

**IMPLEMENTASI ALGORITMA *PARTICLE SWARM*
OPTIMIZATION (PSO) UNTUK OPTIMASI PEMERATAAN
GURU MATA PELAJARAN DI KABUPATEN LUMAJANG**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Risda Amalia Khusna
NIM: 125150101111008



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

IMPLEMENTASI ALGORITMA *PARTICLE SWARM OPTIMIZATION* (PSO) UNTUK
OPTIMASI PEMERATAAN GURU MATA PELAJARAN DI KABUPATEN LUMAJANG

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Risda Amalia Khusna

NIM: 125150101111008

Skripsi ini telah diuji dan dinyatakan lulus pada
15 Agustus 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Imam Cholissodin, S.Si, M.Kom

NIK: 201201 850719 1 001

Randy Cahya W, S.ST, M.Kom

NIK: 201405 880206 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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

Malang, 15 Agustus 2016

Risda Amalia Khusna

NIM: 125150101111008

KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Implementasi *Algoritma Particle Swarm Optimization* (PSO) Untuk Pemerataan Guru Mata Pelajaran di Kabupaten Lumajang” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Imam Cholissodin, S.Si., M.Kom. selaku dosen pembimbing I dan Randy Cahya W., S.ST., M.Kom. selaku dosen pembimbing II yang telah memberikan bimbingan, ilmu dan saran dalam penyusunan tugas akhir ini.
2. Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, selaku Dekan, Ir. Heru Nurwasito M.Kom selaku Wakil Dekan 1, Drs. Marji, M.T selaku Wakil Dekan 2 dan Edy Santoso, S.Si, M.Kom selaku Wakil Dekan 3 Fakultas Ilmu Komputer Universitas Brawijaya.
3. Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
4. Agus Wahyu Widodo, S.T, M.Cs. selaku Ketua Program Studi Teknik Informatika.
5. Novanto Yudistira, S.Kom., M.Sc. selaku dosen penasehat akademik yang selalu memberikan nasehat kepada penulis selama menempuh masa studi.
6. Totok Widodo, SE. selaku staf subbag Sekolah Menengah (SM) Dinas Pendidikan Kabupaten Lumajang dan Kardi selaku staf Subbag Mutasi Badan Kepegawaian Daerah Kabupaten Lumajang yang telah meluangkan waktunya untuk membantu proses penelitian ini.
7. Orang tua dan para sahabat yang telah memberikan motivasi, kasih sayang serta dukungan moril dan materil demi terselesaikannya skripsi ini.
8. Seluruh civitas akademik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
9. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 08 Agustus 2016

Penulis
risda.a.husna@gmail.com

ABSTRAK

Persebaran guru yang tidak merata merupakan masalah mendasar yang masih menjadi persoalan mulai tingkat pusat hingga daerah. Persebaran guru yang tidak merata ditandai dengan adanya sekolah yang kekurangan guru dan sekolah kelebihan yang guru. Persolaan tersebut dapat diatasi dengan melakukan penataan dan pemerataan guru berupa mutasi. Mutasi adalah memindah tugaskan guru dari suatu satuan pendidikan ke satuan pendidikan lain. Mutasi harus dilakukan dengan kriteria yang jelas agar penataan dan pemerataan guru dapat dilakukan secara tepat sasaran. Penataan dan pemerataan yang dilakukan secara tepat sasaran akan menjamin guru dapat bekerja lebih baik.

Berdasarkan hal tersebut, pada penelitian ini dibuat suatu sistem yang mampu mengoptimasi penataan dan pemerataan guru secara otomatis. Optimasi bertujuan untuk menghasilkan persebaran guru yang lebih proporsional dan lebih tepat sasaran serta sesuai dengan peraturan yang telah ditentukan.

Teknik optimasi yang digunakan adalah *Particle Swarm Optimization* (PSO), yaitu teknik optimasi berbasis populasi yang terinspirasi oleh perilaku sosial gerakan kawanan hewan seperti burung atau ikan. PSO menggabungkan metode pencarian lokal dan metode pencarian global, masing-masing partikel akan bergerak mengitari ruang pencarian dan menyesuaikan posisinya terhadap pengalaman pribadi maupun pengalaman partikel lain di sekitarnya.

Dari hasil pengujian didapatkan parameter-parameter PSO yang paling optimal untuk iterasi maksimum=70, ukuran populasi=25, $\theta_{max} = 0.9$, $\theta_{min} = 0.4$, $c_{1i} = 2.5$, $c_{1f} = 0.5$, $c_{2i} = 0.5$, dan $c_{2f} = 2.5$. Optimasi dengan menggunakan nilai parameter optimal PSO dapat meningkatkan nilai *fitness* sebesar 8-12%.

Kata kunci: Penataan dan Pemerataan Guru, Kelebihan dan Kekurangan Guru, Optimasi, Algoritma *Particle Swarm Optimization*.

ABSTRACT

Maldistribution of teachers is a fundamental problem that is still become an issue from central to the regional level. This maldistribution is characterized by a teacher shortages in some schools, but overages in other. These issues can be overcome by making the arrangement and equal distribution of teachers, called as displacement. Displacement are redeployed teachers from school to another school. This removal must be done with clear criteria, so that the arrangement and equal distribution of teachers can be done right on target. Structuring and equalization which done on the right target will ensure teachers can work better.

According to these, this study will created a system that is able to provide structuring and optimization of equal distribution of teachers automatically. Optimization aims to produce a more proportional distribution of teachers and better right on target and in accordance with predetermined rules.

The optimization technique used is Particle Swarm Optimization (PSO), it is a population-based optimization techniques inspired by the social behavior of animals flock movements like a bird or a fish. PSO combines methods of local search and global search methods, each particle will move around the search space and adjust the position of the personal experiences nor experiences of other particles in the surrounding.

The test results obtained the most optimal parameters of PSO, that is maximum iteration = 70, population size=25, $\theta_{max} = 0.9$, $\theta_{min} = 0.4$, $c_{1i} = 2.5$, $c_{1f} = 0.5$, $c_{2i} = 0.5$, and $c_{2f} = 2.5$. Optimization using PSO optimal parameter can improve the fitness by 8-12%.

Keywords: *Arrangement and Equal Distribution of Teachers, Teacher Shortages and Overages, Optimization, Particle Swarm Optimization Algorithm.*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR	iv
ABSTRAK	v
ABSTRACT	vi
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
DAFTAR PERSAMAAN	xii
DAFTAR <i>SOURCE CODE</i>	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	2
1.3 Tujuan	2
1.4 Manfaat	3
1.5 Batasan masalah	3
1.6 Sistematika Penulisan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Guru	7
2.3 Mutasi	8
2.4 Mutasi Guru PNS	8
2.4.1 Mutasi Guru PNS ke Satuan Pendidikan Lain	8
2.4.2 Mutasi Guru PNS ke Kabupaten/Kota Lain	9
2.4.3 Kriteria Guru PNS Dalam Mutasi	9
2.4.4 Waktu Pemindahan	10
2.5 Penataan dan Pemerataan Guru	10
2.6 Optimalisasi Pemenuhan Kebutuhan Guru Mata Pelajaran	12
2.7 Perhitungan Kebutuhan Guru SMP	13
2.8 Optimasi	15
2.9 <i>Particle Swarm Optimization</i> (PSO)	15

2.9.2 Inisialisasi Partikel	17
2.9.3 Bobot Inersia	17
2.9.4 Pembatasan Kecepatan.....	18
2.9.5 Kondisi Berhenti	19
2.9.6 <i>Time Varying Acceleration Coefficients</i> (TVAC).....	19
2.10 Nilai Evaluasi	19
BAB 3 METODOLOGI	21
3.1 Studi Literatur	22
3.2 Analisa Kebutuhan Sistem	22
3.3 Pengumpulan Data.....	22
3.4 Perancangan Sistem.....	22
3.5 Implementasi Sistem.....	22
3.6 Pengujian Sistem.....	23
3.7 Kesimpulan dan Saran.....	23
BAB 4 PERANCANGAN.....	24
4.1 Deskripsi Sistem.....	24
4.2 Alir Perancangan Sistem.....	25
4.2.1 Proses PSO.....	25
4.2.2 Proses Inisialisasi Partikel.....	25
4.2.3 Proses Pemilihan Posisi Secara Acak.....	27
4.2.4 Proses Perhitungan <i>Fitness</i> Awal	28
4.2.5 Proses Penentuan G_{best} Awal.....	29
4.2.6 Proses <i>Update</i> Kecepatan	30
4.2.7 Proses <i>Update</i> Posisi.....	32
4.2.8 Proses Perhitungan <i>Fitness</i>	34
4.2.9 Proses <i>Update</i> P_{best}	35
4.2.10 Proses <i>Update</i> G_{best}	36
4.3 Manualisasi	37
4.4 Perancangan Antarmuka.....	51
4.4.1 Antarmuka Halaman Data Guru dan Sekolah	51
4.4.2 Antarmuka Halaman Data Kekurangan dan Kelebihan.....	51
4.4.3 Antarmuka Halaman Pengujian	51

4.4.4 Antarmuka Halaman Detail Perhitungan	51
BAB 5 IMPLEMENTASI	56
5.1 Implementasi Program.....	56
5.1.1 Inisialisasi Partikel	56
5.1.2 <i>Update</i> Kecepatan	58
5.1.3 <i>Update</i> Posisi.....	59
5.1.4 Perhitungan <i>Fitness</i>	60
5.1.5 <i>Update</i> P_{best}	62
5.1.6 <i>Update</i> G_{best}	62
5.2 Implementasi Antarmuka	63
5.2.1 Halaman Data Sekolah dan Guru	63
5.2.2 Halaman Kelebihan dan Kekurangan	63
5.2.3 Halaman Pengujian	65
5.2.4 Halaman Detail Perhitungan	65
BAB 6 PENGUJIAN DAN ANALISIS.....	67
6.1 Hasil dan Analisis Uji Coba Iterasi Maksimum	67
6.2 Hasil dan Analisis Uji Coba Ukuran Populasi.....	69
6.3 Hasil dan Analisis Uji Coba Kombinasi Bobot Inersia Minimum dan Maksimum	71
6.4 Hasil dan Analisis Uji Coba Kombinasi Koefisien Akselerasi	73
BAB 7 PENUTUP.....	76
7.1 Kesimpulan	76
7.2 Saran.....	76
DAFTAR PUSTAKA	78
LAMPIRAN 1	80
LAMPIRAN 2	81
LAMPIRAN 3	86
LAMPIRAN 4	87
LAMPIRAN 5	88
LAMPIRAN 6	89

DAFTAR TABEL

Tabel 2.2 Kajian Pustaka.....	6
Tabel 4.1 Formasi Kebutuhan Guru	38
Tabel 4.2 Pilihan Guru Mata Pelajaran IPS.....	39
Tabel 4.3 Banyak Pilihan Mutasi Pada Mata Pelajaran IPS	39
Tabel 4.4 Hasil Inisialisasi Partikel.....	40
Tabel 4.5 Hasil Perhitungan <i>Fitness</i> dan Total <i>Fitness</i> Pada Iterasi 0.....	41
Tabel 4.6 P_{best} dan G_{best} Iterasi 0.....	43
Tabel 4.7 Hasil <i>Update</i> Kecepatan Iterasi ke-1	44
Tabel 4.8 Hasil <i>Update</i> Kecepatan Iterasi ke-1 Setelah Pembatasan Kecepatan	45
Tabel 4.9 Hasil <i>Update</i> Posisi Iterasi ke-1	46
Tabel 4.10 Hasil <i>Update</i> Posisi Iterasi ke-1 Setelah Pembatasan Posisi.....	47
Tabel 4.11 Hasil Perhitungan <i>Fitness</i> dan Total <i>Fitness</i> Pada Iterasi 1.....	49
Tabel 4.12 Perbandingan Total <i>Fitness</i> Setiap Partikel di Iterasi ke-0 dan Iterasi ke-1.....	50
Tabel 4.13 P_{best} dan G_{best} Iterasi ke-1.....	50
Tabel 6.1 Hasil Uji Coba Iterasi Maksimum.....	67
Tabel 6.2 Hasil Uji Coba Ukuran Populasi	69
Tabel 6.3 Kombinasi Nilai Bobot Inersia Minimum dan Maksimum.....	71
Tabel 6.4 Hasil Uji Coba Kombinasi Nilai Bobot Inersia Minimum dan Maksimu m	71
Tabel 6.5 Kombinasi Nilai Koefisien Akselerasi.....	73
Tabel 6.6 Hasil Uji Coba Kombinasi Koefisien Akselerasi.....	74

DAFTAR GAMBAR

Gambar 3.1 Metode Penelitian.....	21
Gambar 3.2 Arsitektur Perancangan Sistem.....	23
Gambar 4.1 Diagram Alir Perancangan Sistem.....	25
Gambar 4.2 Diagram Alir Proses PSO.....	26
Gambar 4.3 Proses Inisialisasi Partikel.....	27
Gambar 4.4 Diagram Alir Proses Pemilihan Posisi Secara Acak.....	28
Gambar 4.5 Diagram Alir Proses Perhitungan <i>Fitness</i> Awal.....	29
Gambar 4.6 Diagram Alir Proses Penentuan G_{best} Awal.....	30
Gambar 4.7 Diagram Alir Proses Update Kecepatan.....	32
Gambar 4.8 Diagram Alir Proses <i>Update</i> Posisi.....	34
Gambar 4.9 Diagram Alir Proses Perhitungan <i>Fitness</i>	35
Gambar 4.10 Diagram Alir Proses <i>Update</i> P_{best}	36
Gambar 4.11 Diagram Alir Proses <i>Update</i> G_{best}	37
Gambar 4.12 Antarmuka Halaman Data Guru dan Sekolah.....	52
Gambar 4.13 Antarmuka Halaman Kekurangan dan Kelebihan.....	53
Gambar 4.14 Antarmuka Halaman Pengujian.....	54
Gambar 4.15 Antarmuka Halaman Detail Perhitungan.....	55
Gambar 5.1 Antarmuka Halaman Data Guru dan Data Sekolah.....	64
Gambar 5.2 Antarmuka Halaman Data Kekurangan dan Kelebihan.....	64
Gambar 5.3 Antarmuka Halaman Pengujian.....	65
Gambar 5.4 Antarmuka Halaman Detail Perhitungan.....	66
Gambar 6.1 Grafik Hasil Uji Coba Iterasi Maksimum.....	68
Gambar 6.2 Grafik Hasil Uji Coba Ukuran Populasi.....	70
Gambar 6.3 Grafik Hasil Uji Coba Kombinasi Nilai Bobot Inersia Minimum dan Maksimum.....	72
Gambar 6.4 Grafik Hasil Uji Coba Kombinasi Koefisien Akselerasi.....	74

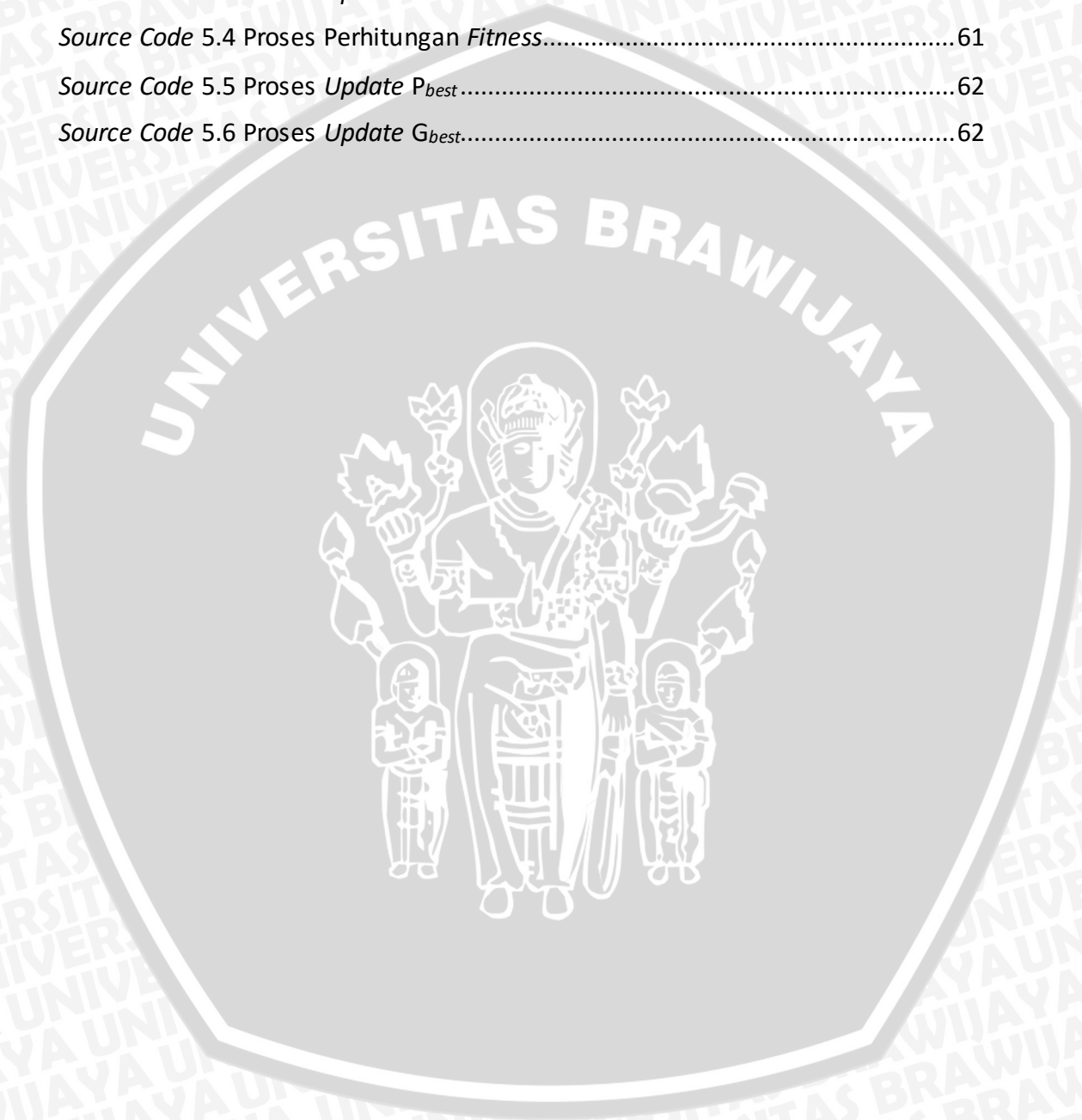
DAFTAR PERSAMAAN

Persamaan 2.1	13
Persamaan 2.2	14
Persamaan 2.3	14
Persamaan 2.4	16
Persamaan 2.5	16
Persamaan 2.6	17
Persamaan 2.7	17
Persamaan 2.8	17
Persamaan 2.9	18
Persamaan 2.10	18
Persamaan 2.11	19
Persamaan 2.12	19
Persamaan 2.13	20
Persamaan 2.14	20
Persamaan 2.15	20
Persamaan 4.1	24



DAFTAR SOURCE CODE

Source Code 5.1 Proses Inisialisasi Partikel	57
Source Code 5.2 Proses <i>Update</i> Kecepatan	59
Source Code 5.3 Proses <i>Update</i> Posisi	60
Source Code 5.4 Proses Perhitungan <i>Fitness</i>	61
Source Code 5.5 Proses <i>Update</i> P_{best}	62
Source Code 5.6 Proses <i>Update</i> G_{best}	62



BAB 1 PENDAHULUAN

1.1 Latar belakang

Penataan dan pemerataan guru merupakan salah satu upaya untuk meningkatkan mutu pendidikan yang dilakukan dengan cara mutasi guru. Mutasi adalah pemindahan guru antar satuan pendidikan dalam lingkup kecamatan, kabupaten, atau provinsi agar komposisi guru sesuai dengan kebutuhan riil setiap satuan pendidikan.

Namun, pada kenyataannya upaya penataan dan pemerataan guru dengan cara mutasi belum bisa memberikan hasil sesuai dengan harapan. Sampai saat ini, masih dapat ditemui persebaran guru yang belum merata pada beberapa daerah di Indonesia. Banyak sekolah yang kekurangan guru, tetapi di sisi lain juga ada sekolah yang kelebihan guru. Umumnya sekolah yang kekurangan guru adalah sekolah yang berada di daerah terpencil atau pinggiran sedangkan penumpukan guru cenderung terjadi di wilayah perkotaan. Ketua Gerakan Indonesia Pintar (GIP) mengungkapkan bahwa di wilayah perkotaan pun masih bisa ditemui sekolah yang kekurangan guru meskipun tidak sebanyak di pedesaan dan daerah terpencil, dimana kekurangan guru di daerah perkotaan mencapai 21%, 37% di daerah pedesaan dan 66% di daerah terpencil (Paat, 2015).

Menteri Pendidikan dan Kebudayaan Muhammad Nuh menegaskan, bahwa kekurangan guru terjadi bukan karena tidak tersedianya guru yang mencukupi, melainkan lebih kepada persebaran guru yang belum merata. Mengangkat guru baru hanya akan memberikan solusi untuk sekolah yang kekurangan guru namun tidak untuk sekolah yang kelebihan guru, selain itu pengangkatan guru baru akan membutuhkan anggaran dana yang lebih banyak. Sehingga, lebih baik dilakukan optimalisasi jumlah guru yang ada untuk didistribusikan merata secara nasional (Akuntono, 2011). Akan tetapi, untuk mendistribusikan guru secara merata membutuhkan proses penataan dan pemerataan guru yang cukup lama, karena banyak hal yang harus dipertimbangkan saat melakukan mutasi guru agar guru dapat dipindahkan sesuai tepat sasaran. Penataan dan pemerataan guru didasarkan pada Surat Keputusan Bersama (SKB) 5 menteri yang mengatur penataan dan pemerataan guru berdasarkan pemenuhan jam mengajar minimal 24 jam per minggu.

Salah satu yang menjadi pertimbangan mutasi guru adalah rasionalitas jarak dan akses dari tempat tinggal guru ke sekolah tujuan mutasi. Penataan dan pemerataan yang dilakukan secara tepat sasaran dapat menjamin guru untuk bekerja lebih baik karena selain memberikan keuntungan berupa pemenuhan kekurangan jam mengajar, guru dipindahkan ke sekolah yang memiliki akses terdekat dengan tempat tinggal guru, sehingga guru tidak merasa diberatkan. Untuk memberikan penataan dan pemerataan guru yang tepat sasaran, maka dirasa sangat membantu apabila dilengkapi perhitungan untuk optimasi penentuan sekolah tujuan mutasi tiap guru yang akan dipindahkan.

Ada banyak macam metode optimasi, beberapa di antaranya seperti algoritma genetika dan algoritma particle swarm optimization (PSO). Algoritma genetika dan algoritma PSO merupakan metode optimasi yang memiliki kemiripan satu sama lain yaitu keduanya diinisialisasi oleh sebuah populasi solusi yang dibangkitkan secara acak dengan tujuan mencari titik optimum dengan cara meng-*update* tiap hasil pembangkitan. Hanya saja, algoritma PSO memiliki kelebihan daripada algoritma genetika yaitu PSO memiliki proses lebih sederhana karena tidak ada proses *crossover* dan mutasi seperti yang ada pada algoritma genetika. Selain itu, kelebihan algoritma PSO adalah memiliki parameter lebih sedikit daripada algoritma optimasi lainnya serta algoritma PSO telah terbukti efektif digunakan untuk permasalahan dengan area pencarian yang sangat luas (Mahmudy, 2014).

Untuk optimasi penentuan dan pemerataan guru, sebelumnya telah dilakukan dengan menggunakan algoritma genetika, seperti pada penelitian yang dilakukan oleh Putra Oky (2015) dimana hasil penelitiannya menunjukkan bahwa dengan menggunakan parameter-parameter terbaik algoritma genetika dapat menghasilkan mutasi tanpa pinalti yang berarti semua guru yang di mutasi sudah berada pada posisi semestinya (Putra, 2015).

Berdasarkan keberhasilan algoritma genetika untuk mengoptimasi pemerataan guru, dimana algoritma genetika merupakan algoritma yang memiliki kemiripan dengan algoritma PSO, maka penulis berinisiatif membuat sistem berbasis komputer dengan menggunakan algoritma PSO untuk melakukan optimasi pemerataan guru mata pelajaran di Kabupaten Lumajang agar pemerataannya lebih proporsional dan tepat sasaran. Pemilihan algoritma PSO juga dikarenakan kelebihanannya yang memiliki proses yang lebih sederhana dari algoritma genetika dan kelebihan-kelebihan lain algoritma PSO seperti yang telah diuraikan sebelumnya.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dipaparkan, dapat ditarik beberapa rumusan masalah sebagai berikut:

1. Bagaimana cara implementasi algoritma PSO untuk optimasi pemerataan guru mata pelajaran di Kabupaten Lumajang?
2. Bagaimana penentuan nilai parameter untuk mendapatkan solusi terbaik menggunakan algoritma PSO untuk optimasi mutasi guru mata pelajaran di Kabupaten Lumajang?

1.3 Tujuan

Adapun maksud dan tujuan dari penelitian ini adalah sebagai berikut:

1. Mengetahui cara penerapan algoritma PSO untuk optimasi pemerataan guru mata pelajaran di Kabupaten Lumajang.
2. Mengetahui nilai parameter PSO yang dapat menghasilkan solusi terbaik dalam mutasi guru mata pelajaran di Kabupaten Lumajang.

1.4 Manfaat

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Membantu mengoptimalkan hasil penataan dan pemerataan guru mata pelajaran di Kabupaten Lumajang agar lebih proporsional dan tepat sasaran.
2. Membantu Badan Pertimbangan Jabatan dan Pangkat (Baperjakat) dan Badan Kepegawaian Daerah (BKD) Kabupaten Lumajang untuk melakukan penatan dan pemerataan guru mata pelajaran secara efektif dan profesional.

1.5 Batasan masalah

Pada penelitian ini terdapat beberapa batasan masalah yang harus terpenuhi, di antara lain:

1. Algoritma optimasi yang digunakan adalah algoritma *Particle Swarm Optimization* (PSO).
2. Acuan penataan dan pemerataan guru dilihat dari jumlah kekurangan dan jumlah kelebihan guru untuk masing-masing mata pelajaran di setiap SMP di Kabupaten Lumajang.
3. Penataan dan pemerataan hanya diberlakukan terhadap guru mata pelajaran yang bestatus Pegawai Negeri Sipil (PNS) dan mengajar pada Sekolah Menengah Pertama (SMP) di Kabupaten Lumajang.
4. Parameter yang menjadi pertimbangan dalam melakukan penataan dan pemerataan terhadap guru mata pelajaran di Kabupaten Lumajang adalah jarak, masa kerja, dan usia.
5. Penataan dan pemerataan tidak diberlakukan kepada guru dengan tambahan tugas sebagai kepala sekolah dan guru yang memiliki masa kerja kurang dari 5 tahun.

1.6 Sistematika Penulisan

Adapun sistematika penulisan yang digunakan dalam menyusun skripsi ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab ini menguraikan latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika penulisan skripsi.

BAB 2 LANDASAN KEPUSTAKAAN

Bab ini berisi kajian pustaka dan teori-teori yang dibutuhkan dalam penyusunan skripsi. Kajian pustaka merupakan pembahasan mengenai penelitian-penelitian terdahulu yang terkait dengan topik skripsi.

BAB 3 METODOLOGI

Bab ini menjelaskan tentang metodologi yang dilakukan dalam proses pengerjaan skripsi.

BAB 4 PERANCANGAN

Bab ini berisi tentang perancangan sistem yang akan diimplementasikan dalam penelitian.

BAB 5 IMPLEMENTASI

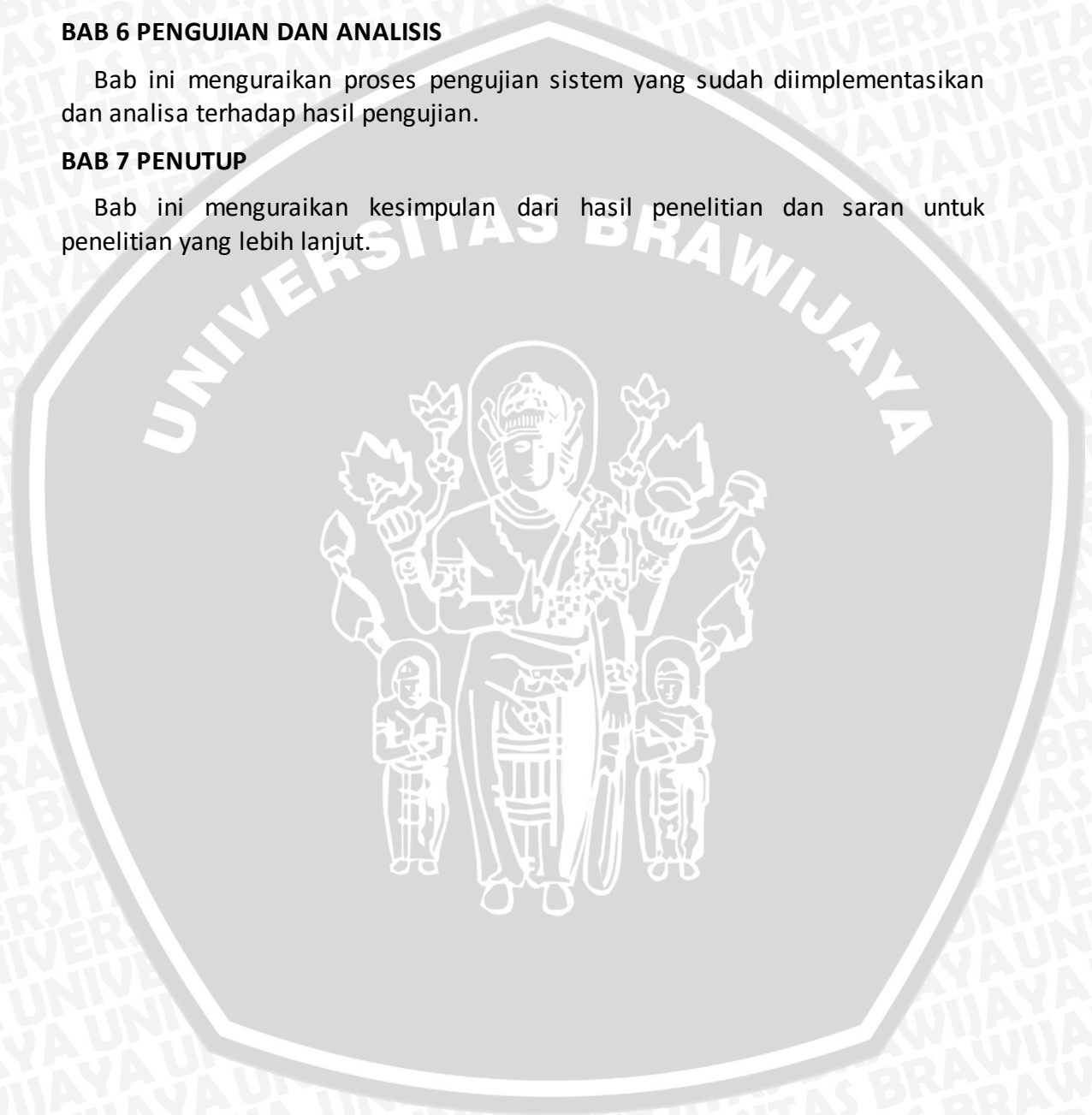
Bab ini membahas implementasi dari perancangan yang telah dilakukan.

BAB 6 PENGUJIAN DAN ANALISIS

Bab ini menguraikan proses pengujian sistem yang sudah diimplementasikan dan analisa terhadap hasil pengujian.

BAB 7 PENUTUP

Bab ini menguraikan kesimpulan dari hasil penelitian dan saran untuk penelitian yang lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakaan terdiri dari kajian pustaka dan dasar teori. Kajian pustaka adalah pembahasan tentang penelitian-penelitian terdahulu yang berkaitan dengan topik skripsi. Dasar teori berisi beberapa teori-teori yang berkaitan dengan topik penelitian penulis.

2.1 Kajian Pustaka

Untuk mengetahui perbandingan antara penelitian penulis dengan penelitian terdahulu yang serupa, maka penulis menghimpun beberapa informasi seperti objek, metode dan hasil dari penelitian lain yang menjadi pembeda dengan penelitian penulis seperti yang ditunjukkan pada Tabel 2.1.

Penelitian pertama adalah penelitian yang melakukan optimasi terdapat alokasi sumber daya manusia di sebuah perusahaan menggunakan algoritma *Multi-objective Particle Swarm Optimization* (MOPSO), dimana alokasi sumber daya manusia mengacu pada *competecenty* model dan prinsip *Pareto*. Sebagai percobaan, dimisalkan perusahaan memiliki 4 posisi baru yang membutuhkan pegawai. Perusahaan memilih 4 pegawai dari departemen yang ada. Perusahaan memiliki 4 departemen dan masing-masing departemen memiliki 12 pegawai. Sehingga, MOPSO akan digunakan untuk memilih 4 pegawai dari 48 pegawai yang paling memungkinkan untuk mengisi posisi yang dibutuhkan. Pada setiap departemen akan mengambil 1 pegawai yang memiliki keuntungan paling besar dan biaya paling sedikit untuk dipilih. Percobaan dilakukan dengan menggunakan 80 partikel, nilai $w_{int}=0.94$, $w_{end}=0.13$, $c_1=c_2=1.28$, dan maksimum $swarm=800$. Hasil percobaan ini menunjukkan bahwa pegawai yang paling optimal untuk mengisi posisi baru adalah pegawai ke-12 dari departemen 1, pegawai ke-7 dari departemen 2, pegawai ke-1 dari departemen 3, pegawai ke-2 dari departemen 4. Berdasarkan percobaan yang dilakukan, dapat disimpulkan bahwa algoritma MOPSO mampu digunakan untuk permasalahan optimasi alokasi sumber daya dengan mencari pegawai dengan nilai keuntungan paling besar dan biaya paling kecil untuk mengisi posisi baru pada suatu perusahaan (Jia & Gong, 2008).

Penelitian kedua adalah penelitian mengenai optimasi pemerataan mutasi guru dengan menggunakan Algoritma Genetika. Guru yang menjadi objek dalam penelitian adalah guru Sekolah Dasar (SD) di Kabupaten Banyuwangi. Penelitian menggunakan data guru dan data kualitas guru. Data guru meliputi nama, NIP, jenis kelamin, pangkat, kelas yang diajar, dan sekolah tempat mengajar. Data kualitas mengacu pada 11 parameter dari Sasaran Kerja Pegawai (SKP) dan 7 parameter dari Penilaian Perilaku Kerja (PPK). Data kualitas yang digunakan merupakan hasil jumlah dari 60% nilai SKP dengan 40% nilai PPK. Pada pengujian parameter Algoritma Genetika pada penelitian ini didapatkan bahwa banyaknya generasi yang optimal adalah 400, ukuran populasi yang optimal adalah 40, nilai kombinasi dari *crossover rate* dengan *mutation rate* yang paling optimal adalah 0.5 dan 0.5. Penelitian ini menunjukkan bahwa percobaan dengan menggunakan

parameter terbaik, dapat menghasilkan mutasi tanpa pinalti yang berarti semua guru yang di mutasi sudah berada pada posisi semestinya (Putra, 2015).

Penelitian ketiga merupakan penelitian yang bertujuan mengoptimasi distribusi guru dengan menggunakan metode Dijkstra. Guru yang menjadi objek penelitian adalah guru Sekolah Menengah Pertama (SMP) di Kota Batu yang mengajar mata pelajaran Pendidikan Kewarganegaraan (PKn). Data yang digunakan adalah data guru dan data sekolah. Data guru yang menjadi parameter mutasi adalah jenis kelamin, status kepegawaian, golongan, jarak antar sekolah, dan masa kerja, dimana guru yang memiliki nilai terendah lebih diprioritaskan untuk dimutasi. Sedangkan, data sekolah yang digunakan adalah data kekurangan dan kelebihan guru mata pelajaran PKn. Penelitian ini mendistribusikan guru mata pelajaran PKn secara merata berdasarkan jarak terpendek dengan cara memutasi guru dari sekolah yang kelebihan guru mata pelajaran PKn ke sekolah yang kekurangan guru mata pelajaran PKn menggunakan graf berbobot. Untuk 35% sekolah yang kekurangan guru, metode Dijkstra berhasil memutasi guru sebanyak 35% dari 37% sekolah yang kelebihan guru mata pelajaran PKn, sedangkan sisanya sebanyak 2% tidak bisa dimutasi karena semua sekolah yang kekurangan guru mata pelajaran PKn sudah tercukupi. Sehingga, kelebihan guru yang tersisa bisa di mutasi ke kabupaten-kabupaten di luar atau sekitar Kota Batu yang masih kekurangan guru mata pelajaran PKn (Santoso, Purnama, & Sumpeno, 2013).

Tabel 2.1 Kajian Pustaka

No.	Judul	Input	Metode	Hasil
1	<i>Multi-criteria Human Resource Allocation for Problems Using Multi-objective Particle Swarm Optimization Algorithm</i> (Jia & Gong , 2008)	Objek: Pegawai Input: Nilai <i>cost</i> dan <i>benefit</i> tiap pegawai	Metode: Algoritma <i>Multi-objective Particle Swarm Optimization</i> (MOPSO)	Hasil: Pegawai dengan nilai keuntungan paling besar dan biaya paling kecil untuk mengisi posisi yang dibutuhkan perusahaan
2	Implementasi Algoritma Genetika Untuk Optimasi Pemerataan Mutasi Guru SD Di Kabupaten Banyuwangi (Putra, 2015)	Objek: Guru SD di Kabupaten Banyuwangi Input: nama, nip, jenis kelamin, pangkat, kelas yang diajar, dan sekolah tempat mengajar, nilai SKP dan PPK	Metode: Algoritma Genetika	Hasil: Sekolah tujuan mutasi untuk 12 guru dengan nilai <i>fitness</i> terbaik

No	Judul	Input	Metode	Hasil
3	Optimasi Distribusi Guru Berbasis Metode Dijkstra (Santoso, Purnama, & Sumpeno, 2013)	Objek: Guru SMP mata pelajaran PKn di Kota Batu <i>Input:</i> Data guru meliputi jenis kelamin, status kepegawaian, golongan, jarak antar sekolah, dan masa kerja. Data sekolah meliputi nama, banyak kekurangan guru PKn, kelebihan guru PKn	Metode: Dijkstra	Hasil: Sekolah tujuan mutasi untuk 9 guru dari sekolah yang kelebihan guru mata pelajaran PKn ke sekolah yang kekurangan guru mata pelajaran PKn dengan jarak terpendek. Metode Dijkstra berhasil memutasi guru sebanyak 35% sesuai kekurangan yang ada
4	Implementasi Algoritma <i>Particle Swarm Optimization</i> (PSO) Untuk Optimasi Pemerataan Mutasi Guru Mata Pelajaran di Kabupaten Lumajang (Usulan).	Objek: Guru SMP berstatus PNS di Kabupaten Lumajang <i>Input:</i> Data guru meliputi nama, mata pelajaran yang diampu, sekolah tempat mengajar, alamat, tanggal Lahir, tanggal diangkat di sekolah tempat mengajar, dan tugas tambahan. Data sekolah meliputi nama, alamat, banyak kekurangan guru per mapel, kelebihan guru per mapel, sedia guru per mata pelajaran	Metode: Algoritma <i>Particle Swarm Optimization</i>	Hasil: Sekolah tempat tujuan mutasi guru untuk guru dari seluruh sekolah yang kelebihan ke seluruh sekolah yang kekurangan guru dengan nilai total <i>fitness</i> terbaik

Sumber : (Jia & Gong , 2008), (Putra, 2015), (Santoso, Purnama, & Sumpeno, 2013)

2.2 Guru

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. Berdasarkan sifat, tugas, dan kegiatannya, guru dibedakan menjadi 3 jenis yakni Guru Kelas, Guru Mata Pelajaran serta Guru Bimbingan dan Konseling (BK) (Kemenristekdikti, 2011).

1. Guru Kelas

Guru Kelas adalah guru yang mempunyai tugas, tanggung jawab, wewenang, dan hak secara penuh dalam proses pembelajaran seluruh mata

pelajaran di kelas tertentu pada satuan pendidikan formal tingkat Taman Kanak-kanak (TK)/Taman Kanak-Kanak Luar biasa (TKLB) dan Sekolah Dasar (SD)/Sekolah Dasar Luar Biasa (SDLB) atau satuan pendidikan formal lain yang sederajat.

2. Guru Mata Pelajaran

Guru Mata Pelajaran adalah guru yang mempunyai tugas, tanggung jawab, wewenang, dan hak secara penuh dalam proses pembelajaran mata pelajaran tertentu pada satuan pendidikan formal tingkat SMP/Sekolah Menengah Pertama Luar biasa (SMPLB), Sekolah Menengah Atas (SMA)/Sekolah Menengah Atas Luar Biasa (SMALB), dan Sekolah Menengah Kejuruan (SMK). Termasuk guru mata pelajaran pendidikan jasmani dan kesehatan, dan guru pendidikan agama pada satuan pendidikan formal tingkat SD/SDLB.

3. Guru Bimbingan dan Konseling

Guru Bimbingan dan Konseling adalah guru yang mempunyai tugas, tanggung jawab, wewenang, dan hak secara penuh dalam kegiatan bimbingan dan konseling terhadap beberapa peserta didik pada satuan pendidikan formal tingkat SMP/SMPLB, SMA/SMALB dan SMK.

2.3 Mutasi

Mutasi guru adalah perpindahan guru antar satuan pendidikan dalam satu kabupaten/kota, dalam kabupaten/kota yang berbeda, maupun dalam provinsi yang berbeda dikarenakan sebab tertentu. Secara umum penyebab terjadinya mutasi adalah (Kemenristekdikti, 2011):

1. Mutasi karena kenaikan jabatan atau disebut dengan mutasi promosi
2. Mutasi karena adanya kebijakan bupati untuk pergantian formasi dalam rangka melakukan pemerataan guru.
3. Mutasi karena pindah tempat tinggal.

2.4 Mutasi Guru PNS

Pada prinsipnya mekanisme mutasi guru PNS antar satuan pendidikan pada kabupaten/kota yang sama, pada kabupaten/kota lain dalam provinsi yang sama, maupun pada kabupaten/kota dalam provinsi yang berbeda mengacu pada ketentuan dan peraturan yang berlaku. Agar tujuan pemerataan guru PNS dapat tercapai maka pemerintah setempat wajib memfasilitasi proses pemindahan guru (Kemenristekdikti, 2011).

2.4.1 Mutasi Guru PNS ke Satuan Pendidikan Lain

Kelebihan guru untuk suatu mata pelajaran di satuan pendidikan tertentu menandakan bahwa jam mengajar untuk mata pelajaran tersebut lebih sedikit daripada jumlah guru yang tersedia yang menyebabkan adanya guru yang memiliki jam mengajar kurang dari standar minimal 24 jam. Sehingga satuan pendidikan yang memiliki kelebihan guru harus menetapkan guru yang akan dimutasi ke

satuan pendidikan yang kekurangan guru. Berikut adalah beberapa pertimbangan dalam menetapkan guru yang akan dimutasi (Kemenristekdikti, 2011):

1. Pertimbangan atas pemenuhan kebutuhan guru dalam rangka peningkatan mutu pendidikan berdasarkan penilaian kinerja.
2. Pertimbangan atas pemenuhan beban mengajar minimum 24 jam per minggu di sekolah tujuan mutasi.
3. Pertimbangan atas rasionalitas jarak dan akses dari tempat tinggal guru ke satuan pendidikan tujuan mutasi.

2.4.2 Mutasi Guru PNS ke Kabupaten/Kota Lain

Mutasi guru PNS ke Kabupaten/kota lain dilakukan apabila di satuan pendidikan lain di kabupaten/kota yang sama tidak memiliki kekurangan guru atau kebutuhan gurunya sudah mencukupi. Maka pemerintah kabupaten/kota mengupayakan untuk memutasi guru dari satuan pendidikan di wilayahnya yang masih kelebihan guru ke satuan pendidikan di kabupate/kota lain yang kekurangan guru. Tahapan mutasi guru antar Kabupaten/kota adalah sebagai berikut (Kemenristekdikti, 2011):

1. Menginformasikan kepada kabupaten/kota lain yang kekurangan guru mengenai kelebihan guru.
2. Menyampaikan data/portofolio guru yang bersangkutan dan memfasilitasi proses mutasi guru yang diterima di kabupaten/kota lain yang membutuhkan guru.
3. Menentukan mutasi guru berdasarkan pertimbangan terhadap penilaian kinerja, pertimbangan terhadap kedekatan tempat tinggal guru dan kemudahan akses ke satuan pendidikan tujuan mutasi, pertimbangan aspek sosial yang kondusif, dan aspek ekonomi yang lebih baik di kabupaten/kota lain tujuan mutasi.

Apabila mutasi guru PNS antar kabupaten/kota tidak bisa dilakukan, maka kabupaten/kota yang kelebihan dan yang kekurangan guru PNS menyampaikan laporan mengenai kelebihan dan/atau kekurangan guru PNS kepada pemerintah provinsi.

2.4.3 Kriteria Guru PNS Dalam Mutasi

Ada beberapa kriteria yang menjadikan guru yang berstatus PNS untuk dimutasi, di antaranya adalah (Kemenristekdikti, 2011):

1. Guru yang bertugas di satuan pendidikan dengan kelebihan guru.
2. Guru yang dibutuhkan oleh satuan pendidikan lain karena mempunyai keterampilan atau keahlian khusus.
3. Guru yang mempunyai sertifikat pendidik tetapi belum dapat memenuhi beban mengajar minimal 24 jam per minggu.
4. Lebih diprioritaskan untuk guru yang masa kerjanya paling sedikit.

5. Permintaan mutasi dari guru sendiri.
6. Lebih diprioritaskan untuk guru yang memiliki jarak terdekat dengan satuan pendidikan yang kekurangan guru.

2.4.4 Waktu Pemindahan

Demi menjaga keberlangsungan proses pembelajaran di sekolah, guru yang dimutasi harus sudah mengajar di sekolah yang baru mulai semester awal. Jika mutasi guru dilakukan dalam kabupaten/kota yang sama maka mutasi dilakukan di akhir semester pada tahun berjalan dan jika mutasi guru dilakukan dalam kabupaten/kota yang berbeda, maka mutasi guru harus dilakukan paling lambat pada bulan Oktober tahun berjalan (Kemenristekdikti, 2011).

2.5 Penataan dan Pemerataan Guru

Terdapat beberapa tahapan yang harus dilakukan saat melakukan penataan dan pemerataan guru. Setiap instansi yang terkait memiliki tahapan yang berbeda sesuai dengan tingkatannya (Kemenristekdikti, 2011).

1. Tingkat Satuan Pendidikan

- a. Menyusun perencanaan pengembangan sekolah terkait perencanaan akan manajemen sekolah, kurikulum, jumlah peserta didik, dan kebutuhan-kebutuhan sekolah seperti kebutuhan guru dan sarana prasarana.
- b. Menyusun perencanaan kebutuhan guru, berdasarkan pola perhitungan yang telah dijelaskan pada petunjuk teknis Surat Keputusan Bersama 5 menteri terkait penataan dan pemerataan guru PNS.
- c. Mengidentifikasi jumlah dan jenis guru PNS pada tahun berjalan untuk mengetahui adanya kekurangan atau kelebihan guru.
- d. Menentukan solusi untuk penyelesaian pemenuhan kekurangan atau kelebihan guru.
- e. Membuat laporan hasil analisis kebutuhan dan optimalisasi guru PNS yang telah ditandatangani oleh kepala satuan pendidikan. Pada tingkat TK dan SD diserahkan kepada UPTD pendidikan kecamatan terlebih dahulu untuk kemudian diteruskan kepada dinas pendidikan kabupaten/kota. Pada tingkat SMP, SMA, dan SMK diserahkan kepada dinas pendidikan kabupaten/kota. Sedangkan, pada tingkat pendidikan luar biasa (PLB) dan satuan pendidikan yang berada dalam naungan dinas pendidikan provinsi diserahkan kepada dinas pendidikan provinsi.

2. Tingkat UPTD Pendidikan Kecamatan

- a. Merekapitulasi data guru dari laporan hasil analisis kebutuhan guru PNS dari seluruh tingkat TK dan SD di kecamatan bersangkutan.
- b. Merekonsiliasi kebutuhan guru PNS dengan melakukan koordinasi dengan seluruh kepala TK dan SD di kecamatan bersangkutan.

- c Melaporkan hasil rekonsiliasi kebutuhan guru PNS kepada dinas pendidikan kabupaten/kota.

3. Dinas Pendidikan Kabupaten/Kota

- a Merekapitulasi laporan hasil analisis kebutuhan guru PNS pada semua tingkat pendidikan dari seluruh UPTD pendidikan kecamatan di kabupaten/kota bersangkutan.
- b Merekonsiliasi kebutuhan guru PNS bersama UPTD pendidikan kecamatan dan kepala sekolah SMP, SMA, dan SMK di kabupaten/kota bersangkutan.
- c Melakukan perencanaan penataan dan pemerataan guru PNS antar satuan pendidikan, antarjenjang, dan antar jenis pendidikan di wilayahnya.
- d Melaporkan hasil perencanaan kepada pemerintah provinsi selambatnya bulan Februari pada tahun berjalan.
- e Memfasilitasi dan memproses pemindahan guru PNS dari sekolah yang kelebihan guru ke sekolah yang kekurangan guru.
- f Menyusun laporan terkait pelaksanaan penataan dan pemerataan guru PNS antar satuan pendidikan, antarjenjang, dan antar jenis pendidikan di wilayahnya untuk diserahkan kepada pemerintah provinsi dengan tembusan kepada Kementerian Pendidikan dan Kebudayaan selambatnya bulan Juni tahun berjalan.

4. Dinas Pendidikan Provinsi

- a Merekapitulasi laporan hasil analisis kebutuhan guru seluruh satuan pendidikan luar biasa (PLB) atau satuan pendidikan yang berada dalam naungan dinas pendidikan provinsi.
- b Merekonsiliasi kebutuhan guru PLB dan guru pada satuan pendidikan yang berada dalam naungan dinas pendidikan provinsi bersama masing-masing kepala sekolah yang bersangkutan.
- c Melakukan pemindahan guru PNS dari sekolah yang kelebihan guru ke sekolah yang kekurangan guru dalam satu kabupaten/kota atau antar kabupaten/kota dalam satu provinsi untuk PLB dan satuan pendidikan yang berada dalam naungan dinas pendidikan provinsi.
- d Melaporkan hasil capaian pelaksanaan penataan dan pemerataan guru PLB dan guru pada satuan pendidikan yang berada dalam naungan dinas pendidikan provinsi melalui Lembaga Penjaminan Mutu Pendidikan (LPMP) untuk diserahkan kepada Kementerian Pendidikan dan Kebudayaan.
- e Menerima hasil capaian pelaksanaan penataan guru PNS dari semua tingkat PLB dari pemerintah kabupaten/kota.
- f Bersama seluruh kepala sekolah PLB dan kepala sekolah satuan pendidikan yang berada dalam naungan dinas pendidikan provinsi melakukan rekonsiliasi kebutuhan guru PNS.

- g Memberikan perencanaan penataan dan pemerataan guru PNS antar satuan pendidikan, antar jenjang, dan antar jenis pendidikan di pemerintah provinsi selambatnya bulan Maret tahun berjalan kepada Kementerian Pendidikan dan Kebudayaan melalui LPMP untuk kemudian diteruskan ke Kementerian Dalam Negeri, Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, dan Kementerian Keuangan.
- h Memfasilitasi pemindahan guru PNS antar kabupaten/kota pada seluruh jenjang PLB.
- i Menyusun laporan terkait pelaksanaan penataan dan pemerataan guru PNS antar satuan pendidikan, antar jenjang, dan jenis pendidikan di pemerintah provinsi kepada Kementerian Pendidikan dan Kebudayaan melalui Lembaga Penjaminan Mutu Pendidikan (LPMP) selambatnya bulan Juli tahun berjalan untuk kemudian diteruskan ke Kementerian Dalam Negeri, Kementerian Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, dan Kementerian Keuangan.

5. Kementerian Pendidikan dan Kebudayaan

- a Menerima hasil capaian pelaksanaan penataan guru PNS dari semua tingkat PLB dan SMK dalam lingkup kabupaten/kota dan provinsi, yang dilengkapi dengan data kekurangan dan/atau kelebihan guru .
- b Merekonsiliasi penataan guru PNS bersama dinas pendidikan kabupaten/kota dan dinas pendidikan provinsi.
- c Melakukan pemetaan kelebihan dan/atau kekurangan guru PNS dari seluruh tingkat PLB dan SMK secara nasional.
- d Memfasilitasi dan mengkoordinasikan pemindahan guru PNS dari semua tingkat PLB dan SMK ke provinsi atau kabupaten/kota yang berbeda.
- e Mengevaluasi laporan dan menetapkan capaian penataan dan pemerataan guru PNS secara nasional.
- f Melaporkan hasil evaluasi capaian penataan dan pemerataan guru PNS secara nasional kepada Kementerian Keuangan, Kementerian Negara Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, dan Kementerian Dalam Negeri untuk dijadikan bahan pertimbangan dalam pengambilan kebijakan.
- g Melakukan koordinasi dalam menetapkan sanksi bersama Kementerian Negara Pendayagunaan Aparatur Negara dan Reformasi Birokrasi, Kementerian Dalam Negeri, Kementerian Keuangan, dan Kementerian Agama.

2.6 Optimalisasi Pemenuhan Kebutuhan Guru Mata Pelajaran

Optimalisasi pemenuhan kebutuhan guru mata pelajaran adalah usaha untuk meminimalkan adanya kekurangan dan kelebihan guru mata pelajaran dengan cara memindahkan guru mata pelajaran tertentu dari satuan pendidikan ke satuan

pendidikan lain yang kekurangan guru dalam kabupaten/kota yang sama atau dalam kabupaten/kota yang berbeda. Jika memungkinkan untuk memindahkan guru ke sekolah dengan kekurangan mata pelajaran yang bukan bidangnya, maka pemindahan guru dapat dilakukan dengan mempertimbangkan kedekatan latar belakang pendidikan dengan mata pelajaran yang akan diampu atau bisa disebut alih fungsi/profesi. Guru alih fungsi/profesi akan diikutsertakan dalam pendidikan/pelatihan/penataran agar mendapatkan kompetensi profesional pada mata pelajaran baru yang akan diampu (Kemenristekdikti, 2011).

2.7 Perhitungan Kebutuhan Guru SMP

Berikut adalah aturan-aturan yang harus diperhatikan dalam melakukan perhitungan kebutuhan guru SMP (Kemenristekdikti, 2011):

1. Setiap rombongan belajar (rombel) dalam mengikuti mata pelajaran tertentu diampu oleh 1 orang guru.
2. Guru mata pelajaran mengampu 1 jenis mata pelajaran sesuai dengan sertifikat pendidik yang dimilikinya.
3. Wajib mengajar minimal 24 jam perminggu.
4. Jumlah rombel yang digunakan adalah jumlah rombel dengan rasio siswa guru yang sesuai dengan Peraturan Pemerintah Nomor 74 tahun 2008 dan Peraturan Menteri Pendidikan Nasional nomor 41 tahun 2007 tentang standar proses untuk satuan pendidikan dasar dan menengah. Rumus perhitungan untuk menentukan jumlah rombel ideal ditunjukkan pada Persamaan 2.1.

$$jr_i = \sum_{i=1}^3 \frac{jm}{rsg} \quad (2.1)$$

Keterangan:

jr_i = jumlah rombel ideal untuk kelas ke- i

jm = jumlah murid

rsg = rasio siswa guru sesuai Standar Pelayanan minimum (SPM)

Jadi, jika suatu sekolah memiliki jumlah murid sebanyak 330 murid yang terdiri dari 110 murid kelas 7, 100 murid kelas 8, dan 120 murid kelas 9 dan rasio siswa guru dari dinas pendidikan sebesar 32 maka jumlah rombel idealnya adalah 12 yang didapatkan dari perhitungan sebagai berikut:

$$jr = \frac{110}{32} + \frac{100}{32} + \frac{100}{32}$$

$$jr = 3.48 + 3.12 + 3.75$$

Dibulatkan keatas

$$jr = 4 + 4 + 4 = 12$$

5. Jumlah jam tersedia merupakan jumlah jam tatap muka berdasarkan kurikulum tingkat satuan pendidikan (KTSP) yang dibutuhkan oleh sekolah sesuai dengan jumlah rombelnya yang dihitung menggunakan Persamaan 2.2.

$$jt = \sum_{k=7}^9 jr_k \times jtm \quad (2.2)$$

Keterangan:

jt = jam tersedia

jr = jumlah rombel pada kelas ke- k

jt = jam tatap muka per minggu sesuai KTSP

jadi, jika jumlah rombel ideal di SMP tertentu adalah 12 rombel dengan masing-masing kelas 7,8,9 ada 4 rombel maka jumlah jam yang dibutuhkan perminggu adalah 48 jam yang didapatkan dari perhitungan sebagai berikut:

$$jt = 4 \times 4 + 4 \times 4 + 4 \times 4 = 48$$

6. Jumlah guru dihitung dengan membagi jam tersedia dengan wajib mengajar minimal 24 jam. Sesuai Pemerdiknas nomor 39 tahun 2010, setiap SMP harus terdapat 1 orang guru untuk setiap mata pelajaran. Apabila jam yang tersedia tidak habis dibagi dengan wajib mengajar minimal 24 jam, maka dilakukan pembulatan kebawah dengan ketentuan sebagai berikut:

- Jika setelah dibulatkan ke bawah, tatap muka per minggu untuk masing-masing guru tidak lebih dari 40 jam, maka angka yang diambil adalah hasil pembulatan ke bawah
- Jika setelah dibulatkan ke bawah, tatap muka per minggu untuk masing-masing guru melebihi 40 jam, maka nilai yang diambil adalah pembulatan keatas dimana akan ada 1 orang guru yang belum mengajar 24 jam. Untuk sekolah yang berada di daerah khusus, kebutuhan gurunya disesuaikan dengan peraturan yang berlaku

Rumus perhitungan kebutuhan guru SMP dihitung menggunakan Persamaan 2.3.

$$kg = \frac{jt}{24} \quad (2.3)$$

Keterangan:

kg = kebutuhan guru

jt = jam tersedia

Misalkan jika suatu SMP tertentu mempunyai jumlah jam mengajar sebanyak 44 jam untuk mata pelajaran Bahasa Indonesia, maka mata pelajaran Bahasa Indonesia hanya membutuhkan 1 orang guru.

$$kg = \frac{44}{23} = 1.83 \approx 1$$

Namun, setelah dibulatkan ke bawah ternyata guru mengalami kelebihan jam mengajar sebanyak 4 jam, Sehingga perlu 1 guru tambahan lagi dengan catatan 1 orang guru mengajar 24 jam dan 1 orang guru yang lain hanya mengajar 20 jam. Jadi, setidaknya perlu tambahan 4 jam lagi untuk guru yang belum memenuhi jam minimal wajib mengajar.

2.8 Optimasi

Optimasi adalah usaha untuk mendapatkan hasil yang terbaik dari beberapa pilihan, sehingga dengan menggunakan optimasi diharapkan bisa mendapatkan keputusan dengan keuntungan yang besar dan *cost* seminim mungkin. Keuntungan yang ingin dicapai dinyatakan sebagai fungsi tujuan dengan variabel keputusan untuk dicari nilai optimalnya. Terdapat dua teknik optimasi yang sering digunakan yaitu teknik *mathematical programming* dan *combinatorial optimization* (Zerda, 2009).

2.9 Particle Swarm Optimization (PSO)

Algoritma PSO adalah algoritma yang terinspirasi oleh perilaku sosial sekawanan hewan seperti burung atau ikan. Seekor hewan dalam PSO dianggap sebagai partikel. Perilaku partikel dipengaruhi oleh kecerdasan dari partikel itu sendiri dan perilaku dari partikel lain di kelompoknya. Jika satu partikel menemukan jalan tepat dan pendek menuju ke suatu sumber makanan, maka partikel-partikel yang lain akan mengikuti jalan tersebut. Pada PSO, kawanan (*swarm*) mempunyai ukuran tertentu dimana posisi awal setiap partikel terletak di suatu lokasi yang acak pada suatu ruang multidimensi. Setiap partikel mempunyai 2 karakteristik yaitu posisi dan kecepatan. Setiap partikel bergerak dalam ruang tertentu dengan cara mengingat posisi terbaik yang pernah ditemui dan menyampaikan informasi mengenai posisi terbaik tersebut kepada partikel lain agar semua partikel dapat menyamakan kecepatan antara satu dengan yang lainnya. Hal ini bisa digambarkan seperti pada perilaku burung saat bersama kawanannya, beberapa perilaku tersebut antara lain (Santosa, 2011):

1. Jarak antar burung tidak terlalu dekat (kohesi).
2. Arah terbang burung sesuai dengan arah rata-rata keseluruhan burung yang ada pada kelompoknya (*alignment*).
3. Posisi burung akan menyesuaikan posisi rata-rata burung yang lain dengan (separasi).

Pencarian solusi dalam algoritma PSO dilakukan oleh *swarm*. Inisialisasi *swarm* dibangkitkan secara acak dengan batasan nilai terkecil dan terbesar dari

dimensinya. Partikel-partikel dalam populasi merepresentasikan solusi untuk permasalahan yang dihadapi. Pencarian solusi paling optimal dilakukan partikel dengan melintasi ruang pencarian. Saat melintasi ruang pencarian, partikel akan menyesuaikan posisi terbaik dirinya sendiri (P_{best}) dan posisi terbaik seluruh partikel di kelompoknya (G_{best}) hingga mencapai batas maksimum iterasi atau didapat posisi yang konvergen. Solusi yang dipresentasikan oleh partikel di setiap iterasi akan dievaluasi menggunakan fungsi objektif yang telah ditetapkan. Posisi partikel akan semakin mengarah ke target yang dituju (minimasi atau maksimasi fungsi) di setiap iterasinya. Pada PSO, P_{best} dan G_{best} akan disimpan nilainya untuk keseluruhan iterasi. Secara singkat, Tahapan Algoritma PSO terdiri dari (Santosa, 2011):

1. Membangkitkan posisi awal sejumlah partikel lengkap dengan kecepatan awal secara acak.
2. Mengevaluasi *fitness/cost* posisi setiap partikel.
3. Menentukan P_{best} dan G_{best} .
4. Memperbarui kecepatan setiap partikel iterasi ke- $t+1$ menggunakan Persamaan 2.4.

$$v_{ij}(t) = v_{ij}(t-1) + c_1 r_1 [x_{ij}^L - x_{ij}(t-1)] + c_2 r_2 [x^G - x_{ij}(t-1)] \quad (2.4)$$

$$x_{ij}(t) = x_{i1}(t), x_{i2}(t), \dots, x_{ij}(t)$$

$$v_{ij}(t) = v_{i1}(t), v_{i2}(t), \dots, v_{ij}(t)$$

Keterangan:

$x_{ij}(t)$ = posisi partikel ke- i , dimensi ke- j pada iterasi ke- t

$v_{ij}(t)$ = kecepatan partikel ke- i , dimensi ke- j pada iterasi ke- t

x_{ij}^L = P_{best} dari partikel ke- i

x^G = G_{best} dari seluruh *swarm*

c_1 = konstanta *learning factor* untuk partikel

c_2 = konstanta *learning factor* untuk *swarm*

r_1, r_2 = bilangan acak pada antara 0 - 1

5. Memperbarui Posisi setiap partikel iterasi ke- $t+1$ menggunakan Persamaan 2.5.

$$x_{ij}(t) = x_{ij}(t-1) + v_{ij}(t) \quad (2.5)$$

6. Mengevaluasi *fitness/cost* posisi setiap partikel pada iterasi ke- $t+1$.
7. Menentukan P_{best} dan G_{best} di iterasi ke- $t+1$
8. Mengulangi langkah 4 sampai langkah 7 hingga kondisi berhenti terpenuhi.

2.9.2 Inisialisasi Partikel

Posisi awal sebuah partikel harus berada dalam *range* $[x_{min}, x_{max}]$. Jika posisi yang dihasilkan kurang dari x_{min} maka posisi di kembalikan kepada x_{min} , dan jika posisi melebihi x_{max} maka posisi dikembalikan pada x_{max} . Formula untuk inisialisasi partikel sesuai dengan *range* dimensinya dapat dituliskan menggunakan Persamaan 2.6.

$$x_{ij}(t) = x_{min,j} + r(x_{max,j} - x_{min,j}) \quad (2.6)$$

Keterangan :

x_{max} = posisi maksimum partikel pada dimensi ke- j

x_{min} = posisi minimum partikel pada dimensi ke- j

r = angka random 0-1

Banyaknya partikel sangat berpengaruh terhadap hasil optimasi yang dilakukan. Pada kebanyakan masalah, penggunaan 10 partikel sudah cukup untuk mendapatkan hasil yang bagus. Namun, untuk hasil yang lebih bagus bisa digunakan 20 sampai 40 partikel. Semakin kompleks masalah yang dihadapi bisa digunakan partikel yang lebih banyak lagi (Zerda, 2009).

2.9.3 Bobot Inersia

Untuk mengatasi *update* kecepatan yang terlalu cepat pada algoritma PSO dilakukan modifikasi atau perbaikan dengan menambahkan suatu variabel pada formula *update* kecepatan yang disebut dengan θ (bobot inersia). *Update* kecepatan yang terlalu cepat akan menyebabkan nilai optimal dari fungsi tujuan yang dicari sering terlewatkan. Sehingga, perbaikan formula *update* kecepatan pada PSO dapat dituliskan seperti Persamaan 2.7.

Diperlukan untuk menentukan nilai θ yang mampu menjaga keseimbangan antara pencarian global dan lokal dikarenakan nilai θ yang terlalu tinggi akan menambah porsi pencarian global dan untuk nilai θ yang rendah akan lebih menekankan pencarian lokal. Oleh karena itu, ditentukan suatu persamaan seperti yang dituliskan pada Persamaan 2.8 untuk menentukan nilai θ agar nilainya semakin mengecil seiring dengan bertambahnya iterasi dan dapat digunakan mempercepat konvergensi (Santoso, 2011). Persamaan 2.8 bisa disebut juga dengan persamaan *Time Varying Inertia Weight* (TVIW).

$$v_{ij}(t) = \theta v_{ij}(t-1) + c_1 r_1 [x_{ij}^L - x_{ij}(t-1)] + c_2 r_2 [x^G - x_{ij}(t-1)] \quad (2.7)$$

$$\theta_t = \theta_{max} - \left(\frac{\theta_{max} - \theta_{min}}{t_{max}} \right) t \quad (2.8)$$

atau

$$\theta_t = \theta_{min} + (\theta_{max} - \theta_{min}) * \left(\frac{t_{max} - t}{t_{max}} \right)$$

Keterangan:

θ_{min} = nilai awal

θ_{max} = nilai akhir

biasanya digunakan $\theta_{min} = 0.4$ dan $\theta_{max} = 0.9$

t = iterasi sekarang

t_{max} = iterasi maksimum

Penentuan nilai θ dapat dilakukan dengan beberapa cara di antaranya adalah sebagai berikut (Huang, 2008):

1. Konstan

θ akan ditentukan menggunakan nilai antara 0-1.

2. Penurunan

Nilai awal θ dikurangi nilai akhir θ secara linear menggunakan Persamaan 2.8, biasanya digunakan nilai awal $\theta = 0.9$ dan nilai akhir $\theta = 0.4$.

3. Acak

θ ditentukan menggunakan Persamaan 2.9.

$$\theta = 0.5 - \left(\frac{rnd}{2.0} \right) \quad (2.9)$$

Keterangan:

rnd = nilai acak antara 0-1.

2.9.4 Pembatasan Kecepatan

Setiap nilai kecepatan yang didapatkan harus dipastikan berada dalam *range* $[-v_{max}, v_{max}]$, dimana v_{max} merupakan kecepatan maksimum yang diperbolehkan pada suatu dimensi. *Range* $[-v_{max}, v_{max}]$ bertujuan untuk mencegah partikel meninggalkan daerah pencarian. Jika kecepatan yang dihasilkan kurang dari $-v_{max}$ maka nilai kecepatan di kembalikan kepada $-v_{max}$, dan jika kecepatan melebihi v_{max} maka nilai kecepatan dikembalikan pada v_{max} . Pada umumnya, besar nilai v_{max} adalah 10-20% dari posisi terbesar yang bisa diambil pada suatu dimensi (Eberhart & Shi, 2001). Perhitungan v_{max} bisa dilakukan dengan menggunakan Persamaan 2.10.

$$v_{max} = (60\% * x_{max}) \quad (2.10)$$

2.9.5 Kondisi Berhenti

Kondisi berhenti dalam PSO merupakan syarat yang digunakan untuk mengakhiri iterasi pencarian. Beberapa syarat pemberhentian iterasi yang bisa digunakan dalam PSO adalah sebagai berikut:

1. Iterasi berhenti apabila iterasi telah mencapai maksimum iterasi.
2. Iterasi berhenti apabila ditemukan solusi yang telah memenuhi kriteria.
3. Iterasi berhenti apabila tidak ada perubahan nilai atau disebut dengan kondisi konvergen.
4. Iterasi berhenti apabila nilai radius *swarm* yang dinormalisasi mendekati 0.
5. Iterasi berhenti apabila grafik fungsi obyekif mendekati 0 seiring bertambahnya iterasi.

2.9.6 Time Varying Acceleration Coefficients (TVAC)

TVAC berguna untuk menyeimbangkan ruang pencarian antara *global exploration* dengan *local exploitation* dengan cara menentukan nilai c_1 menggunakan Persamaan 2.11 dan menentukan nilai c_2 menggunakan Persamaan 2.12. Dalam TVAC, nilai c_{1i} ke c_{1f} akan semakin menurun nilai dari c_{2i} ke c_{2f} akan semakin meningkat seiring bertambahnya iterasi. Dengan kondisi nilai komponen kognitif lebih besar dari komponen sosial di awal iterasi memungkinkan partikel-partikel untuk bergerak mengelilingi ruang pencarian daripada bergerak menuju nilai populasi terbaik, sebaliknya dengan nilai komponen kognitif lebih kecil dari komponen sosial pada iterasi akhir maka akan membuat partikel-partikel bertemu pada nilai global optimum pada proses terakhir optimasi.

$$c_1 = (c_{1f} - c_{1i}) * \frac{t}{t_{max}} + c_{1i} \quad (2.11)$$

$$c_2 = (c_{2f} - c_{2i}) * \frac{t}{t_{max}} + c_{2i} \quad (2.12)$$

2.10 Nilai Evaluasi

Evaluasi bertujuan untuk mengetahui seberapa jauh model peramalan itu mampu memprediksi data yang telah diketahui dengan cara melakukan evaluasi terhadap kesesuaian model peramalan terhadap suatu kumpulan data yang diberikan. Hasil peramalan yang akurat adalah peramalan yang mampu kesalahan meramal seminim mungkin. Beberapa persamaan yang dapat digunakan dalam menghitung kesalahan adalah:

1. *Mean Absolute Error* (MAE).

MAE adalah rata-rata *absolute* dari kesalahan meramal, tanpa menghiraukan tanda positif atau negatif.

$$MAE = \sum_{i=1}^n \frac{|e_i|}{n} \quad (2.13)$$

Keterangan:

$e_i = x_i - f_i$ = Kesalahan ramalan periode ke- i

x_i = Data aktual periode ke- i

f_i = Data hasil ramalan periode ke- i

2. *Mean Squared Error (MSE).*

MSE adalah rata-rata kesalahan meramal yang dikuadratkan.

$$MSE = \sum_{i=1}^n \frac{e_i^2}{n} \quad (2.14)$$

3. *Mean Absolute Percentage Error (MAPE).*

MAPE merupakan nilai tengah kesalahan persentase *absolute* dari suatu peramalan.

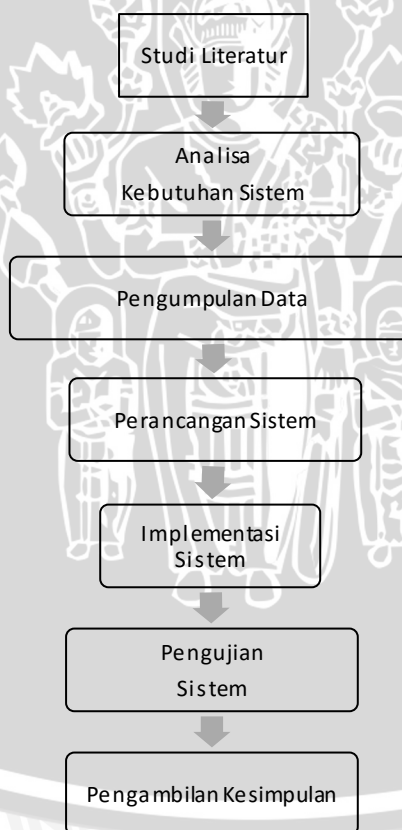
$$MAPE = \sum_{i=1}^n \frac{\frac{e_i}{x_i}}{n} \quad (2.15)$$

BAB 3 METODOLOGI

Bab metodologi menjelaskan metode penelitian dan langkah-langkah yang dilakukan dalam penelitian tentang optimasi pemerataan mutasi guru menggunakan algoritma PSO. Adapun tahapannya, yaitu:

1. Mempelajari literatur yang berhubungan dengan optimasi menggunakan algoritma PSO.
2. Melakukan pengumpulan data.
3. Melakukan perancangan sistem.
4. Mengimplementasi sistem berdasarkan perancangan yang telah dilakukan.
5. Melakukan uji coba terhadap sistem yang telah dibuat.
6. Menganalisis hasil optimasi.

Adapun tahapannya dapat dilihat seperti pada Gambar 3.1



Gambar 3.1 Metode Penelitian

3.1 Studi Literatur

Studi literatur adalah tahapan mencari dan mempelajari teori-teori yang berkaitan dengan topik penelitian penulis guna mempermudah dalam proses analisa, perancangan dan yang didapatkan. Teori-teori tersebut dapat berasal dari jurnal, *paper*, buku, dan *e-book*.

3.2 Analisa Kebutuhan Sistem

Analisa kebutuhan merupakan tahapan untuk menentukan kebutuhan-kebutuhan untuk membangun sistem. Kebutuhan untuk membangun sistem optimasi pemerataan mutasi guru menggunakan algoritma PSO adalah laptop dengan sistem operasi minimal Windows 7 64 bit dan RAM 4 GB, XAMPP server for Windows, Database MySQL, Netbeans IDE 7.3, data yang berkaitan dengan guru PNS mata pelajaran pada tingkat SMP yang berstatus PNS di Kabupaten Lumajang, Data kebutuhan guru PNS semua SMP di Kabupaten Lumajang yang berisi informasi jumlah ketersediaan, kekurangan dan kelebihan guru.

3.3 Pengumpulan Data

Pengumpulan data dilakukan untuk mendapatkan data yang dibutuhkan untuk implementasi sistem. Pada penelitian ini, pengumpulan data dilakukan dengan cara melakukan permohonan data skripsi ke Badan Kepegawaian Daerah (BKD) Kabupaten Lumajang dan Dinas Pendidikan Kabupaten Lumajang. Permohonan data skripsi ke BKD Kabupaten Lumajang bertujuan untuk mendapatkan data semua SMP Negeri di Kabupaten Lumajang yang berisi informasi jumlah kekurangan, ketersediaan, dan kelebihan guru, sedangkan permohonan data skripsi ke Dinas Pendidikan Kabupaten Lumajang bertujuan untuk mendapatkan data semua guru PNS tingkat SMP yang berisi identitas kepegawaian guru.

3.4 Perancangan Sistem

Perancangan merupakan tahapan untuk menggambarkan rancangan sistem yang akan dibangun. Perancangan dilakukan setelah semua kebutuhan sudah ditetapkan saat tahapan analisis kebutuhan. Perancangan sistem pada peniliti ini meliputi manualisasi, perancangan program yang digambarkan dengan menggunakan diagram alir, dan perancangan antar muka. Arsitektur perancangan sistem untuk optimasi pemerataan mutasi guru SMP di Kabupaten Lumajang ditunjukkan pada Gambar 3.2.

3.5 Implementasi Sistem

Implementasi dilakukan dengan mengacu pada perancangan sistem. Untuk membangun sistem optimasi pemerataan mutasi guru dengan menggunakan bahasa pemrograman Java dan *database* MySQL, maka langkah-langkah yang dilakukan adalah sebagai berikut:

1. Menerapkan Algoritma PSO kedalam *code* program menggunakan Netbeans IDE 7.3.

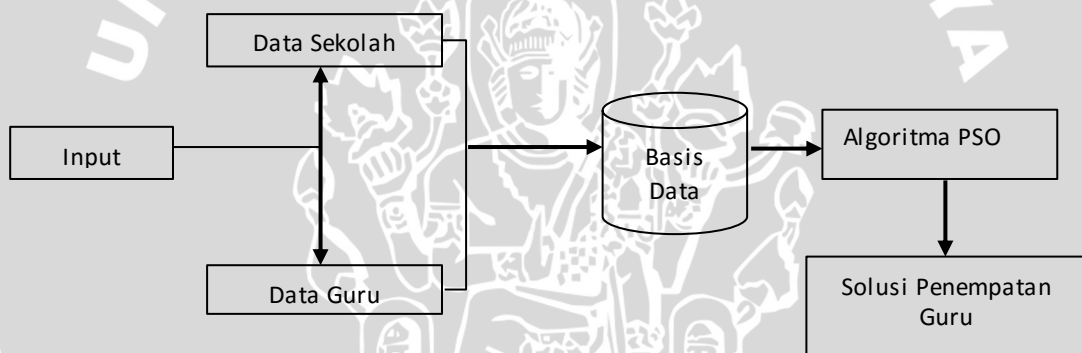
2. Membuat antarmuka.
3. Membuat *database*.

3.6 Pengujian Sistem

Pengujian sistem yang dilakukan dalam penelitian ini merupakan tahapan yang bertujuan untuk mengetahui apakah algoritma PSO yang diimplementasikan dalam program sudah benar dan bertujuan untuk mengetahui nilai terbaik dari parameter-parameter PSO seperti menguji berapa banyak iterasi dan banyak partikel yang paling optimal.

3.7 Kesimpulan dan Saran

Kesimpulan bertujuan untuk menjelaskan secara singkat mengenai sistem yang dibangun dan hasil pengujian sistem yang diperoleh. Kesimpulan diambil untuk menjawab rumusan masalah yang telah ditentukan. Saran ditujukan untuk pengembangan penelitian ke depannya agar menjadi lebih baik.



Gambar 3.2 Arsitektur Perancangan Sistem

BAB 4 PERANCANGAN

Pada bab ini membahas mengenai perancangan dan contoh perhitungan manual. Perancangan yang dilakukan meliputi perancangan program dan perancangan antarmuka. Perancangan program digambarkan dalam bentuk diagram alir setiap proses PSO.

4.1 Deskripsi Sistem

Sistem yang dibangun merupakan sistem yang digunakan untuk melakukan optimasi pemerataan jumlah guru SMP PNS di wilayah kabupaten Lumajang dengan menggunakan algoritma PSO. Pemerataan guru didasarkan pada data formasi kebutuhan guru setiap sekolah yang berisi keterangan jumlah kekurangan dan kelebihan guru untuk masing-masing mata pelajaran sehingga guru dari sekolah yang kelebihan dipindahkan ke sekolah yang kekurangan sesuai dengan ketentuan mutasi yang berlaku di Kabupaten Lumajang. Kriteria guru yang diprioritaskan untuk dimutasi adalah sebagai berikut:

1. Guru dengan masa kerja terlama, terhitung dari mulai mengajar di sekolah yang bersangkutan. Batas minimal masa kerja yang diberlakukan untuk mutasi adalah 5 tahun.
2. Guru yang memiliki akses terdekat ke sekolah tujuan mutasi di antara guru yang lainnya.
3. Guru yang tidak berstatus sebagai kepala sekolah.

Fungsi obyektif yang digunakan pada penelitian ini adalah *fitness* (maksimasi nilai) seperti yang dituliskan pada Persamaan 4.1

$$Fitness = mk + \left(\frac{1000}{u + j} \right) \quad (4.1)$$

Keterangan:

mk = masa kerja

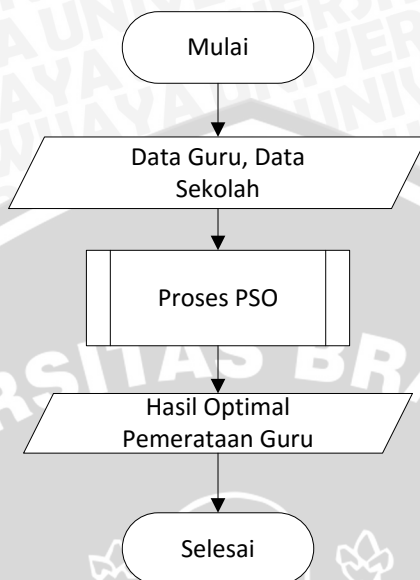
u = usia

j = jarak

Persamaan 4.1 berfungsi untuk mencari masa kerja guru sebesar mungkin dengan jarak dan usia sekecil mungkin sesuai dengan kriteria prioritas yang telah ditentukan. Sedangkan untuk nilai 1000 merupakan pembobot untuk menyeimbangkan antara nilai pinalti (usia dan jarak) dengan besar masa kerja. Penentuan pembobot dengan nilai 1000 ditentukan agar lebih besar dari hasil jumlah antara nilai usia guru paling tua dan jarak terjauh. Jarak terjauh bisa dilihat jarak dari ujung ke ujung wilayah Kabupaten Lumajang dan jarak dari kabupaten-kabupaten lain sekitar kabupaten Lumajang ke Kabupaten Lumajang.

4.2 Alir Perancangan Sistem

Alir perancangan sistem optimasi pemerataan guru mata pelajaran dengan menggunakan algoritma PSO ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram Alir Perancangan Sistem

4.2.1 Proses PSO

Perhitungan menggunakan algoritma PSO untuk sistem optimasi pemerataan guru mata pelajaran mempunyai proses-proses sebagai berikut :

1. Inisialisasi partikel.
2. *Update* kecepatan.
3. *Update* posisi.
4. Perhitungan *fitness*.
5. *Update* P_{best} .
6. *Update* G_{best} .

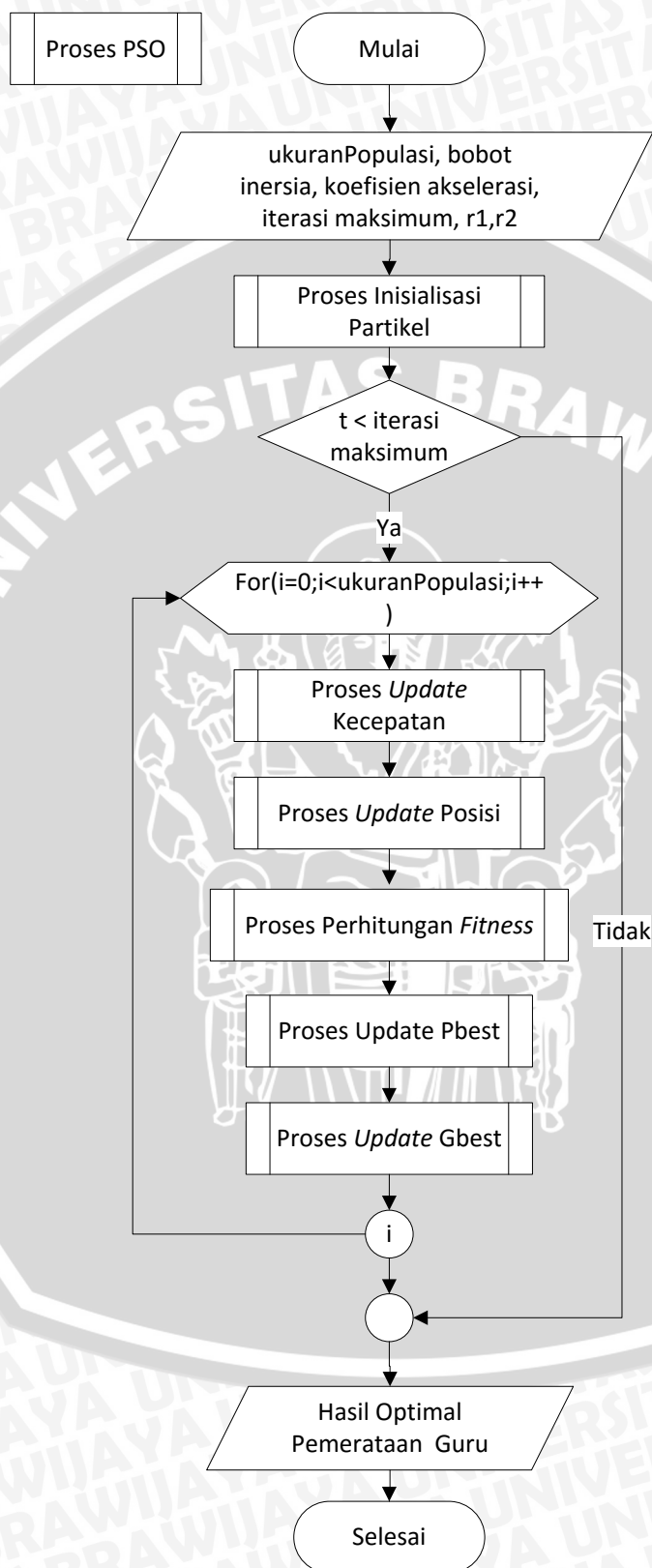
Diagram Alir proses PSO ditunjukkan pada Gambar 4.2.

4.2.2 Proses Inisialisasi Partikel

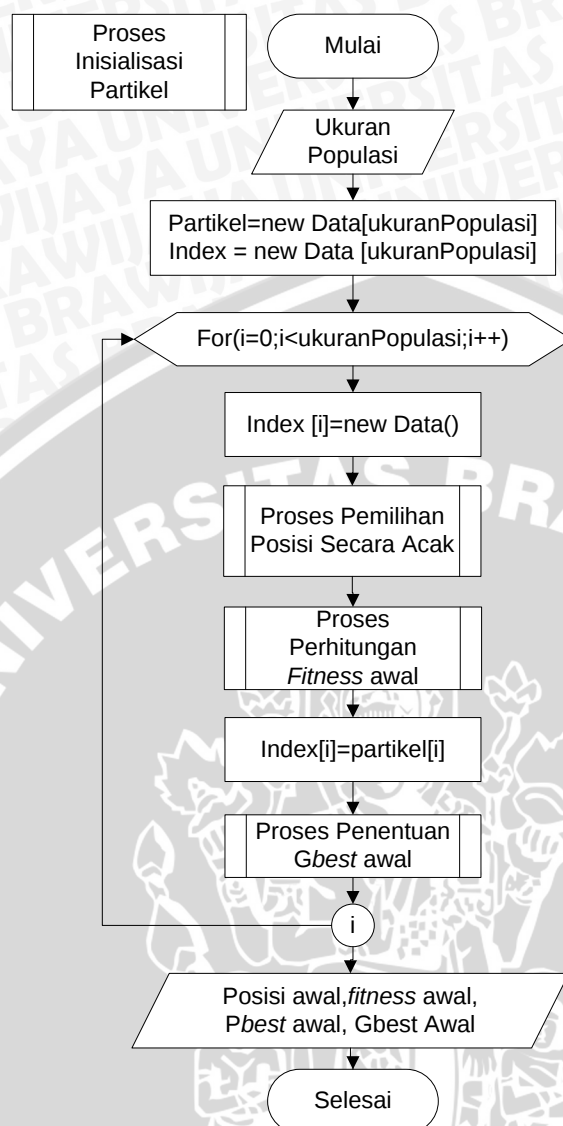
Proses Inisialisasi partikel merupakan tahap awal dalam proses PSO. Pada sistem optimasi pemerataan guru mata pelajaran, Inisialisasi partikel bertujuan untuk menentukan beberapa hal sebagai berikut:

1. Posisi awal secara acak seluruh partikel dalam populasi.
2. Nilai *fitness* awal.
3. P_{best} awal.
4. G_{best} awal.

Diagram alir pada proses inialisasi partikel dapat dilihat pada Gambar 4.3.



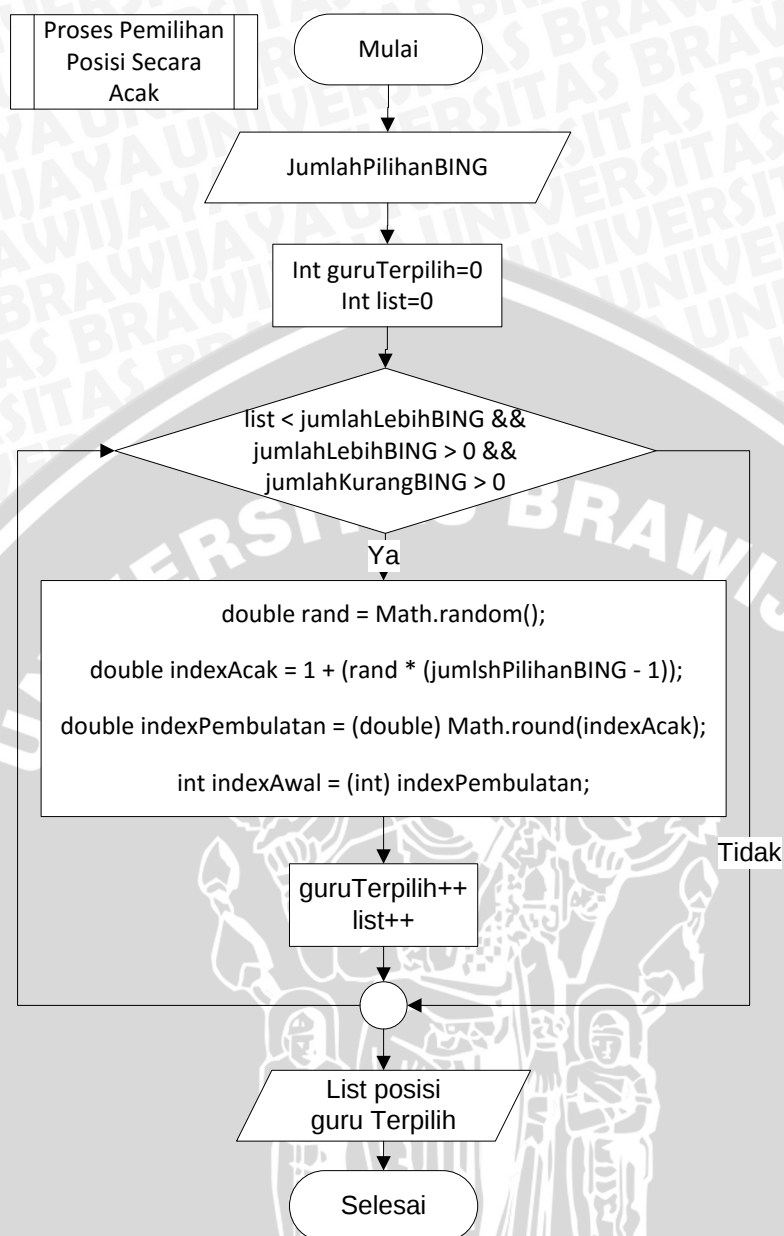
Gambar 4.2 Diagram Alir Proses PSO



Gambar 4.3 Proses Inisialisasi Partikel

4.2.3 Proses Pemilihan Posisi Secara Acak

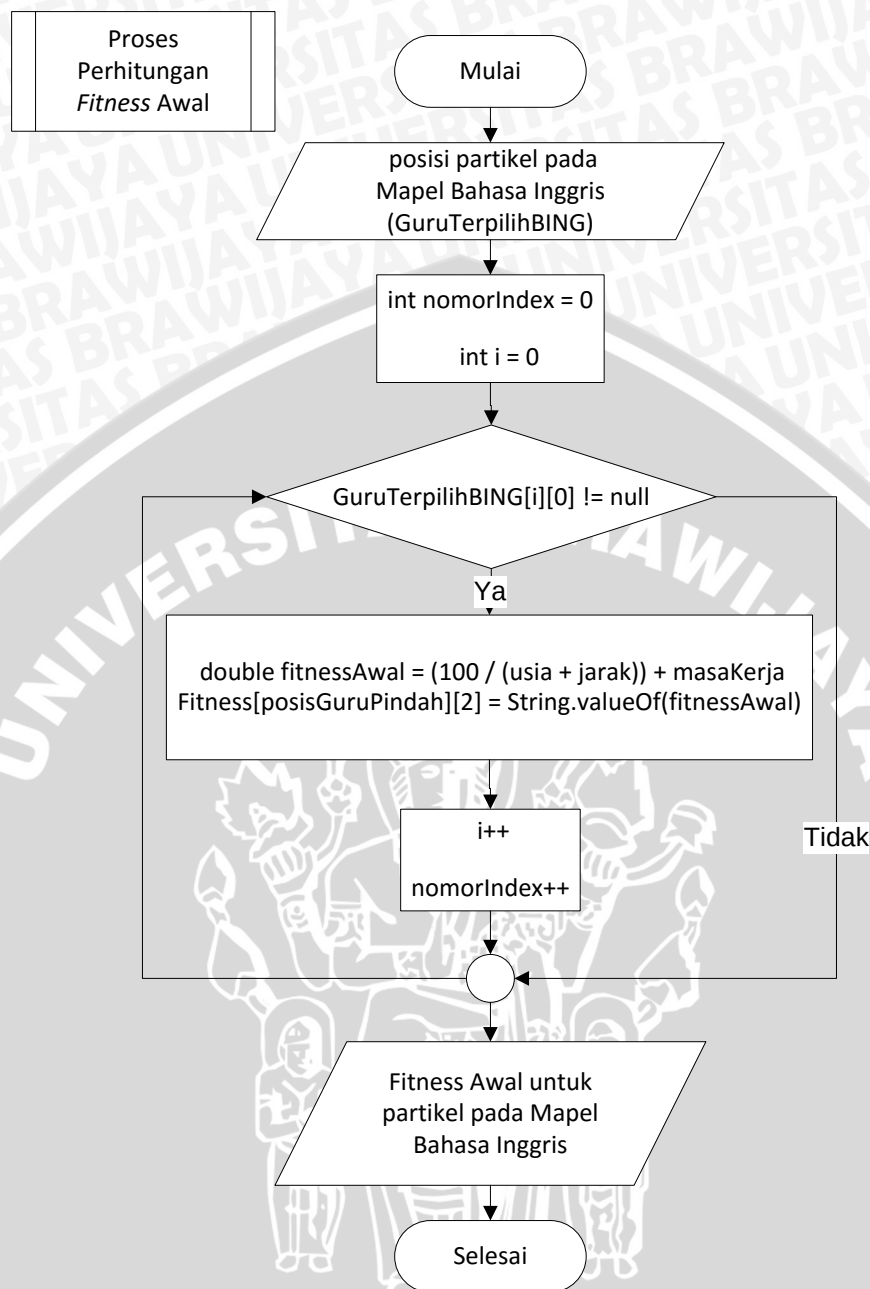
Pemilihan secara acak posisi dilakukan dengan menggunakan Persamaan 2.6. nilai random untuk menentukan posisi partikel adalah nilai random dengan *range* 0-1. Pada implementasi PSO untuk sistem optimasi pemerataan guru mata pelajaran, pemilihan posisi secara acak dilakukan terhadap partikel sesuai dengan mata pelajarannya. Diagram alir proses penentuan posisi partikel secara acak dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Alir Proses Pemilihan Posisi Secara Acak

4.2.4 Proses Perhitungan *Fitness* Awal

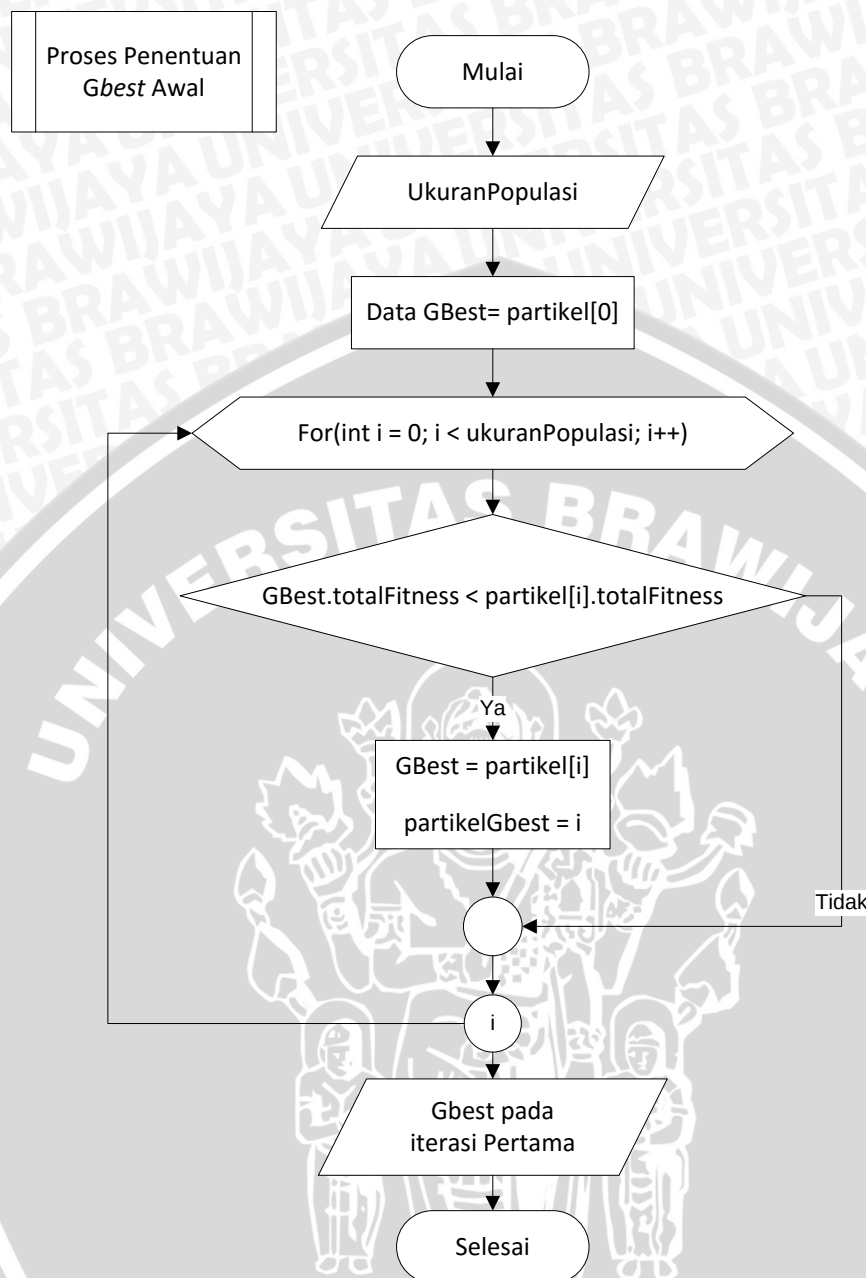
Proses perhitungan *fitness* awal bertujuan untuk mengetahui nilai *fitness* dari posisi awal yang telah didapatkan. Pada implementasi PSO untuk sistem optimasi pemerataan guru mata pelajaran, perhitungan *fitness* dilakukan terhadap partikel sesuai dengan mata pelajarannya. Diagram alir proses perhitungan *fitness* awal ditunjukkan pada Gambar 4.5.



Gambar 4.5 Diagram Alir Proses Perhitungan *Fitness Awal*

4.2.5 Proses Penentuan G_{best} Awal

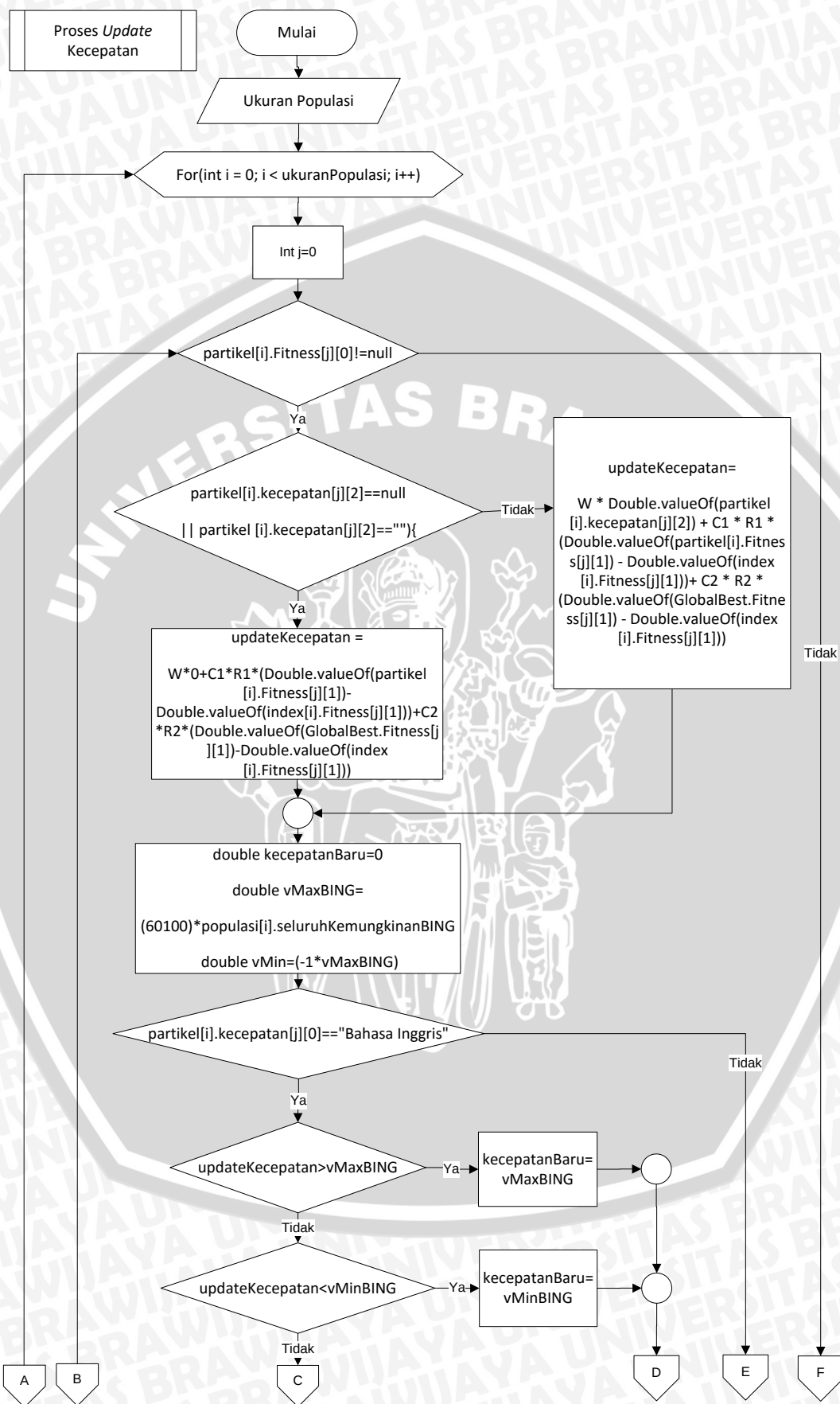
Penentuan G_{best} awal bertujuan untuk mencari partikel terbaik pada iterasi pertama. Diagram alir proses penentuan G_{best} awal ditunjukkan pada Gambar 4.6.

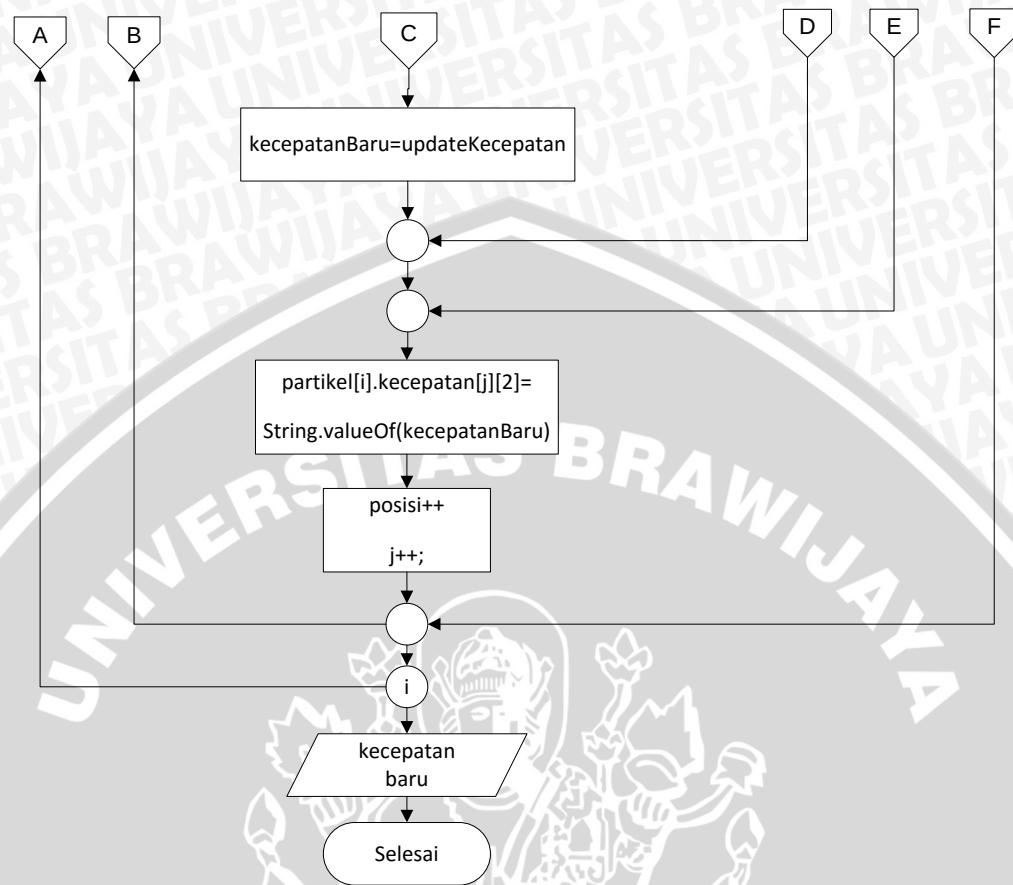


Gambar 4.6 Diagram Alir Proses Penentuan G_{best} Awal

4.2.6 Proses *Update* Kecepatan

Kecepatan untuk setiap partikel akan di-*update* menggunakan Persamaan 2.7. Kecepatan tiap partikel harus dipastikan dalam *range* $[-v_{max}, v_{max}]$ dimana v_{max} adalah kecepatan maksimum sedangkan $-v_{max}$ merupakan kecepatan minimum yang diperbolehkan pada suatu partikel. Nilai v_{max} yang digunakan merupakan 60% dari x_{max} . Diagram alir proses *update* kecepatan dapat dilihat pada Gambar 4.7.

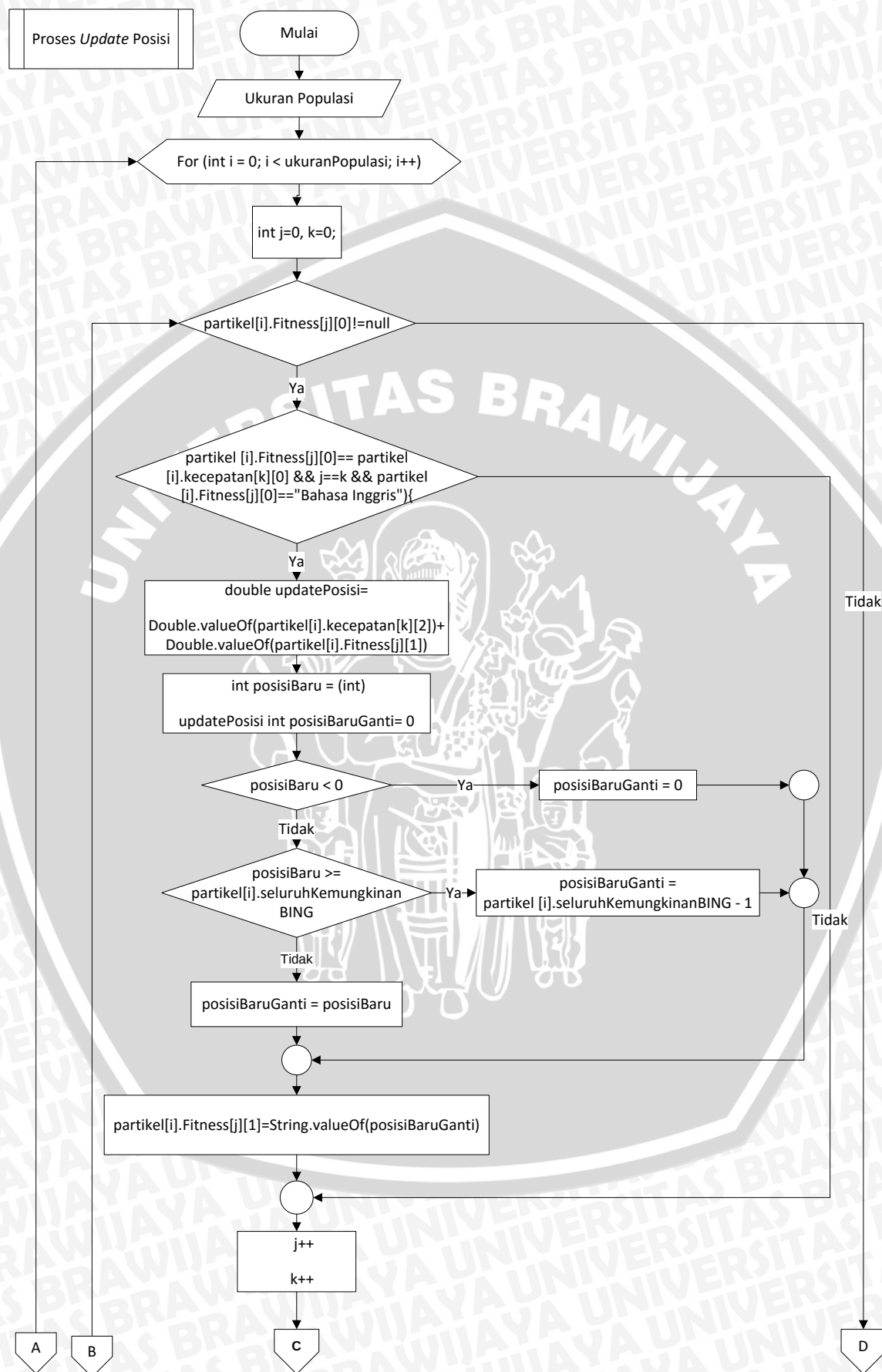


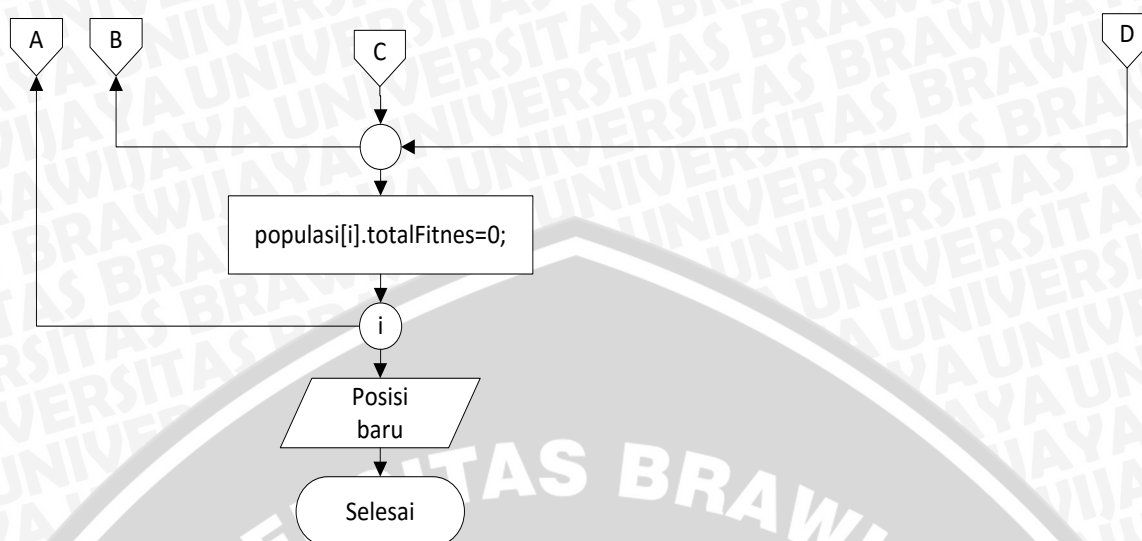


Gambar 4.7 Diagram Alir Proses Update Kecepatan

4.2.7 Proses Update Posisi

Posisi partikel akan di-*update* per iterasinya menggunakan Persamaan 2.6. Posisi baru yang didapatkan harus dipastikan dalam *range* $[x_{min}, x_{max}]$. Posisi baru partikel tidak boleh kurang dari posisi minimum (x_{min}) atau lebih besar dari posisi maksimum (x_{max}) yang diperbolehkan pada mata pelajaran tertentu. Jika posisi baru yang didapatkan lebih besar dari x_{max} maka posisi baru akan dikembalikan pada x_{max} , sebaliknya jika posisi baru yang didapatkan lebih kecil dari x_{min} maka posisi baru akan dikembalikan pada x_{min} . Diagram alir proses *update* posisi dapat dilihat pada Gambar 4.8.

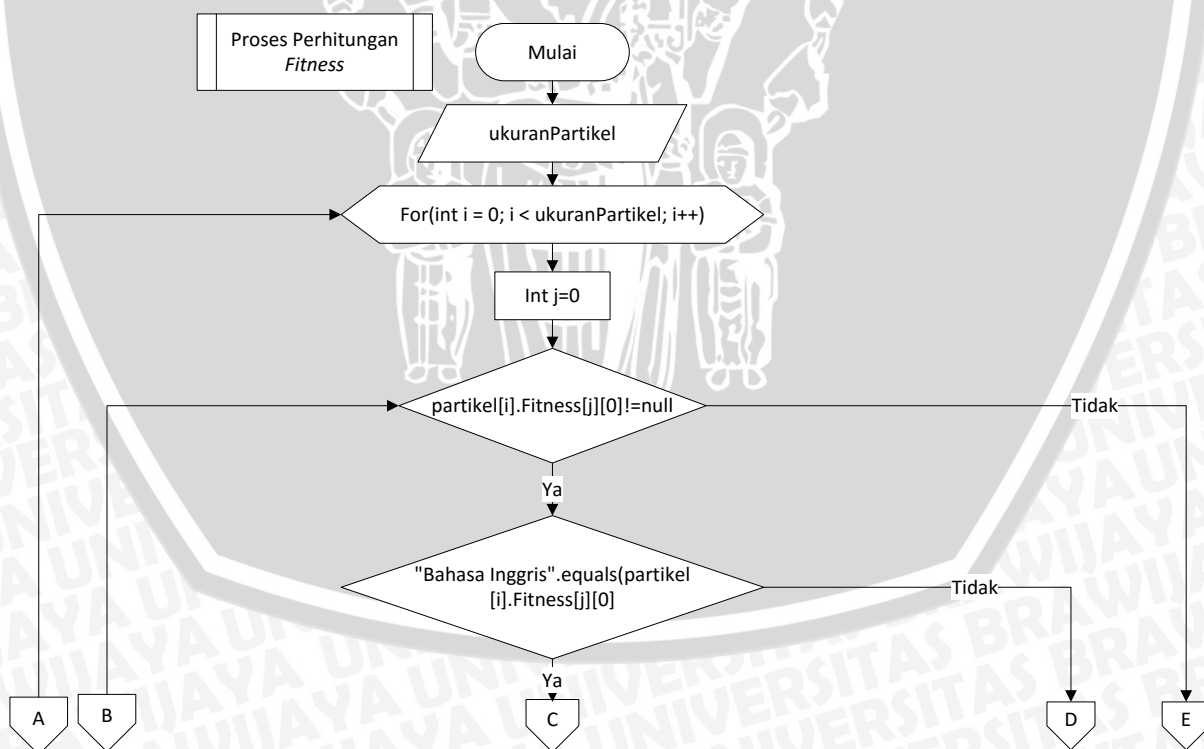


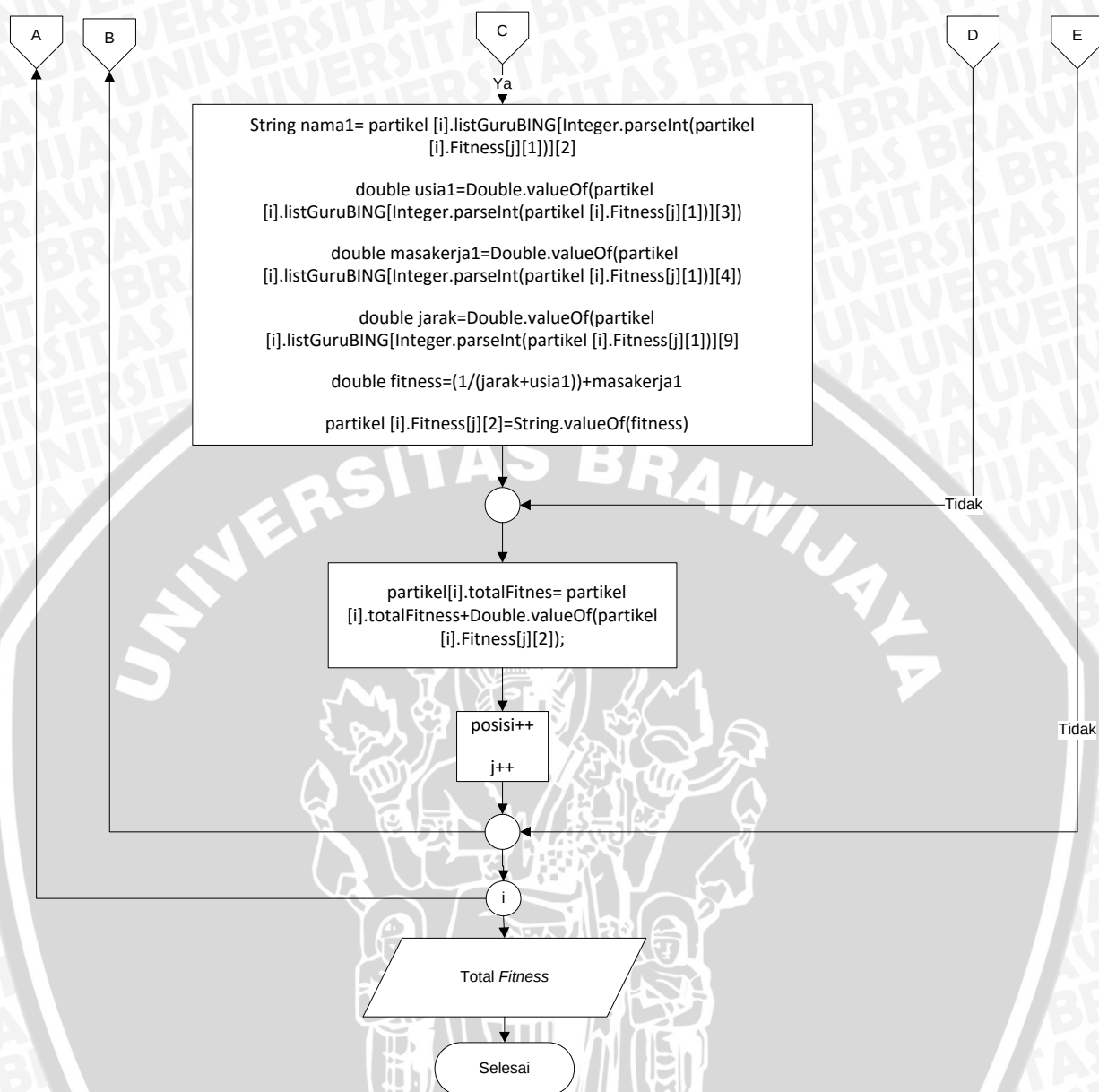


Gambar 4.8 Diagram Alir Proses *Update* Posisi

4.2.8 Proses Perhitungan *Fitness*

Perhitungan *fitness* merupakan proses untuk mencari nilai *fitness* dari posisi baru partikel. Diagram alir proses perhitungan *fitness* dapat dilihat pada Gambar 4.9.

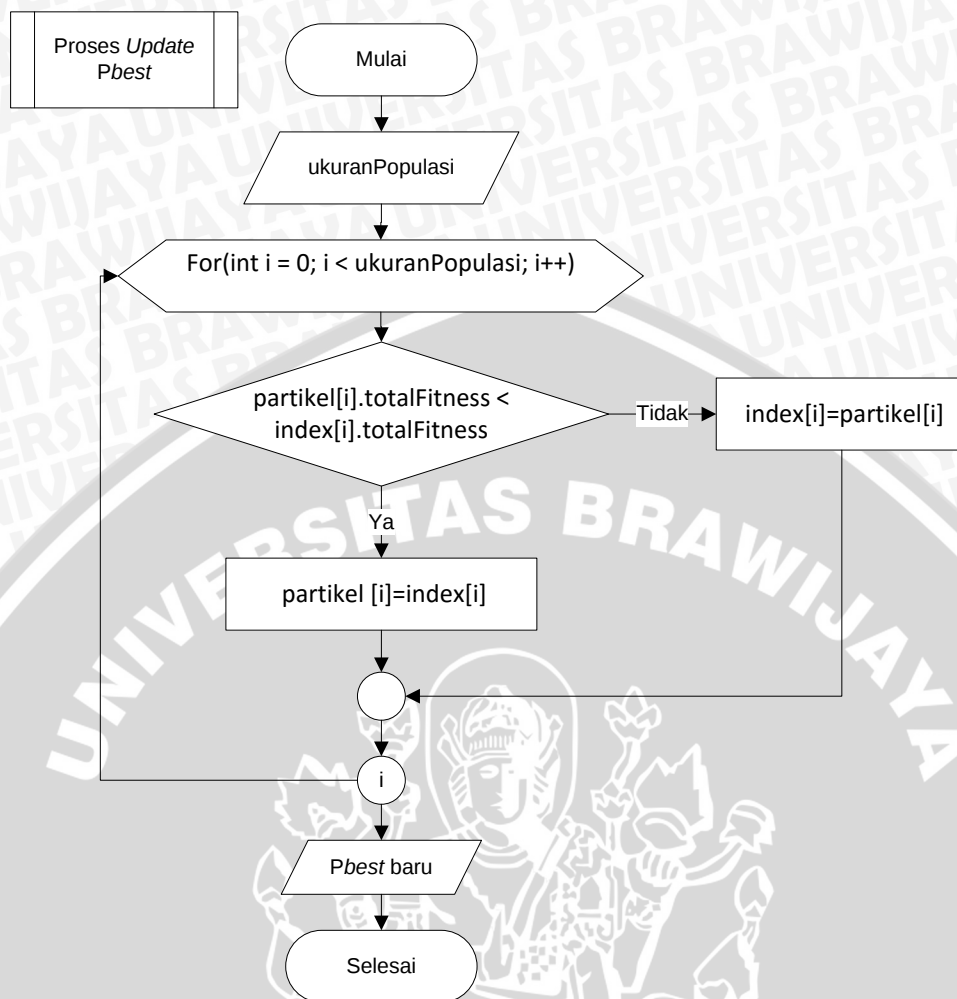




Gambar 4.9 Diagram Alir Proses Perhitungan *Fitness*

4.2.9 Proses *Update P_{best}*

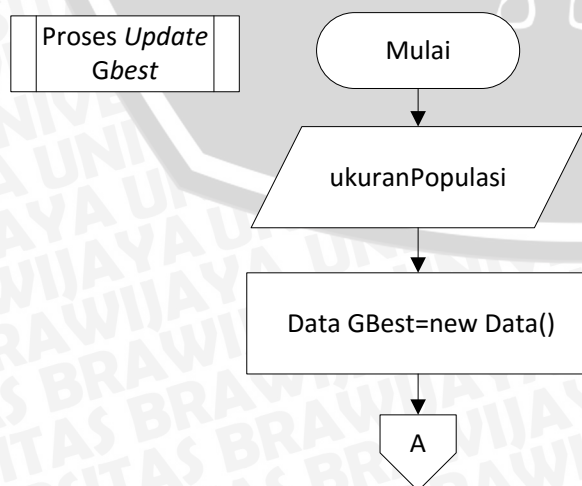
Proses *update P_{best}* bertujuan untuk menentukan *P_{best}* dari populasi yang baru. Diagram alir proses *update P_{best}* ditunjukkan pada Gambar 4.10.

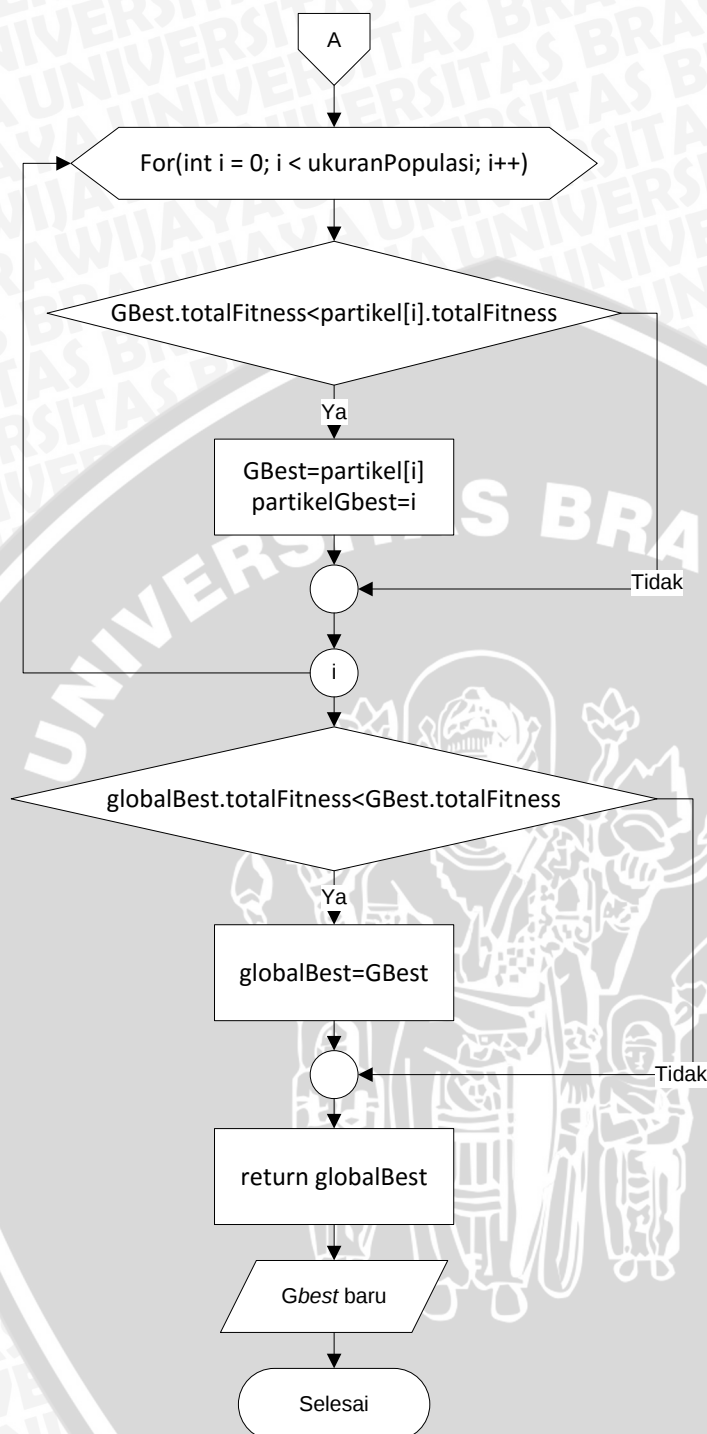


Gambar 4.10 Diagram Alir Proses Update P_{best}

4.2.10 Proses Update G_{best}

Update G_{best} merupakan proses yang bertujuan untuk mencari partikel terbaik yaitu partikel yang memiliki nilai total *fitness* terbesar yang ditemukan sampai iterasi ke- t . Diagram alir proses *update* G_{best} ditunjukkan pada Gambar 4.11.





Gambar 4.11 Diagram Alir Proses *Update Gbest*

4.3 Manualisasi

Manualisasi bertujuan untuk menerapkan algoritma PSO ke dalam kasus sederhana dan menjelaskan perhitungan manual setiap tahapan PSO. Kasus sederhana yang digunakan untuk manualisasi adalah optimasi mutasi pemerataan guru untuk 4 mata pelajaran yaitu Ilmu Pengetahuan Sosial (IPS), Matematika (MTK), Teknologi Informasi dan Komputer (TIK), dan Bahasa Daerah (BD) pada 3

sekolah yaitu SMPN 2, SMPN 5, dan SMPN 6. Masing-masing sekolah memiliki data kelebihan dan kekurangan guru mata pelajaran seperti yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Formasi Kebutuhan Guru

No	Sekolah	Mapel	Kurang	Lebih	Ketersediaan Guru
1	SMP 2	IPS	1	0	3
		MTK	0	1	3
		TIK	1	0	0
		BD	1	0	0
2	SMP 5	IPS	0	1	4
		MTK	0	2	4
		TIK	1	0	0
		BD	0	2	4
3	SMP 6	IPS	0	1	4
		MTK	1	0	2
		TIK	0	2	4
		BD	2	0	0

Sebelum melakukan perhitungan menggunakan algoritma PSO, ditentukan nilai awal parameter-parameter PSO, seperti menentukan maksimum iterasi, ukuran populasi, nilai bobot inersia, nilai koefisien akselerasi, nilai r_1 dan r_2 . Detail nilai parameter-parameter PSO yang digunakan dalam kasus ini adalah sebagai berikut:

Maksimum iterasi = 3
 Ukuran populasi = 4
 $\theta_{min}, \theta_{max}$ = 0.4, 0.9
 c_{1i}, c_{1f} = 2.5, 0.5
 c_{2i}, c_{2f} = 0.5, 2.5
 r_1, r_2 = 0.5, 0.5

Berikut adalah tahapan-tahapan berdasarkan algoritma PSO:

ITERASI 0

1. Inisialisasi Partikel

Inisialisasi partikel digunakan untuk menentukan posisi awal setiap partikel secara acak. Dalam kasus ini, sebelum melakukan inisialisasi partikel, ada beberapa tahapan yang dilakukan yaitu sebagai berikut:

- Menghitung Banyak Pilihan Mutasi Guru Per Mata Pelajaran

Banyaknya pilihan diperoleh dari jumlah sedia guru dari sekolah yang kelebihan dikali jumlah sekolah yang kekurangan. Pilihan tersebut berisi informasi terkait nama usia, masa kerja guru mata pelajaran tertentu dari sekolah yang kelebihan beserta jarak dari tempat tinggal guru ke setiap sekolah

tujuan mutasi yang ada. Misalkan berdasarkan Tabel 4.1, sekolah yang kekurangan guru IPS adalah SMP 2 sebanyak 1 guru, sedangkan sekolah yang kelebihan mata pelajaran IPS adalah SMP 5 dan SMP 6 dimana jumlah ketersediaan di SMP 5 adalah 4 dan di SMP 6 juga 4. Sehingga jumlah guru pindah yang bisa dipilih untuk mengisi 1 kekurangan guru IPS di SMP 2 adalah sebanyak 8 pilihan seperti yang ditunjukkan pada Tabel 4.2. Untuk jumlah pilihan mata pelajaran lainnya bisa dilihat pada Lampiran 1.

Tabel 4.2 Pilihan Guru Mata Pelajaran IPS

PILIHAN	GURU	USIA	MASA KERJA (thn)	JARAK (km)
				SMP 2
1	GURU 1 SMP 5	51	5	4
2	GURU 2 SMP 5	49	8	6
3	GURU 3 SMP 5	34	4	3
4	GURU 4 SMP 5	37	8	7
5	GURU 1 SMP 6	42	9	7
6	GURU 2 SMP 6	32	7	1.5
7	GURU 3 SMP 6	53	3	2
8	GURU 4 SMP 6	44	5	2

- Menentukan Banyak Dimensi Setiap Partikel

Menentukan banyak dimensi setiap partikel bertujuan untuk menentukan banyak dimensi yang akan mengisi 1 partikel. Pada kasus ini, banyak dimensi pada 1 partikel disesuaikan dengan banyak guru yang bisa dimutasi. Apabila kekurangan guru lebih banyak daripada kelebihan guru, maka banyak guru yang bisa dimutasi hanya sebanyak kelebihan yang ada, sedangkan jika kekurangan guru tidak sebanyak kelebihan guru, maka banyak guru yang bisa dimutasi hanya sebanyak kekurangan yang ada, seperti yang ditunjukkan pada Tabel 4.3.

Tabel 4.3 Banyak Pilihan Mutasi Pada Mata Pelajaran IPS

Mata Pelajaran	Kurang	Lebih	Banyak mutasi
Ilmu Pengetahuan Sosial	1	2	1
Matematika	1	1	1
Teknologi Informasi & Komputer	2	2	2
Bahasa Daerah	2	2	2
TOTAL	6	6	6

Sehingga, berdasarkan Tabel 4.3, maka satu partikel akan terdiri dari 6 dimensi.

Setelah proses menghitung banyak pilihan mutasi guru per mata pelajaran dan menentukan banyak dimensi setiap partikel maka tahap inisialisasi partikel dapat dilakukan. Tahapan inisialisasi partikel dilakukan hingga semua partikel pada

semua populasi telah mendapatkan posisi awalnya. Hasil dari proses inialisai pada kasus ini dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Inialisasi Partikel

Partikel ke- i	Mata Pelajaran	Nilai r	Dimensi ke- j	Posisi Awal	Kecepatan Awal
1	IPS	0.4	1	4	0
	MTK	0.4	2	3	0
	TIK	0.9	3	7	0
	TIK	0.2	4	2	0
	BD	1	5	8	0
	BD	0.1	6	2	0
2	IPS	0.3	1	3	0
	MTK	0.3	2	3	0
	TIK	0.3	3	3	0
	TIK	0.7	4	6	0
	BD	0.8	5	7	0
	BD	0.4	6	4	0
3	IPS	0.7	1	5	0
	MTK	0.7	2	4	0
	TIK	1	3	8	0
	TIK	0	4	1	0
	BD	0.6	5	5	0
	BD	0.9	6	7	0
4	IPS	0.1	1	2	0
	MTK	0.1	2	2	0
	TIK	0.5	3	5	0
	TIK	0.3	4	3	0
	BD	0	5	1	0
	BD	1	6	8	0

Berdasarkan Tabel 4.4, posisi dimensi ke-1 pada partikel ke-1 adalah 4, dimana posisi 4 menunjukan guru yang terpilih untuk di mutasi ke sekolah yang kekurangan guru IPS adalah adalah GURU 4 SMP 5 yang berarti guru IPS ke-4 dari SMP 5 seperti yang ditunjukkan Tabel 4.2.

Posisi dimensi ke-1 pada partikel ke-1 diperoleh dengan menggunakan persamaan 2.6. sehingga perhitungannya diperoleh sebagai berikut:

$$\begin{aligned}x_{i,j} &= 1 + 0.4 * (8 - 1) \\&= 1 + 0.4 * 7 \\&= 3.8 \approx 4\end{aligned}$$

Dimana 1 adalah x_{min} untuk mata pelajaran IPS dan 8 x_{max} untuk mata pelajaran IPS seperti yang ditunjukkan Tabel 4.2 bahwa jumlah pilihan untuk mata pelajaran IPS ada sebanyak 8 pilihan sehingga pilihan terkecil yang bisa diambil adalah pilihan ke-1 dan pilihan terbesar adalah pilihan ke-8. Sedangkan nilai 0.4 adalah nilai *random* yang didapatkan untuk menentukan posisi seperti yang ditunjukkan pada Tabel 4.4.

2. Menentukan kecepatan Awal

Kecepatan setiap partikel pada iterasi pertama akan bernilai 0 seperti yang ditunjukkan pada Tabel 4.4.

3. Perhitungan Nilai *Fitness*

Setiap posisi akan dihitung nilai *fitness* nya menggunakan Persamaan 4.1. Misalkan pada Tabel 4.4 posisi dimensi ke-1 pada partikel ke-1 berisi pilihan ke -4, dimana berdasarkan Tabel 4.2 pilihan ke-4 adalah guru yang memiliki masa kerja selama 8, usia 37 tahun, dan jarak tempat tinggal ke SMP 2 adalah 7 km, maka perhitungan *fitness* nya adalah seperti berikut:

$$\begin{aligned}fitness &= 8 + \left(\frac{100}{37 + 7} \right) \\&= 8 + 0.0227 \\&= 10.27273\end{aligned}$$

Kemudian setelah nilai *fitness* setiap dimensi dihitung, maka akan dijumlahkan, sehingga setiap partikel akan mempunyai 1 nilai total *fitness* seperti yang ditunjukkan pada Tabel 4.5.

Tabel 4.5 Hasil Perhitungan *Fitness* dan Total *Fitness* Pada Iterasi 0

Partikel ke- <i>i</i>	Mata Pelajaran	Dimensi ke- <i>j</i>	Posisi	<i>Fitness</i>	Total <i>Fitness</i>
1	IPS	1	4	10.27273	54.35447
	MTK	2	3	4.380952	
	TIK	3	7	11.7027	
	TIK	4	2	12.32558	
	BD	5	8	4.785714	
	BD	6	2	10.88679	

Partikel ke- i	Mata Pelajaran	Dimensi ke- j	Posisi	<i>Fitness</i>	Total <i>Fitness</i>
2	IPS	1	3	6.702703	51.29008
	MTK	2	3	4.380952	
	TIK	3	3	15.89655	
	TIK	4	6	12.5	
	BD	5	7	6.886792	
	BD	6	4	4.923077	
3	IPS	1	5	11.04082	52.12663
	MTK	2	4	6.020202	
	TIK	3	8	11.7027	
	TIK	4	1	9.325581	
	BD	5	5	7.150538	
	BD	6	7	6.886792	
4	IPS	1	2	9.818182	67.75307
	MTK	2	2	11.92398	
	TIK	3	5	9.173913	
	TIK	4	3	11.66667	
	BD	5	1	20.38462	
	BD	6	8	4.785714	

4. Menentukan P_{best} dan G_{best}

P_{best} adalah posisi terbaik yang pernah dicapai partikel sampai iterasi ke- i dan G_{best} adalah posisi terbaik untuk semua partikel yang ditemukan sampai iterasi ke- i . Pada kasus ini, G_{best} menunjukkan partikel yang memiliki nilai total *fitness* terbesar.

P_{best} setiap partikel pada iterasi ke-0 sama dengan posisi awal karena iterasi ke-0 merupakan iterasi pertama. Berdasarkan Tabel 4.5, partikel yang memiliki nilai total *fitness* terbesar adalah partikel ke-4 yaitu sebesar 67.75307, sehingga G_{best} pada iterasi ke 0 adalah partikel ke-4. P_{best} dan G_{best} pada iterasi ke-0 dapat dilihat pada Tabel 4.6.

Tabel 4.6 P_{best} dan G_{best} Iterasi 0

Partikel ke- i	Mata Pelajaran	Dimensi ke- j	P_{best}	$Fitness$	Total $Fitness$
1	IPS	1	4	10.27273	54.35447
	MTK	2	3	4.380952	
	TIK	3	7	11.7027	
	TIK	4	2	12.32558	
	BD	5	8	4.785714	
	BD	6	2	10.88679	
2	IPS	1	3	6.702703	51.29008
	MTK	2	3	4.380952	
	TIK	3	3	15.89655	
	TIK	4	6	12.5	
	BD	5	7	6.886792	
	BD	6	4	4.923077	
3	IPS	1	5	11.04082	52.12663
	MTK	2	4	6.020202	
	TIK	3	8	11.7027	
	TIK	4	1	9.325581	
	BD	5	5	7.150538	
	BD	6	7	6.886792	
G_{best} 4	IPS	1	2	9.818182	67.75307
	MTK	2	2	11.92398	
	TIK	3	5	9.173913	
	TIK	4	3	11.66667	
	BD	5	1	20.38462	
	BD	6	8	4.785714	

5. Naikan Iterasi

ITERASI 1

6. Update Kecepatan

Ada beberapa hal yang harus dilakukan sebelum menghitung kecepatan, yakni:

- Menghitung nilai Bobot inersia

Bobot inersia dihitung menggunakan Persamaan 2.8. Pada kasus iterasi dilakukan sebanyak 3 kali. Jadi, jika menghitung nilai θ pada iterasi 1 maka perhitungannya diperoleh sebagai berikut:

$$\begin{aligned}\theta_1 &= 0.4 + (0.9 - 0.4) * \left(\frac{3 - 1}{3}\right) \\ &= 0.7333\end{aligned}$$

- Menghitung nilai c_1 dan c_2

Nilai c_1 dihitung menggunakan Persamaan 2.11 dan c_2 dihitung menggunakan Persamaan 2.12 nilai $c_{1i} = 2.5, c_{1f} = 0.5, c_{2i} = 0.5, c_{2f} = 2.5$. Dengan maksimum iterasi sebanyak 3, nilai c_1 dan c_2 pada iterasi 1 diperoleh sebagai berikut:

$$c_1 = (0.5 - 2.5) * \left(\frac{1}{3}\right) + 2.5 = 1.8333$$

$$c_2 = (2.5 - 0.5) * \left(\frac{1}{3}\right) + 0.5 = 1.1667$$

Nilai bobot inersia, c_1 , dan c_2 yang telah dihitung digunakan untuk melakukan *update* kecepatan dengan Persamaan 2.7. Misalkan untuk menghitung kecepatan baru dari dimensi ke-1 pada partikel ke-1 dengan nilai r_1 dan r_2 yang telah ditetapkan sebesar 0.5, maka perhitungannya sebagai berikut:

$$\begin{aligned}v_{1,1}(1) &= (0.7333 * 0) + (1.8333 * 0.5)(4 - 4) + (1.1667 * 0.5)(2 - 4) \\ &= -1.225\end{aligned}$$

Nilai G_{best} yang digunakan pada *update* kecepatan dimensi ke-1 pada partikel ke-1 di iterasi ke-1 adalah 2, dimana 2 adalah dimensi ke-1 pada partikel ke-4 (Partikel G_{best}) yang dapat dilihat pada Tabel 4.6. Hasil *update* kecepatan untuk setiap partikel pada iterasi ke-1 ditunjukkan pada Tabel 4.7.

Tabel 4.7 Hasil *Update* Kecepatan Iterasi ke-1

Partikel ke-i	Mata Pelajaran	Dimensi ke-j	Kecepatan
1	IPS	1	-1.225
	MTK	2	-1.05
	TIK	3	-1.63333
	TIK	4	0.408333
	BD	5	-4.08333
	BD	6	3.675
2	IPS	1	-0.81667
	MTK	2	-0.7
	TIK	3	0.816667
	TIK	4	-1.63333
	BD	5	-3.26667
	BD	6	2.45

Partikel ke-i	Mata Pelajaran	Dimensi ke-j	Kecepatan
3	IPS	1	-1.86667
	MTK	2	-1.51667
	TIK	3	-2.04167
	TIK	4	1.225
	BD	5	-2.45
	BD	6	0.408333
4	IPS	1	0
	MTK	2	0
	TIK	3	0
	TIK	4	0
	BD	5	0
	BD	6	0

Kemudian dilakukan pembatasan kecepatan terhadap hasil *update* kecepatan untuk memastikan kecepatan yang diperoleh masih dalam batas kecepatan yang diperbolehkan. Sebagai contoh, berdasarkan Tabel 4.7 untuk dimensi ke-1 pada partikel ke-1 mempresentasikan kekurangan untuk mata Pelajaran IPS. Sehingga dengan menggunakan Persamaan 2.10, kecepatan maksimum yang bisa digunakan oleh dimensi ke-1 adalah sebagai berikut:

$$v_{max} = \left(\frac{60}{100} \times 8 \right) = 4.8$$

Nilai 8 merupakan x_{max} atau pilihan terbesar yang ada pada mata pelajaran IPS. Jadi, berdasarkan Tabel 4.7 kecepatan dari dimensi ke-1 pada partikel ke-1 masih dalam *range* $[-4.8 ; 4.8]$ karena -1.225 masih lebih besar dari -4.8 dan masih lebih kecil dari 4.8.

Tabel 4.8 Hasil *Update* Kecepatan Iterasi ke-1 Setelah Pembatasan Kecepatan

Partikel ke-i	Mata Pelajaran	Dimensi ke-j	Kecepatan Sebelum	Kecepatan Sesudah
1	IPS	1	-1.225	-1.225
	MTK	2	-1.05	-1.05
	TIK	3	-1.63333	-1.63333
	TIK	4	0.408333	0.408333
	BD	5	-4.08333	-4.08333
	BD	6	3.675	3.675
2	IPS	1	-0.81667	-0.81667
	MTK	2	-0.7	-0.7
	TIK	3	0.816667	0.816667
	TIK	4	-1.63333	-1.63333
	BD	5	-3.26667	-3.26667
	BD	6	2.45	2.45

Partikel ke-i	Mata Pelajaran	Dimensi ke-j	Kecepatan Sebelum	Kecepatan Sesudah
3	IPS	1	-1.86667	-1.86667
	MTK	2	-1.51667	-1.51667
	TIK	3	-2.04167	-2.04167
	TIK	4	1.225	1.225
	BD	5	-2.45	-2.45
	BD	6	0.408333	0.408333
4	IPS	1	0	0
	MTK	2	0	0
	TIK	3	0	0
	TIK	4	0	0
	BD	5	0	0
	BD	6	0	0

Hasil *update* kecepatan dan *range* kecepatan untuk masing-masing mata pelajaran dapat dilihat pada Lampiran 3.

7. Update Posisi

Update posisi dilakukan untuk mendapatkan posisi baru dengan menggunakan Persamaan 2.5. Sebagai contoh, dengan menggunakan Persamaan 2.5 posisi baru untuk dimensi ke-1 pada partikel ke-1 didapatkan dengan perhitungan sebagai berikut:

$$x_{1,1} = 4 + -1.225 \approx 3$$

4 adalah posisi lama dari dimensi ke-1 pada partikel ke-1 seperti yang ditunjukkan pada Tabel 4.4 dan -1.225 adalah hasil *update* kecepatan dimensi ke-1 pada partikel ke-1 seperti yang ditunjukkan pada Tabel 4.8. Hasil *update* posisi dapat dilihat pada Tabel 4.9.

Tabel 4.9 Hasil *Update* Posisi Iterasi ke-1

Partikel ke-i	Mata Pelajaran	Dimensi ke-j	Posisi
1	IPS	1	3
	MTK	2	2
	TIK	3	6
	TIK	4	3
	BD	5	4
	BD	6	5

Partikel ke-i	Mata Pelajaran	Dimensi ke-j	Posisi
2	IPS	1	2
	MTK	2	2
	TIK	3	4
	TIK	4	4
	BD	5	3
	BD	6	6
3	IPS	1	3
	MTK	2	3
	TIK	3	6
	TIK	4	2
	BD	5	3
	BD	6	8
4	IPS	1	2
	MTK	2	2
	TIK	3	5
	TIK	4	3
	BD	5	1
	BD	6	8

Kemudian dilakukan pengecekan terhadap posisi yang baru. Berdasarkan Tabel 4.2, x_{min} untuk mata pelajaran IPS adalah 1 dan x_{max} nya adalah 8, jadi berdasarkan tabel 4.9 jika diketahui posisi baru dari dimensi ke-1 pada partikel ke-1 bernilai 3 maka posisi baru dari dimensi ke-1 pada partikel-1 masih berada dalam range 1 sampai 8. Hasil *update* posisi pada iterasi 1 ditunjukkan pada Tabel 4.10.

Tabel 4.10 Hasil *Update* Posisi Iterasi ke-1 Setelah Pembatasan Posisi

Partikel ke-i	Mata Pelajaran	Dimensi ke-j	Posisi Sebelum	Posisi Sesudah
1	IPS	1	3	3
	MTK	2	2	2
	TIK	3	6	6
	TIK	4	3	3
	BD	5	4	4
	BD	6	5	5

Partikel ke-i	Mata Pelajaran	Dimensi ke-j	Posisi Sebelum	Posisi Sesudah
2	IPS	1	2	2
	MTK	2	2	2
	TIK	3	4	4
	TIK	4	4	4
	BD	5	3	3
	BD	6	6	6
3	IPS	1	3	3
	MTK	2	3	3
	TIK	3	6	6
	TIK	4	2	2
	BD	5	3	3
	BD	6	8	8
4	IPS	1	2	2
	MTK	2	2	2
	TIK	3	5	5
	TIK	4	3	3
	BD	5	1	1
	BD	6	8	8

Hasil *update* posisi tiap iterasi dan *range* posisi untuk masing-masing mata pelajaran dapat dilihat pada Lampiran 4.

8. Perhitungan Nilai *Fitness*

Setiap posisi baru akan dihitung nilai *fitness*-nya menggunakan Persamaan 4.1. Misalkan pada Tabel 4.10, posisi dimensi ke-1 pada partikel ke-1 berisi pilihan ke-3, dimana berdasarkan Tabel 4.2 pilihan ke 3 adalah guru yang memiliki masa kerja selama 4 tahun, usia 34 tahun, dan jarak tempat tinggal ke SMP 2 adalah 3 km, maka perhitungan *fitness* nya adalah seperti berikut:

$$\begin{aligned}
 fitness &= 4 + \left(\frac{100}{34 + 3} \right) \\
 &= 4 + 0.0227 \\
 &= 6.7027
 \end{aligned}$$

Kemudian setelah nilai *fitness* setiap dimensi di hitung, maka akan dijumlahkan, sehingga setiap partikel akan mempunyai 1 nilai total *fitness* seperti yang ditunjukkan pada Tabel 4.11.

Tabel 4.11 Hasil Perhitungan *Fitness* dan Total *Fitness* Pada Iterasi 1

Partikel ke- i	Mata Pelajaran	Dimensi ke- j	Posisi	<i>Fitness</i>	Total <i>Fitness</i>
1	IPS	1	3	6.702702703	54.86696054
	MTK	2	2	11.92397661	
	TIK	3	6	12.5	
	TIK	4	3	11.66666667	
	BD	5	4	4.923076923	
	BD	6	5	7.150537634	
2	IPS	1	2	9.818181818	55.82199036
	MTK	2	2	11.92397661	
	TIK	3	3	11.66666667	
	TIK	4	4	4.785714286	
	BD	5	3	6.666666667	
	BD	6	6	10.96078431	
3	IPS	1	3	6.702702703	47.36161743
	MTK	2	3	4.380952381	
	TIK	3	6	12.5	
	TIK	4	2	12.3255814	
	BD	5	3	6.666666667	
	BD	6	8	4.785714286	
4	IPS	1	2	9.818181818	54.56625462
	MTK	2	2	11.92397661	
	TIK	3	5	9.173913043	
	TIK	4	3	11.66666667	
	BD	5	1	7.197802198	
	BD	6	8	4.785714286	

9. Penentuan P_{best} dan G_{best}

P_{best} iterasi ke-1 adalah posisi terbaik masing-masing partikel sampai pada iterasi ke-1. Pada kasus ini, posisi terbaik partikel dicari dengan cara membandingkan masing-masing nilai total *fitness* partikel pada iterasi ke-0 dan iterasi ke-1, partikel terbaik adalah partikel yang memiliki nilai total *fitness* terbesar. Perbandingan nilai partikel pada iterasi ke-0 dan iterasi ke-1 dapat dilihat pada Tabel 4.12.

Tabel 4.12 Perbandingan Total *Fitness* Setiap Partikel di Iterasi ke-0 dan Iterasi ke-1

Partikel ke- <i>i</i>	Total Fitness iterasi ke-0	Total fitness iterasi ke-1
1	54.35447049	54.86696054
2	51.29007618	55.82199036
3	52.12663253	47.36161743
4	67.75306781	54.56625462

Sebagai contoh, berdasarkan Tabel 4.12 dapat dilihat bahwa partikel ke-1 yang memiliki nilai total *fitness* terbesar sampai pada iterasi ke-1 adalah partikel ke-1 pada iterasi ke-1, sehingga P_{best} untuk iterasi ke-1 sama dengan partikel ke-1 pada iterasi ke-1. Sedangkan G_{best} adalah partikel yang memiliki nilai total *fitness* terbesar dari P_{best} iterasi ke-1. Hasil P_{best} dan G_{best} untuk iterasi ke-1 dapat dilihat pada Tabel 4.13. P_{best} dan G_{best} per iterasi dapat dilihat pada Lampiran 2.

Tabel 4.13 P_{best} dan G_{best} Iterasi ke-1

Partikel ke- <i>i</i>	Mata Pelajaran	Dmensi ke- <i>j</i>	P_{best}	<i>Fitness</i>	Total <i>Fitness</i>
1	IPS	1	3	6.702702703	54.86696054
	MTK	2	2	11.92397661	
	TIK	3	6	12.5	
	TIK	4	3	11.66666667	
	BD	5	4	4.923076923	
	BD	6	5	7.150537634	
2	IPS	1	2	9.818181818	55.82199036
	MTK	2	2	11.92397661	
	TIK	3	3	11.66666667	
	TIK	4	4	4.785714286	
	BD	5	3	6.666666667	
	BD	6	6	10.96078431	
3	IPS	1	5	11.04082	52.12663253
	MTK	2	4	6.020202	
	TIK	3	8	11.7027	
	TIK	4	1	9.325581	
	BD	5	5	7.150538	
	BD	6	7	6.886792	

Partikel ke- i	Mata Pelajaran	Dmensi ke- j	P_{best}	<i>Fitness</i>	Total <i>Fitness</i>
G_{best} 4	IPS	1	2	9.818182	67.75306781
	MTK	2	2	11.92398	
	TIK	3	5	9.173913	
	TIK	4	3	11.66667	
	BD	5	1	20.38462	
	BD	6	8	4.785714	

10. Cek kondisi Berhenti

Syarat pemberhentian pada kasus ini adalah berdasarkan maksimum iterasi. Sehingga, iterasi berhenti pada iterasi ke-3 karena iterasi maksimum ditentukan sebanyak 3. Jika kondisi tidak terpenuhi maka ulangi perhitungan dimulai dari langkah ke-5. Jika kondisi berhenti terpenuhi maka iterasi diberhentikan.

4.4 Perancangan Antarmuka

Sistem optimasi pemerataan mutasi guru mata pelajaran di Kabupaten Lumajang memiliki 4 halaman, yaitu halaman data guru dan data sekolah, data kekurangan dan kelebihan, pengujian, dan detail perhitungan.

4.4.1 Antarmuka Halaman Data Guru dan Sekolah

Halaman data guru dan sekolah adalah halaman yang bertujuan untuk menampilkan data guru dan data sekolah yang digunakan dalam sistem. Perancangan antarmuka halaman data guru dan sekolah ditunjukkan pada Gambar 4.12.

4.4.2 Antarmuka Halaman Data Kekurangan dan Kelebihan

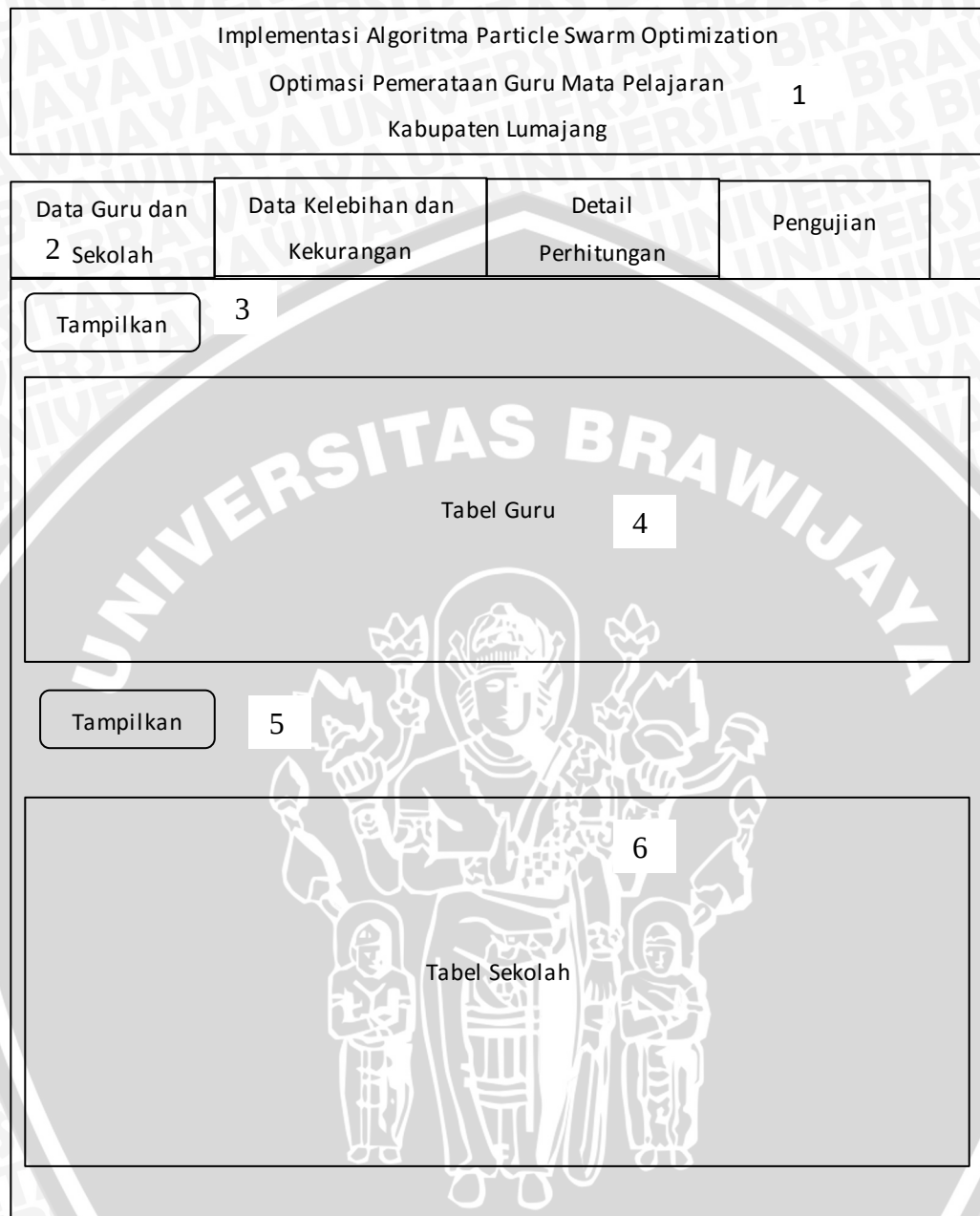
Halaman data kekurangan dan kelebihan terdiri dari 3 tabel yaitu tabel sekolah kekurangan, tabel sekolah kelebihan dan jumlah partikel per mapel. Rancangan antarmuka halaman data kekurangan dan kelebihan ditunjukkan pada Gambar 4.13.

4.4.3 Antarmuka Halaman Pengujian

Halaman pengujian menyediakan panel untuk memasukkan nilai parameter PSO sebagai pengujian. Rancangan antarmuka halaman pengujian dapat dilihat pada Gambar 4.14.

4.4.4 Antarmuka Halaman Detail Perhitungan

Halaman detail perhitungan digunakan untuk menampilkan perhitungan lebih detail dari proses pengujian yang telah dilakukan, meliputi perhitungan *fitness* partikel di tiap iterasi, G_{best} per iterasi dan hasil akhir sebagai solusi terbaik. Rancangan antarmuka halaman detail perhitungan dapat dilihat pada Gambar 4.15.



Gambar 4.12 Antarmuka Halaman Data Guru dan Sekolah

Keterangan:

1. *Panel* untuk menampilkan judul sistem
2. *Tabbed Pane* untuk menampilkan data guru dan data sekolah
3. *Button* untuk menampilkan data guru
4. *Panel* untuk menampilkan tabel data guru
5. *Button* untuk menampilkan data sekolah
6. *Panel* untuk menampilkan tabel data sekolah

Implementasi Algoritma Particle Swarm Optimization

Optimasi Pemerataan Mutasi Guru SMP

Kabupaten Lumajang

Data Guru dan Sekolah	Data Kelebihan dan Kekurangan	Detail Perhitungan	Pengujian
2 Tabel Sekolah Kekurangan Jumlah Kurang :	3 Tabel Sekolah Kelebihan Jumlah Lebih :	4 Tabel Jumlah Partikel per Mapel Jumlah Guru Pindah :	

Gambar 4.13 Antarmuka Halaman Kekurangan dan Kelebihan

Keterangan:

1. *Tabbed Pane* untuk menampilkan sekolah kelebihan sekolah kekurangan, jumlah partikel per mata pelajaran
2. *Panel* untuk menampilkan tabel sekolah kekurangan
3. *Panel* untuk menampilkan tabel sekolah kelebihan
4. *Panel* untuk menampilkan jumlah partikel per mapel

Implementasi Algoritma Particle Swarm Optimization

Optimasi Pemerataan Mutasi Guru SMP

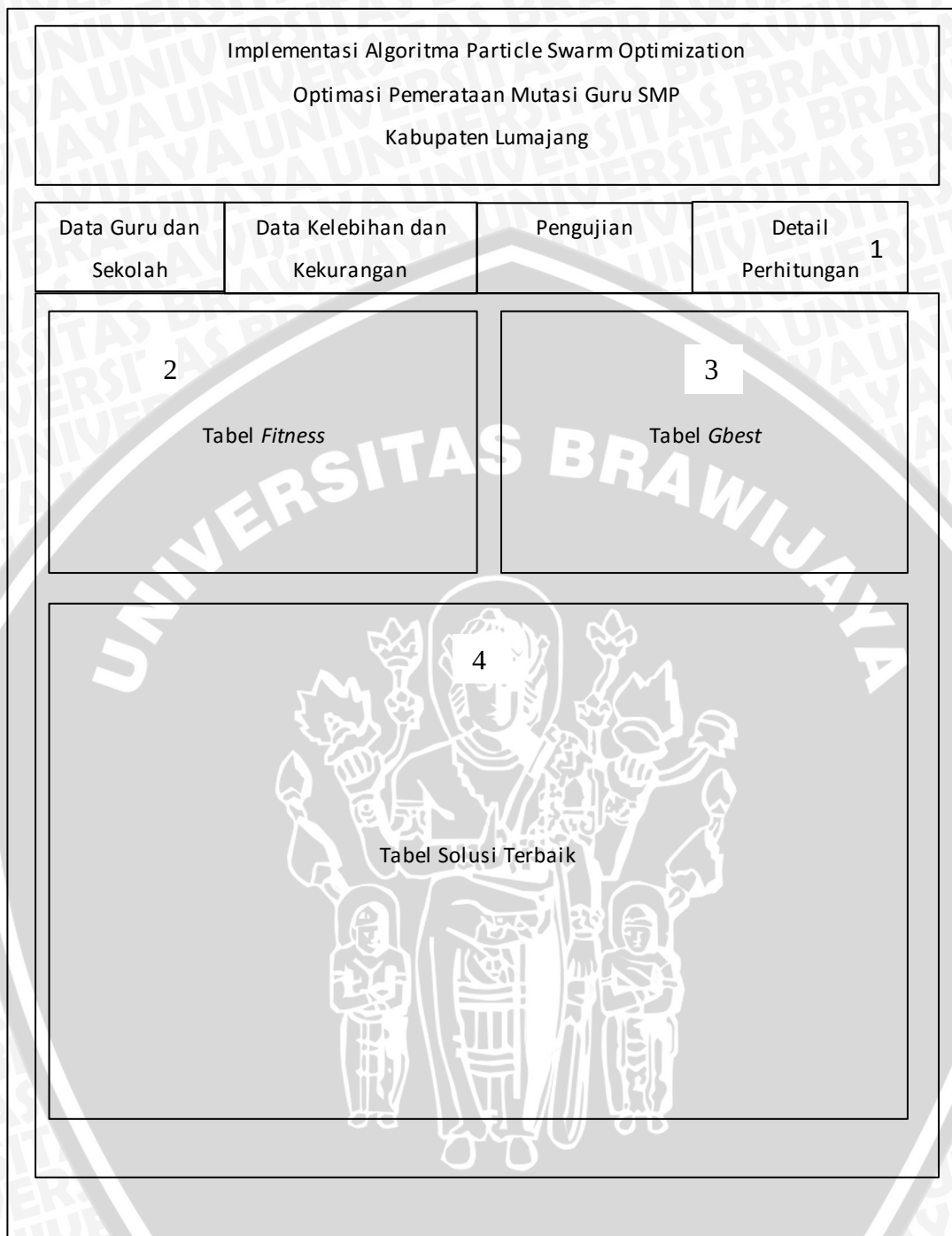
Kabupaten Lumajang

Data Guru dan Sekolah	Data Kelebihan dan Kekurangan	Pengujiar 1	Detail Perhitungan
Ukuran Populasi : 2 Max Populasi : Max Iterasi : W min : W max : C1i : C1f : C2i : C2f : r1 : r2 : 4 Tabel Hasil Pengujian 3 PROCESS Total Fitness Gbest :			

Gambar 4.14 Antarmuka Halaman Pengujian

Keterangan:

1. *Tabbed Pane* untuk menampilkan panel *inputan* parameter PSO dan hasil Pengujian
2. *Panel* untuk memasukan parameter PSO
3. *Button* untuk menjalankan algoritma PSO
4. *Panel* untuk menampilkan tabel hasil pengujian (iterasi terakhir)



Gambar 4.15 Antarmuka Halaman Detail Perhitungan

Keterangan:

1. *Tabbed Pane* untuk menampilkan detal perhitungan
2. *Panel* untuk menamopilkan tabel *fitness*
3. *Panel* untuk menamopilkan tabel *Gbest*
4. *Panel* untuk menamopilkan tabel solusi terbaik

BAB 5 IMPLEMENTASI

Bab Implementasi membahas hasil implementasi dari perancangan yang sudah dilakukan sebelumnya guna mengetahui sejauh mana sistem dapat dibangun sesuai dengan yang direncanakan. Bab implementasi terdiri dari pembahasan implementasi program dan implementasi antarmuka.

5.1 Implementasi Program

Subbab ini membahas hasil dari implementasi program sistem optimasi mutasi guru menggunakan PSO yang telah dirancang sebelumnya dengan diagram alir. Implementasi program terdiri dari implementasi proses inisialisasi partikel, *update* kecepatan, *update* posisi, perhitungan nilai *fitness*, *update* P_{best} dan *update* G_{best} , dan implementasi antarmuka.

5.1.1 Inisialisasi Partikel

Pada sisitem optimasi pemerataan guru mata pelajaran, proses Inisialisasi partikel menentukan posisi awal secara acak seluruh partikel dalam populasi, nilai *fitness* awal, P_{best} awal dan G_{best} awal. Proses Inisialisasi partikel dapat dilihat pada *Source Code* 5.1.

```

1 //Inisialisasi Partikel
2 public void init(int ukuranPopulasi)
3 throws SQLException, IOException{
4     partikel=new Data[ukuranPopulasi];
5     index= new Data[ukuranPopulasi];
6     for (int i = 0; i < ukuranPopulasi; i++) {
7         index[i] = new Data();
8         index [i].connect();
9         index[i] = partikel[i];
10    }
11 //Pemilihan Posisi Awal secara Acak
12 int guruTerpilih = 0;
13 int list = 0;
14 while (list < jumlahLebihBING &&
15     jumlahLebihBING > 0 && jumlahKurangBING > 0) {
16     double rand = Math.random();
17     double indexAcak =
18     1 + (rand * (seluruhKemungkinanBING - 1));
19     double indexPembulatan =
20     (double) Math.round(indexAcak);
21     int indexAwal = (int) indexPembulatan;
22     int cek = 0;
23     guruTerpilih++;
24     list++;

```

```

25     }
26     // Perhitungan Fitness Awal
27     private void FitnessAwal() {
28         int topPSO = 0;
29         int i = 0;
30         while (GuruTerpilihBING[i][0] != null) {
31             Fitness[topPSO][0] = GuruTerpilihBING[i][0];
32             Fitness[topPSO][1] = GuruTerpilihBING[i][7];
33             double masaKerja =
34 Double.valueOf(GuruTerpilihBING[i][4]);
35             double usia =
36 Double.valueOf(GuruTerpilihBING[i][3]);
37             double jarak =
38 Double.valueOf(GuruTerpilihBING[i][6]);
39             double fitnessAwal = (100 / (usia + jarak))
40 + masaKerja;
41             Fitness[topPSO][2] =
42 String.valueOf(fitnessAwal);
43             i++;
44             topPSO++;
45         }
46         for (int loop = 0; loop < topPSO; loop++) {
47             totalFitness = totalFitness +
48 Double.valueOf(Fitness[loop][2]);
49         }
50
51     }
52     // Penentuan Gbest pada iterasi pertama
53     Data GBest = partikel[0];
54     for (int i = 0; i < ukuranPopulasi; i++) {
55         if (GBest.totalFitness <
56 partikel[i].totalFitness) {
57             GBest = partikel[i];
58             partikelGbest = i;
59         }
60     }
61     globalBest = GBest;

```

Source Code 5.1 Proses Inisialisasi Partikel

Keterangan *Source Code* 5.1:

1. Baris 1-5 : Pembangkitan partikel dalam populasi sebanyak ukuran populasi yang ditentukan.
2. Baris 6-10 : Menentukan P_{best} iterasi pertama.

3. Baris 16-20 : Pemilihan posisi secara acak menggunakan Persamaan 2.6.
4. Baris 33-40 : Perhitungan *fitness* awal menggunakan Persamaan 4.1 untuk partikel yang termasuk dalam mata pelajaran bahasa inggris.
5. Baris 52-61 : Penentuan G_{best} untuk iterasi pertama.

5.1.2 Update Kecepatan

Implementasi *update* kecepatan digunakan untuk mengupdate kecepatan partikel di setiap iterasinya menggunakan Persamaan 2.7. *Update* kecepatan partikel dilakukan sesuai mata pelajarannya karena batas kecepatan setiap mata pelajaran berbeda. Implementasi untuk proses *update* kecepatan dapat dilihat pada *Source Code* 5.2.

```

1 // Update Kecepatan
2 public void kecepatan(int ukuranPopulasi){
3     for (int i = 0; i < ukuranPopulasi; i++) {
4         int j=0;
5         while (partikel[i].Fitness[j][0]!=null){
6             double updateKecepatan;
7             if(partikel[i].kecepatan[j][2]==null ||
8 partikel [i].kecepatan[j][2]==""){
9                 updateKecepatan=W*0+C1*R1*D
10 (Double.valueOf(partikel [i].Fitness[j][1]))-
11 ouble.valueOf(index[i].Fitness[j][1]))
12                 +C2*R2*
13 (Double.valueOf(GlobalBest.Fitness[j][1]))-
14 Double.valueOf(index [i].Fitness[j][1]));
15             }else{
16                 updateKecepatan= W *
17 Double.valueOf(partikel [i].kecepatan[j][2]) + C1 * R1 *
18 (Double.valueOf(partikel[i].Fitness[j][1]) -
19 Double.valueOf(index [i].Fitness[j][1]))
20                 + C2 * R2 *
21 (Double.valueOf(GlobalBest.Fitness[j][1]) -
22 Double.valueOf(index [i].Fitness[j][1]));
23             }
24             partikel[i].kecepatan[j][0]=
25 partikel [i].Fitness[j][0];
26             partikel [i].kecepatan[j][1]=
27 partikel [i].Fitness[j][1];
28 //Pembatasan Kecepatan
29             double kecepatanBaru=0;
30             double vMaxBING=
31 (60/100)*populasi[i].seluruhKemungkinanBING;
32             double vMin=(-1*vMaxBING);

```

```

33  if(partikel[i].kecepatan[j][0]=="Bahasa Inggris"){
34      if(UpdateKecepatan>vMaxBING){
35          kecepatanBaru=vMaxBING;
36      }
37      else if(updateKecepatan<vMinBING){
38          kecepatanBaru=vMinBING;
39      }
40      else{
41          kecepatanBaru=updateKecepatan;
42      }
43      }
44      partikel[i].kecepatan[j][2]=
45      String.valueOf(kecepatanBaru);
46      posisi++;
47      j++;
48      }
49      }
50      }

```

Source Code 5.2 Proses Update Kecepatan

Keterangan Source Code 5.2:

1. Baris 9-22 : Proses perhitungan kecepatan baru menggunakan Persamaan 2.7.
2. Baris 28-43 : Proses pembatasan posisi untuk partikel yang termasuk dalam mata pelajaran bahasa Inggris.

5.1.3 Update Posisi

Implementasi *update* posisi digunakan untuk meng-*update* posisi partikel di setiap iterasinya menggunakan Persamaan 2.5. *Update* posisi partikel dilakukan sesuai mata pelajarannya karena batas posisi setiap mata pelajaran berbeda. Implementasi untuk *update* posisi dapat dilihat pada Source Code 5.3.

```

1  //Update Posisi
2  public void posisi(int ukuranPopulasi){
3      for (int i = 0; i < ukuranPopulasi; i++) {
4          int j=0, k=0;
5          while(partikel[i].Fitness[j][0]!=null){
6              if(partikel [i].Fitness[j][0]==
7              partikel [i].kecepatan[k][0] && j==k
8              && partikel [i].Fitness[j][0]=="Bahasa Inggris"){
9                  double updatePosisi=
10                 Double.valueOf(partikel[i].kecepatan[k][2])+
11                 Double.valueOf(partikel[i].Fitness[j][1]);
12                 // Pembatasan Posisi
13                 int posisiBaru = (int) updatePosisi;

```

```

14  int posisiBaruGanti= 0;
15  if (posisiBaru < 0) {
16  posisiBaruGanti = 0;
17  } else if (posisiBaru >=
18 partikel[i].seluruhKemungkinanBINA) {
19  posisiBaruGanti =
20  partikel [i].seluruhKemungkinanBING - 1;
21  } else {
22  posisiBaruGanti = posisiBaru;
23  }
24  partikel[i].Fitness[j][1]
25 =String.valueOf(posisiBaruGanti);
26  }
27  j++;
28  k++;
29  }
30  partikel[i].totalFitnes=0;
31  }
32  }

```

Source Code 5.3 Proses Update Posisi

Keterangan Source Code 5.3:

1. Baris 9-11 : Proses perhitungan posisi baru menggunakan Persamaan 2.5.
2. Baris 13-26 : Proses pembatasan posisi untuk partikel yang termasuk dalam mata pelajaran bahasa Inggris.

5.1.4 Perhitungan *Fitness*

Implementasi perhitungan nilai *fitness* digunakan untuk menghitung nilai *fitness* dari posisi setiap dimensi pada suatu partikel menggunakan Persamaan 4.1. Nilai *fitness* setiap partikel akan dihitung berdasarkan mata pelajaran. Implementasi perhitungan nilai *fitness* dapat dilihat pada Sorce Code 5.4.

```

1  // Perhitungan Fitness
2  public void fitnessBaru(int ukuranPartikel)
3  throws SQLException, IOException{
4  try {
5  // mengambil url,user,pass
6  connection = DriverManager.getConnection
7  (url, user, password);
8  // System.out.println("Connection Database");
9  } catch (SQLException e) {
10 System.out.println("Not Connected");
11 }
12 for (int i = 0; i < ukuranPartikel; i++) {

```

```

13   int j=0;
14   while (partikel[i].Fitness[j][0]!=null){
15   if("Bahasa Inggris".equals(partikel [i].Fitness[j][0])){
16     String nama1=
17     partikel [i].listGuruBING
18     [Integer.parseInt(partikel [i].Fitness[j][1]))[2];
19     double usia1=
20     Double.valueOf(partikel [i].listGuruBING
21     [Integer.parseInt(partikel [i].Fitness[j][1]))[3]);
22     double masakerjal=
23     Double.valueOf(partikel [i].listGuruBING
24     [Integer.parseInt(partikel [i].Fitness[j][1]))[4]);
25     double jarak=
26     Double.valueOf(partikel [i].listGuruBING
27     [Integer.parseInt(partikel [i].Fitness[j][1]))[9]);
28     double fitness=(1/(jarak+usia1))
29     +masakerjal;
30     partikel [i].Fitness[j][2]=
31     String.valueOf(fitness);
32   }
33   partikel[i].totalFitnes= partikel [i].totalFitness+
34   Double.valueOf(partikel [i].Fitness[j][2]);
35
36   posisi++;
37   j++;
38   }
39   }
40   }

```

Source Code 5.4 Proses Perhitungan *Fitness*

Keterangan Source Code 5.4:

1. Baris 19-21 : Mengambil data usia dari partikel yang termasuk dalam mata pelajaran bahasa Inggris.
2. Baris 22-24 : Mengambil data masa kerja dari partikel yang termasuk dalam mata pelajaran bahasa Inggris.
3. Baris 25-27 : Mengambil data jarak dari partikel yang termasuk dalam mata pelajaran bahasa Inggris.
4. Baris 28-31 : Perhitungan nilai *fitness* partikel yang termasuk dalam mata pelajaran bahasa Inggris menggunakan Persamaan 4.1.
5. Baris 33-34 : Menghitung total *fitness* setiap partikel.

5.1.5 Update P_{best}

Proses *update* P_{best} merupakan implementasi dari diagram alir pada Gambar 4.10. Implementasi proses *update* P_{best} dapat dilihat pada *Source Code* 5.5.

Source Code 5.5 Proses Update P_{best}

```

1 // Update Pbest
2 Data gbest(int ukuranPopulasi){
3
4     for (int i = 0; i < ukuranPopulasi; i++) {
5         if(partikel[i].totalFitness <
6             index[i].totalFitness){
7             partikel [i]=index[i];
8         }else{
9             index[i]=partikel[i];
10        }
11    }

```

Keterangan *Source Code* 5.5:

1. Baris 5-10 : Proses untuk membandingkan dan memperbarui, jika nilai total *fitness* partikel yang baru lebih besar dari P_{best} maka P_{best} digantikan dengan partikel yang baru.

5.1.6 Update G_{best}

Proses *update* G_{best} merupakan implementasi dari diagram alir pada Gambar 4.11. Implementasi proses *update* G_{best} dapat dilihat pada *Source Code* 5.6.

```

1 // Update Gbest
2 Data GBest=new Data();
3     for (int i = 0; i < ukuranPopulasi; i++) {
4         if(GBest.totalFitness<
5             partikel[i].totalFitness){
6             GBest=partikel[i];
7             partikelGbest=i;
8         }
9     }
10     if(globalBest.totalFitnes<GBest.totalFitnes){
11         globalBest=GBest;
12     }
13     return globalBest;
14 }
15 }

```

Source Code 5.6 Proses Update G_{best}

Keterangan *Source Code* 5.6:

1. Baris 4-7 : Proses mencari G_{best} pada populasi yang baru

2. Baris 10-11 : Proses membandingkan G_{best} populasi yang baru dengan G_{best}

populasi yang lama, jika G_{best} pada populasi yang baru lebih besar dari G_{best} pada populasi yang lama maka $G_{best} = G_{best}$ pada populasi yang baru.

5.2 Implementasi Antarmuka

Pada sub bab ini akan dibahas mengenai hasil dari implementasi antar muka untuk sistem optimasi mutasi guru menggunakan PSO yang telah dirancang sebelumnya. Implementasi antarmuka terdiri dari implementasi antarmuka halaman data sekolah dan guru, halaman kelebihan dan kekurangan, halaman pengujian dan halaman detail perhitungan.

5.2.1 Halaman Data Sekolah dan Guru

Halaman data sekolah dan guru adalah halaman yang digunakan untuk menampilkan data sekolah dan data guru dalam tabel. Tabel data guru berisi informasi mengenai unit kerja yaitu sekolah tempat guru mengajar, mata pelajaran yang diampu guru, nama guru, tanggal lahir guru, tanggal mulai mengajar di sekolah guru yang bersangkutan, alamat tempat tinggal, jenis PTK, jenis kepegawaian guru, dan tugas tambahan yang dimiliki guru. Sedangkan tabel sekolah berisi informasi nama sekolah, mata pelajaran yang diajarkan, nama guru sesuai mata pelajarannya, jumlah kurang, lebih dan sedia untuk masing-masing mata pelajaran. Implementasi halaman dan sekolah dapat dilihat pada Gambar 5.1.

5.2.2 Halaman Kelebihan dan Kekurangan

Halaman kelebihan dan kekurangan berisi 3 tabel yaitu tabel sekolah kekurangan, tabel sekolah kelebihan, dan tabel jumlah partikel per mapel. Tabel sekolah menampilkan daftar sekolah yang memiliki kekurangan, lengkap dengan mata pelajaran dan jumlah kurangnya. Tabel sekolah kelebihan menampilkan daftar sekolah yang memiliki kelebihan, lengkap dengan mata pelajaran dan jumlah lebihnya. Tabel banyak dimensi per mapel menampilkan daftar semua mata pelajaran lengkap dengan jumlah kekurangan, jumlah kekelebihan, dan jumlah sedia guru nya, dari jumlah kekurangan dan kelebihan bisa dihitung berapa banyak guru yang bisa dipindah untuk mata pelajaran tersebut yang akan ditampilkan pada kolom banyak guru pindah, sedangkan dari jumlah sedia dan jumlah kekurangan bisa didapatkan jumlah pilihan mutasi yang bisa diambil tiap mata pelajarannya dimana jumlah pilihan mutasi yang bisa diambil digunakan sebagai banyak dimensi saat implementasi algoritma PSO pada penelitian ini. Antarmuka halaman kelebihan dan kekurangan dapat dilihat pada Gambar 5.2.

Implementasi Algoritma Particle Swarm Optimization
Optimasi Mutasi Guru SMP
Kabupaten Lumajang

DATA GURU DAN SEKOLAH DATA KEKURANGAN DAN KELEBIHAN PENGUJIAN DETAIL PERHITUNGAN

DATA GURU

No	Unit Kerja	Mapel	Nama	TGL Lahir	TTM Kerja	Alamat	Jenis PTK	Kepe...	Tugas Tambahan
1	SMPN 2 RANDUAGU...	Matematika	SUTOMO	1954-08-15	1977-11-01	Sumbersuko, Lu...	Guru Mata P...	PNS	Kepala Sekolah
2	SMPN 1 KLAKAH	Bahasa Indo...	SUHARYANTO	1954-08-19	1989-03-01	Purwanto, Bli...	Guru Mata P...	PNS	
3	SMPN 1 CANDIPURO	PAI	NASRON	1954-11-04	1987-12-01	Candipuro, Lum...	Guru Mata P...	PNS	
4	SMPN 1 KLAKAH	IPS	SRI WILUDIENG	1954-11-10	1987-03-01	Mlawang, Klak...	Guru Mata P...	PNS	Wakil Kepala Seko...
5	SMPN 1 CANDIPURO	Matematika	HERU PURNOMO...	1955-02-06	1980-05-01	Pasirian, Lumaj...	Guru Mata P...	PNS	
6	SMPN 2 LUMAJANG	IPS	SUKIMI	1955-02-23	1981-03-01	Tompokersan, ...	Guru Mata P...	PNS	
7	SMPN 1 YOSOWILAN...	PMPKn	SUTJIPTO	1955-06-17	1985-02-01	Cakru, Kencong...	Guru Mata P...	PNS	
8	SMPN 1 ROWOKANG...	PMPKn	PAMBUDIDOYO, ...	1955-08-15	1977-03-01	Tanjung Rejo, Y...	Guru Mata P...	PNS	
9	SMPN 1 JATIROTO	Bahasa Inggris...	MADYADI	1955-08-18	1986-01-01	Wringin Agung, ...	Guru Mata P...	PNS	
10	SMPN 3 LUMAJANG	IPA	BUDI RAHAYU	1955-08-28	2008-03-03	Kepuharjo, Lum...	Guru Mata P...	PNS	Kepala Sekolah

DATA SEKOLAH

No	Nama Sekolah	Mata Pelajaran	Kurang	Lebih	Sedia
1	SMPN 2 SUMBERSUKO	PAI	0	0	0
2	SMPN 2 SUMBERSUKO	PMPKn	0	0	1
3	SMPN 2 SUMBERSUKO	PJOK	1	0	0
4	SMPN 2 SUMBERSUKO	Bahasa Indonesia	0	1	3
5	SMPN 2 SUMBERSUKO	Bahasa Inggris	1	0	1
6	SMPN 2 SUMBERSUKO	IPS	0	0	2
7	SMPN 2 SUMBERSUKO	IPA	0	0	2
8	SMPN 2 SUMBERSUKO	Matematika	0	0	2
9	SMPN 2 SUMBERSUKO	TIK	1	0	0
10	SMPN 2 SUMBERSUKO	Bahasa Daerah	1	0	0

Gambar 5.1 Antarmuka Halaman Data Guru dan Data Sekolah

Implementasi Algoritma Particle Swarm Optimization
Optimasi Mutasi Guru SMP
Kabupaten Lumajang

DATA GURU DAN SEKOLAH DATA KEKURANGAN DAN KELEBIHAN PENGUJIAN DETAIL PERHITUNGAN

SEKOLAH KEKURANGAN

Nama Sekolah	Mapel	Kurang
SMPN 2 PASRUJAMBE	Bahasa Daerah	1
SMPN 2 PASRUJAMBE	Seni Budaya	1
SMPN 2 PASRUJAMBE	Prakarya	1
SMPN 2 LUMAJANG	PAI	1
SMPN 2 LUMAJANG	TIK	2
SMPN 2 LUMAJANG	Bahasa Daerah	1
SMPN 2 LUMAJANG	Seni Budaya	1
SMPN 2 LUMAJANG	Prakarya	1
SMPN 1 CANDIPURO	PJOK	1
SMPN 1 CANDIPURO	TIK	2

JUMLAH KURANG: 255

SEKOLAH KELEBIHAN

Nama Sekolah	Mapel	Lebih
SMPN 2 SUMBERSUKO...	Bahasa Indonesia	1
SMPN 1 LUMAJANG	IPS	1
SMPN 1 LUMAJANG	IPA	1
SMPN 1 LUMAJANG	Matematika	1
SMPN 2 PASRUJAM...	Bahasa Inggris	1
SMPN 2 PASRUJAM...	Matematika	1
SMPN 2 LUMAJANG	PMPKn	1
SMPN 2 LUMAJANG	IPS	1
SMPN 1 CANDIPURO	Bahasa Indonesia	1
SMPN 1 GUCIALIT	PMPKn	1
SMPN 1 GUCIALIT	IPS	1

JUMLAH LEBIH: 138

JUMLAH PARTIKEL per MAPEL

No	Mapel	JML Kurang	JML Lebih	JML Sedia	Banyak Guru Pindah	JML Pilihan
1	Bahasa Indonesia	7	15	47	7	329
2	Bahasa Inggris	16	11	25	11	400
3	Bahasa IPA	5	20	61	5	305
4	IPS	4	40	111	4	444
5	TIK	62	0	0	0	0
6	PJOK	24	4	8	4	192
7	Matematika	7	25	78	7	546
8	Bahasa Daerah	54	0	0	0	0
9	PMPKn	5	21	50	5	250

JUMLAH GURU PINDAH: 45

Gambar 5.2 Antarmuka Halaman Data Kekurangan Dan Kelebihan

5.2.3 Halaman Pengujian

Halaman pengujian terdiri dari 2 *panel* yaitu *panel* parameter *input* dan *panel* untuk menampilkan hasil. *Panel* parameter *input* digunakan untuk memasukkan nilai- nilai parameter PSO yang akan di proses pada sistem. Parameter PSO yang dimasukkan adalah ukuran populasi, iterasi maksimum, θ_{min} , θ_{max} , c_{1i} , c_{1f} , c_{2i} , c_{2f} , r_1 , dan r_2 . Sedangkan tabel hasil akan menampilkan hasil sesuai dengan nilai yang di-*input*-kan, hasil merupakan perhitungan yang didapatkan pada iterasi terakhir berupa P_{best} dan G_{best} yang didapatkan berserta nilai *fitness*-nya. Antarmuka halaman pengujian dapat dilihat pada Gambar 5.3.

Iterasi	Populasi G Best	Partikel	Index Terpilih	Fitness
1	1	1	189	8.025641025641026
4	1	2	206	14.02
4	1	3	227	26.015873015873016
4	1	4	201	11.029411764705882
4	1	5	169	23.02
4	1	6	196	7.022222222222222
4	1	7	202	26.0188679245283
4	1	8	182	4.020408163265306
4	1	9	275	19.00943396226415
4	1	10	207	4.020408163265306
4	1	11	116	31.013157894736842
4	1	12	220	12.014925373134329
4	1	13	179	31.00980392156863
4	1	14	118	12.020833333333334
4	1	15	158	30.0125
4	1	16	116	13.02
4	1	17	205	30.01923076923077
4	1	18	78	7.014285714285714
4	1	19	112	29.017241379310345
4	1	20	103	34.016949152542374
4	1	21	162	29.017241379310345
4	1	22	242	31.017543859649123
4	1	23	90	28.01818181818182

TOTAL FITNESS G best: 930.7998732949912

Gambar 5.3 Antarmuka Halaman Pengujian

5.2.4 Halaman Detail Perhitungan

Halaman detail perhitungan merupakan halaman yang menampilkan perhitungan yang lebih detail dari percobaan yang dilakukan. Halaman detail perhitungan terdiri dari 3 tabel yaitu tabel *fitness*, tabel G_{best} dan tabel solusi terbaik. Tabel *fitness* menampilkan posisi setiap partikel beserta *fitness*-nya untuk masing-masing populasi sampai pada iterasi terakhir. Tabel G_{best} menampilkan G_{best} setiap iterasi, sedangkan tabel solusi terbaik adalah populasi yang menjadi G_{best} sampai pada iterasi terakhir yang berisi informasi mengenai daftar guru yang dipindah lengkap dengan asal sekolah, sekolah tujuan mutasi, jarak menuju sekolah tujuan mutasi dan *besar fitness*-nya. Antarmuka halaman detail perhitungan dapat dilihat pada Gambar 5.4.



Implementasi Algoritma Particle Swarm Optimization
Optimasi Mutasi Guru SMP
Kabupaten Lumajang



DATA GURU DAN SEKOLAH
DATA KEKURANGAN DAN KELEBIHAN
PENGUJIAN
DETAIL PERHITUNGAN

FITNESS

Iterasi	Populasi	Partikel	Index Terpilih	Mapel	Fitness
1	1	1	189	Bahasa Ingg...	8.02564102...
1	1	2	206	Bahasa Ingg...	14.02
1	1	3	227	Bahasa Ingg...	26.0158730...
1	1	4	201	Bahasa Ingg...	11.0294117...
1	1	5	169	Bahasa Ingg...	23.02
1	1	6	188	Bahasa Ingg...	7.02222222...
1	1	7	202	Bahasa Ingg...	26.0188673...
1	1	8	182	Bahasa Ingg...	4.02040816...
1	1	9	275	Bahasa Ingg...	19.0094339...
1	1	10	207	Bahasa Ingg...	4.02040816...
1	1	11	116	Bahasa Ingg...	31.0131578...

G best

Iterasi	Populasi	Total fitness
0	1	930.799873294...
1	1	930.799873294...
2	1	930.799873294...
3	1	930.799873294...
4	1	930.799873294...

SOLUSI TERBAIK

Nama	Mapel	Usia	Lama Kerja	Asal Sekolah	Sekolah Baru	Fitness
NOVITA ARIEF HIDAYATI	Bahasa Inggris	37	8	SMPN 1 TEMPURSA...	SMPN 2 PADANG	8.025641025641026
ROCHMI ISTYOWATI	Bahasa Inggris	48	14	SMPN 1 PASIRIAN	SMPN 2 PASIRIAN	14.02
SAIFUDIN ZUHRI	Bahasa Inggris	51	26	SMPN 2 KLAKAH	SMPN 2 RANDUAGU...	26.015873015873016
VIBRIAN IKHWAN AL-KHASANI	Bahasa Inggris	32	11	SMPN 2 PASRUJAM...	SMPN 2 PASIRIAN	11.029411764705882
BASTONO SUBEKTI	Bahasa Inggris	48	23	SMPN 2 KLAKAH	SMPN 2 CANDIPURO	23.02
ENDIARTI SUSILOWATI	Bahasa Inggris	43	7	SMPN 1 SUMBERS...	SMPN 2 PADANG	7.022222222222222
SAIFUDIN ZUHRI	Bahasa Inggris	51	26	SMPN 1 KLAKAH	SMPN 2 PASIRIAN	26.0188675245283
YUSUF WIJANARKO	Bahasa Inggris	44	19	SMPN 2 PASRUJAM...	SMPN 2 SUMBERSU...	4.020408163265306
YUSUF WIJANARKO	Bahasa Inggris	44	19	SMPN 2 PASRUJAM...	SMPN 2 TEMPURSARI	19.00943396226415
BM TEJO PRAMONO	Bahasa Inggris	59	31	SMPN 2 KLAKAH	SMPN 3 YOSOWILA...	4.020408163265306
BM TEJO PRAMONO	Bahasa Inggris	59	31	SMPN 2 KLAKAH	SMPN 1 SENDIRO	31.013157894736842
PRIYO SANTOSO	Bahasa Indonesia	47	12	SMPN 3 PASIRIAN	SMPN 2 RANDUAGU...	12.014925373134329

Gambar 5.4 Antarmuka Halaman Detail Perhitungan

BAB 6 PENGUJIAN DAN ANALISIS

Bab pengujian dan analisis membahas berbagai pengujian terhadap algoritma PSO beserta analisa terhadap hasil pengujian. Pada penelitian ini, pengujian dilakukan untuk mengetahui parameter-parameter PSO yang paling optimal untuk menghasilkan nilai total *fitness* terbaik. Parameter PSO yang akan diuji adalah, iterasi maksimum, ukuran populasi, kombinasi nilai minimum (θ_{min}) dan nilai maksimum (θ_{max}) bobot inersia dan pengujian kombinasi Koefisien akselerasi yaitu c_{1i} , c_{1f} , c_{2i} , c_{2f} .

6.1 Hasil dan Analisis Uji Coba Iterasi Maksimum

Pengujian maksimum iterasi digunakan untuk mengetahui banyak iterasi yang dapat menghasilkan nilai total *fitness* terbaik. Iterasi maksimum yang diuji adalah iterasi kelipatan 10, dimulai dari 10 sampai 100, masing-masing maksimum iterasi diuji sebanyak 5 kali. Secara detail parameter-parameter PSO yang digunakan untuk uji coba maksimum iterasi adalah sebagai berikut:

Ukuran populasi	= 10
$\theta_{min}, \theta_{max}$	= 0.4, 0.9
c_{1i}, c_{1f}	= 2.5, 0.5
c_{2i}, c_{2f}	= 0.5, 2.5
r_1, r_2	= 0.5, 0.5

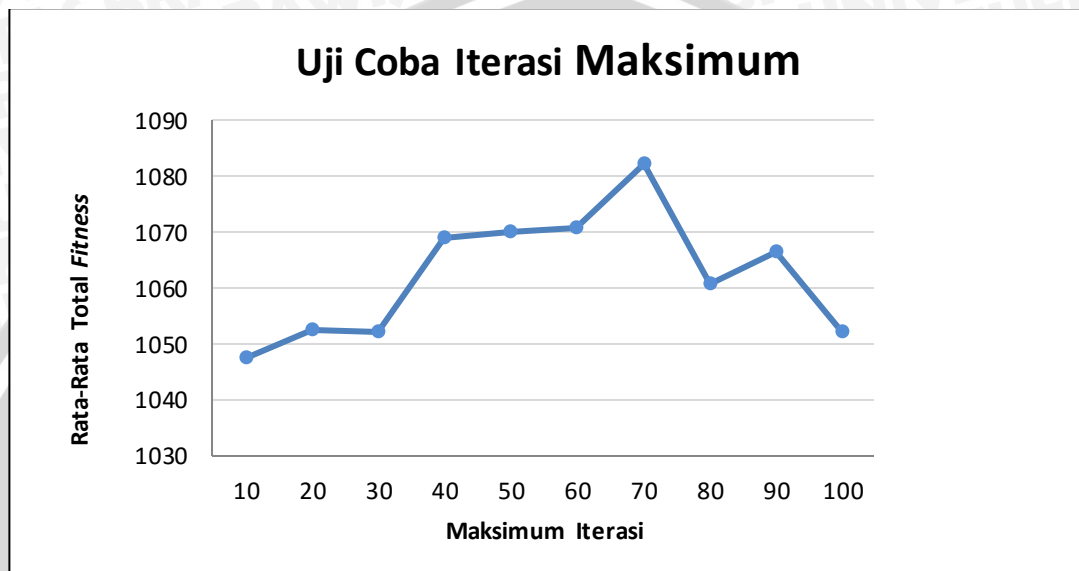
Hasil pengujian untuk mengetahui iterasi maksimum paling optimal dapat dilihat pada Tabel 6.1.

Tabel 6.1 Hasil Uji Coba Iterasi Maksimum

Maksimum Iterasi	Total Fitness					Rata-Rata Total Fitness
	Percobaan ke- i					
	1	2	3	4	5	
10	1073.742	1065.763	1049.745	1018.732	1029.768	1047.550
120	1020.709	1060.71	1055.778	1046.777	1078.754	1052.546
30	1040.744	1058.785	1099.761	1006.773	1053.693	1051.951
40	1094.75	1049.745	1009.764	1074.761	1115.738	1068.952
50	1071.785	1051.723	1052.707	1106.715	1067.749	1070.136
60	1055.741	1040.762	1118.744	1079.741	1058.778	1070.753
70	1097.75	1040.728	1101.749	1053.753	1116.739	1082.144
80	1040.735	1048.74	1104.754	1029.772	1079.728	1060.746
90	1039.79	1083.745	1148.732	1042.742	1017.764	1066.555
100	1090.734	1049.768	1070.736	1015.756	1032.738	1051.946

Iterasi maksimum yang berbeda menghasilkan nilai rata-rata total *fitness* yang berbeda. Pada Tabel 6.1 dapat diketahui bahwa nilai rata-rata total *fitness* yang

paling baik dihasilkan saat iterasi dilakukan sebanyak 70 kali dengan rata-rata total *fitness* yang paling baik yaitu 1082.144 nilai rata-rata total *fitness* cenderung mengalami kenaikan mulai iterasi maksimum sebanyak 10 sampai 70, yang puncak kenaikannya berhenti pada iterasi maksimum sebanyak 70 dikarenakan nilai rata-rata total *fitness* iterasi maksimum setelahnya yaitu 80 sampai 100 tidak mengalami kenaikan. Kondisi tersebut dapat digambarkan dengan grafik seperti pada Gambar 6.1.



Gambar 6.1 Grafik Hasil Uji Coba Iterasi Maksimum

Dari grafik pada Gambar 6.1, dapat diketahui bahwa banyaknya iterasi akan mempengaruhi nilai rata-rata total *fitness* yang dihasilkan. Pada kebanyakan kasus, sering ditemukan bahwa dengan bertambahnya iterasi yang dilakukan maka nilai *fitness* yang dihasilkan akan semakin baik yang berarti semakin banyak iterasi yang dilakukan maka kemungkinan untuk mendapatkan nilai terbaik juga semakin besar karena proses *update* posisi yang lebih sering, seperti yang ditunjukkan pada maksimum iterasi sebanyak 10 sampai 70, grafik rata-rata total *fitness* cenderung mengalami kenaikan. Namun, kelemahan untuk iterasi yang banyak adalah membutuhkan waktu komputasi yang lebih lama, akan terasa sia-sia jika sebelum mencapai batas maksimum iterasi sudah mendapatkan nilai *fitness* yang konvergen, dimana nilai yang konvergen menandakan posisi partikel tidak berubah karena sudah dianggap menemukan solusi optimal, terlebih lagi solusi yang dianggap optimal tersebut belum pasti merupakan solusi optimal sesuai yang diharapkan. Hasil optimasi juga dipengaruhi oleh posisi awal partikel, seberapa baik posisi awal yang terpilih secara acak, jika posisi awal yang dibangkitkan sudah cukup baik atau mendekati nilai optimal maka proses pencarian akan berhenti lebih awal atau dengan kata lain proses pencarian tidak membutuhkan iterasi terlalu banyak, seperti yang ditunjukkan pada Gambar 6.1 dimana iterasi yang lebih besar dari 70 tidak menghasilkan nilai rata-rata total *fitness* yang lebih baik atau sama, bahkan cenderung menurun. Penentuan maksimum iterasi sangat bergantung pada kompleksitas masalah yang dihadapi, disarankan untuk

menggunakan iterasi yang tidak terlalu sedikit dan tidak terlalu besar. Iterasi yang sedikit memungkinkan peluang pencarian solusi optimal semakin sedikit dan iterasi yang banyak hanya akan menambah waktu komputasi yang tidak diperlukan jika kriteria pemberhentian hanya berdasarkan maksimum iterasi (Englebercth, 2007).

6.2 Hasil dan Analisis Uji Coba Ukuran Populasi

Pengujian terhadap ukuran populasi bertujuan untuk mengetahui banyak partikel paling optimal yang mampu mendapatkan nilai *fitness* terbaik. Banyak partikel yang diuji adalah banyak partikel dengan kelipatan 5, dimulai dari 5 partikel sampai 50 partikel. Secara detail, parameter-parameter PSO yang digunakan untuk uji coba ukuran populasi adalah sebagai berikut:

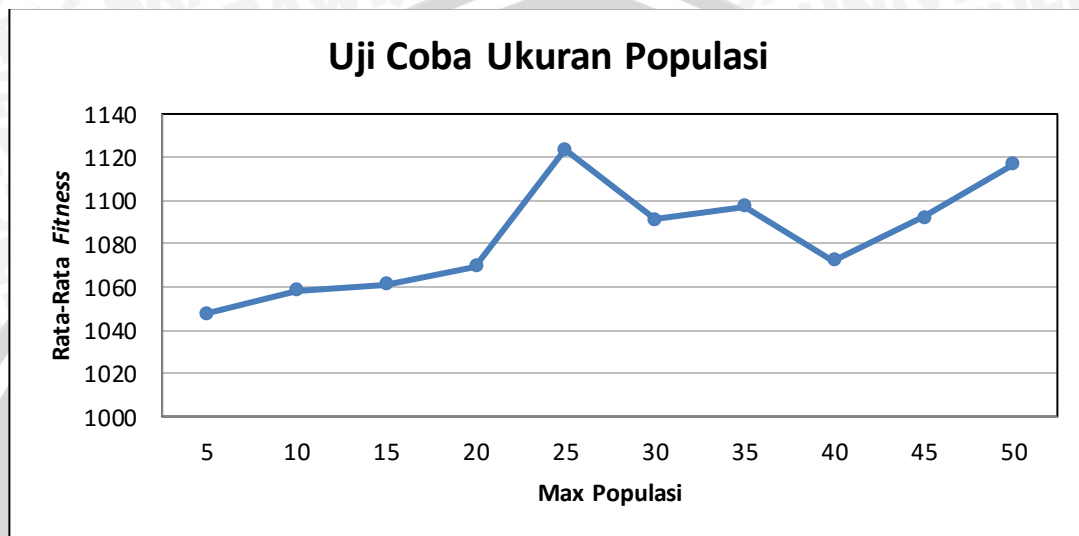
$$\begin{aligned} \text{Iterasi Maksimum} &= 70 \\ \theta_{\min}, \theta_{\max} &= 0.4, 0.9 \\ c_{1i}, c_{1f} &= 2.5, 0.5 \\ c_{2i}, c_{2f} &= 0.5, 2.5 \\ r_1, r_2 &= 0.5, 0.5 \end{aligned}$$

Setiap ukuran populasi akan di uji sebanyak 5 kali. Hasil uji coba terhadap maksimum populasi dapat dilihat pada Tabel 6.2.

Tabel 6.2 Hasil Uji Coba Ukuran Populasi

Ukuran Populasi	Total <i>Fitness</i>					Rata-Rata Total <i>Fitness</i>
	Percobaan ke- <i>i</i>					
	1	2	3	4	5	
5	1039.762	1082.74	1052.771	1034.735	1028.758	1047.753
10	1018.732	1065.763	1049.745	1127.747	1029.768	1058.351
15	1086.750	1056.716	1064.744	1040.712	1055.726	1060.930
20	1066.776	1067.708	1038.784	1076.766	1097.715	1069.550
25	1120.724	1119.751	1125.768	1132.734	1118.757	1123.547
30	1044.724	1106.759	1114.734	1120.731	1068.700	1091.130
35	1095.743	1066.775	1111.718	1083.779	1127.692	1097.141
40	1080.778	1041.739	1052.799	1078.713	1106.724	1072.151
45	1083.775	1080.740	1102.728	1060.764	1132.723	1092.146
50	1139.704	1141.712	1119.710	1081.761	1099.755	1116.528

Pada Tabel 6.2 dapat diketahui bahwa nilai rata-rata total *fitness* yang paling baik dihasilkan saat banyak partikel = 25 dengan rata-rata total *fitness* yang paling besar yaitu 1123.547. Nilai rata-rata total *fitness* cenderung mengalami kenaikan sampai pada partikel sebanyak 25. Banyak partikel yang lebih besar dari 25 tidak ada yang menghasilkan nilai rata-rata total *fitness* yang sama atau lebih baik dari partikel sebanyak 25. Kondisi tersebut dapat digambarkan dengan grafik seperti pada Gambar 6.2.



Gambar 6.2 Grafik Hasil Uji Coba Ukuran Populasi

Berdasarkan grafik pada Gambar 6.2, semakin banyak partikel yang digunakan maka semakin baik solusi yang didapatkan. Partikel yang banyak berarti akan menyediakan pilihan posisi partikel yang banyak juga, sehingga besar kemungkinan untuk mendapatkan solusi optimal karena adanya ruang pencarian yang lebih besar untuk dijelajahi tiap iterasinya. Partikel yang banyak pasti akan membutuhkan waktu komputasi yang lebih lama (Englebercth, 2007). Terkadang komputasi yang lama tidak sebanding dengan peningkatan nilai *fitness* yang didapatkan seperti yang terlihat pada Gambar 6.2, bahwa rata-rata total *fitness* terbesar ketika ditentukan banyak partikel sebanyak 25 sedangkan untuk banyak partikel yang lebih besar tidak didapatkan nilai rata-rata total *fitness* yang lebih baik dari partikel sebanyak 20. Pada umumnya populasi dengan partikel sebanyak 20 sampai 40 sudah mampu mendapatkan nilai yang cukup baik (Zerda, 2009). Jumlah partikel ditentukan sesuai dengan kompleksitas masalah yang dihadapi. Pada uji coba ini didapatkan banyak populasi yang paling optimal adalah 25 karena menghasilkan nilai rata-rata total *fitness* terbesar yaitu 1123.547. Sehingga, penggunaan 25 partikel dirasa cukup optimal untuk permasalahan optimasi yang dihadapi dan akan lebih menghemat waktu dari pada banyak populasi yang lebih besar lagi.

6.3 Hasil dan Analisis Uji Coba Kombinasi Bobot Inersia Minimum dan Maksimum

Pengujian ini bertujuan untuk mengetahui kombinasi nilai minimum (θ_{min}) dan nilai maksimum (θ_{max}) bobot inersia yang paling optimal untuk menghasilkan nilai *fitness* terbaik. Dalam pengujian ini digunakan ukuran populasi adalah 25, karena terbukti optimal pada pengujian sebelumnya, sedangkan kombinasi nilai θ_{max} dan θ_{min} yang digunakan dalam pengujian ini yang dapat dilihat pada Tabel 6.3.

Tabel 6.3 Kombinasi Nilai Bobot Inersia Minimum dan Maksimum

θ_{min}	θ_{max}
0.4	0.5
0.4	0.6
0.4	0.7
0.4	0.8
0.4	0.9
0.5	0.9
0.6	0.9
0.7	0.9
0.8	0.9

Detail nilai parameter PSO yang digunakan adalah sebagai berikut:

Iterasi Maksimum = 70

Ukuran Populasi = 25

c_{1i}, c_{1f} = 2.5, 0.5

c_{2i}, c_{2f} = 0.5, 2.5

r_1, r_2 = 0.5, 0.5

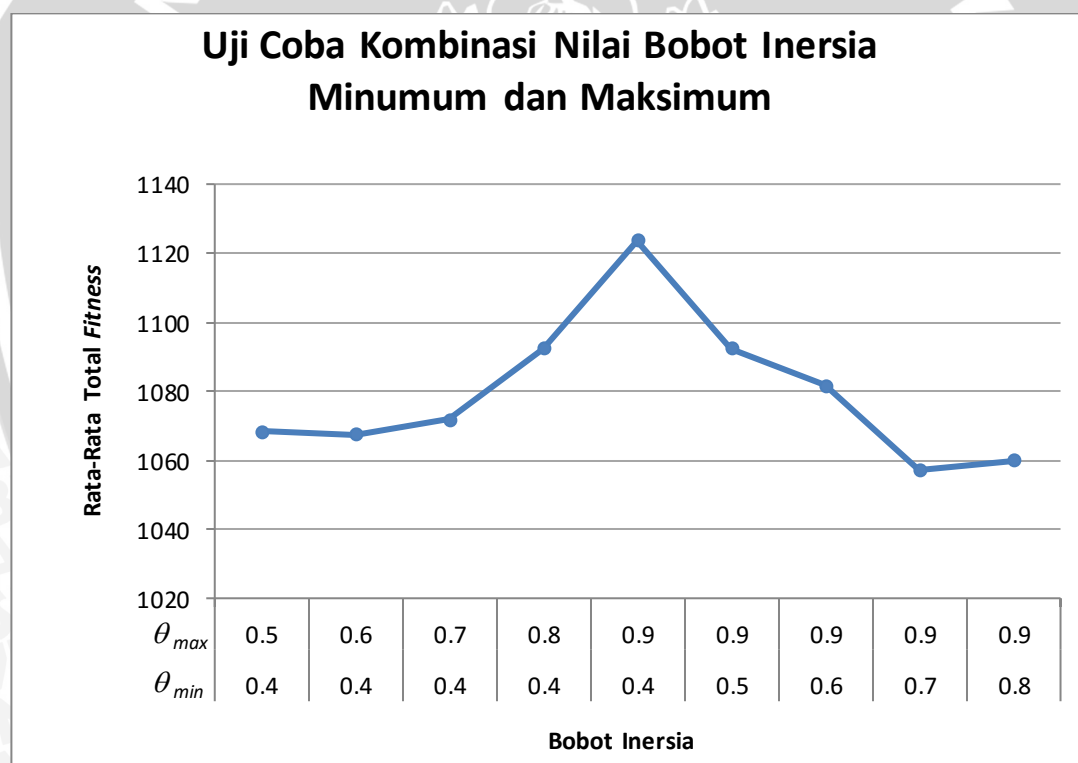
Masing-masing nilai kombinasi akan diuji sebanyak 5 kali percobaan yang hasilnya dapat dilihat seperti pada Tabel 6.4.

Tabel 6.4 Hasil Uji Coba Kombinasi Nilai Bobot Inersia Minimum dan Maksimum

θ_{min}	θ_{max}	Total <i>Fitness</i>					Rata-Rata Total <i>Fitness</i>
		Percobaan ke- i					
		1	2	3	4	5	
0.4	0.5	1072.745	1070.732	1043.713	1049.773	1104.781	1068.349
0.4	0.6	1114.726	1114.723	1018.776	1045.746	1043.738	1067.542
0.4	0.7	1079.749	1030.743	1079.774	1079.708	1088.746	1071.744
0.4	0.8	1170.739	1114.764	1044.751	1067.769	1063.767	1092.358

θ_{min}	θ_{max}	Total Fitness					Rata-Rata Total Fitness
		Percobaan ke-					
		1	2	3	4	5	
0.4	0.9	1120.724	1119.751	1125.768	1132.734	1118.757	1123.547
0.5	0.9	1144.735	1126.739	1069.742	1106.800	1013.779	1092.359
0.6	0.9	1152.745	1035.753	1066.761	1083.721	1069.759	1081.748
0.7	0.9	1061.759	1011.792	1106.732	1047.731	1057.754	1057.154
0.8	0.9	1076.737	1045.729	1066.768	1054.727	1055.767	1059.945

Pada Tabel 6.4 dapat dilihat bahwa setiap kombinasi nilai bobot inersia yang berbeda akan menghasilkan nilai rata-rata total *fitness* berbeda. Kombinasi nilai maksimum dan minimum bobot inersai yang menghasilkan nilai rata-rata total *fitness* terbesar adalah $\theta_{max} = 0.9$ dan $\theta_{min} = 0.4$ yaitu mencapai 1123.547. Nilai rata-rata total *fitness* cenderung menurun ketika selisih atau jarak nilai θ_{max} dan θ_{min} semakin sedikit. Kondisi ini dapat digambarkan dengan grafik seperti Gambar 6.3.



Gambar 6.3 Grafik Hasil Uji Coba Kombinasi Nilai Bobot Inersia Minimum dan Maksimum

Pada Gambar 6.3 dapat dilihat bahwa grafik terus meningkat seiring bertambahnya selisih atau jarak antara nilai θ_{max} dan θ_{min} dan sebaliknya grafik terus menurun seiring berkurangnya selisih nilai θ_{max} dan θ_{min} . Ketika jarak antara nilai θ_{max} dan θ_{min} semakin besar maka jarak nilai penurunan kecepatan dari iterasi ke iterasi yang lain akan juga semakin besar yang berarti kecepatan

partikel semakin diperlambat di awal pencarian. Kecepatan yang semakin diperlambat diawal pencarian akan memberikan kesempatan eksploitasi lokal lebih besar. Eksploitasi akan berguna untuk mencari solusi optimal pada suatu wilayah sebelum melakukan eksplorasi ke wilayah lain, namun jika kesempatan eksploitasi terlalu kecil maka partikel akan cenderung melakukan eksplorasi ke wilayah baru dan kehilangan kesempatan eksploitasi lebih dalam terhadap wilayah-wilayah yang dikunjungi, akibatnya solusi optimal pada suatu wilayah tertentu sering terlewatkan (Novitasari, 2015). Pada hasil uji coba dapat dibuktikan bahwa nilai $\theta_{max} = 0.9$ dan $\theta_{min} = 0.4$ menghasilkan rata-rata total *fitness* terbesar karena jarak/selisih antara kedua nilai tersebut paling besar daripada nilai kombinasi θ_{max} dan θ_{min} lain. Pada penelitian lain juga telah dibuktikan bahwa dengan penentuan nilai $\theta_{max} = 0.9$ dan $\theta_{min} = 0.4$ dapat menghasilkan solusi yang lebih baik untuk permasalahan yang sedang dihadapi (Ratnaweera, Halgamuge, & Watson, 2004).

6.4 Hasil dan Analisis Uji Coba Kombinasi Koefisien Akselerasi

Pada penelitian ini nilai c_1 dan c_2 dihitung menggunakan Persamaan 2.11 dan Persamaan 2.12, sehingga untuk mengetahui nilai c_1 dan c_2 yang paling optimal, dilakukan uji coba terhadap koefisien yang ada dalam Persamaan 2.11 dan Persamaan 2.12 yaitu untuk c_1 adalah nilai c_{1i} dan c_{1f} sedangkan untuk c_2 adalah c_{2i} dan c_{2f} . Nilai yang diuji untuk kombinasi keempat nilai tersebut dapat dilihat pada Tabel 6.5. Untuk ukuran populasi yang digunakan adalah 25, $\theta_{max} = 0.9$ dan $\theta_{min} = 0.4$, karena nilai-nilai tersebut terbukti optimal pada pengujian sebelumnya.

Tabel 6.5 Kombinasi Nilai Koefisien Akselerasi

c_1		c_2	
c_{1i}	c_{1f}	c_{2i}	c_{2f}
2	1	1	2
2.25	0.5	0.5	2.25
2.5	0.5	0.5	2.5
2.5	0.75	0.75	2.5

Sumber : (Ratnaweera, Halgamuge, & Watson, 20)

Detail nilai parameter PSO yang digunakan adalah sebagai berikut:

Iterasi Maksimum = 70

Ukuran Populasi = 25

$\theta_{min}, \theta_{max}$ = 0.4, 0.9

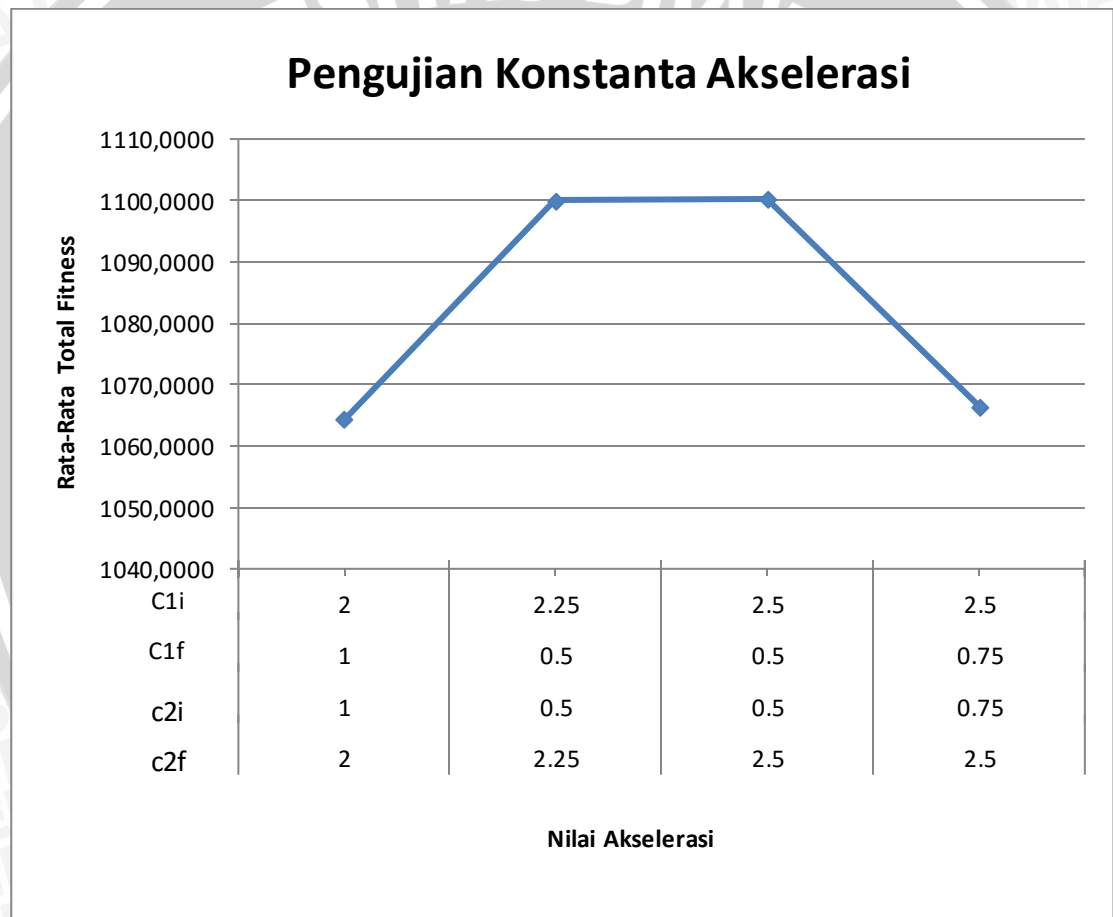
r_1, r_2 = 0.5, 0.5

Masing-masing nilai kombinasi akan diuji sebanyak 10 kali percobaan dengan hasil yang ditunjukkan pada Tabel 6.6 dan lebih lengkapnya dapat dilihat pada Lampiran 5.

Tabel 6.6 Hasil Uji Coba Kombinasi Koefisien Akselerasi

c_1		c_2		Total <i>Fitness</i>				Rata-Rata Total <i>Fitness</i>
				Percobaan ke- i				
c1i	cif	c2i	c2f	1	2		10	
2	1	1	2	1064.7082	1017.7961		1050.7467	1064.2525
2.25	0.5	0.5	2.25	1108.7237	1121.7382		1137.6970	1099.9400
2.5	0.5	0.5	2.5	1120.7239	1119.7505		1054.7426	1100.2398
2.5	0.75	0.75	2.5	1062.7511	1011.7339		1063.7702	1066.4451

Dari hasil uji coba pada Tabel 6.6, diperoleh grafik hasil uji coba untuk mengetahui pengaruh Kombinasi nilai koefisien akselerasi terhadap rata-rata total *fitness* yang ditunjukkan pada Gambar 6.4.



Gambar 6.4 Grafik Hasil Uji Coba Kombinasi Koefisien Akselerasi

Berdasarkan Gambar 6.4, nilai rata-rata total *fitness* cenderung besar saat jarak/selisih nilai antar koefisien akselerasi mendekatai atau bernilai 2. dapat dilihat bahwa nilai rata-rata total *fitness* yang terbaik didapatkan ketika nilai koefisien $c_{1i}, c_{1f} = 2.5, 0.5$ dan koefisien $c_{2i}, c_{2f} = 0.5, 2.5$ yaitu sebesar 1100.2398, dimana selisih antar koefisiennya bernilai 2. Pada umumnya, dari

range 1.5-2 untuk c_1 dan dari range 2-2.5 untuk c_2 , dipilih nilai 2 pada keduanya, karena dapat menghasilkan solusi yang lebih optimal. Nilai 2 dianggap cukup karena tidak terlalu kecil dan tidak terlalu besar karena apabila c_1 dan c_2 yang terlalu besar membutuhkan waktu yang lebih lama untuk mendapatkan solusi optimal karena konvergensi akan lebih lama didapatkan, sedangkan untuk nilai c_1 dan c_2 yang semakin kecil akan mengurangi kemungkinan mendapat solusi paling optimal karena terjadinya konvergensi dini. Pada uji coba penelitian lain juga didapatkan bahwa saat menggunakan nilai $c_{1i}, c_{1f} = 2.5, 0.5$ dan $c_{2i}, c_{2f} = 0.5, 2.5$ dapat menghasilkan nilai rata-rata *fitness* yang lebih baik dengan standar deviasi yang kecil (Ratnaweera, Halgamuge & Watson, 2004).



BAB 7 PENUTUP

Bab Penutup terdiri dari kesimpulan dan saran. Kesimpulan berguna untuk menjelaskan hasil penerapan dan uji coba pada penelitian. Saran bertujuan memberikan masukan yang lebih baik untuk pengembangan sistem ke depannya.

7.1 Kesimpulan

Berdasarkan penelitian dan hasil pengujian didapatkan beberapa kesimpulan sebagai berikut:

1. Implementasi algoritma PSO terbukti mampu digunakan dengan baik untuk sistem optimasi pemerataan guru mata pelajaran di Kabupaten Lumajang. Implementasi algoritma PSO dilakukan melalui beberapa tahap sebagai berikut:
 - Inisialisasi partikel, yaitu mendefinisikan sejumlah partikel sebagai solusi dalam setiap ruang pencarian secara *random*. Posisi partikel mempresentasikan guru yang di mutasi beserta sekolah yang menjadi tujuan mutasinya.
 - Perhitungan nilai *fitness*, yaitu menghitung nilai *fitness* setiap posisi partikel menggunakan fungsi tujuan yang telah ditentukan.
 - *Update* kecepatan, yaitu menghitung kecepatan untuk perpindahan posisi selama proses pencarian.
 - *Update* posisi, menentukan posisi partikel setelah dilakukan perpindahan.
 - Penentuan P_{best} , yaitu mencari posisi terbaik partikel dengan cara mengevaluasi dan membandingkan setiap posisi hasil *update* posisi.
 - Penentuan G_{best} , yaitu mencari solusi terbaik dari posisi terbaik global dengan cara membandingkan posisi seluruh posisi terbaik partikel.
2. Penentuan nilai parameter PSO yang tepat sangat penting untuk mencapai solusi yang terbaik, sehingga dilakukan uji coba terhadap parameter-parameter PSO yang meliputi uji coba iterasi maksimum, ukuran populasi, kombinasi nilai bobot inersia minimum dan maksimum, serta pengujian terhadap kombinasi koefisien akselerasi. Dari hasil pengujian didapatkan parameter-parameter PSO yang paling optimal di antaranya untuk iterasi maksimum =70, ukuran populasi =25, $\theta_{max} = 0.9$, $\theta_{min} = 0.4$, $c_{1i} = 2.5$, $c_{1f} = 0.5$, $c_{2i} = 0.5$, dan $c_{2f} = 2.5$. Optimasi dengan menggunakan parameter PSO paling optimal dapat meningkatkan *fitness* sebesar 8-12%.

7.2 Saran

Saran untuk pengembangan lebih lanjut agar sistem optimasi pemerataan mutasi guru dapat menghasilkan solusi pemerataan yang lebih optimal dan memiliki performa yang lebih baik, antara lain:

1. Menggunakan lebih banyak parameter mutasi guru, agar pertimbangan pemindahan guru tidak hanya didasarkan pada usia, masa kerja, dan jarak.
2. Pencarian jarak antara tempat tinggal guru ke masing-masing sekolah menggunakan data alamat yang sudah dikonversi menjadi data *longitude* dan *latitude*.
3. Sistem optimasi pemerataan guru mata pelajaran bisa dikembangkan untuk penataan dan pemerataan guru mata pelajaran di jenjang pendidikan lain.
4. Sistem optimasi pemerataan guru mata pelajaran bisa dikembangkan dengan menggunakan metode PSO jenis lainnya atau PSO yang dikombinasikan dengan metode lain.
5. Ditambahkan analisis global terhadap hasil optimasi, yakni membandingkan hasil optimasi penataan dan pemerataan oleh sistem dengan hasil penataan dan pemerataan yang dilakukan secara manual, sehingga dapat mengetahui seberapa baik solusi yang dihasilkan algoritma PSO untuk mengoptimasi penataan dan pemerataan guru.



DAFTAR PUSTAKA

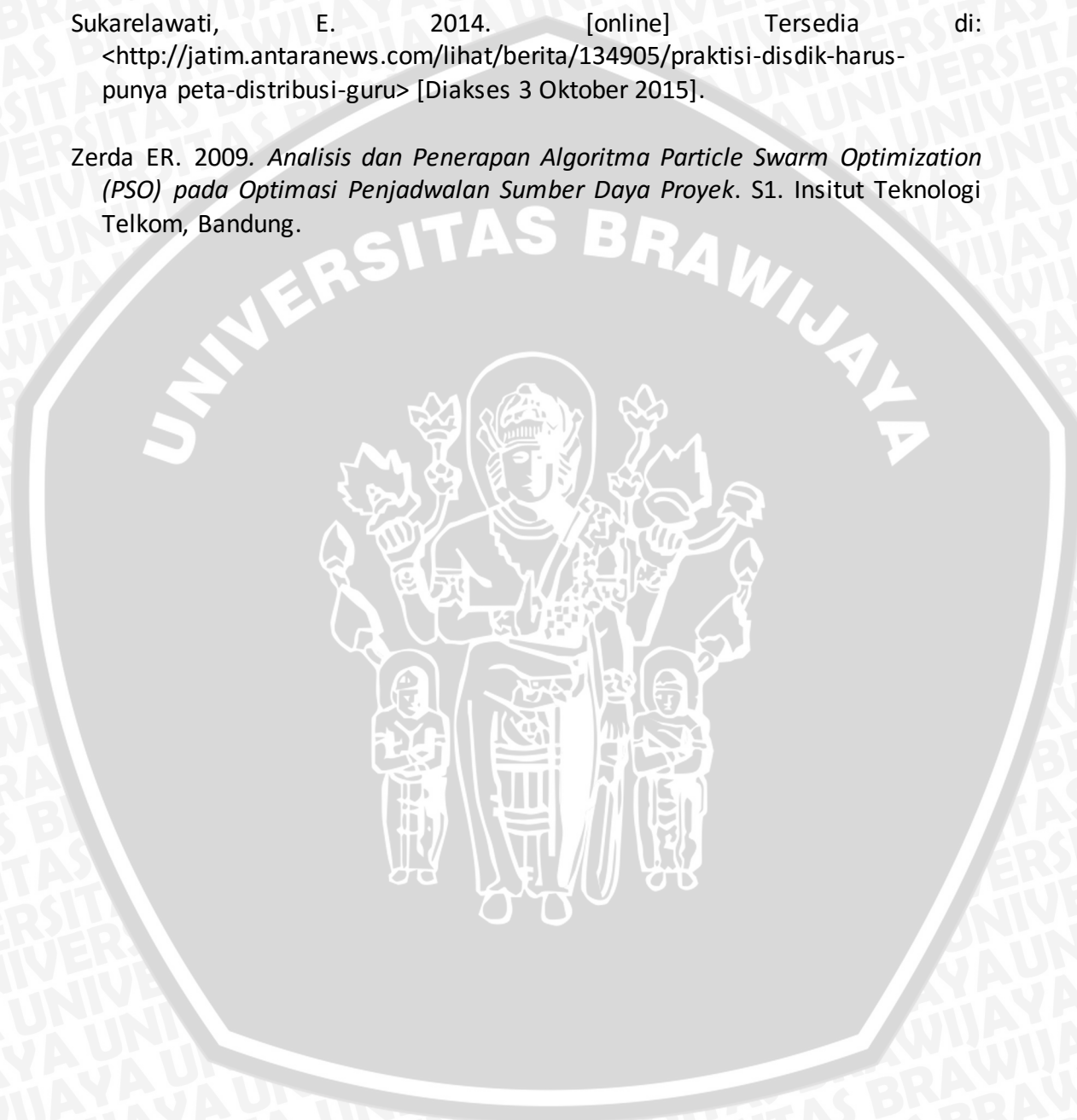
- Akuntono I., 2011. [online] Tersedia di: <<http://edukasi.kompas.com/read/2011/11/29/10143063/SKB.5.Menteri.Fokus.Kelola.Distribusi.Guru>>[Diakses 12 Desember 2015].
- Eberhart R.C, Y. Shi Y., 2011. *Particle Swarm Optimization: Developments, Applications and Resources. Proceedings of 2001 Congress on evolutionary computation.*
- Englebercth AP., 2007. *Computational Intelligent: An Introduction 2nd ed.*, West Sussex: John Willwy & Sons Ltd.
- Huang C., Dun J., 2008. *A Distributed PSO SVM Hybrid System With Feature Seletion and Parameter.* Huaan University, Taiwan.
- Jia Z., Gong L., 2008. *Multi-criteria Human Resource Allocation for Optimization Problems Using Multi-objective Particle Swarm Optimization Algorithm.* IEEE.
- Kementerian Riset, Teknologi, dan Pendidikan Tinggi, 2011. *Petunjuk Teknis Pelaksanaan Peraturan Bersama5 menteri tentang Penataan dan Pemerataan Guru PNS.* [pdf] Kementerian Riset, Teknologi, dan Pendidikan Tinggi. Tersedia di:<<http://www.kopertis12.or.id/wp-content/uploads/2012/01/juknis-peraturanbersama-lima-menteri-tentang-penataan-pemerataan-guru-pns>>[Diakses 12 Desember 2015].
- Mahmudy WF., 2014. *Optimasi Part Type Selection and Machine Loading Problems Pada Fms Menggunakan Metode Particle Swarm Optimization.* Konferensi Nasional Sistem Informasi (KNSI).STMIK Dipanegara, Makassar.
- Novitasari D., 2015. *Optimasi Support Vector Regression dengan Particle Swarm Optimization untuk Software Effort Estimations.* S1. Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
- Paat Y., 2015. [online] Tersedia di: <<http://www.beritasatu.com/pendidikan/299679-di-bidang-pendidikan-indonesia-masih-belum-merdeka.html>>[Diakses 12 Desember 2015].
- Putra OP., 2015. *Implementasi Algoritma Genetika Untuk Optimasi Pemerataan Mutasi Guru Sd Di Kabupaten Banyuwangi.* S1. Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya.
- Ratnaweera A., Halgamuge, SK., Watson HC., 2004. *Self Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients.* IEEE.

Santosa B., Willy P., 2011. *Metoda Metaheuristik, Konsep dan Implementasi*. Graha Ilmu. Surabaya.

Santoso DA., Purnama IK., Sumpeno S., 2013. *Optimasi Distribusi Guru Berbasis Metode Dijkstra*. ITS, Surabaya.

Sukarelawati, E. 2014. [online] Tersedia di:
<<http://jatim.antaranews.com/lihat/berita/134905/praktisi-disdik-harus-punya-peta-distribusi-guru>> [Diakses 3 Oktober 2015].

Zerda ER. 2009. *Analisis dan Penerapan Algoritma Particle Swarm Optimization (PSO) pada Optimasi Penjadwalan Sumber Daya Proyek*. S1. Insitut Teknologi Telkom, Bandung.



LAMPIRAN 1

Daftar Pilihan Mutasi Guru Mata Pelajaran MTK

NO	GURU	MASA KERJA	USIA	JARAK
				SMP 6
1	GURU 1 SMP 2	6	54	5
2	GURU 2 SMP 2	9	30	4.2
3	GURU 3 SMP 2	2	37	5
4	GURU 1 SMP 5	4	47	2.5
5	GURU 2 SMP 5	8	42	1
6	GURU 3 SMP 5	4	52	3
7	GURU 4 SMP 5	7	51	9

Daftar Pilihan Mutasi Guru Mata Pelajaran TIK

NO	GURU	MASA KERJA	USIA	JARAK (km)	
				SMP 2	SMP 5
1	GURU 1 SMP 6	7	40	3	
2	GURU 2 SMP 6	10	38	5	
3	GURU 3 SMP 6	9	32	5.5	
4	GURU 4 SMP 6	3	47	9	
5	GURU 1 SMP 6	7	40		6
6	GURU 2 SMP 6	10	38		2
7	GURU 3 SMP 6	9	32		5
8	GURU 4 SMP 6	3	47		11

Daftar Pilihan Mutasi Guru Mata Pelajaran Bahasa Daerah

NO	GURU	MASA KERJA	USIA	JARAK (km)	
				SMP 2	SMP 6
1	GURU 1 SMP 5	5	44	1.5	
2	GURU 2 SMP 5	9	49	4	
3	GURU 3 SMP 5	5	52	8	
4	GURU 4 SMP 5	3	50	2	
5	GURU 1 SMP 5	5	44		2.5
6	GURU 2 SMP 5	9	49		2
7	GURU 3 SMP 5	5	52		1
8	GURU 4 SMP 5	3	50		6

LAMPIRAN 2

P_{best} per Iterasi

P_{best} Iterasi ke-0

Partikel ke- i	Mata Pelajaran	Dimensi ke- j	P_{best}	<i>Fitness</i>	Total <i>Fitness</i>
1	IPS	1	4	10.27273	54.35447
	MTK	2	3	4.380952	
	TIK	3	7	11.7027	
	TIK	4	2	12.32558	
	BD	5	8	4.785714	
	BD	6	2	10.88679	
2	IPS	1	3	6.702703	51.29008
	MTK	2	3	4.380952	
	TIK	3	3	15.89655	
	TIK	4	6	12.5	
	BD	5	7	6.886792	
	BD	6	4	4.923077	
3	IPS	1	5	11.04082	52.12663
	MTK	2	4	6.020202	
	TIK	3	8	11.7027	
	TIK	4	1	9.325581	
	BD	5	5	7.150538	
	BD	6	7	6.886792	
4	IPS	1	2	9.818182	67.75307
	MTK	2	2	11.92398	
	TIK	3	5	9.173913	
	TIK	4	3	11.66667	
	BD	5	1	20.38462	
	BD	6	8	4.785714	

P_{best} Iterasi ke-1

Partikel ke- j	Mata Pelajaran	Dimensi ke- j	P_{best}	<i>Fitness</i>	Total <i>Fitness</i>
1	IPS	1	3	6.702702703	54.86696054
	MTK	2	2	11.92397661	
	TIK	3	6	12.5	
	TIK	4	3	11.66666667	
	BD	5	4	4.923076923	
	BD	6	5	7.150537634	
2	IPS	1	3	6.702703	51.29007618
	MTK	2	3	4.380952	
	TIK	3	3	15.89655	
	TIK	4	6	12.5	
	BD	5	7	6.886792	
	BD	6	4	4.923077	
3	IPS	1	5	11.04082	52.12663253
	MTK	2	4	6.020202	
	TIK	3	8	11.7027	
	TIK	4	1	9.325581	
	BD	5	5	7.150538	
	BD	6	7	6.886792	
4	IPS	1	2	9.818182	67.75306781
	MTK	2	2	11.92398	
	TIK	3	5	9.173913	
	TIK	4	3	11.66667	
	BD	5	1	20.38462	
	BD	6	8	4.785714	

P_{best} Iterasi 2

Partikel ke- i	Mata Pelajaran	Dimensi ke- j	P_{best}	<i>Fitness</i>	Total <i>Fitness</i>
1	IPS	1	4	6.702702703	54.86696054
	MTK	2	3	11.92397661	
	TIK	3	7	12.5	
	TIK	4	2	11.66666667	
	BD	5	8	4.923076923	
	BD	6	2	7.150537634	
2	IPS	1	2	9.818181818	62.69400717
	MTK	2	1	7.694915254	
	TIK	3	3	11.66666667	
	TIK	4	3	11.66666667	
	BD	5	2	10.88679245	
	BD	6	6	10.96078431	
3	IPS	1	2	9.818181818	61.01523777
	MTK	2	2	11.92397661	
	TIK	3	5	9.173913043	
	TIK	4	2	12.3255814	
	BD	5	2	10.88679245	
	BD	6	7	6.886792453	
4	IPS	1	2	9.818182	67.75306781
	MTK	2	2	11.92398	
	TIK	3	5	9.173913	
	TIK	4	3	11.66667	
	BD	5	1	20.38462	
	BD	6	8	4.785714	

P_{best} Iterasi 3

Partikel ke- i	Mata Pelajaran	Dimensi ke- j	P_{best}	<i>Fitness</i>	Total <i>Fitness</i>
1	IPS	1	4	6.702702703	54.86696054
	MTK	2	3	11.92397661	
	TIK	3	7	12.5	
	TIK	4	2	11.66666667	
	BD	5	8	4.923076923	
	BD	6	2	7.150537634	
2	IPS	1	2	9.818181818	62.69400717
	MTK	2	1	7.694915254	
	TIK	3	3	11.66666667	
	TIK	4	3	11.66666667	
	BD	5	2	10.88679245	
	BD	6	6	10.96078431	
3	IPS	1	2	9.818181818	61.01523777
	MTK	2	2	11.92397661	
	TIK	3	5	9.173913043	
	TIK	4	2	12.3255814	
	BD	5	2	10.88679245	
	BD	6	7	6.886792453	
4	IPS	1	2	9.818182	67.75306781
	MTK	2	2	11.92398	
	TIK	3	5	9.173913	
	TIK	4	3	11.66667	
	BD	5	1	20.38462	
	BD	6	8	4.785714	

G_{best} per Iterasi

Iterasi	Partikel	Gbest			
		Dimensi ke- j	Posisi	Fitness	Total Fitness
0	4	1	2	9.818182	67.75306781
		2	2	11.92398	
		3	5	9.173913	
		4	3	11.66667	
		5	1	20.38462	
		6	8	4.785714	
1	4	1	2	9.818182	67.75306781
		2	2	11.92398	
		3	5	9.173913	
		4	3	11.66667	
		5	1	20.38462	
		6	8	4.785714	
2	4	1	2	9.818182	67.75306781
		2	2	11.92398	
		3	5	9.173913	
		4	3	11.66667	
		5	1	20.38462	
		6	8	4.785714	
3	4	1	2	9.818182	67.75306781
		2	2	11.92398	
		3	5	9.173913	
		4	3	11.66667	
		5	1	20.38462	
		6	8	4.785714	

LAMPIRAN 3

Range Kecepatan $[-v_{max}, v_{max}]$ per Mata Pelajaran

Mata Pelajaran	v_{max}	$-v_{max}$
IPS	4.8	-4.8
MTK	4.2	-4.2
TIK	4.8	-4.8
BD	4.8	-4.8

Kecepatan per Iterasi

Partikel	Iterasi			
	0	1	2	3
1	0	-1.225	-1.49625	0.463896
	0	-1.05	-1.38167	0.442333
	0	-1.63333	-1.76361	0.421847
	0	0.408333	-0.42681	0.63209
	0	-4.08333	-3.36778	-0.39433
	0	3.675	1.712083	0.968479
	0	-0.81667	-0.7525	-0.2801
2	0	-0.7	-0.74417	-0.15579
	0	0.816667	-0.63583	1.819625
	0	-1.63333	-0.81083	-1.15024
	0	-3.26667	-0.9275	-2.89051
	0	2.45	-0.51917	3.200458
	0	-1.86667	-0.8275	-0.96329
3	0	-1.51667	-0.8025	-0.67204
	0	-2.04167	-0.84	-1.10892
	0	1.225	-0.60667	1.609417
	0	-2.45	-0.86917	-1.44871
	0	0.408333	-0.665	0.929833
	0	0	-0.69417	0.590042
4	0	0	-0.69417	0.472333
	0	0	-0.69417	0.590042
	0	0	-0.69417	0.590042
	0	0	-0.69417	0.590042
	0	0	-0.69417	-0.27767
	0	0	-0.69417	0.590042

LAMPIRAN 4

Range Posisi [x_{min} , x_{max}] per Mata Pelajaran

Mata Pelajaran	x_{min}	x_{max}
IPS	1	8
MTK	1	7
TIK	1	8
BD	1	8

Posisi per Iterasi

Populasi	Iterasi			
	0	1	2	3
1	4	3	1	5
	3	2	1	4
	7	6	4	8
	2	3	2	5
	8	4	1	8
	2	5	7	8
2	3	2	2	3
	3	2	1	3
	3	4	3	7
	6	4	3	7
	7	3	2	5
	4	6	6	8
3	5	3	2	4
	4	3	2	4
	8	6	5	8
	1	2	2	3
	5	3	2	4
	7	8	7	8
4	2	2	1	3
	2	2	1	3
	5	5	4	8
	3	3	2	6
	1	1	1	2
	8	8	7	8

LAMPIRAN 5

Hasil Uji Coba Kombinasi Koefesien Akselerasi

c1		c2		Total Fitness										Rata-Rata
				Percobaan ke- <i>i</i>										Total Fitness
c1i	cif	c2i	c2f	1	2	3	4	5	6	7	8	9	10	
2	1	1	2	1064,7082	1017,7961	1041,7492	1088,7561	1090,7506	1053,7334	1086,7813	1042,7480	1104,7558	1050,7467	1064,2525
2,25	0,5	0,5	2,25	1108,7237	1121,7382	1132,7525	1076,7393	1100,7278	1017,7825	1093,7789	1091,7248	1117,7358	1137,6970	1099,9400
2,5	0,5	0,5	2,5	1120,7239	1119,7505	1125,7684	1132,7336	1118,7574	1108,7154	1039,7257	1126,7524	1054,7277	1054,7426	1100,2398
2,5	0,75	0,75	2,5	1062,7511	1011,7339	1077,7363	1093,7631	1115,7487	1110,7483	1035,7294	1043,7626	1048,7072	1063,7702	1066,4451



LAMPIRAN 6

Hasil optimasi tidak menggunakan nilai paremeter optimal PSO

iterasi maksimum=10; ukuran populasi=5; $\theta_{min}, \theta_{max} = 0.4, 0.5$; $c_{1i}, c_{1f} = 2, 1$; $c_{2i}, c_{2f} = 1, 2$;

$r_1, r_2 = 0.5, 0.5$

Nama	Mata Pelajaran	Usia	Masa Kerja	Sekolah Asal	Sekolah Tujuan Mutasi	Fitness
AKSANUL KHAK	Bahasa Inggris	49	27	SMPN 1 PASIRIAN	SMPN 2 TEMPEH	28.96078
ADE ADITYA IRAWAN	Bahasa Inggris	32	6	SMPN 1 TEMPURSARI	SMPN 2 TEMPURSARI	7.041667
SAIFUDIN ZUHRI	Bahasa Inggris	51	26	SMPN 1 KLAKAH	SMPN 2 CANDIPURO	27.88679
AHMAD EFENDY	Bahasa Inggris	48	11	SMPN 2 TEKUNG	SMPN 3 PASIRIAN	13
YUSUF WIJANARKO	Bahasa Inggris	44	19	SMPN 2 PASRUJAMBE	SMPN 2 SUMBERSUKO	4.083333
BM.TEJO PRAMONO	Bahasa Inggris	59	31	SMPN 2 KLAKAH	SMPN 1 PRONOJIWO	31.93458
SUBROTO	Bahasa Inggris	48	21	SMPN 1 RANUYOSO	SMPN 2 PADANG	23
BASTONO SUBEKTI	Bahasa Inggris	48	23	SMPN 2 KLAKAH	SMPN 1 KEDUNGAJANG	20.6129
ENDIARTI SUSILOWATI	Bahasa Inggris	43	7	SMPN 1 SUMBERSUKO	SMPN 2 RANDUAGUNG	8.818182
POERWONO	Bahasa Inggris	54	30	SMPN 1 PASIRIAN	SMPN 2 SUMBERSUKO	31.42857
HALIL	Bahasa Inggris	50	25	SMPN 2 KLAKAH	SMPN 1 PASRUJAMBE	26.28205
MOHAMMAD SYAIFUDDIN	Bahasa Indonesia	47	9	SMPN 1 JATIROTO	SMPN 1 SUKODONO	10.51515
SUBARI	Bahasa Indonesia	55	31	SMPN 3 CANDIPURO	SMPN 2 PRONOJIWO	31.97087
SARWONO	Bahasa Indonesia	57	29	SMPN 4 LUMAJANG	SMPN 2 PASIRIAN	30.69492
MUHAMMAD KHOIRUL RIJAL	Bahasa Indonesia	52	24	SMPN 2 YOSOWILANGUN	SMPN 1 SUKODONO	25.38889
GHONIYUL KHUSNAH	Bahasa Indonesia	54	31	SMPN 4 LUMAJANG	SMPN 1 SUKODONO	32.72414
NINIK USAWATUN CHASANA	Bahasa Indonesia	50	27	SMPN 1 PASIRIAN	SMPN 1 SUKODONO	28.35135
SADELI	Bahasa Indonesia	54	33	SMPN 1 PASIRIAN	SMPN 1 SUKODONO	34.26582
SAMPUN	PMPKn	53	29	SMPN 2 LUMAJANG	SMPN 1 RANDUAGUNG	4.886792
SIYATI	PMPKn	54	27	SMPN 1 KLAKAH	SMPN 2 PRONOJIWO	28.78571
SETYO HADI SANTOSA	PMPKn	56	29	SMPN 1 TEKUNG	SMPN 2 GUCIALIT	30.72414
IMAM ASMUNI	PMPKn	51	28	SMPN 3 CANDIPURO	SMPN 2 PRONOJIWO	29.88679
MOCH. FADJAR SANTOSO	PMPKn	56	34	SMPN 2 PASIRIAN	SMPN 2 GUCIALIT	35.72414
ANDY TRISTIADI	PJOK	52	27	SMPN 1 TEMPURSARI	SMPN 2 SUMBERSUKO	28.85185
WAWAN SYAIFUROI, S.PD	PJOK	44	7	SMPN 1 ROWOKANGKUNG	SMPN 2 PASIRIAN	9.173913
ADI SUSIYANTO	PJOK	51	25	SMPN 2 JATIROTO	SMPN 3 LUMAJANG	26.88679
AKHMAD HASANUDIN	PJOK	48	11	SMPN 1 TEMPURSARI	SMPN 4 LUMAJANG	13
SOCHIF PRASETIYA	Matematika	30	6	SMPN 2 PADANG	SMPN 3 CANDIPURO	9.125
ANJAR SRI WAHYUNI	Matematika	49	26	SMPN 2 KLAKAH	SMPN 1 KLAKAH	27.96078
ENNY WIDYAWATI	Matematika	47	11	SMPN 5 LUMAJANG	SMPN 2 PRONOJIWO	13.04082
ADI CAHYANTO	Matematika	48	13	SMPN 2 YOSOWILANGUN	SMPN 2 CANDIPURO	14.1236
SIDIK WINOTO	Matematika	52	31	SMPN 1 TEMPURSARI	SMPN 1 KLAKAH	31.86207
EDIANA KOESTATININGTYAS	Matematika	54	21	SMPN 2 YOSOWILANGUN	SMPN 1 PASRUJAMBE	22.78571
PRIYO HADI IMANTO	Matematika	46	19	SMPN 3 TEMPEH	SMPN 1 KLAKAH	20.5625
ENDAH WULANDARI	IPA	45	11	SMPN 3 LUMAJANG	SMPN 3 YOSOWILANGUN	12.69492
YULIASIH	IPA	47	20	SMPN 1 TEKUNG	SMPN 2 PASRUJAMBE	21.38889
EKO PURWANTORO	IPA	44	18	SMPN 1 LUMAJANG	SMPN 3 YOSOWILANGUN	19.72414
NINIK QURROTUL AINI	IPA	55	32	SMPN 1 YOSOWILANGUN	SMPN 2 PRONOJIWO	32.97087
AHWAN FAQIH	IPA	58	33	SMPN 1 JATIROTO	SMPN 3 YOSOWILANGUN	34.25
MUALIMAH	IPS	55	27	SMPN 2 SUKODONO	SMPN 3 CANDIPURO	28.75439
TRI RETNO WIDIASTUTI	IPS	43	8	SMPN 1 RANUYOSO	SMPN 2 CANDIPURO	9.388889
SRI HARTINI	IPS	46	19	SMPN 1 TEMPEH	SMPN 3 CANDIPURO	20.78571
MOHAMMAD NURUL HABIBIE	IPS	33	7	SMPN 1 SENDURO	SMPN 2 CANDIPURO	9.272727
YULIANTO	Seni Budaya	48	18	SMPN 1 RANUYOSO	SMPN 2 YOSOWILANGUN	20
ACHMAD SUBIANTO	Seni Budaya	51	27	SMPN 1 RANUYOSO	SMPN 1 TEKUNG	28.88679
Total Fitness						1002.458

Hasil optimasi menggunakan nilai parameter optimal PSO

iterasi maksimum=70; ukuran populasi=25; $\theta_{min}, \theta_{max} = 0.4, 0.9$; $c_{1i}, c_{1f} = 2.5, 0.5$; $c_{2i}, c_{2f} = 0.5, 2.5$; $r_1, r_2 = 0.5, 0.5$

Nama	Mata Pelajaran	Usia	Masa Kerja	Sekolah Asal	Sekolah Tujuan Mutasi	Fitness
SAIFUDIN ZUHRI	Bahasa Inggris	51	26	SMPN 1 KLAKAH	SMPN 1 TEMPEH	27.492537
YUSUF WIJANARKO	Bahasa Inggris	44	19	SMPN 2 PASRUJAMBE	SMPN 2 SUMBERSUKO	4.0833333
ENDIARTI SUSILOWATI	Bahasa Inggris	43	7	SMPN 1 SUMBERSUKO	SMPN 1 PASRUJAMBE	8.7241379
BASTONO SUBEKTI	Bahasa Inggris	48	23	SMPN 2 KLAKAH	SMPN 2 PADANG	25
VIBRIAN IKHWAN AL-KHASANI	Bahasa Inggris	32	11	SMPN 2 PASRUJAMBE	SMPN 3 YOSOWILANGUN	13.941176
AHMAD EFENDY	Bahasa Inggris	48	11	SMPN 2 TEKUNG	SMPN 1 TEMPEH	21.173913
LULUK KHURROTUL INSIYAH	Bahasa Inggris	44	16	SMPN 1 SUMBERSUKO	SMPN 2 CANDIPURO	5.0526316
ARIF WIRDANA	Bahasa Inggris	43	15	SMPN 3 CANDIPURO	SMPN 2 CANDIPURO	17.222222
AKSANUL KHAK	Bahasa Inggris	49	27	SMPN 1 PASIRIAN	SMPN 2 SUMBERSUKO	28.694915
SUBROTO	Bahasa Inggris	48	21	SMPN 1 RANUYOSO	SMPN 2 YOSOWILANGUN	23
EVI INDRAMATI	Bahasa Inggris	46	11	SMPN 1 SUMBERSUKO	SMPN 2 RANDUAGUNG	12.5625
NAIMAH	Bahasa Indonesia	54	32	SMPN 4 LUMAJANG	SMPN 1 SUKODONO	33.785714
CANDRA HASAN	Bahasa Indonesia	55	30	SMPN 1 PRONOJIWO	SMPN 2 TEMPEH	31.754386
SUSIYONO	Bahasa Indonesia	56	20	SMPN 2 YOSOWILANGUN	SMPN 3 SENDURO	21.724138
ENDANG SULASTRININGSIH	Bahasa Indonesia	45	20	SMPN 1 RANUYOSO	SMPN 2 PRONOJIWO	21.25
NANING MAGHFIROH	Bahasa Indonesia	38	13	SMPN 3 PASIRIAN	SMPN 2 RANDUAGUNG	14.639344
PARDJITO	Bahasa Indonesia	56	30	SMPN 1 KLAKAH	SMPN 2 TEMPEH	31.724138
AGUSTINA WULANDARI	Bahasa Indonesia	47	11	SMPN 2 YOSOWILANGUN	SMPN 2 PASIRIAN	13.040816
INDRA PANCA SEPUTRA	PMPKn	54	29	SMPN 3 PASIRIAN	SMPN 2 GUCIALIT	30.785714
SUGENG PRASETYO	PMPKn	58	33	SMPN 2 SUKODONO	SMPN 1 YOSOWILANGUN	34.666667
YULIARTO	PMPKn	54	23	SMPN 2 YOSOWILANGUN	SMPN 2 PRONOJIWO	24.785714
AMI SUHARTI	PMPKn	56	30	SMPN 5 LUMAJANG	SMPN 2 GUCIALIT	31.724138
MOCH. FADJAR SANTOSO	PMPKn	56	34	SMPN 2 PASIRIAN	SMPN 2 PRONOJIWO	35.724138
BAGYA HERMAWAN	PJOK	53	27	SMPN 2 JATIROTO	SMPN 1 SENDURO	28.818182
ENDAH YUNIARTI	PJOK	45	18	SMPN 1 SUMBERSUKO	SMPN 2 SUKODONO	20.12766
ANDY TRISTIADI	PJOK	52	27	SMPN 1 TEMPURSARI	SMPN 1 SENDURO	28.851852
ADI SUSIYANTO	PJOK	51	25	SMPN 2 JATIROTO	SMPN 1 PASRUJAMBE	26.886792
EDY WINARTO	Matematika	59	37	SMPN 1 TEMPEH	SMPN 1 PASRUJAMBE	38.639344
IRAWANTI	Matematika	49	19	SMPN 1 JATIROTO	SMPN 2 CANDIPURO	20.25
SUTOMO	Matematika	62	38	SMPN 2 RANDUAGUNG	SMPN 3 CANDIPURO	39.5625
IDAM DJUNAEDI, S.PD	Matematika	49	18	SMPN 1 ROWOKANGKUNG	SMPN 2 PRONOJIWO	19.030928
SUHARIANTO, S.PD	Matematika	58	35	SMPN 1 ROWOKANGKUNG	SMPN 2 CANDIPURO	35.917431
SAIFUL RIDJAL	Matematika	54	27	SMPN 2 SUKODONO	SMPN 3 CANDIPURO	28.785714
PRIYO HADI IMANTO	Matematika	46	19	SMPN 3 TEMPEH	SMPN 1 PASRUJAMBE	21.083333
NURUL HIDAYATI	IPA	44	17	SMPN 2 TEKUNG	SMPN 2 PASRUJAMBE	18.470588
ADI SISWANTONO	IPA	49	11	SMPN 1 JATIROTO	SMPN 2 PRONOJIWO	11.877193
NINIK ENDAH SULISTIOWATI	IPA	56	32	SMPN 1 TEMPURSARI	SMPN 1 PASRUJAMBE	32.952381
AGUS TURBANDONO	IPA	55	34	SMPN 1 PASIRIAN	SMPN 2 PRONOJIWO	35.333333
SITI ASIAH	IPA	58	32	SMPN 1 YOSOWILANGUN	SMPN 2 PASRUJAMBE	33.075269
KANIAWATI	IPS	54	31	SMPN 2 SUKODONO	SMPN 2 CANDIPURO	32.136364
NURUL AKHMAD YANI	IPS	45	18	SMPN 2 SUKODONO	SMPN 1 PRONOJIWO	19.041667
LILIS SUMARNI	IPS	52	8	SMPN 1 SUMBERSUKO	SMPN 2 CANDIPURO	9.2658228
LILIK SUPRIYANTA	IPS	51	20	SMPN 3 YOSOWILANGUN	SMPN 1 PRONOJIWO	20.961538
YULIANTO	Seni Budaya	48	18	SMPN 1 RANUYOSO	SMPN 2 SUMBERSUKO	20
ACHMAD SUBIANTO	Seni Budaya	51	27	SMPN 1 RANUYOSO	SMPN 3 SENDURO	28.886792
Total fitness						1082.507