedi	70	115	3	34	3	50
62	Mujiati	70	115	3	33	3	80
63	Badriyah	100	100	2	39	2	70
64	Khotijah	115	100	2	35	1	30
65	Kholil Multazam	100	100	2	40	0	0
66	Muslichah	100	100	2	38	1	10
67	Sholihin	130	100	2	30	2	0
68	Mashudi	70	115	2	33	3	15
69	Khoirudin	100	85	2	34	4	90
70	Istiqomah	115	85	1	40	1	30
71	Sumiarsih	55	100	2	38	0	90
72	Siti Aisyah	55	100	2	39	4	0

No	Nama	Skor Usia	Skor Masa Kerja	Skor Pangkat	Beban kerja	Skor Tugas tambaha	Skor Prestasi kerja
73	Khoiriyah	70	100	2	28	2	190
74	Edy Sucipto	85	85	2	40	0	0
75	Umi Zahriyah	55	100	2	35	1	70
76	Mujadi	70	100	2	30	0	90
77	Istibsyaroh	85	100	2	28	2	20
78	Tri Susilowati	70	100	2	32	1	20
79	Choiriyah	55	100	2	34	0	30
80	Taufiq	100	85	1	31	2	20
81	M. Syahid	85	85	1	39	1	0
82	Sutrisno	85	85	1	39	0	0
83	Syamsul Hadi	100	85	1	33	1	20
84	Hadiyati	55	100	2	28	4	0
85	Moh. Salam	55	85	2	34	2	40
86	Siti Aminah	40	85	2	38	1	0
87	Suaeib	85	85	2	24	1	80
88	Fathonah	55	85	2	31	2	40
89	Abdul Rosyid	85	85	1	32	0	10
90	Supaat	55	85	2	32	0	10
91	Ahmad Bakri	55	85	1	36	2	20
92	Masuda	40	85	2	33	0	20
93	Syamsul Hadi	100	85	1	33	1	20
93	Abdul Halim	55	85	2	29	0	0
95	Mamik Idah Wahyuni	40	85	1	36	0	0
96	Ali Musyafa	70	85	1	24	0	10
97	Mansur	40	85	1	30	0	0
98	Arifin	40	85	1	29	0	20
99	Mas'ud	40	85	1	26	0	60
100	Khoiriyah	70	100	2	28	2	190