

**IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMASI  
PEMERATAAN MUTASI GURU SD DI KABUPATEN BANYUWANGI**

**SKRIPSI**

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

**OKY PRIMADE SAPTO**

**NIM. 105060800111101**

PROGRAM STUDI TEKNIK INFORMATIKA

PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2015

## LEMBAR PERSETUJUAN

### IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMASI PEMERATAAN MUTASI GURU SD DI KABUPATEN BANYUWANGI

### SKRIPSI

Diajukan untuk memenuhi persyaratan memperoleh gelar Sarjana Komputer



Disusun Oleh :

**OKY PRIMADE SAPTO**

**NIM. 105060800111101**

Skripsi ini telah disetujui oleh dosen pembimbing pada tanggal 10 Juni 2015

Dosen Pembimbing I

Dosen Pembimbing II

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

NIP. 197209191997021001

Edy Santoso, S.Si., M.Kom

NIP. 19740414 200312 1 004

## LEMBAR PENGESAHAN

**IMPLEMENTASI ALGORITMA GENETIKA UNTUK OPTIMASI  
PEMERATAAN MUTASI GURU SD DI KABUPATEN BANYUWANGI**

**SKRIPSI**

**LABORATORIUM KOMPUTASI DAN SISTEM CERDAS**

Untuk memenuhi sebagian persyaratan untuk mencapai gelar Sarjana Komputer

Disusun oleh:

**OKY PRIMADE SAPTO**

**105060800111101**

Skrripsi ini telah dinyatakan lulus pada tanggal 26 Juni 2015

**Penguji I,**

**Penguji II,**

**Suprpto, S.T., M.T.**

**NIP. 19710727 199603 1 001**

**Ismiarta Aknuranda, S.T., M.Sc.,**

**Ph.D.**

**NIK. 74071906110079**

**Penguji III,**

**Indriati, S.T., M.Kom.**

**NIK. 83101306120035**

**Mengetahui,**

**Ketua Program Studi Informatika/Ilmu Komputer**

**Drs. Mardji, MT**

NIP. 19670801 199203 1 001



## PERNYATAAN ORISINALITAS SKRIPSI

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

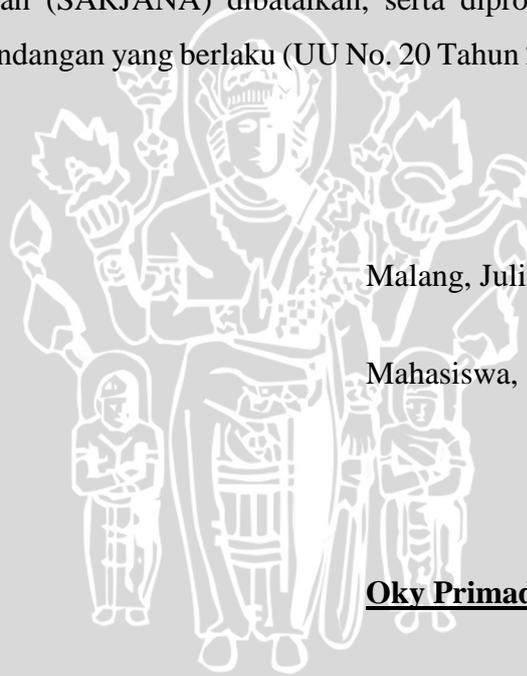
Apabila ternyata di dalam naskah SKRIPSI ini dapat dibuktikan terdapat unsur-unsur PLAGIASI, saya bersedia SKRIPSI ini digugurkan dan gelar akademik yang telah saya peroleh (SARJANA) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat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

Malang, Juli 2015

Mahasiswa,

**Oky Primade Spto**

**NIM. 105060800111101**



## KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih lagi Maha Penyayang. Segala puji bagi Allah SWT karena atas rahmat dan hidayahNya penulis dapat menyelesaikan Tugas Akhir yang berjudul “**Implementasi Algoritma Genetika Untuk Optimalisasi Pemerataan Mutasi Guru SD di Kabupaten Banyuwangi**”. Shalawat atas junjungan Nabi Muhammad S.A.W beserta keluarga dan sahabatnya sekalian. Tugas akhir ini disusun untuk memenuhi sebagai persyaratan gelar Sarjana Komputer di Program Studi Informatika Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terima kasih penulis sebesar-besarnya kepada pihak yang telah memberikan bantuan lahir maupun batin selama penulisan tugas akhir ini. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih penulis kepada :

1. Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., selaku dosen pembimbing I dan Edy Santoso, S.Si., M.Kom. selaku dosen pembimbing II yang telah memberikan bimbingan, ilmu dan saran dalam penyusunan tugas akhir ini.
2. Orang tua Penulis, Sarbini dan Insiyah yang telah memberikan motivasi, kasih sayang serta dukungan moril dan materil kepada penulis.
3. Abdul Latif, S.Pd. selaku Kepala SDN 2 Lemahbangdewo dan Mukhlishin selaku Kasubbag Mutasi Badan Kepegawaian Daerah Kabupaten Banyuwangi yang telah meluangkan waktunya untuk membantu proses penelitian ini.
4. Raden Arief Setyawan, S.T., M.T., selaku dosen pembimbing akademik yang memberikan bimbingan, ilmu dan saran selama penulis belajar.
5. Seluruh dosen dan staff Program Studi Teknik Informatika atas kesediaan membagi ilmunya kepada penulis.
6. Semua teman-teman angkatan 2010 terima kasih atas segala bantuan selama menjadi mahasiswa.

7. Semua warga kos MT Hariyono Gang X1 No 356 yang membantu memberikan dukungan sehingga terselesaikannya skripsi ini.
8. Semua keluarga KMOBM yang memberikan dorongan untuk menyelesaikan skripsi ini dan menjadi keluarga kecil di Malang.
9. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT. Penulis menyadari bahwa tugas akhir ini masih banyak kekurangan dan masih jauh dari kata sempurna. Untuk itu, saran dan kritik yang membangun, sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.



Malang, 10 Juni 2015

Penulis

## ABSTRAK

**Oky Primade Sapto. 2015 : Implementasi Algoritma Genetika Untuk Optimasi Pemerataan Guru SD di Kabupaten Banyuwangi. Skripsi Program Studi Teknik Informatika/Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya. Pembimbing : Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. dan Edy Santoso, S.Si., M.Kom.**

Pendidikan adalah usaha sadar dan terencana untuk mengembangkan potensi diri agar tercipta seseorang yang berkualitas dan berkarakter. Di Indonesia, pendidikan nasional bertujuan untuk mencerdaskan kehidupan bangsa dan warga negara berhak untuk mendapatkan pendidikan seperti tercantum dalam UUD 1945. Kenyataannya, banyak sekali kendala untuk melaksanakan pendidikan yang seperti kita harapkan, terutama pada Sekolah Dasar yang merupakan awal mula seseorang mendapatkan pendidikan secara formal. Menurut Anies Baswedan, untuk meningkatkan kualitas pendidikan di Indonesia, peningkatan kualitas guru adalah yang harus diutamakan. Salah satu upaya yang sering dilakukan tetapi tidak membuahkan hasil yang optimal adalah mutasi guru. Solusi yang dapat diberikan adalah dengan membuat system optimasi pemerataan guru SD.

Optimasi yang dilakukan menggunakan metode Algoritma Genetika. Algoritma genetika (AG) adalah algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah yang terdiri dari beberapa fase yaitu pencarian nilai fitness, *crossover*, dan mutasi sehingga menghasilkan individu baru yang lebih baik dan nilai yang mendekati optimal. Output yang dihasilkan algoritma genetika adalah sebuah solusi optimal yang direpresentasikan dalam sebuah kromosom permutasi. Pada algoritma genetika ini menggunakan metode one cut point *crossover* dan reciprocal exchange *mutation* sedangkan untuk metode seleksi menggunakan elitism. Dari hasil penelitian didapatkan hasil paling optimal ukuran populasi sebanyak 40, jumlah generasi 400, kombinasi *crossover rate* dan *mutation rate* secara berturut-turut 0.5 dan 0.5.

**Kata Kunci :** Algoritma Genetika, Mutasi Guru, Optimasi

## ABSTRACT

**Okky Primade Sapto. 2015 : *Implementation of Genetic Algorithm for Optimization Equalizing Elementary School Teachers in Banyuwangi City. Study program of Information Engineering / Computer Science, Information Technology and Computer Science Program of Brawijaya University. Advisors : Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. dan Edy Santoso, S.Si., M.Kom.***

*Education is a conscious and deliberate effort to develop own potential in order to create a quality and characterized human. In Indonesia, national education aims to educate the nation and its citizen is entitled to get education as stated in UUD 1945. In fact, there are many obstacles for implementing education as our expectation, especially in the elementary school which is beginning of a person's formal education. According Anies Baswedan, improving the quality of teachers is preferred to increase the quality of education in Indonesia. One of those efforts often done but it does not produce optimal results is mutation of teachers. The Solution that can be given is creating the optimization of equalization elementary school teacher system.*

*Optimization is performed using Genetic Algorithms. Genetic Algorithms (GA) are search algorithms based on the mechanism of natural selection and natural genetics consisting of several phases, namely the search value of fitness, crossover, and mutation to produce new and better individual and a value close to the optimum. Output generated genetic algorithm is an optimal solution that is represented in a chromosome permutation. At this genetic algorithm using one cut point crossover and mutation reiprincipal exchange for selection method while using elitism. This research showed that the results of the most optimal population was 40, the amount of generation was 400, a combination of crossover rate and mutation rate was respectively 0.5 and 0.5.*

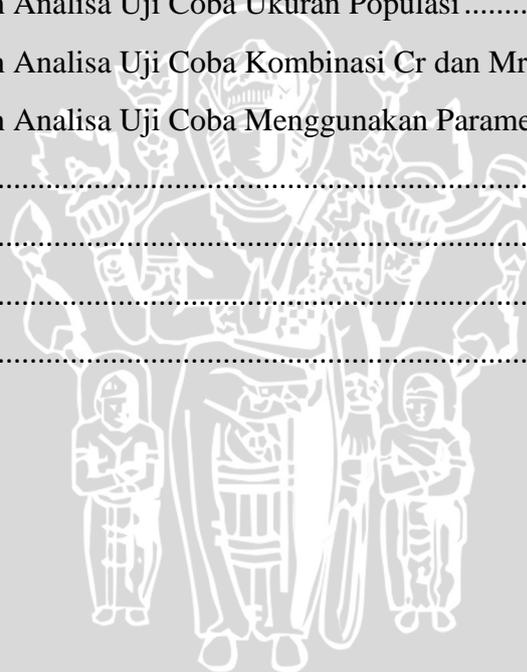
**Keywords :** *Genetic Algorithm, Mutation of Teacher, Optimization.*

## DAFTAR ISI

KATA PENGANTAR .....	i
ABSTRAK .....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR .....	viii
DAFTAR TABEL.....	ix
1. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	4
1.3. Batasan Masalah.....	4
1.4. Tujuan.....	5
1.5. Manfaat.....	5
1.6. Sistematika Penulisan.....	6
2. KAJIAN PUSTAKA DAN TEORI.....	7
2.1. Pengertian Guru.....	7
2.2. Penilaian Kinerja Guru .....	8
2.2.1. Sasaran Kerja Pegawai.....	8
2.2.2. Penilaian Perilaku Kerja.....	14
2.3. Mutasi Guru.....	15
2.4. Proses Mutasi Guru .....	18
2.5. Algoritma Genetika .....	19
2.6. Struktur Umum Algoritma Genetika .....	20
2.7. Parameter-Parameter Algoritma Genetika .....	21
2.7.1. Ukuran Populasi .....	21
2.7.2. Jumlah Generasi .....	22
2.7.3. <i>Crossover Rate</i> .....	22
2.7.3. <i>Mutasi Rate</i> .....	22
2.8. Penerapan Algoritma Genetika.....	23
2.8.1. Membangun Generasi Awal.....	23

2.8.2.	Representasi Kromosom .....	23
2.8.3.	Operator Genetika .....	24
2.8.4.	Seleksi .....	26
2.9.	Bahasa Pemrograman C# .....	26
2.10.	Database .....	27
3.	METODOLOGI PENELITIAN .....	29
3.1.	Studi Literatur .....	30
3.2.	Analisa Kebutuhan .....	30
3.3.	Perancangan Perangkat Lunak .....	31
3.4.	Pengumpulan Data .....	31
3.5.	Implementasi .....	32
3.6.	Perancangan Uji Coba .....	32
3.7.	Pengambilan Kesimpulan .....	32
4.	PERANCANGAN .....	33
4.1.	Formulasi Permasalahan .....	33
4.2.	Manualisasi .....	34
4.2.1.	Perhitungan Manual .....	34
4.2.2.	Membuat Populasi Awal .....	37
4.2.3.	Menghitung Nilai Fitness .....	44
4.3.	Proses Algoritma Genetika .....	49
4.2.1.	<i>Crossover</i> .....	49
4.2.2.	Mutasi .....	51
4.2.3.	Evaluasi .....	53
4.2.4.	Seleksi .....	55
4.	IMPLEMENTASI .....	58
5.1	Implementasi Program .....	58
5.1.1	Pembangkitan Populasi Awal .....	58
5.1.2	Proses <i>Crossover</i> .....	59
5.1.3.	Proses Mutasi .....	60
5.1.4.	Perhitungan Nilai <i>Fitness</i> .....	61

5.1.5.	Proses Seleksi.....	64
5.2.	Implementasi Antarmuka Pengguna.....	64
5.2.1.	Implementasi Halaman Data Guru.....	65
5.2.2.	Implementasi Halaman Data Sekolah.....	66
5.2.3.	Implementasi Halaman Mutasi Guru.....	67
5.2.4.	Implementasi Halaman Proses Algoritma Genetika.....	68
6.	PENGUJIAN DAN ANALISA.....	69
6.1	Hasil Pengujian Sistem.....	69
6.2	Hasil dan Analisa Pengujian Parameter Algoritma Genetika.....	70
6.2.1	Hasil dan Analisa Uji Coba Banyaknya Generasi.....	71
6.2.2	Hasil dan Analisa Uji Coba Ukuran Populasi.....	72
6.2.3	Hasil dan Analisa Uji Coba Kombinasi Cr dan Mr.....	74
6.2.4	Hasil dan Analisa Uji Coba Menggunakan Parameter Terbaik.....	76
6.	PENUTUP.....	78
6.1.	Kesimpulan.....	78
6.2.	Saran.....	78
	DAFTAR PUSTAKA.....	80



## DAFTAR GAMBAR

Gambar 2.1 Struktur Populasi .....	20
Gambar 2.2 Siklus Algoritma Genetika .....	21
Gambar 3.1 Diagram Blok Metode Penelitian .....	29
Gambar 3.2 Arsitektur Perancangan Sistem .....	31
Gambar 4.1 Proses Crossover .....	49
Gambar 4.2 Sub Proses One-cut point crossover .....	50
Gambar 4.3 Proses Mutasi .....	51
Gambar 4.4 Subproses Reciprocal Exchange Mutation .....	52
Gambar 4.5 Proses Evaluasi Fitness .....	54
Gambar 4.6 Proses Elitism .....	56
Gambar 5.1 Implementasi Halaman Data Guru .....	65
Gambar 5.2 Implementasi Halaman Penilaian Kinerja Guru SKP .....	66
Gambar 5.3 Implementasi Halaman Penilaian Kinerja Guru PPK .....	66
Gambar 5.4 Implementasi Halaman Data Sekolah .....	67
Gambar 5.5 Implementasi Halaman Mutasi .....	67
Gambar 5.6 Implementasi Halaman Proses Algoritma Genetika .....	68
Gambar 6.1 Grafik Uji Coba Banyak Generasi .....	72
Gambar 6.2 Grafik Uji Coba Ukuran Populasi .....	74
Gambar 6.3 Grafik Uji Coba Kombinasi Crossover Rate dan Mutation Rate .....	76
Gambar 6.4 Input Parameter Terbaik .....	77
Gambar 6.5 Output Sistem Menggunakan Parameter Terbaik .....	77

## DAFTAR TABEL

Tabel 2.1 Contoh Penilaian Menggunakan SKP .....	11
Tabel 2.2 Contoh Penilaian Perilaku Kerja.....	15
Tabel 2.3 Representasi Kromosom .....	24
Tabel 2.4 Contoh Proses Crossover .....	25
Tabel 2.5 Contoh Proses Mutasi .....	25
Tabel 4.1 Rentang Nilai Kualitas Guru.....	33
Tabel 4.2 Perhitungan Nilai Prestasi Kerja.....	37
Tabel 4.3 Tabel Kinerja Pegawai .....	39
Tabel 4.4 Guru Terpilih.....	40
Tabel 4.5 Inisialisasi Guru Terpilih .....	41
Tabel 4.6 Representasi Kromosom .....	41
Tabel 4.7 Contoh Pembangkitan Kromosom Secara Acak.....	42
Tabel 4.8 Hasil Konversi Kromosom Menjadi Solusi .....	43
Tabel 4.9 Contoh Perhitungan Nilai Fitness .....	46
Tabel 4.10 Proses Crossover .....	51
Tabel 4.11 Proses Mutasi .....	53
Tabel 4.12 Hasil Evaluasi .....	55
Tabel 4.13 Tabel Proses Seleksi Elitism .....	57
Tabel 4.14 Populasi Baru Untuk Generasi Berikutnya .....	57
Tabel 6.1 Guru yang Dimutasi Pada Pengujian .....	69
Tabel 6.2 Hasil Pengujian Sistem .....	70
Tabel 6.3 Hasil Percobaan Banyak Generasi .....	71
Tabel 6.4 Hasil Percobaan Ukuran Populasi.....	73
Tabel 6.5 Hasil Percobaan Kombinasi Crossover Rate dan Mutation Rate.....	75

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Pendidikan adalah usaha sadar dan terencana untuk mewujudkan suasana belajar dalam proses pembelajaran agar peserta didik secara aktif mengembangkan potensi diri untuk memiliki kekuatan spiritual keagamaan, pengendalian diri, kepribadian, kecerdasan, akhlak mulia serta keterampilan yang diperlukan oleh diri sendiri, masyarakat dan bangsa. Tujuan dari pendidikan itu sendiri adalah menciptakan seseorang yang berkualitas dan berkarakter sehingga memiliki pandangan yang luas kedepan untuk mencapai suatu cita-cita yang diharapkan dan mampu beradaptasi secara cepat dan tepat di dalam berbagai lingkungan.

Pendidikan nasional memiliki peranan yang sangat penting bagi warga negara. Pendidikan nasional bertujuan untuk mencerdaskan kehidupan bangsa dan mengembangkan manusia Indonesia seutuhnya, yaitu manusia yang beriman dan bertakwa terhadap Tuhan Yang Maha Esa dan berbudi pekerti luhur, memiliki pengetahuan dan ketrampilan, kesehatan jasmani dan rohani, kepribadian yang mantap dan mandiri serta rasa tanggung jawab kemasyarakatan dan kebangsaan. Oleh karena itu setiap warga Negara berhak untuk mendapatkan pendidikan. Seperti tercantum didalam Undang Undang Dasar 1945 dinyatakan bahwa setiap warga negara mempunyai kesempatan yang sama memperoleh pendidikan.

Dalam praktiknya, ternyata banyak sekali kendala untuk melaksanakan pendidikan yang seperti kita harapkan. Menurut Anies Baswedan, untuk meningkatkan kualitas pendidikan di Indonesia, peningkatan kualitas guru adalah yang harus diutamakan. Menurutnya guru merupakan ujung tombak masalah kualitas pendidikan di Indonesia. Tambahnya lagi, terdapat tiga masalah terkait guru di Indonesia adalah distribusi penempatan guru yang tidak merata, kualitas guru yang juga tidak merata, serta kesejahteraan guru yang tidak memadai. [HER-13]

Salah satu kasus yang sering terjadi dalam kehidupan nyata yang merupakan salah satu solusi tetapi tidak membuahkan hasil yang seperti diharapkan adalah adanya proses perpindahan tenaga pengajar atau yang sering disebut mutasi. Mutasi merupakan salah satu hal yang lazim dilakukan dalam upaya pemerataan tenaga pengajar dalam suatu daerah. Selain itu proses ini juga bertujuan agar tenaga pengajar memahami posisinya, terkait dengan perhitungan penataan dan ketercukupan tenaga pengajar pada suatu sekolah. Tanpa adanya mutasi, proses pemerataan ini mustahil dilakukan. Dalam upaya pemerataan ini di satu sisi jika dikaitkan dengan efektivitas kinerja sangat beralasan, namun disisi lainnya muncul dampak yang negatif apabila tidak disertai dengan perencanaan yang matang. Pada pelaksanaannya proses mutasi tenaga pengajar muncul permasalahan yaitu pemerataan yang kurang baik atau kurang optimal, dimana tenaga pengajar yang mempunyai kinerja yang baik berkumpul pada satu sekolah sehingga sekolah tersebut akan menghasilkan peserta didik yang baik dibandingkan dengan sekolah yang memiliki tenaga pengajar dengan kinerja yang kurang.

Berbagai alasan muncul sebagai sebab mengapa seorang tenaga pengajar mengalami mutasi. Tetapi dari pihak tenaga pengajar sendiri, mereka tidak tahumenuh alasan mereka dimutasi, mereka menganggap itu merupakan “nasib”. Akibat dari mutasi ini, banyak tenaga pengajar yang merasa lebih optimal dalam mengajar tetapi tidak sedikit pula yang merasa kurang optimal dalam mengajar setelah dimutasi. Hal tersebut sudah pasti akan berpengaruh kepada pendidikan siswa-siswa yang mereka ajar. Apalagi jika masalah tersebut terjadi pada pemerataan pada sekolah dasar. Dimana sekolah dasar merupakan pendidikan formal pertama yang didapatkan oleh orang-orang. Disini karakter dan pendidikan seseorang dibentuk. Jika terkendala dengan masalah tenaga pengajar yang kurang optimal dalam segi kualitas, maka siswa tersebut juga akan menerima pelajaran yang kurang optimal pula.

Tidak hanya penempatan tenaga pengajar saja yang mengalami kendala, tenaga kerja seperti karyawan pada umumnya juga mengalaminya. Pengaruh seleksi karyawan dan penempatan kerja menjadi sangat berpengaruh terhadap kinerja dan prestasi karyawan tersebut. Menurut Basuki (2007) dalam penelitiannya “Optimasi

Penempatan Pegawai Dengan Metode Algoritma Genetika” menyimpulkan bahwa mampu menempatkan pegawai dengan nilai error minimum dengan nilai fitness terbaik diperoleh dengan maksimum generasi 1000, *crossover rate* 0.9, mutasi *rate* 0.9, dan jumlah kromosom 40 [DHA-07].

Dalam setiap keputusan untuk menempatkan tenaga kerja ke tempat lain terdapat kemungkinan ketidaktepatan dalam suatu penempatan tenaga kerja. Selain itu belum adanya aplikasi pengolahan data yang baik membuat pemerataan tenaga pengajar dirasa guru kurang begitu jelas. Sehingga dibutuhkan suatu aplikasi yang dapat mengoptimalkan pemerataan tenaga pengajar yang diharapkan dapat mengatasi masalah ini. Selain itu, hasil yang dihasilkan oleh aplikasi ini nantinya dapat dipertanggungjawabkan kepada semua pihak.

Algoritma evolusi merupakan teknik optimasi yang meniru proses evolusi biologi. Tujuan dari algoritma evolusi ini adalah menghasilkan keturunan yang lebih baik dari generasi ke generasi sehingga menghasilkan individu yang optimal [WAY-13]. Terdapat beberapa metode dalam algoritma evolusi yaitu Algoritma Genetika (*Genetic Algorithm*, *GA*), *Evolution Strategies* (ES), *Genetic Programming* (GP), *Evolutionary Programming* (EP). Salah satu metode yang banyak diterapkan pada masalah yang kompleks dan yang paling populer adalah algoritma genetika. Algoritma genetika (AG) adalah algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah [SUY-11]. Konsep dari algoritma genetika itu sendiri terdiri dari beberapa fase yaitu pencarian nilai fitness, *crossover*, dan mutasi sehingga menghasilkan individu baru yang lebih baik dan nilai yang mendekati optimal.

Karena algoritma genetika ini telah berhasil diterapkan dalam berbagai masalah yang kompleks, disini penulis melakukan penelitian dengan judul “*Implementasi Algoritma Genetika Untuk Optimasi Pemerataan Guru SD di Kabupaten Banyuwangi*”. Diharapkan dengan menerapkan metode algoritma genetika dapat menyelesaikan permasalahan tersebut.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, dapat ditarik beberapa rumusan masalah diantara lain:

- a. Faktor apa saja yang mempengaruhi seorang guru SD akan di mutasi atau tidak.
- b. Bagaimana mengimplementasikan metode algoritma genetika dalam menyelesaikan pemerataan guru SD untuk menghasilkan kualitas guru yang optimal.
- c. Bagaimana perhitungan mendapatkan solusi optimal menggunakan metode algoritma genetika.

## 1.3. Batasan Masalah

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

- a. Penentuan parameter yang dibutuhkan untuk mengetahui seorang guru di mutasi sesuai dengan peraturan Kementerian Pendidikan dan Badan Kepegawaian Daerah Kabupaten Banyuwangi.
- b. Data guru dibangkitkan secara random seperti nama guru, penilaian setiap guru, dan penempatan guru.
- c. Daftar SD yang digunakan hanya berada di ruang lingkup wilayah kabupaten Banyuwangi yaitu pada Kecamatan Banyuwangi, Kabat, Glagah, dan Giri.
- d. Aturan yang digunakan adalah aturan yang berlaku pada guru PNS.
- e. Acuan pemerataan dilihat dari total kualitas guru setiap sekolah, tidak melihat batasan setiap guru pada satu sekolah.

#### 1.4. Tujuan

Adapun maksud dan tujuan dari penelitian ini adalah sebagai berikut :

1. Membuat aplikasi optimasi pemerataan mutasi guru SD menggunakan metode algoritma genetika.
2. Menerapkan metode algoritma genetika dalam optimasi pemerataan guru SD.
3. Menganalisa parameter yang dibutuhkan sebagai acuan mutasi seorang guru SD dengan studi kasus Kementerian Pendidikan Kabupaten Banyuwangi.

#### 1.5. Manfaat

Adapun manfaat dari penelitian ini adalah sebagai berikut :

1. Menciptakan suatu rancangan aplikasi optimasi pemerataan mutasi guru SD yang dapat di implementasikan di kantor pemerintahan terkait.
2. Membantu pihak Kementerian Pendidikan dan Badan Kepegawaian Daerah dalam membantu menempatkan seorang guru SD.
3. Mahasiswa dapat menerapkan pengetahuan teoritis yang telah diperoleh di bangku perkuliahan dalam berbagai kasus di lapangan.
4. Memperoleh pengalaman-pengalaman praktis dan mengenal lebih jauh relevansi ilmu yang diterima selama kuliah, dimana teori pernah didapat dan kemudian diterapkan dalam situasi yang sesungguhnya.
5. Merupakan salah satu tolak ukur bagi perguruan tinggi untuk menilai sejauh mana kualitas yang dimiliki oleh mahasiswa, sehingga dapat menjadi acuan bagi perguruan tinggi untuk meningkatkan mutu pendidikan selanjutnya.

## 1.6. Sistematika Penulisan

Adapun sistematika penulisan yang digunakan dalam menyusun Skripsi ini adalah:

### **BAB I : PENDAHULUAN**

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

### **BAB II : KAJIAN PUSTAKA DAN DASAR TEORI**

Bab ini menguraikan tentang landasan teori dan referensi yang mendasari proses perancangan dari penelitian ini.

### **BAB III : METODOLOGI PENELITIAN**

Bab ini menguraikan tentang metode yang dilakukan dalam proses pada penelitian ini.

### **BAB IV : PERANCANGAN**

Bab ini menguraikan tentang perancangan yang dilakukan dalam proses penelitian ini.

### **BAB V : IMPLEMENTASI**

Bab ini menguraikan tentang proses implementasi dari perancangan yang dilakukan pada penelitian ini.

### **BAB VI : PENGUJIAN DAN ANALISIS**

Bab ini menguraikan tentang pengujian dan analisa dari system yang sudah dibuat pada penelitian ini.

### **BAB VII : PENUTUP**

Bab ini menguraikan kesimpulan dan saran dari perancangan penelitian ini.

## BAB II

### KAJIAN PUSTAKA DAN TEORI

#### 2.1. Kajian Pustaka

Pencarian solusi dalam sebuah masalah merupakan hal yang sangat penting untuk memecahkan masalah tersebut. Dalam suatu masalah umumnya terdapat beberapa solusi yang dapat dihasilkan. Untuk mendapatkan solusi yang terbaik maka digunakanlah suatu langkah untuk mengoptimasi solusi tersebut. Optimasi digunakan untuk menghasilkan sebuah solusi yang paling optimal dengan nilai *error* yang sedikit. Salah satu metode optimasi yang sering digunakan adalah algoritma genetika.

Dhany Rahadian Basuki telah mengembangkan sebuah sistem optimasi penempatan pegawai dengan menggunakan algoritma genetika. Kasus pada sistem ini yaitu PT. Indoniles Electric Parts yang merupakan sebuah perusahaan yang bergerak pada bidang produksi komponen listrik kendaraan bermotor. Pada sistem tersebut memanfaatkan data nilai *error* tiap pegawai pada sebuah proses produksi barang. Sistem ini nantinya akan menempatkan pegawai proses produksi tertentu pada perusahaan tersebut ke bagian produksi lain. Karena tujuan dari aplikasi tersebut untuk meminimalkan tingkat kerusakan pada produksi, maka solusi yang dihasilkan harus mempunyai tingkat *error* yang paling kecil. Sebagai keluarannya, sistem ini mampu menghasilkan solusi yang optimal pada parameter algoritma genetika yaitu pada generasi 1000, *crossover rate* 0.9, mutasi *rate* 0.9, dan jumlah kromosom 40 [DHA-07].

#### 2.2. Pengertian Guru

Guru merupakan seseorang yang sangat berpengaruh kehidupan orang lain. Guru mempunyai peran yang penting dalam proses belajar mengajar seorang anak dan ikut berperan aktif dalam perkembangannya. Menurut Djamarah [MAS-13], “Guru adalah orang yang memberikan ilmu pengetahuan kepada anak didik. Guru dalam pandangan masyarakat adalah orang yang melaksanakan pendidikan di

tempat-tempat tertentu, tidak mesti di lembaga pendidikan formal, tetapi bisa juga di masjid, di surau/mushalla, di rumah, dan sebagainya”.

Sedangkan Dalam pengertian di Kamus Besar Bahasa Indonesia guru merupakan orang yg pekerjaannya (mata pencahariannya, profesinya) mengajar;-  
- kelakuan murid (orang bawahan) selalu mencontoh guru (orang atasannya).

Dari penjabaran tersebut maka dapat disimpulkan bahwa guru merupakan orang yang berwenang dan bertanggung jawab untuk membimbing dan membina anak didik, di sekolah maupun di luar sekolah.

### 2.3. Penilaian Kinerja Guru

Penilaian prestasi kerja PNS dilaksanakan oleh Pejabat Penilai (dalam kasus guru Pejabat Penilai merupakan pengawas dari UPTD dan Kepala Sekolah) sekali dalam 1 (satu) tahun yang dilakukan setiap akhir desember pada tahun yang bersangkutan atau paling lama akhir Januari tahun berikutnya. Penilaian prestasi kinerja PNS tersiri atas unsur:

1. Sasaran Kerja Pegawai (SKP) dengan bobot nilai 60% (enam puluh persen); dan
2. Perilaku kerja dengan bobot nilai 40% (empat puluh persen) [BKN-13].

#### 2.3.1. Sasaran Kerja Pegawai

Tata cara penilaian SKP [BKN-13]:

- a. Setiap PNS wajib menyusun SKP berdasarkan RKT instansi. Dalam menyusun SKP harus memperhatikan hal-hal sebagai berikut:
  1. Jelas  
Kegiatan yang dilakukan harus dapat diuraikan secara jelas.
  2. Dapat diukur  
Kegiatan yang harus dapat diukur secara kuantitas dalam bentuk angka seperti jumlah satuan, jumlah hasil, dan lain-lain maupun secara kualitas seperti hasil kerja sempurna, tidak ada kesalahan,

tidak ada revisi dan pelayanan kepada masyarakat memuaskan, dan lain-lain.

3. Relevan

Kegiatan yang dilakukan harus berdasarkan lingkup tugas jabatan masing-masing.

4. Dapat dicapai

Kegiatan yang dilakukan harus disesuaikan dengan kemampuan PNS.

5. Memiliki target waktu

Kegiatan yang dilakukan harus dapat ditentukan waktunya.

- b. SKP memuat kegiatan tugas jabatan dan target yang harus dicapai dalam kurun waktu penilaian yang bersifat nyata dan dapat diukur. Setiap kegiatan tugas jabatan yang akan dilakukan harus didasarkan pada tugas dan fungsi, wewenang, tanggung jawab dan uraian tugasnya secara umum telah ditetapkan dalam struktur organisasi dan tata kerja (SOTK).
- c. SKP yang telah disusun harus disetujui dan ditetapkan oleh Pejabat Penilai sebagai kontrak kerja.
- d. Dalam hal SKP yang disusun oleh PNS tidak disetujui oleh Pejabat Penilai maka keputusannya diserahkan kepada Atasan Pejabat Penilai dan bersifat final.
- e. SKP ditetapkan setiap tahun pada awal Januari.
- f. Dalam hal terjadi perpindahan pegawai setelah bulan Januari maka yang bersangkutan tetap menyusun SKP pada awal bulan sesuai dengan surat perintah melaksanakan tugas atau surat suara perintah menduduki jabatan.
- g. PNS yang tidak menyusun SKP dijatuhi hukuman sesuai dengan ketentuan peraturan perundang-undangan yang mengatur mengenai disiplin PNS.

Unsur-unsur SKP [BKN-13] :

a. Kegiatan Tugas Jabatan

Setiap kegiatan tugas jabatan yang akan dilakukan harus mengacu pada Penetapan Kinerja/RKT, sebagai implementasi kebijakan dalam rangka

mencapai tujuan dan sasaran organisasi yang telah ditetapkan dan harus berorientasi pada hasil secara nyata dan terukur.

b. Angka Kredit

Satuan nilai dari tiap butir kegiatan dan/atau akumulasi nilai butir-butir kegiatan yang harus dicapai oleh seorang pejabat fungsional dalam rangka pembinaan karier yang bersangkutan ditetapkan dengan jumlah angka kredit yang akan dicapai. Oleh sebab itu pejabat fungsional tertentu harus menetapkan target angka kredit yang akan dicapai dalam satu tahun. Angka kredit berfungsi sebagai nilai patokan terhadap suatu pegawai yang akan naik jabatan. Jadi dalam penelitian ini, angka kredit tidak begitu berpengaruh.

c. Target

1. Kuantitas (Target Output)

Dalam menentukan target output dapat berupa dokumen, konsep, naskah, surat keputusan, paket, laporan, dan lain-lain.

2. Kualitas (Target Kualitas)

Dalam menetapkan target kualitas harus memprediksi pada mutu hasil kerja yang terbaik, target kualitas diberikan nilai paling tinggi 100.

3. Waktu (Target Waktu)

Dalam menetapkan target waktu harus memperhitungkan berapa waktu yang dibutuhkan untuk menyelesaikan suatu pekerjaan, misalnya bulan, triwulan, kwartal, semester dan tahunan.

4. Biaya (Target Biaya)

Dalam menetapkan target biaya harus memperhitungkan berapa biaya yang dibutuhkan untuk menyelesaikan suatu pekerjaan dalam 1 tahun, misal jutaan, ratusan juta, miliaran, dan lain-lain.

Tabel 2.1 Contoh Penilaian Menggunakan SKP

NO	I. PEJABAT PENILAI		NO	II. PEGAWAI NEGERI SIPIL YANG DINILAI		
1	Nama	HADI HASAN, S.Pd., M.Pd.	1	Nama	ABDUL LATIF, S.Pd	
2	NIP	19610401 198201 1 006	2	NIP	19611114 198112 1 001	
3	Pangkat/Gol.Ruang	Pembina (IV/a)	3	Pangkat/Gol.Ruang	Pembina/Ivb	
4	Jabatan	Kepala UPTD Pendidikan Kec. Rogojampi	4	Jabatan	Kepala Sekolah / Guru Madya	
5	Unit Kerja	UPTD Kec. Rogojampi	5	Unit Kerja	SDN 2 Lemahbangdewo	
NO	I.KEGIATAN TUGAS JABATAN	TARGET				
		AK	KUANTI/OUTP UT	KUAL/ MUTU	WAKTU	BIAYA
1	Merencanakan dan melaksanakan pembelajaran	7.44	2 Laporan	100	12 bulan	-
2	Menjadi Kepala Sekolah/Madrasah per Tahun	22.31	1 Laporan	100	12 bulan	-
3	Menyusun kurikulum pada satuan pendidikannya	1.49	1 Laporan	100	12 bulan	-
4	Membimbing siswa dalam kegiatan ekstrakurikuler	0.6	1 SK	100	12 bulan	-
5	Mengikuti diklat fungsional lamanya antara 30 s.d 80 jam	1	1 1.Surat Tugas 2. Laporan 3. Sertifikat	100	1 bulan	-
6	Lokakarya atau kegiatan bersama (seperti KKG/MGMP/KKKS/MKKS)	0.15	1 Surat ket dan laporan	100	1 bulan	-
7	Membuat makalah berupa tinjauan ilmiah dalam bidang pendidikan formal dan pembelajaran pada satuan pendidikannya, tidak diterbitkan, disimpan di perpustakaan	4	2 Makalah	100	3 bulan	-
8	Membuat alat peraga Kategori sederhana	3	3 Alat peraga	100	2 bulan	-
9	Menjadi anggota organisasi profesi, sebagai anggota aktif	0.75	1 KTA	100	12 bulan	-
10	Menjadi anggota kegiatan kepramukaan, sebagai Pengurus aktif	1	1 SK	100	12 bulan	-
11	Menjadi tim penilai angka kredit	0.04	1 PAK	100	3 bulan	-
<b>JUMLAH ANGKA KREDIT</b>		<b>41.78</b>				

### 2.3.1.1. Cara Perhitungan SKP

Nilai capaian SKP dinyatakan dengan angka dan sebutan sebagai berikut :

- a. 91 – ke atas : Sangat Baik
- b. 76 – 90 : Baik
- c. 61 – 75 : Cukup
- d. 51 – 60 : Kurang
- e. 50 – ke bawah : Buruk

Penilaian SKP dilakukan dengan menghitung tingkat capaian SKP yang telah ditetapkan untuk setiap pelaksanaan kegiatan tugas jabatan, yang diukur dengan 4 aspek yaitu aspek kuantitas, aspek kualitas, waktu dan biaya [BKN-13]. Perhitungan 4 aspek tersebut ditunjukkan pada Persamaan 2-1 sampai Persamaan 2-10.

- a. Aspek kuantitas

$$\text{Penilaian SKP (aspek kuantitas)} = \frac{\text{Realisasi Output}}{\text{Target Output}} \times 100 \quad (2-1)$$

- b. Aspek kualitas

$$\text{Penilaian SKP (aspek kualitas)} = \frac{\text{Realisasi Kualitas}}{\text{Target Kualitas}} \times 100 \quad (2-2)$$

- c. Aspek waktu

1. Dalam hal kegiatan tidak dilakukan maka realisasi waktu 0 (nol), penhitungannya menggunakan rumus :

$$\text{Nilai capaian SKP waktu} = \frac{1.76 \times \text{Target Waktu} - \text{Realisasi Waktu}}{\text{Target Waktu}} \times 0 \quad (2-3)$$

2. Untuk aspek waktu dengan tingkat efisiensi waktu sebesar  $\leq 24\%$  menggunakan rumus :

$$\text{Nilai capaian SKP waktu} = \frac{1.76 \times \text{Target Waktu} - \text{Realisasi Waktu}}{\text{Target Waktu}} \times 100 \quad (2-4)$$

3. Untuk aspek waktu dengan tingkat efisiensi waktu sebesar  $\geq 24\%$  menggunakan rumus :

$$\begin{aligned} \text{Nilai} \\ \text{capaian} \\ \text{SKP waktu} \end{aligned} = 76 - \left( \left( \frac{1.76 \times \text{Target Waktu} - \text{Realisasi Waktu}}{\text{Target Waktu}} \times 100 \right) - 100 \right) \quad (2-5)$$

4. Untuk menghitung presentase tingkat efisiensi waktu dari target waktu, perhitungannya menggunakan rumus :

$$\begin{aligned} \text{Presentase} \\ \text{Efisiensi Waktu} \end{aligned} = 100\% - \left( \frac{\text{Realisasi Waktu}}{\text{Target Waktu}} \times 100\% \right) \quad (2-6)$$

d. Aspek biaya

1. Dalam hal kegiatan tidak dilakukan maka realisasi biaya 0 (nol), perhitungannya menggunakan rumus :

$$\begin{aligned} \text{Nilai} \\ \text{capaian} \\ \text{SKP biaya} \end{aligned} = \frac{1.76 \times \text{Target Biaya} - \text{Realisasi Biaya}}{\text{Target Biaya}} \times 0 \times 100 \quad (2-7)$$

2. Untuk aspek biaya dengan tingkat efisiensi biaya sebesar  $\leq 24\%$  menggunakan rumus :

$$\begin{aligned} \text{Nilai} \\ \text{capaian} \\ \text{SKP biaya} \end{aligned} = \frac{1.76 \times \text{Target Biaya} - \text{Realisasi Biaya}}{\text{Target Biaya}} \times 100 \quad (2-8)$$

3. Untuk aspek biaya dengan tingkat efisiensi biaya sebesar  $\geq 24\%$  menggunakan rumus :

$$\begin{aligned} \text{Nilai} \\ \text{capaian} \\ \text{SKP biaya} \end{aligned} = 76 - \left( \left( \frac{1.76 \times \text{Target Biaya} - \text{Realisasi Biaya}}{\text{Target Biaya}} \times 100 \right) - 100 \right) \quad (2-9)$$

4. Untuk menghitung presentase tingkat efisiensi biaya dari target biaya, perhitungannya menggunakan rumus :

$$\begin{aligned} \text{Presentase} \\ \text{Efisiensi Biaya} \end{aligned} = 100\% - \left( \frac{\text{Realisasi Biaya}}{\text{Target Biaya}} \times 100\% \right) \quad (2-10)$$

### 2.3.2. Penilaian Perilaku Kerja

1. Nilai perilaku kerja PNS dinyatakan dengan angka dan sebutan sebagai berikut:
  - a. 91-100 : Sangat Baik
  - b. 76-90 : Baik
  - c. 61-75 : Cukup
  - d. 51-60 : Kurang
  - e. 50 ke bawah : Buruk
2. Penilaian perilaku kerja meliputi aspek:
  - a. Kesetiaan
  - b. Prestasi kerja
  - c. Tanggung jawab
  - d. Ketaatan
  - e. Kejujuran
  - f. Kerjasama
  - g. Prakarsa
3. Cara menilai perilaku kerja dilakukan melalui pengamatan oleh pejabat penilai terhadap PNS yang dinilai, penilaian perilaku kerja dapat mempertimbangkan masukan dari Pejabat Penilai lain yang setingkat di lingkungan unit kerja masing-masing.
4. Untuk mendapatkan nilai akhir dari Penilaian Perilaku Kerja seorang PNS didapatkan dari mendapatkan nilai rata-rata dari aspek-aspek tersebut.
5. Nilai perilaku kerja dapat diberikan paling tinggi 100 (seratus) [BKN-13].

**Tabel 2.2** Contoh Penilaian Perilaku Kerja

PENILAIAN			
UNSUR YANG DINILAI	NILAI		KETERANGAN
	ANGKA	SEBUTAN	
a. Kesetiaan	91	Amat Baik	
b. Prestasi Kerja	82	Baik	
c. Tanggung Jawab	82	Baik	
d. Ketaatan	81	Baik	
e. Kejujuran	81	Baik	
f. Kerjasama	81	Baik	
g. Prakarsa	81	Baik	
i. JUMLAH	579	-	
j. NILAI RATA-RATA	83	Baik	

#### 2.4. Mutasi Guru

Mutasi adalah suatu kegiatan memindahkan karyawan dari unit/bagian yang kelebihan tenaga ke unit/bagian yang kekurangan tenaga atau yang lebih memerlukan.

Mutasi karyawan dapat terjadi karena 2 hal yaitu:

1. Keinginan karyawan sendiri terjadi karena :
  - Pegawai yang bersangkutan merasa tidak sesai dengan bidang tugasnya atau jabatannya.
  - Pegawai yang bersangkutan merasa tidak dapat bekerja sama dengan teman sekerjanya atau dengan atasannya.
  - Pegawai yang bersangkutan merasa bahwa tempat atau lingkungan kerja tidak sesuai dengan kondisi fisik atau keinginannya.

2. Keinginan perusahaan, atau keinginan atasan terjadi karena :

- Perusahaan ingin menunjukkan kepada pegawai yang bersangkutan bahwa mutasi tersebut bukan hukuman, melainkan upaya untuk menjamin kelangsungan pekerjaan pegawai tersebut.
- Perusahaan ingin menyakinkan pegawai bahwa ia tidak akan diberhentikan karena kekurangan atau kekurangcakapan pegawai yang bersangkutan.
- Perusahaan ingin menghindari rasa jenuh pegawai pada jenis pekerjaan, jabatan

Berikut merupakan manfaat dari mutasi [JEN-10]:

1. Memenuhi kebutuhan tenaga di bagian yang kekurangan tanpa merekrut dari luar.
2. Memenuhi keinginan karyawan sesuai dengan minat dan bidang tugasnya
3. Menjamin keyakinan karyawan bahwa tidak akan diberhentikan karena kurang mampu atau kurang cakapan.
4. Sebagai motivasi karyawan
5. Mengatasi rasa bosan pada jabatan atau pekerjaan yang sama

Segi negatif anggapan sebagian orang bahwa mutasi merupakan suatu hukuman terutama bagi karyawan yang kurang mampu, kurang cakap, kurang berhasil dan karyawan yang merasa bersalah/melakukan kesalahan. Namun juga ada segi positif dari mutasi diantaranya adalah:

1. Usaha menempatkan pegawai pada pekerjaan dan jabatan yang sesuai dengan kecakapan dan kemampuannya.
2. Usaha meningkatkan semangat dan gairah kerja.
3. Usaha menciptakan persaingan yang sehat diantara pegawai.

Mutasi promosi yaitu mutasi yang diikuti dengan kenaikan jabatan. Tugas dan tanggung jawab seorang pegawai yang mendapat mutasi tersebut bertambah besar.

Tujuannya adalah:

- a. Mengisi suatu formasi jabatan dengan mengambil sumber tenaga dari dalam.
- b. Membina karier pegawai
- c. Mengembangkan kemampuan pegawai.

Macam-macam Mutasi [JEN-10]:

1. Ditinjau dari aktivitas tempat pegawai bekerja:

Mutasi antar urusan, Mutasi antar seksi, mutasi antar bagian, mutasi antar biro, mutasi antar unit.

2. Ditinjau dari tujuan dan maksud mutasi ada lima yaitu:

- a) Production transfer (mutasi dalam jabatan yang sama karena produksi menurun)
- b) Replacemant transfer (mutasi dari jabatan yang sudah lama di pegang ke jabatan yang sama pada unit/bagian lain, untuk menggantikan pegawai baru atau pegawai yang diberhentikan)
- c) Versatility transfer (mutasi dari jabatan yang satu ke jabatan yang lain untuk menambah pengetahuan pegawai yang bersangkutan)
- d) Shife transfer (mutasi dalam jabatan yang sama, tetapi berbeda shife, misalnya dari shife A (malam) ke shife B (siang))
- e) Remedial transfer (mutasi pegawai ke bagian mana saja dengan tujuan untuk memupuk atau memperbaiki kerjasama antar pegawai).

3. Ditinjau dari masa kerja pegawai

- a) Temporary transfer (mutasi yang bersifat sementara, untuk menggantikan yang berhalangan).
- b) Permanent transfer (mutasi yang bersifat tetap).

## 2.5. Proses Mutasi Guru

Pemutasian seorang guru tidaklah melalui proses yang singkat, melainkan terdapat beberapa tahap yang dilakukan oleh dinas-dinas terkait sehingga seorang guru tersebut patut di mutasi ke tempat lain. Berikut ini merupakan proses mekanisme yang terjadi sehingga seorang guru tersebut diputuskan untuk dimutasikan.

Unit Pelaksana Tingkat Dinas (UPTD) akan mengumpulkan kepala sekolah dari SD yang berada pada daerah tertentu. Kemudian mereka akan mengkaji kebutuhan-kebutuhan dari tiap SD pada daerah tersebut. Selain itu, mereka juga mengumpulkan semua data-data guru pada tiap-tiap sekolah. Kemudian akan didapatkan siapa sajakah guru yang akan dimutasi dan akan menempati sekolah mana nantinya. Hasil tersebut kemudian akan diserahkan ke Kementerian Pendidikan untuk diverifikasi terlebih dahulu. Setelah lolos tahap ini kemudian akan langsung diserahkan ke Bupati. Dari Bupati kemudian akan dilemparkan ke Badan Kepegawaian Daerah (BKD) untuk mengkaji ulang. Setelah itu dikembalikan lagi kepada Bupati untuk langsung dibuatkan SK tersebut.

Terdapat 5 faktor pertimbangan seseorang dimutasi.

1.
  - Dalam rangka penyegaran, sudah  $\pm 15$  tahun bertugas di sekolah lama
  - Jumlah rombongan belajar yang ditinggal 6 rombongan belajar, guru kelas yang ada 7 orang, kelebihan 1 orang guru kelas.
  - Jumlah rombongan belajar pada sekolah yang dituju 6 rombongan belajar, guru kelas yang ada 5 orang, kekurangan 1 orang guru kelas.
2.
  - Dalam rangka penyegaran, sudah  $\pm 10$  tahun bertugas di sekolah lama.
  - Jumlah rombongan belajar pada sekolah yang ditinggal 6 rombongan belajar, guru kelas yang ada 6 orang, apabila yang bersangkutan diusulkan mutasi pengganti a.n. Sdr.....

- Jumlah rombongan belajar pada sekolah yang dituju 6 rombongan belajar, guru kelas yang ada 5 orang, kekurangan 1 orang guru kelas.
- 3.
  - Dalam rangka penyegaran, sudah  $\pm 8$  tahun bertugas di sekolah lama
  - Jumlah rombongan belajar pada sekolah yang ditinggal 6 rombongan belajar, guru kelas yang ada 6 rombongan belajar, apabila yang diusulkan mutasi pengganti a.n. Sdr.....
  - Jumlah rombongan belajar pada sekolah yang dituju 6 rombongan belajar, guru kelas yang ada 6 orang, 1 orang diusulkan mutasi ke SDN.....
- 4.
  - Dalam rangka penyegaran/Mendekatkan dengan tempat tinggal.
  - Mutasi tukar tempat dengan Sdr.....
- 5.
  - SDN Lama regrouping
  - Jumlah rombongan belajar pada sekolah yang dituju 6 rombongan belajar, guru kelas yang ada 4 orang, kekurangan guru kelas 2 orang.

Sumber Dinas Pendidikan Kabupaten Banyuwangi

## 2.6. Algoritma Genetika

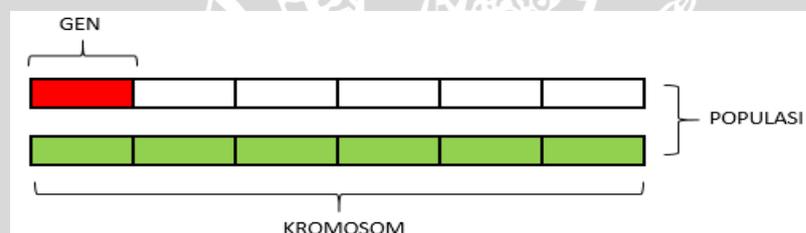
Algoritma genetika (AG) adalah algoritma pencarian yang didasarkan pada mekanisme seleksi alamiah dan genetika alamiah. Algoritma genetika merupakan algoritma pencarian heuristic yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Variasi kromosom ini akan mempengaruhi laju reproduksi dan tingkat kemampuan organisme untuk tetap hidup. Pada dasarnya ada 4 kondisi yang sangat mempengaruhi proses evaluasi, yaitu [SRI-03] :

- a. Kemampuan organisme untuk melakukan reproduksi.
- b. Keberadaan populasi organisme yang bias melakukan reproduksi.
- c. Keberagaman organisme dalam suatu populasi.
- d. Perbedaan kemampuan untuk *survive*.

Pelopop pertama penggunaan metode algoritma genetika adalah John Holland pada tahun 60-an [FAC-10]. Algoritma genetika menggunakan analogi seleksi alam yang bekerja dari suatu populasi yang terdiri dari berbagai individu (gen), yang masing-masing individu mempresentasikan suatu solusi yang mungkin muncul dari persoalan yang dihadapi. Dalam hal ini, individu yang terpilih dilambangkan dengan sebuah nilai *fitness* yang digunakan untuk mencari solusi terbaik dari persoalan yang ada.

## 2.7. Struktur Umum Algoritma Genetika

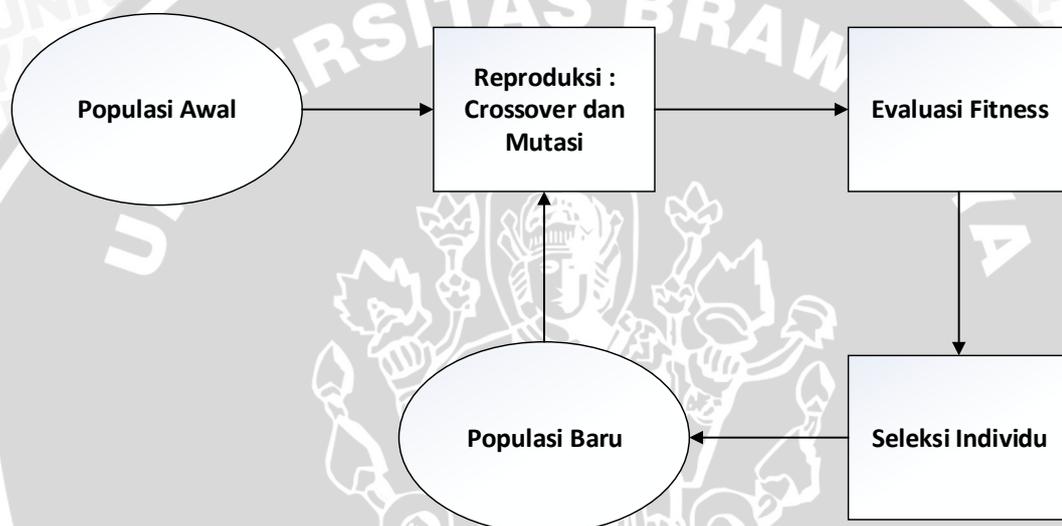
Terdapat beberapa struktur umum yang ada pada algoritma genetika. Untuk sejumlah solusi yang merepresentasikan sebuah masalah biasa disebut dengan istilah populasi. Dalam sebuah populasi terdapat beberapa individu-individu yang disebut dengan kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Panjang simbol sebuah kromosom tergantung pada jumlah gen yang ada pada kromosom itu sendiri. Gen merupakan bagian terkecil dari sebuah solusi.



**Gambar 2.1** Struktur Populasi

Pada awalnya populasi akan dibangkitkan secara acak yang biasa disebut dengan populasi awal. Sedangkan untuk populasi berikutnya didapatkan dari hasil evolusi tiap kromosom-kromosomnya melalui iterasi yang disebut dengan generasi. Pada setiap generasi, kromosom akan di evaluasi dengan parameter yang disebut fungsi fitness. Semakin bagus nilai fitness maka semakin besar kemungkinan kromosom tersebut lolos tahap seleksi. Nilai fitness juga berbanding lurus dengan kualitas sebuah kromosom dalam sebuah populasi [SRI-03].

Generasi berikutnya yang akan terbentuk biasa disebut dengan anak (*offspring*) yang terbentuk karena gabungan 2 kromosom sekarang yang bertindak sebagai induk (*parent*) dengan melalui proses penyilangan (*crossover*) dan mutasi. Setelah didapatkan individu baru, kemudian dilakukan seleksi dengan melihat nilai fitness tiap kromosom induk (*parent*) dan anak (*offspring*). Untuk kromosom yang tidak lolos tahap seleksi akan dibuang sehingga menghasilkan nilai fitness yang cenderung lebih baik tiap generasinya. Setelah melalui beberapa generasi, maka algoritma ini akan konvergen ke kromosom terbaik.



**Gambar 2.2** Siklus Algoritma Genetika

## 2.8. Parameter-Parameter Algoritma Genetika

Metode ini bekerja berdasarkan parameter-parameter tertentu yang akan mempengaruhi kinerja dan perilaku dari algoritma ini. Parameter-parameter penting yang mempengaruhi langsung terhadap performa dari algoritma ini diantaranya adalah ukuran populasi, jumlah generasi dan probabilitas mutasi.

### 2.8.1. Ukuran Populasi

Populasi merupakan sejumlah solusi yang mempresentasikan sebuah masalah. Masalah disini ialah kromosom itu sendiri. Kromosom mewakili sebuah masalah. Jadi dalam satu populasi terdapat beberapa kromosom. Semakin banyak kromosom maka semakin sulit algoritma ini untuk mendapatkan solusinya.

### 2.8.2. Jumlah Generasi

Generasi merupakan istilah lain untuk iterasi. Generasi merupakan proses evaluasi dari tiap-tiap populasi. Sama halnya dengan populasi, besarnya suatu generasi akan mempengaruhi kecepatan konvergensi. Semakin besar jumlah generasi, maka konvergensi akan lambat, tetapi bila jumlah generasi terlalu kecil maka dapat terjadi konvergensi prematur. Untuk itu jumlah generasi yang tepat harus diperhitungkan terlebih dahulu agar mendapatkan hasil yang diharapkan. Proses pencarian optimasi menggunakan algoritma genetika ini akan dihentikan ketika jumlah generasi sudah terpenuhi [MON-14].

### 2.8.3. Crossover Rate

*Crossover rate* pada penelitian ini berfungsi untuk mencari seberapa banyak kromosom yang akan menjadi induk. *Crossover rate* menunjukkan seberapa persen dari total kromosom yang akan terpilih menjadi induk. Semakin besar nilai *crossover rate* dan total kromosom, semakin banyak pula kromosom yang akan menjadi induk. Nantinya *crossover rate* pada proses *crossover* ini akan berpengaruh pada banyaknya *offspring* yang dihasilkan.

### 2.8.4. Mutasi Rate

Mutasi *rate* pada metode mutasi berfungsi untuk mencari banyaknya kromosom yang akan menjadi induk. Sama halnya dengan *crossover rate*, mutasi *rate* akan menunjukkan seberapa persen dari total kromosom yang akan terpilih untuk menjadi induk. Semakin besar nilai mutasi *rate* maka semakin banyak kromosom yang terpilih untuk menjadi induk dan semakin banyak juga *offspring* yang akan dihasilkan.

## 2.9. Penerapan Algoritma Genetika

### 2.9.1. Membangun Generasi Awal

Hal yang paling pertama dilakukan dalam algoritma ini adalah membentuk sejumlah populasi awal yang digunakan untuk penyelesaian optimasi. Populasi awal yang dibangun dalam penelitian ini dengan menggunakan bilangan *random* (acak) dengan *range* bilangan yang sudah ditentukan.

### 2.9.2. Representasi Kromosom

Representasi kromosom merupakan proses pengkodean dari penyelesaian asli suatu permasalahan. Pengkodean kandidat ini disebut dengan kromosom. Pengkodean tersebut meliputi penyandian gen, dengan satu gen mewakili satu variable [MON-14].

Ada berbagai macam representasi kromosom, diantaranya adalah representasi biner, representasi real coded, representasi integer, representasi matriks dan representasi permutasi. Pada penelitian ini, representasi kromosom menggunakan representasi permutasi yaitu dimana calon solusi dinyatakan dalam bilangan integer tetapi angkanya terbatas pada suatu keadaan tertentu.

Cara mempresentasikan suatu permasalahan dalam kromosom merupakan hal yang sangat penting dalam algoritma ini. Berikut contoh representasi kromosom pada penelitian ini :

Misal terdapat 10 guru dan dimana 5 guru tersebut akan dimutasi yaitu secara urut guru 1, guru 2, guru 3, guru 4 dan guru 5. Guru yang akan dimutasi tersebut akan mewakili setiap gen pada semua kromosom nantinya. Pada proses mutasi, guru yang akan dimutasi akan menempati posisi guru lain yang akan dimutasi juga. Maka inisialisasi kromosom akan dibangkitkan secara random yang terdiri dari 5 gen setiap kromosomnya dikarenakan guru yang akan dimutasi berjumlah 5. Contoh pembangkitan kromosom awal ditunjukkan pada Tabel 2.3.

**Tabel 2.3** Representasi Kromosom

Kromosom 1	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5
	3	2	1	5	4

Pada Tabel 2.3 menunjukkan kromosom 1 memiliki 5 gen yang berarti terdapat 5 guru yang akan dimutasi. Masing-masing gen mewakili tiap guru yang akan dimutasi antara lain gen 1 mewakili guru 3, gen 2 mewakili guru 2, gen 3 mewakili guru 1, gen 4 mewakili guru 5 dan gen 5 mewakili guru 4.

Seperti yang sudah dijelaskan sebelumnya bahwa guru yang akan dimutasi akan mewakili setiap gen pada kromosom dan juga guru yang akan dimutasi akan menempati posisi guru lain yang juga akan dimutasi. Jadi pada gen 1 yang awalnya mewakili guru 1 mempunyai nilai 3 yang berarti berisi guru 3, maka guru 3 akan menempati posisi guru 1. Untuk gen 2 yang mewakili guru 2 mempunyai nilai 2 yang berarti berisi guru 2, maka guru 2 tidak mengalami mutasi atau pemindahan. Begitu seterusnya untuk gen selanjutnya.

### 2.9.3. Operator Genetika

Pada algoritma ini menggunakan dua operator untuk mengkombinasikan (modifikasi) individu dalam suatu populasi untuk menentukan populasi pada generasi berikutnya. Operator tersebut adalah *crossover* dan mutasi.

#### 2.9.3.1. Persilangan (*crossover*)

Perkawinan silang atau *crossover* berfungsi untuk menghasilkan *offspring* dari dua buah induk yang terpilih. *Offspring* yang dihasilkan dari perkawinan silang mewarisi gen-gen dari kedua induk. Metode *crossover* pada penelitian ini menggunakan *one cut point crossover*. Metode ini merupakan metode yang sederhana dimana segmen kiri pada *offspring* berisi gen induk 1 sedangkan segmen kanan berisi gen tersisa dari induk 2 dengan patokan *crossover point* yang sudah ditentukan [WAY-13]. Jumlah *offspring* dan induk yang terpilih tergantung pada *crossover rate* yang sudah ditentukan sebelumnya. Contoh dari proses *crossover* ditunjukkan pada Tabel 2.4.

**Tabel 2.4** Contoh Proses *Crossover*

	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5	Gen 6	Gen 7	Gen 8
Parent 1	3	8	2	7	5	6	4	1
Parent 2	8	1	4	2	5	3	6	1
Child	3	8	2	7	1	4	5	6

### 2.9.3.2. Mutasi

Mutasi digunakan untuk melakukan modifikasi satu atau lebih dari nilai gen pada suatu kromosom. Dari proses mutasi ini akan tercipta individu baru. Mutasi ini berperan untuk menggantikan gen yang tidak ada dalam populasi awal sehingga mutasi akan meningkatkan variasi populasi [SRI-03].

Untuk proses mutasi, disini penulis akan menggunakan metode mutasi yang paling sederhana yaitu *reciprocal exchange mutation*. Dua gen akan terpilih secara random kemudian saling menukarkan nilai [WAY-13]. Jumlah kromosom yang terpilih tergantung pada mutasi *rate* yang sudah ditentukan sebelumnya. Contoh proses mutasi ditunjukkan pada Tabel 2.5.

**Tabel 2.5** Contoh Proses Mutasi

	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5	Gen 6	Gen 7	Gen 8
Parent	5	3	7	6	2	8	4	1
Offspring	5	3	4	6	2	8	7	1

### 2.9.3.3. Perhitungan Fitness

Nilai *fitness* adalah inversi dari biaya pelaksanaan kromosom. Nilai ini membedakan kualitas dari kromosom untuk mengetahui seberapa baik kromosom yang dihasilkan [MON-14]. Pada penelitian ini, semakin besar nilai fitness pada sebuah kromosom, maka semakin besar juga peluang kromosom tersebut untuk lolos pada generasi selanjutnya.

Nilai fitness menjadi parameter utama untuk menentukan apakah sebuah kromosom tersebut layak untuk lolos pada generasi selanjutnya atau tidak pada proses seleksi. Tidak semua kromosom mempunyai nilai fitness yang tinggi, suatu kromosom dapat terkena penalti dalam suatu keadaan tertentu. Nilai penalti diberikan dengan tujuan agar kromosom yang tidak bagus nantinya tidak ikut terseleksi dalam populasi selanjutnya. Untuk itu perlu diberikan penalti agar nilai fitness pada kromosom menjadi kecil.

Pada penelitian ini, kromosom yang terkena penalti adalah kromosom yang dimana satu atau lebih gennya berisi mutasi guru dengan ketentuan berikut :

1. Guru Penjaskes / Agama menempati posisi menjadi Guru Kelas
2. Guru Kelas menempati posisi menjadi Guru Penjaskes / Agama
3. Guru Penjaskes menempati posisi menjadi Guru Agama atau sebaliknya.

#### 2.9.4. Seleksi

Seleksi akan menentukan individu-individu mana saja yang akan dipilih untuk dilakukan rekombinasi dan bagaimana *offspring* terbentuk dari individu-individu tersebut [SRI-03]. Langkah pertama yang dilakukan dalam seleksi ini adalah pencarian nilai *fitness*. Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit.

Dalam penelitian ini, semakin besar nilai *fitness* suatu kromosom maka semakin besar peluang kromosom tersebut menjadi kromosom baru pada generasi berikutnya. Untuk menjaga kromosom-kromosom yang bernilai *fitness* terendah tidak hilang maka dibutuhkan proses penyeleksian untuk mengambil kromosom terbaik. Proses ini dikenal sebagai *elitism*.

#### 2.10. Bahasa Pemrograman C#

Microsoft Visual C-Sharp atau yang lebih dikenal dengan Visual C# adalah sebuah bahasa yang tidak diragukan lagi dalam proses pengembangan aplikasi berbasis .NET Framework, dimana C# bebas dari masalah kompatibilitas dilengkapi

dengan berbagai fitur yang sebagian besar merupakan fitur baru, menarik, dan tentu saja menjanjikan.

Visual C# dibuat berdasarkan pemrograman C# yang merupakan bahasa pemrograman berorientasi objek dan mempunyai banyak kesamaan dengan C++, Java, dan VB. C# pada faktanya merupakan kombinasi antara efisiensi pemrograman C++, kesederhanaan pemrograman Java, dan penyederhanaan dari pemrograman Visual Basic.

Seperti pemrograman Java, C# juga tidak memperbolehkan multiple inheritance atau penggunaan pointer (pada safe/managed code), tetapi C# menyediakan garbage memory collection pada saat runtime dan pada saat pengecekan akses memori.

Meskipun bertentangan dengan pemrograman Java, C# tetap mempertahankan operasi unik yang terdapat pada bahasa pemrograman C++ seperti overloading, enumerations, pre-processor directive, pointer (pada unmanaged/unsafe code), dan fungsi pointer. Seperti halnya Visual Basic, bahasa pemrograman C# juga dilengkapi dengan properties.

Sebagai tambahan, bahasa pemrograman C# juga datang dengan beberapa fitur baru dan sangat menarik seperti reflections, attributes, marshalling, remote, threads, streams data acces dengan ADO.NET, dan masih banyak lagi. [SIT-10]

## 2.11. Database

Database adalah kumpulan informasi yang disimpan di dalam computer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut [IGD-09].

Konsep dasar dari database adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundancy) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Sebuah database memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di

dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu database, dan hubungan di antara obyek tersebut.

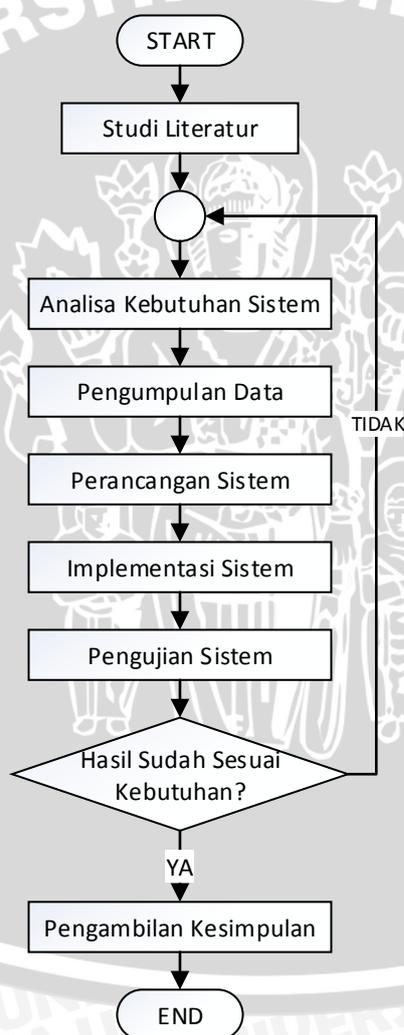
Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur database: ini dikenal sebagai database model atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah yaitu mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.



### BAB III

## METODOLOGI PENELITIAN

Pada bab ini dan selanjutnya akan menjelaskan langkah-langkah yang akan ditempuh dalam penyusunan dalam skripsi, seperti perancangan, implementasi dan pengujian dari sistem yang akan dibuat. Untuk kesimpulan dan saran akan disertakan sebagai catatan atas sistem dan kemungkinan arah pengembangan sistem selanjutnya.



**Gambar 3.1** Diagram Blok Metode Penelitian

### 3.1. Studi Literatur

Dalam penelitian ini, studi literatur membahas mengenai teori-teori yang digunakan untuk menunjang penulisan implementasi algoritma genetika untuk optimasi pemerataan guru SD di Kabupaten Banyuwangi. Teori yang dipaparkan dalam studi literature yaitu mengenai konsep Algoritma Genetika (GA) dan dalam dasar teori akan dipaparkan juga mengenai teori yang berhubungan dengan penilaian kinerja guru, mutasi, pemrograman C#, dan database. Teori-teori tersebut diperoleh dari buku, jurnal, e-book, dan dokumentasi project pada peneletian sebelumnya.

### 3.2. Analisa Kebutuhan

Analisis kebutuhan bertujuan untuk mengetahui secara keseluruhan kebutuhan yang diperlukan. Analisis kebutuhan dilakukan dengan mengidentifikasi kebutuhan dan siapa saja yang terlibat di dalam implementasi penelitian ini.

Secara keseluruhan, kebutuhan yang diperlukan dalam implementasi penelitian ini terdiri dari 2 macam yaitu kebutuh fungsional dan kebutuhan non-fungsional sebagai berikut :

#### 1. Kebutuhan Non-Fungsional

##### a. Kebutuhan hardware, meliputi :

- Laptop.

##### b. Kebutuhan software, meliputi :

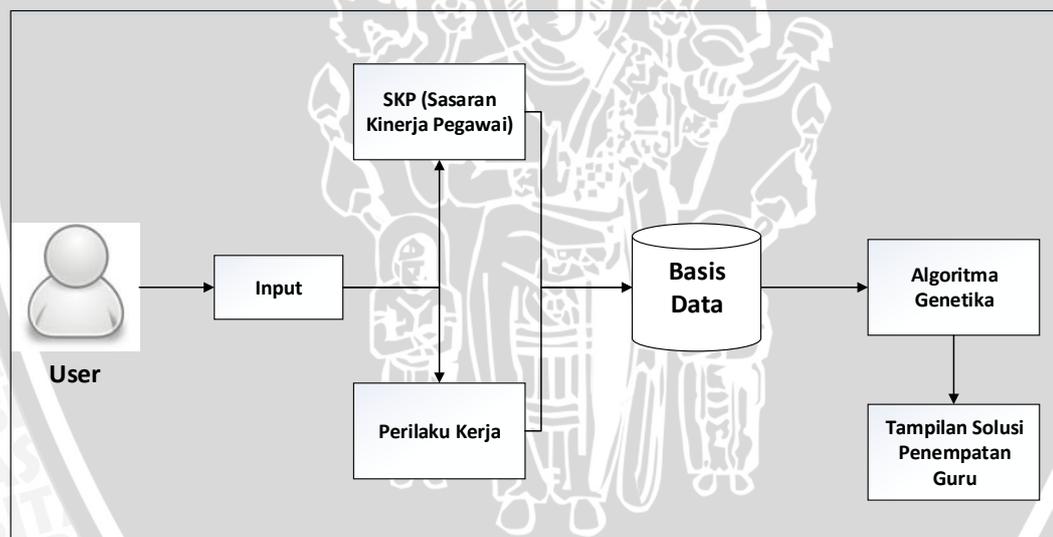
- XAMPP Server for Windows Versi 1.8.1.
- MySQL sebagai server Database Management System.
- Microsoft Visual Studio 2013.
- Windows 8.1 sebagai Operating System.

## 2. Kebutuhan Fungsional

- Data sekolah di Kabupaten Banyuwangi.
- Contoh penilaian Sasaran Kerja Pegawai (SKP) PNS.
- Contoh penilaian perilaku kerja PNS.

### 3.3. Perancangan Perangkat Lunak

Perancangan dalam penelitian ini dilakukan setelah semua kebutuhan sistem sudah didapatkan melalui tahap analisis kebutuhan. Pada metode perancangan sistem ini merupakan tahap awal dari implementasi sistem. Perancangan perangkat lunak akan menggunakan perancangan dengan *flowchart* atau diagram alir. Diagram alir akan menjelaskan alur proses implementasi Algoritma Genetika untuk optimasi pemerataan mutasi guru SD. Untuk arsitektur perancangan sistem ini akan ditunjukkan pada Gambar 3.2.



Gambar 3.2 Arsitektur Perancangan Sistem

### 3.4. Pengumpulan Data

Berdasarkan analisis kebutuhan data di atas, pengumpulan data dilakukan dengan mengajukan surat permohonan permintaan data skripsi kepada kementerian pendidikan Kabupaten Banyuwangi. Adapun aturan secara teknis penilaian kinerja PNS didapatkan dari Badan Kepegawaian Kabupaten Banyuwangi.

### 3.5. Implementasi

Implementasi dalam penelitian ini dilakukan dengan mengacu kepada perancangan perangkat lunak. Implementasi perangkat lunak dilakukan dengan menggunakan Bahasa pemrograman C#, database MySQL dan tools pendukung lainnya. Implementasi optimasi pemerataan guru SD meliputi :

- Penerapan Algoritma Genetika dalam program yang dibuat dengan menggunakan bahasa pemrograman C#.
- Pembuatan antarmuka program.
- Memasukkan data yang diperlukan ke database MySQL.

### 3.6. Perancangan Uji Coba

Beberapa uji coba yang akan dilakukan untuk mengevaluasi program ini antara lain:

1. Uji Coba untuk menentukan ukuran generasi yang optimal untuk proses algoritma genetika pada optimasi pemerataan mutasi guru.
2. Uji Coba untuk menentukan ukuran *crossover rate* dan *mutation rate* yang optimal untuk proses algoritma genetika pada optimasi pemerataan mutasi guru.
3. Uji Coba untuk menentukan ukuran populasi yang optimal untuk proses algoritma genetika pada optimasi pemerataan mutasi guru.

### 3.7. Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan, implementasi dan pengujian metode yang diterapkan telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis metode yang diterapkan. Kesimpulan diambil untuk menjawab rumusan masalah yang telah ditetapkan sebelumnya. Tahap terakhir dari penulisan adalah saran yang dimaksudkan untuk memperbaiki kesalahan-kesalahan yang terjadi serta untuk memberikan pertimbangan atas penelitian selanjutnya.

## BAB IV

### PERANCANGAN

#### 4.1. Formulasi Permasalahan

Kasus yang akan diselesaikan pada penelitian ini adalah masalah pemutasian guru SD di wilayah Kabupaten Banyuwangi. Untuk data guru, pada penelitian ini data guru meliputi nama, nip, jenis kelamin, pangkat, kelas yang diajar, dan sekolah tempat mengajar. Untuk penilaian kualitas guru menggunakan 2 unsur yaitu Sasaran Kinerja Guru (SKP) dan Penilaian Perilaku Kerja (PPK). Untuk penilaian SKP menggunakan 11 parameter sedangkan untuk penilaian PPK menggunakan 7 parameter. Nilai dari kedua unsur tersebut kemudian akan dijumlahkan dengan acuan pada SKP memiliki bobot 60% sedangkan untuk PPK memiliki bobot 40%.

Setelah mendapatkan nilai kualitas dari tiap guru, maka nilai tersebut akan masuk ke tahap awal dari Algoritma Genetika sebelum membangkitkan populasi baru. Rentang nilai pada guru akan dijelaskan pada Tabel 4.1.

**Tabel 4.1** Rentang Nilai Kualitas Guru

Kriteria	Range
Sangat Baik	91 – 100
Baik	76 - 90
Cukup	61 - 75
Kurang	51 – 60
Buruk	50 ke bawah

## 4.2. Manualisasi

### 4.2.1. Perhitungan Manual

Dimisalkan seorang guru SD dengan jabatan sebagai guru agama (4A). Pada awal tahun, yang bersangkutan menyusun SKP dengan atasan langsungnya sesuai dengan penetapan kinerja/RKT tahun 2014 antara lain merencanakan dan melaksanakan pembelajaran dengan target sebagai berikut :

- a. Aspek kuantitas/output = 1 dokumen
- b. Aspek kualitas = 100
- c. Aspek waktu = 6 bulan
- d. Aspek biaya = Rp. 75.000.000.-

Kemudian pada akhir tahun realisasinya sebagai berikut :

- a. Aspek kuantitas/output = 1 dokumen
- b. Aspek kualitas = 90
- c. Aspek waktu = 5 bulan
- d. Aspek biaya = Rp. 74.000.000.-

Dengan melihat kembali Persamaan 2-1 hingga Persamaan 2-10 maka perhitungan penilaian capaian SKP sebagai berikut :

$$\begin{aligned}
 \text{Penilaian SKP} &= \frac{\text{Realisasi Output (RO)}}{\text{Target Output (TO)}} \times 100 \\
 \text{(Aspek kuantitas)} &= \frac{1}{1} \times 100 \\
 &= 100
 \end{aligned}$$

$$\begin{aligned}
 \text{Penilaian SKP} &= \frac{\text{Realisasi Kuantitas (RK)}}{\text{Target Kuantitas (TK)}} \times 100 \\
 \text{(Aspek kualitas)} &= \frac{90}{100} \times 100 \\
 &= 90
 \end{aligned}$$

Untuk mengetahui presentase efisiensi waktu dengan menggunakan persamaan

$$\begin{aligned}
 \text{Presentase Efisiensi Waktu} &= 100\% \times \left[ \frac{\text{Realisasi Waktu (RW)}}{\text{Target Waktu (TW)}} \times 100\% \right] \\
 &= 100\% \times \left[ \frac{5}{6} \times 100\% \right] \\
 &= 100\% - 83,33\% \\
 &= 16,67\%
 \end{aligned}$$

Jadi tingkat efisiensi waktu = 16,67% sehingga merupakan bagian dari tingkat efisiensi biaya  $\leq 24\%$ . Kemudian dari hasil tersebut maka akan mencari nilai capaian SKP aspek waktu sebagai berikut :

$$\begin{aligned}
 \text{Nilai Capaian SKP Aspek Waktu} &= \frac{1.76 \times \text{Target Waktu} - \text{Realisasi Waktu}}{\text{Target Waktu}} \times 100 \\
 &= \frac{1.76 \times 6 - 5}{6} \times 100 \\
 &= \frac{5,56}{6} \times 100 \\
 &= 92,67
 \end{aligned}$$

Untuk mengetahui presentase efisiensi biaya adalah sebagai berikut :

$$\begin{aligned}
 \text{Presentasi Efisiensi Biaya} &= 100\% - \left[ \frac{\text{Realisasi Biaya (RB)}}{\text{Target Biaya (TB)}} \times 100\% \right] \\
 &= 100\% - \left[ \frac{74.000.000}{75.000.000} \times 100\% \right] \\
 &= 100\% - 98,67\% \\
 &= 1,33\%
 \end{aligned}$$

Karena presentase efisiensi biaya 1,33% merupakan bagian tingkat efisiensi biaya  $\leq 24\%$  dari target yang ditentukan, maka perhitungan nilai capaian SKP aspek biaya adalah sebagai berikut :

$$\begin{aligned}
 \text{Nilai Capaian} &= \frac{1,76 \times \text{Target Biaya (TB)} - \text{Realisasi Biaya (RB)}}{\text{Target Biaya (TB)}} \\
 \text{SKP Aspek} & \\
 \text{Biaya} & \times 100 \\
 &= \frac{1,76 \times 75.000.000 - 74.000.000}{75.000.000} \times 100 \\
 &= \frac{58.000.000}{75.000.000} \times 100 \\
 &= 77,33
 \end{aligned}$$

Dengan demikian dengan semua aspek yang sudah diketahui nilainya, maka akan didapatkan hasil akhir sebagai berikut :

$$\begin{aligned}
 \text{Nilai Capaian} &= \frac{100 + 90 + 92,67 + 77,33}{4} \\
 \text{SKP} & \\
 &= \frac{360}{4} \\
 &= 90 \text{ (Baik)}
 \end{aligned}$$

Langkah-langkah diatas dilakukan untuk setiap pencarian penilaian dari sebuah kegiatan yang akan dilaksanakan. Dalam penelitian ini setiap guru akan disama ratakan untuk jenis kegiatannya yang berjumlah 11 buah seperti pada tabel 2.1 dengan kegiatan yang sama.

Setelah mendapatkan nilai SKP dari tiap guru, maka langkah selanjutnya adalah mencari nilai Penilaian Perilaku Kerja. Untuk parameter Perilaku Kerja, pada penelitian ini setiap guru menggunakan 7 parameter penilaian dan untuk perhitungannya hanya mencari nilai rata-rata dari ketujuh nilai parameter tersebut, seperti pada Tabel 2.2. Setelah itu maka akan dicari Nilai Prestasi Kerja dari setiap guru yang akan ditunjukkan pada tabel 4.2.

**Tabel 4.2** Perhitungan Nilai Prestasi Kerja

Unsur yang dinilai				Jumlah
a. Sasaran Kerja Pegawai (SKP)		90 × 60%		54
b. Perilaku Kerja	a. Kesetiaan	91	Amat Baik	
	b. Prestasi Kerja	82	Baik	
	c. Tanggung Jawab	82	Baik	
	d. Ketaatan	81	Baik	
	e. Kejujuran	81	Baik	
	f. Kerjasama	81	Baik	
	g. Prakarsa	81	Baik	
	Jumlah	579	-	
	Nilai rata-rata	83	Baik	
Nilai Perilaku Kerja		83 × 40%		33,2
<b>NILAI PRESTASI KERJA</b>				<b>87,2 (Baik)</b>

#### 4.2.2. Membuat Populasi Awal

Pada tahap ini merupakan proses pembuatan populasi awal sebanyak jumlah populasi yang sudah ditentukan. Panjang dari masing-masing kromosom tergantung dengan guru yang terpilih untuk dimutasikan. Nilai dari tiap kromosom didapatkan dari nilai random.. Tiap gen berisi nilai random yang merepresentasikan guru yang terpilih.

Sebelum membagikan populasi awal, yang pertama kali dilakukan adalah mencari kinerja dari seluruh guru. Misal ingin melakukan pemerataan guru SD 10 sekolah dasar di wilayah Kabupaten Banyuwangi. Diketahui tiap sekolah

mempunyai 6 guru SD, 1 guru penjaskes dan 1 guru agama. Setelah melakukan proses perhitungan SKP dan Penilaian Perilaku Kinerja Guru, maka tabel kinerja guru akan ditunjukkan pada Tabel 4.3 dengan range nilai 0-100.



Tabel 4.3 Tabel Kinerja Pegawai

Range 0-100	SDN 1 L ATENG	SDN 3 L ATENG	SDN 5 L ATENG	SDN 1 SING OTRUNAN	SDN 2 SING OTRUNAN	SDN 3 SING OTRUNAN	SDN 4 SING OTRUNAN	SDN PENG ANTIGAN	SDN KAMPUNG MANDAR	SDN KAMPUNG MELAYU
Guru 1	61	68	77	5	6	7	19	68	68	39
Guru 2	24	97	81	44	35	43	66	20	51	71
Guru 3	39	77	88	95	32	6	89	5	100	14
Guru 4	64	86	32	19	39	32	9	65	22	35
Guru 5	90	47	18	11	17	52	43	70	53	95
Guru 6	89	93	91	36	92	24	10	57	100	98
Guru Penjaskes	29	78	100	34	72	98	84	15	66	65
Guru Agama	13	56	56	100	15	61	22	91	23	46

#### 4.2.2.1. Representasi Kromosom

Dari hasil tabel kinerja pegawai pada Tabel 4.3, selanjutnya akan dipilih guru mana saja yang akan melakukan mutasi pegawai. Guru yang tidak terpilih tidak akan diproses dalam sistem ini. Misal dari 10 sekolah tersebut terpilih 8 guru yang akan dimutasi dari 3 sekolah. Nilai pada tabel merupakan nilai kinerja pegawai dari hasil perhitungan pada subbab sebelumnya yang akan ditunjukkan pada Tabel 4.4.

**Tabel 4.4** Guru Terpilih

Range 0-100	SD 1	SD 2	SD 3
Guru Kelas 1	61	68	77
Guru Kelas 2	24	97	81
Guru Kelas 3	39	77	88
Guru Kelas 4	64	86	32
Guru Kelas 5	90	47	18
Guru Kelas 6	89	93	91
Guru Penjaskes	29	78	100
Guru Agama	13	56	56

Pada Tabel 4.4., sel yang berwarna merah merupakan guru yang terpilih untuk dimutasi. Ditunjukkan terdapat 8 guru yang akan dimutasi. Pada sekolah pertama terdapat 3 guru, sekolah kedua terdapat 2 guru, dan sekolah ketiga terdapat 3 guru yang akan dimutasikan. Tabel tersebut akan direpresentasikan menjadi sebuah kromosom yang ditunjukkan pada Tabel 4.5.

**Tabel 4.5** Inisialisasi Guru Terpilih

Id	SD	Status	
1	SD 1	Guru Kelas 1	G1
2	SD 1	Guru Kelas 3	G2
3	SD 1	Guru Penjaskes	G3
4	SD 2	Guru Kelas 2	G4
5	SD 2	Guru Kelas 4	G5
6	SD 3	Guru Kelas 1	G6
7	SD 3	Guru Kelas 5	G7
8	SD 3	Guru Kelas 6	G8



G1	G2	G3	G4	G5	G6	G7	G8
1	2	3	4	5	6	7	8

Pada tabel 4.5, semua guru yang terpilih akan disimbolkan dengan G1, G2, G3... Gn. Misal Guru kelas 1 pada SD 1 akan dimisalkan G1, Guru kelas 3 pada SD 1 dimisalkan G2, begitu seterusnya sampai setiap guru diinisialisasikan kedalam G. Tabel 4.5 didapatkan sebuah kromosom dengan panjang 8 gen yang merepresentasikan tiap guru. Tiap guru juga mempunyai identitas sendiri yang akan digunakan untuk proses pada subbab selanjutnya. Contoh representasi kromosom akan ditunjukkan pada Tabel 4.6.

**Tabel 4.6** Representasi Kromosom

Kromosom 1	G1	G2	G3	G4	G5	G6	G7	G8
	4	7	6	1	8	5	2	3

Tabel 4.6 merupakan contoh pembangkitan kromosom secara acak. Kromosom akan mempunyai panjang 8 gen dimana memawikili jumlah guru yang akan mengalami mutasi. Nilai dari setiap gen juga tidak boleh lebih dari 8 dengan alasan yang sama..

**4.2.2.2. Konversi Kromosom Menjadi Solusi**

Pada subbab ini akan dijelaskan bagaimana proses pencarian solusi dari sebuah kromosom. Pada sistem ini, jumlah kromosom akan ditentukan oleh user dan nilai dari kromosom tersebut akan bernilai random. Untuk panjang gen dari tiap kromosom tergantung pada banyaknya guru yang terpilih untuk dimutasikan.

Menggunakan kromosom pada Tabel 4.6 dijelaskan lebih lanjut pada Tabel 4.7 untuk pembacaan kromosom dan dimisalkan kromosom menggunakan patokan dari inisialisasi guru yang terpilih pada Tabel 4.5..

**Tabel 4.7** Contoh Pembangkitan Kromosom Secara Acak

	Gen 1 (G1)	Gen 2 (G2)	Gen 3 (G3)	Gen 4 (G4)	Gen 5 (G5)	Gen 6 (G6)	Gen 7 (G7)	Gen 8 (G8)
Kromosom	4	7	6	1	8	5	2	3
Keterangan	Guru 2 SD2	Guru 5 SD3	Guru 1 SD3	Guru 1 SD1	Guru 6 SD3	Guru 4 SD2	Guru 3 SD1	Guru Penjaskes SD1

Dari Tabel 4.7 merupakan kromosom yang dibangkitkan secara acak dan keterangan dari masing-masing gen. Pada kromosom yang dibangkitkan random tersebut mempunyai panjang 8 gen dengan angka yang saling berbeda. Angka-angka tersebut merupakan identitas guru yang merepresentasikan guru yang sudah terpilih untuk dimutasi.

Misal pada gen pertama pada Tabel 4.7 bernilai 4 dan terdapat pada G1, berarti Guru 2 pada SD 2 yang memiliki id = 4 menempati G1 yang merupakan posisi awal dari Guru 1 pada SD 1. Berarti semua atribut yang melekat pada Guru 2 SD 2 akan dipindah ke Guru 1 SD 1. Untuk gen selanjutnya akan dilakukan proses yang sama. Untuk lebih jelasnya dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil Konversi Kromosom Menjadi Solusi

Data Guru			Solusi		
Guru 1 SD 1	Guru 1 SD 2	Guru 1 SD 3	Guru 2 SD 2	Guru 1 SD 2	Guru 4 SD 2
Guru 2 SD 1	Guru 2 SD 2	Guru 2 SD 3	Guru 2 SD 1	Guru 1 SD 1	Guru 2 SD 3
Guru 3 SD 1	Guru 3 SD 2	Guru 3 SD 3	Guru 5 SD 3	Guru 3 SD 2	Guru 3 SD 3
Guru 4 SD 1	Guru 4 SD 2	Guru 4 SD 3	Guru 4 SD 1	Guru 6 SD 3	Guru 4 SD 3
Guru 5 SD 1	Guru 5 SD 2	Guru 5 SD 3	Guru 5 SD 1	Guru 5 SD 2	Guru 3 SD 1
Guru 6 SD 1	Guru 6 SD 2	Guru 6 SD 3	Guru 6 SD 1	Guru 6 SD 2	Guru Penjaskes SD 1
Guru Penjaskes SD 1	Guru Penjaskes SD 2	Guru Penjaskes SD 3	Guru 1 SD 3	Guru Penjaskes SD 2	Guru Penjaskes SD 3
Guru Agama SD 1	Guru Agama SD 2	Guru Agama SD 3	Guru Agama SD 1	Guru Agama SD 2	Guru Agama SD 3

Pada Tabel 4.8 diatas dapat dilihat bagaimana guru yang terpilih akan dimutasi sesuai dengan kromosom yang dibangkitkan secara random tersebut. Guru yang terpilih berada pada sel yang berwarna. Jika dilihat, tabel kiri merupakan representasi kromosom awal yang menjadi patokan awal, sedangkan tabel yang sebelah kanan merupakan bagaimana kondisi guru yang terpilih tersebut setelah dimutasi sesuai dengan kromosom yang dibangkitkan. Bisa dilihat warna dari masing-masing guru yang terpilih berbeda-beda untuk mewakili masing-masing guru yang terpilih.

Untuk tahap selanjutnya kemudian akan dibangkitkan kromosom acak sebanyak inputan user pada sistem. Nantinya kromosom-kromosom ini akan menjadi populasi awal untuk melakukan proses algoritma genetika.

#### 4.2.3. Menghitung Nilai Fitness

Setelah membangkitkan populasi awal, tiap kromosom akan dicari nilai *fitness* dari masing-masing kromosom terlebih dahulu. Nilai *fitness* ini nantinya akan digunakan untuk seleksi *elitism* pada proses selanjutnya. Semakin besar nilai *fitness* pada suatu kromosom maka semakin berpeluang untuk lolos ke generasi berikutnya.

Nilai *fitness* pada sistem ini didapatkan menggunakan beberapa rumus yang terkait. Pada penelitian ini, akan dicari terlebih dahulu nilai standar deviasi pada nilai tiap sekolah yang terpilih. Perhitungan standar deviasi ditunjukkan pada Persamaan 4-1 dan Persamaan 4-2.

$$s = \sqrt{s^2} \quad (4-11)$$

$$s^2 = \frac{n \times \sum_{i=1}^n X_i^2 - (\sum_{i=1}^n X_i)^2}{n \times (n - 1)} \quad (4-2)$$

Keterangan :

$s$  = Standar deviasi

$s^2$  = Varian

$n$  = Jumlah sekolah

$\sum_{i=1}^n X_i^2$  = Jumlah nilai kuadrat dari tiap sekolah

$\left(\sum_{i=1}^n X_i\right)^2$  = Nilai kuadrat dari jumlah nilai sekolah

Setelah melakukan pencarian nilai standar deviasi, kemudian akan dilakukan pencarian nilai *Coefficient of Variation*. *Coefficient of Variation* berfungsi untuk melihat sebaran data dari rata-rata hitungannya. Pada kasus ini, semakin kecil nilai *Coefficient of Variation* maka semakin seragam nilai kualitas sekolah pada sekolah-sekolah yang terpilih. Perhitungan *Coefficient of Variation* akan ditunjukkan pada Persamaan 4-3.

$$Cv = \frac{s}{\bar{X}} \quad (4-3)$$

Keterangan :

$Cv$  = *Coefficient of Variation*

$S$  = Standar deviasi

$\bar{X}$  = Rata-rata

Setelah mendapatkan nilai *Coefficient of Variation*, langkah selanjutnya adalah mencari nilai *fitness*. Dikarenakan kondisi  $Cv$  dimana semakin kecil nilai  $Cv$  maka nilai sekolah semakin seragam, maka perhitungan untuk mencari nilai *fitness* akan ditunjukkan pada Persamaan 4-4.

$$\text{fitness} = (1 - Cv) - \text{Penalti} \quad (4-4)$$

Sekolah yang digunakan adalah sekolah yang dimana salah satu atau lebih gurunya terpilih untuk dimutasikan. Contoh pencarian nilai fitness pada suatu kromosom yang ditunjukkan pada Tabel 4.9.

**Tabel 4.9** Contoh Perhitungan Nilai *Fitness*

Range 0-100	SD 1	SD 2	SD 3
Guru 1	97	68	86
Guru 2	24	61	81
Guru 3	18	77	88
Guru 4	64	91	32
Guru 5	90	47	39
Guru 6	89	93	29
Guru Penjaskes	77	78	100
Guru Agama	13	56	56
jumlah	472	571	511
fitness	-0.096273804		

Dengan menggunakan contoh kromosom yang dibangkitkan seperti pada Tabel 4.6 maka didapatkan nilai *fitness* sebesar -0.096273804. Perhitungan ini menghasilkan sebuah nilai fitness dari satu buah kromosom, pada contoh ini menggunakan kromosom pada Tabel 4.6. Menggunakan Persamaan 4-1 sampai Persamaan 4-4, maka proses pencarian nilai *fitness* akan dijelaskan seperti berikut.

$$\begin{aligned}
 s^2 &= \frac{n \times \sum_{i=1}^n X_i^2 - (\sum_{i=1}^n X_i)^2}{n \times (n - 1)} \\
 &= \frac{3 \times 809946 - 2414916}{3 \times (3 - 1)} \\
 &= \frac{2429838 - 2414916}{3 \times 2} \\
 &= \frac{14932}{6} \\
 &= 2488.667
 \end{aligned}$$

$$s = \sqrt{2488.667}$$

$$= 49.886$$

$$Cv = \frac{s}{\bar{X}}$$

$$= \frac{49.886}{518}$$

$$= 0.0963$$

$$\text{Fitness} = (1 - Cv) - \text{Penalti}$$

$$= (1 - 0.003162) - 1$$

$$= -0.963$$

Nilai *fitness* pada kromosom tersebut dikurangi dengan penalti. Pada penelitian ini penalti bernilai 1 dikarenakan nilai *fitness* maksimum dari sebuah kromosom bernilai 1. Kondisi nilai *fitness* sebuah kromosom dapat dikatakan bernilai 1 jika hasil mutasi guru tersebut menghasilkan nilai jumlah sekolah yang sama persis antar sekolah. Untuk itu pada penelitian ini nilai penalti bernilai 1 karena nantinya nilai *fitness* untuk kromosom yang terkena penalty akan berada pada rentang -1 sampai 0 sehingga kemungkinan untuk terseleksi menjadi individu baru pada populasi selanjutnya akan semakin kecil.

Nilai *fitness* pada kromosom tersebut dikurangi dengan nilai 1 pada perhitungan *fitness* karena kromosom tersebut terkena penalti sehingga hasil perhitungan *fitness* tersebut dikurangi dengan nilai penalti yang bernilai 1 dan menghasilkan nilai minus.

Jika dilihat pada Tabel 4.9, sel yang berwarna merah merupakan gen yang terkena penalti. Karena gen yang bernilai 29 tersebut berisi hasil mutasi guru penjaskes SD 1 dimana setelah dilakukan mutasi guru tersebut menempati posisi guru kelas 6 SD 3, sedangkan gen yang bernilai 77 merupakan hasil mutasi guru kelas 1 SD 3 dimana setelah dilakukan mutasi guru tersebut menempati posisi guru

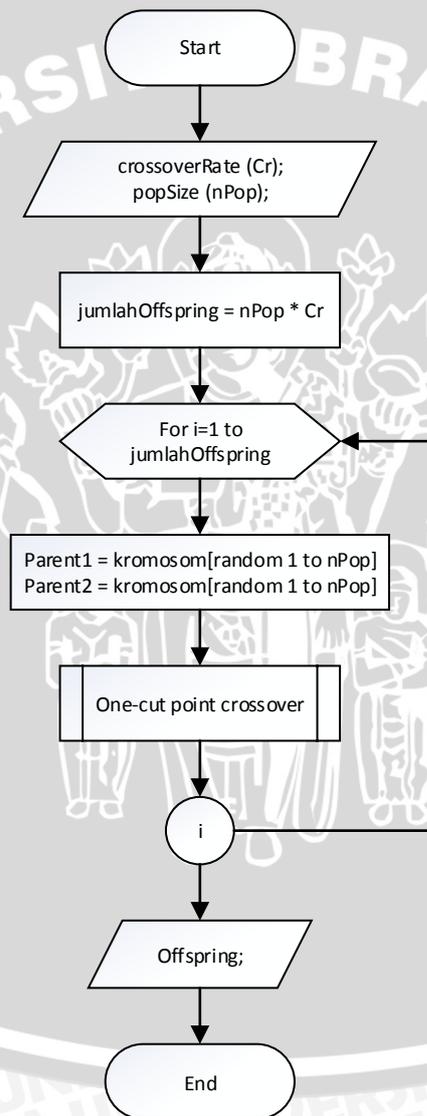
Penjaskes SD 1. Cara yang sama diulangi untuk mendapatkan nilai *fitness* pada kromosom yang lainnya.



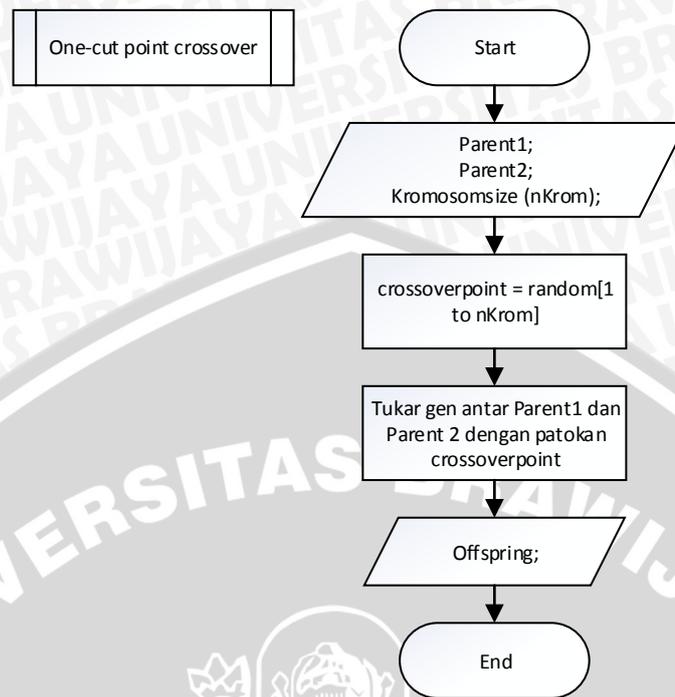
### 4.3. Proses Algoritma Genetika

#### 4.2.1. Crossover

Pada proses *crossover* akan melakukan persilangan antara dua buah induk yang sudah terpilih secara acak. Dua buah induk ini nantinya akan menghasilkan anak (*offspring*). *Flowchart* dalam melakukan proses *crossover* ditunjukkan pada pada Gambar 4.1 dan Gambar 4.2.



Gambar 4.1 Proses Crossover



**Gambar 4.2** Sub Proses One-cut point *crossover*

Pada *flowchart* diatas dijelaskan bagaimana proses *crossover*, dalam penelitian ini metode yang digunakan untuk melakukan *crossover* adalah *one-cut point crossover*. *One-cut point crossover* merupakan metode persilangan yang paling sederhana.

Prinsip kerja metode ini adalah dengan menentukan *crossover point* tertentu terlebih dahulu. *Offspring* yang akan dihasilkan akan berisi gen hasil kawin silang antara kedua induknya. *Offspring* akan berisi gen induk pertama sampai dengan *crossover point* yang sudah ditentukan sebelumnya, untuk gen selanjutnya akan memilih gen pada induk kedua dengan ketentuan gen tersebut belum terdapat pada gen yang sudah terisi pada *offspring* sebelumnya.

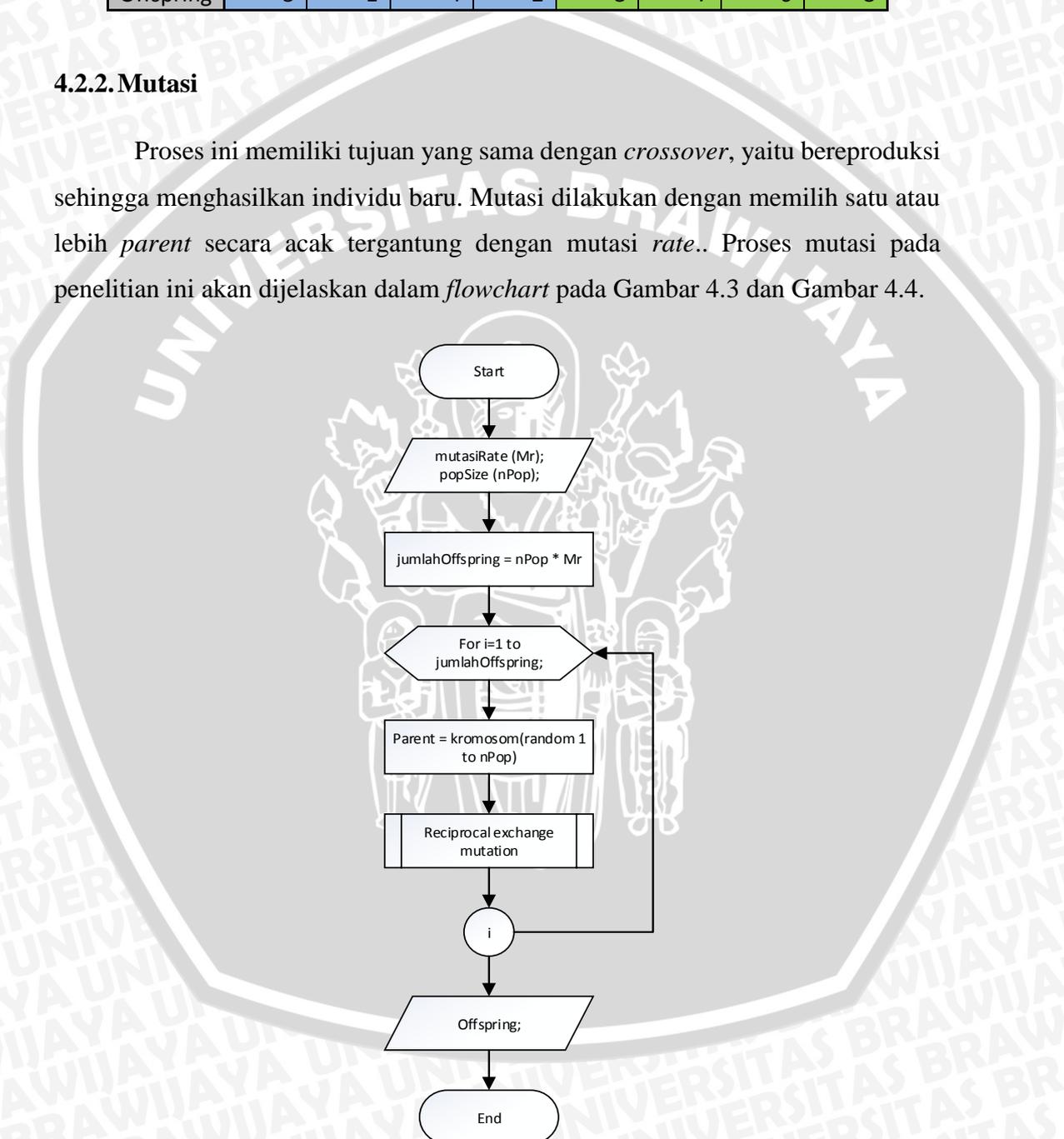
Banyaknya *offspring* yang dihasilkan proses *crossover* pada penelitian ini tergantung pada *crossover rate*, yaitu  $Cr \times$  banyak kromosom. Misal  $Cr$  bernilai 0.2 dan banyak kromosom 10 maka  $0.1 \times 10 = 1$ . Maka akan dihasilkan 1 *offspring*. Proses *crossover* akan ditunjukkan pada Tabel 4.10..

**Tabel 4.10** Proses *Crossover*

	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5	Gen 6	Gen 7	Gen 8
Parent 1	5	1	4	2	8	3	6	7
Parent 2	5	3	7	6	2	8	4	1
Offspring	5	1	4	2	3	7	6	8

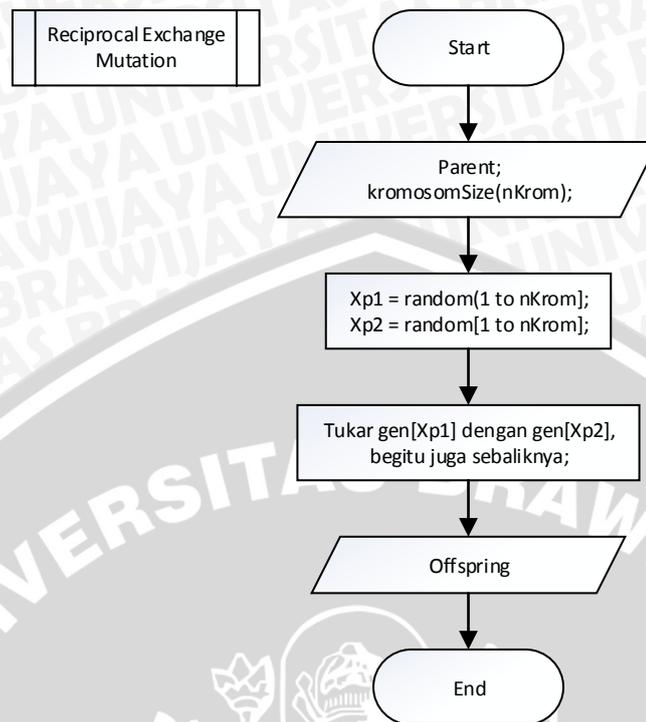
**4.2.2. Mutasi**

Proses ini memiliki tujuan yang sama dengan *crossover*, yaitu bereproduksi sehingga menghasilkan individu baru. Mutasi dilakukan dengan memilih satu atau lebih *parent* secara acak tergantung dengan mutasi *rate*.. Proses mutasi pada penelitian ini akan dijelaskan dalam *flowchart* pada Gambar 4.3 dan Gambar 4.4.



**Gambar 4.3** Proses Mutasi





**Gambar 4.4** Subproses *Reciprocal Exchange Mutation*

Pada Gambar 4.3 dan Gambar 4.4 dijelaskan bagaimana proses mutasi pada penelitian ini dilakukan. Dalam penelitian ini metode pada proses mutasi menggunakan *Reciprocal Exchange Mutation*. Metode ini merupakan metode yang paling sederhana. Prinsip kerja metode ini adalah menentukan dua *exchange point* terlebih dahulu pada sebuah induk yang sudah didapatkan dari hasil random sebelumnya. Kemudian gen yang terpilih untuk menjadi *exchange point* akan saling ditukar dan akan menghasilkan *offspring*. Banyaknya *offspring* yang dihasilkan proses mutasi pada penelitian ini tergantung pada mutasi *rate*, yaitu  $M_r \times \text{banyak kromosom}$ . Misal  $M_r$  bernilai 0.2 dan banyak kromosom 10 maka  $0.2 \times 10 = 2$ . Maka akan dihasilkan 2 *offspring*. Proses mutasi akan ditunjukkan pada Tabel 4.11.

Tabel 4.11 Proses Mutasi

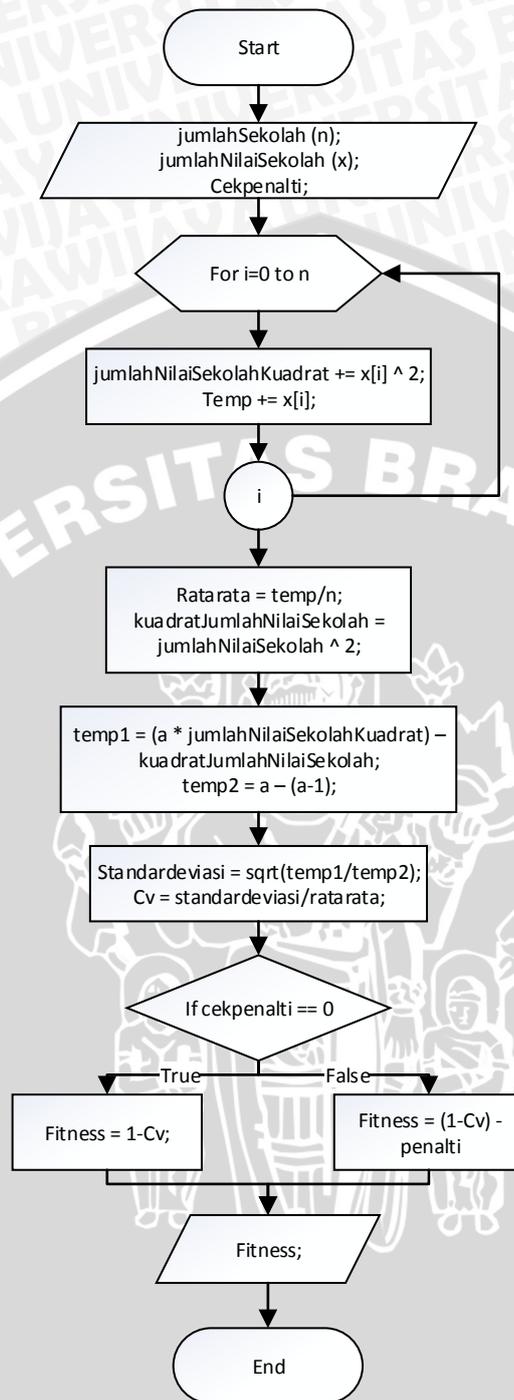
	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5	Gen 6	Gen 7	Gen 8
Parent	2	4	6	8	1	7	5	3
Offspring 1	2	5	6	8	1	7	4	3

Parent	8	5	2	4	6	3	1	7
Offspring 2	8	5	3	4	6	2	1	7

### 4.2.3. Evaluasi

Evaluasi digunakan untuk menghitung nilai *fitness* dari seluruh kromosom, baik itu parent maupun *offspring* dari hasil *crossover* dan mutasi. Masing-masing kromosom memiliki sebuah nilai *fitness*. Proses evaluasi pada penelitian ini akan dijelaskan dalam flowchart pada Gambar 4.5.





**Gambar 4.5** Proses Evaluasi *Fitness*

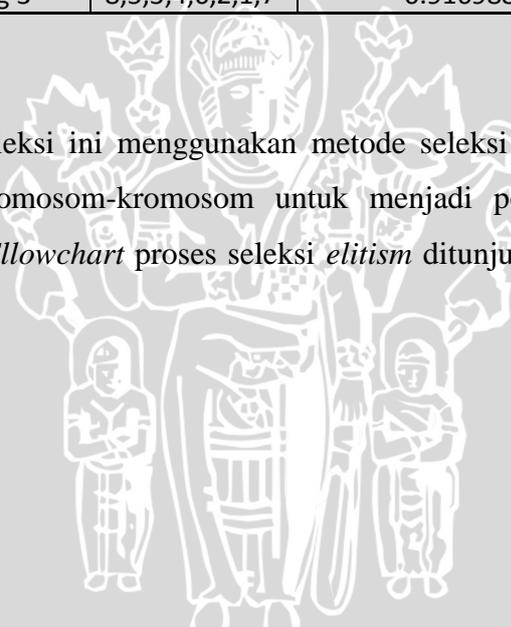
Nilai *fitness* yang didapatkan pada proses evaluasi akan digunakan menjadi acuan untuk proses seleksi pada tahap selanjutnya. Hasil evaluasi dapat dilihat pada Tabel 4.12.

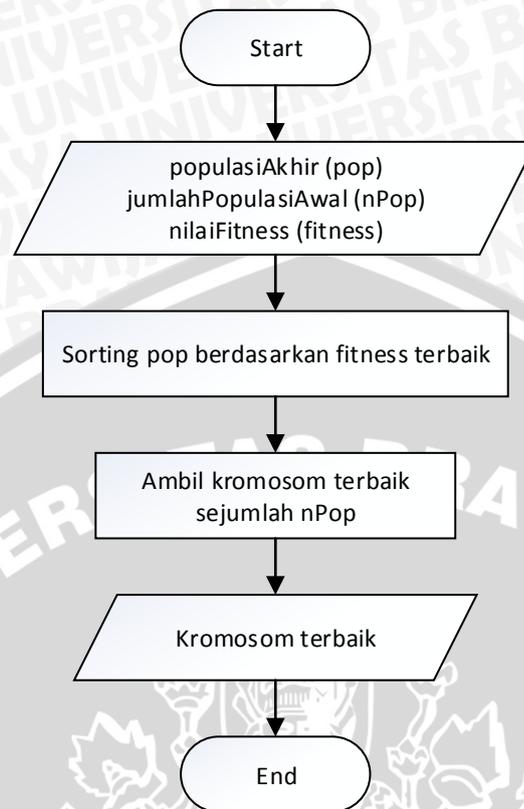
**Tabel 4.12** Hasil Evaluasi

No	Kromosom	Fitness
Parent 1	3,6,2,7,5,8,4,1	-0.17953668
Parent 2	4,8,3,5,7,2,1,6	0.959459459
Parent 3	5,1,4,2,8,3,6,7	-0.071428571
Parent 4	6,1,2,7,9,4,8,3	-0.106948229
Parent 5	5,3,7,6,2,8,4,1	-0.202702703
Parent 6	1,6,5,2,3,8,4,7	-0.05984556
Parent 7	2,4,6,8,1,7,5,3	-0.174386921
Parent 8	7,4,3,5,1,6,2,8	0.818532819
Parent 9	8,5,2,4,6,3,1,7	-0.102316602
Parent 10	5,6,3,2,7,4,1,8	0.911196911
Offspring 1	5,1,4,2,3,7,6,8	-0.05984556
Offspring 2	2,5,6,8,1,7,4,3	-0.069498069
Offspring 3	8,5,3,4,6,2,1,7	0.916988417

#### 4.2.4. Seleksi

Pada proses seleksi ini menggunakan metode seleksi *elitism*. Proses ini bertujuan memilih kromosom-kromosom untuk menjadi populasi baru pada generasi selanjutnya. *Flowchart* proses seleksi *elitism* ditunjukkan pada Gambar 4.6.





**Gambar 4.6** Proses Elitism

Penyeleksian ini dilakukan dengan cara mencari nilai *fitness* terbaik kemudian dari semua kromosom termasuk *offspring* dari proses *crossover* dan mutasi akan dipilih yang terbaik sebanyak jumlah kromosom awal. Hasil dari proses tersebut akan menjadi kromosom akhir yang terpilih dan akan menjadi populasi baru pada generasi selanjutnya. Untuk contoh penerapannya dapat dilihat pada Tabel 4.13.

**Tabel 4.13** Tabel Proses Seleksi Elitism

No	Kromosom	Fitness
Parent 2	4,8,3,5,7,2,1,6	0.959459459
Offspring 3	8,5,3,4,6,2,1,7	0.916988417
Parent 10	5,6,3,2,7,4,1,8	0.911196911
Parent 8	7,4,3,5,1,6,2,8	0.818532819
Parent 6	1,6,5,2,3,8,4,7	-0.05984556
Offspring 1	5,1,4,2,3,7,6,8	-0.05984556
Offspring 2	2,5,6,8,1,7,4,3	-0.069498069
Parent 3	5,1,4,2,8,3,6,7	-0.071428571
Parent 9	8,5,2,4,6,3,1,7	-0.102316602
Parent 4	6,1,2,7,9,4,8,3	-0.106948229
Parent 7	2,4,6,8,1,7,5,3	-0.174386921
Parent 1	3,6,2,7,5,8,4,1	-0.17953668
Parent 5	5,3,7,6,2,8,4,1	-0.202702703

Dari Tabel 4.13 diatas, terdapat 13 kromosom akhir yang sudah diurutkan berdasarkan nilai *fitness* dari yang terbesar. Diketahui sebelumnya kromosom awal berjumlah 10, jadi dari populasi akhir pada Tabel 4.13 akan diambil 10 kromosom terbaik untuk menjadi populasi baru pada generasi selanjutnya. Setelah melakukan proses sorting, diketahui Parent 7, Parent 1 dan Parent 5 tidak akan lolos menjadi individu baru pada generasi selanjutnya. Populasi baru pada generasi selanjutnya akan ditunjukkan pada Tabel 4.14.

**Tabel 4.14** Populasi Baru Untuk Generasi Berikutnya

No	Kromosom	Fitness
Parent 2	4,8,3,5,7,2,1,6	0.959459459
Offspring 3	8,5,3,4,6,2,1,7	0.916988417
Parent 10	5,6,3,2,7,4,1,8	0.911196911
Parent 8	7,4,3,5,1,6,2,8	0.818532819
Parent 6	1,6,5,2,3,8,4,7	-0.05984556
Offspring 1	5,1,4,2,3,7,6,8	-0.05984556
Offspring 2	2,5,6,8,1,7,4,3	-0.069498069
Parent 3	5,1,4,2,8,3,6,7	-0.071428571
Parent 9	8,5,2,4,6,3,1,7	-0.102316602
Parent 4	6,1,2,7,9,4,8,3	-0.106948229

## BAB V

### IMPLEMENTASI

#### 5.1 Implementasi Program

Berdasarkan metodologi penelitian dan perancangan sistem yang telah dijelaskan pada Bab 3 dan Bab 4, maka selanjutnya akan diimplementasikan proses tersebut ke dalam sebuah sistem.

##### 5.1.1 Pembangkitan Populasi Awal

Proses awal dari metode ini yaitu pembangkitan populasi awal. Kromosom akan dibangkitkan secara otomatis oleh sistem sebanyak jumlah individu yang telah diinputkan oleh user sebelumnya. Tiap kromosom akan memiliki panjang gen yang sama tergantung pada jumlah guru yang terpilih yang sudah dipilih oleh user sebelumnya untuk dimutasikan.

Suatu individu diperoleh dengan membangkitkan angka 1 sampai n yang menyatakan jumlah guru yang terpilih untuk dimutasikan. Setiap gen dalam sebuah individu tidak akan ada yang bernilai sama. Proses pembangkitan populasi baru dapat dilihat pada Source Code 5.1.

```
//membangkitkan populasi awal
for (int i = 0; i < jmlindividu; i++)
{
    for (int j = 0; j < jumlahguru; j++)
    {
        kromosom[i, j] = j + 1;
    }
}

for (int i = 0; i < jmlindividu; i++)
{
    for (int j = 0; j < jumlahguru; j++)
    {
        int index = data.Next(j, jumlahguru - 1);
        int temp = kromosom[i, index];
        kromosom[i, index] = kromosom[i, j];
        kromosom[i, j] = temp;
    }
}
```

Source Code 5.1 Pembangkitan Populasi Awal

### 5.1.2 Proses Crossover

Pada proses *crossover*, metode yang digunakan adalah *one cut point crossover* dimana *offspring* didapatkan dari perkawinan antara dua induk yang sudah terpilih secara random. Segmen kiri dari *offspring* akan berisi gen pada induk pertama sampai batas yang sudah ditentukan yaitu *crossover point*. Untuk segmen kanan pada *offspring* akan berisi gen dari induk 2 dengan ketentuan gen tersebut belum terdapat pada segmen kiri.

Jumlah *offspring* yang dihasilkan pada proses ini tergantung pada hasil perkalian antara *crossover rate* dengan jumlah individu awal yang telah dibangkitkan. Proses *crossover* dapat dilihat pada Source Code 5.2.

```
Int jmloffspringcross = 0;
int pointcrossover = 0;
Random kromosomcross = new Random();
Random pointcross = new Random();
int[] idparentcross1 = new int[jmlindividu];
int[] idparentcross2 = new int[jmlindividu];
int[,] parentcross = new int[jmlindividu, jumlahguru];
int[,] offspringcross = new int[jmlindividu, jumlahguru];

jmloffspringcross = (int)(crossrate * jmlindividu);

for (int i = 0; i < jmloffspringcross; i++)
{
    idparentcross1[i] = kromosomcross.Next(1, jmlindividu);
    do
    {
        idparentcross2[i] = kromosomcross.Next(1, jmlindividu);
    }
    while (idparentcross2[i] == idparentcross1[i]);

    richTextBox7.Text += idparentcross1[i] + " " + idparentcross2[i] +
    "\n";

    pointcrossover = pointcross.Next(1, jumlahguru);
    int p2 = 0;
    for (int j = 0; j < jumlahguru; j++)
    {
        if (j < pointcrossover)
        {
            parentcross[i, j] = kromosom[idparentcross1[i] - 1, j];
        }
        else
        {
            int cek;
            do
            {
                cek = 0;
            }
        }
    }
}
```

```

        for (int k = 0; k < j; k++)
        {
            if (parentcross[i, k] == kromosom[idparentcross2[i] -
1, p2])
            {
                cek++;
            }
        }
        if (cek > 0) p2++;
    } while (cek > 0);
    parentcross[i, j] = kromosom[idparentcross2[i] - 1, p2];
}
richTextBox6.Text += parentcross[i, j] + " ";
}
richTextBox6.Text += "\n";
offspringcross = parentcross;
}

```

**Source Code 5.2** Proses Crossover

### 5.1.3. Proses Mutasi

Metode mutasi yang digunakan pada penelitian ini adalah *reciprocal exchange mutation*. Setelah menentukan induk yang terpilih untuk melakukan proses mutasi, kemudian dua gen akan terpilih secara acak pada induk tersebut. Dua gen tersebut kemudian akan saling menukarkan nilainya sehingga akan menghasilkan sebuah *offspring* baru.

Jumlah offspring yang dihasilkan pada proses mutasi ini didapatkan dari hasil perkalian *mutation rate* dengan jumlah individu awal yang telah dibangkitkan. Proses mutasi akan ditunjukkan pada Source Code 5.3.

```

int jmlOffspringmutasi;
Random kromosommutasi = new Random();
Random exchangepoint1 = new Random();
Random exchangepoint2 = new Random();
int xp1, xp2;

int[] idparentmutasi = new int[jmlindividu];
int[,] parentmutasi = new int[jmlindividu, jumlahguru];
int[,] offspringmutasi = new int[jmlindividu, jumlahguru];

jmlOffspringmutasi = (int)(mutasirate * jmlindividu);

for (int i = 0; i < jmlOffspringmutasi; i++)
{
    idparentmutasi[i] = kromosommutasi.Next(1, jmlindividu);
    xp1 = exchangepoint1.Next(1, jumlahguru);

```

```

do
{
    xp2 = exchangepoint2.Next(1, jumlahguru);
}
while (xp2 == xp1);

for (int j = 0; j < jumlahguru; j++)
{
    parentmutasi[i, j] = kromosom[idparentmutasi[i] - 1, j];
}

richTextBox5.Text += "id " + idparentmutasi[i] + " xp " + xp1 + xp2 +
"\n";

int temp;
temp = parentmutasi[i, xp1 - 1];
parentmutasi[i, xp1 - 1] = parentmutasi[i, xp2 - 1];
parentmutasi[i, xp2 - 1] = temp;

for (int j = 0; j < jumlahguru; j++)
{
    richTextBox4.Text += parentmutasi[i, j]+ " ";
}
richTextBox4.Text += "\n";
offspringmutasi = parentmutasi;
}

```

Source Code 5.3 Proses Mutasi

#### 5.1.4. Perhitungan Nilai *Fitness*

Nilai *fitness* diperoleh dengan menggunakan Persamaan 4-1 sampai Persamaan 4-4. Pada awalnya mencari nilai setiap sekolah dimana salah satu atau lebih guru pada sekolah tersebut terpilih untuk dimutasikan. Nilai setiap sekolah berisi jumlah dari nilai setiap guru pada sekolah tersebut. Dari nilai-nilai sekolah tersebut kemudian dicari nilai yang terkecil dari seluruh sekolah. Nilai minimum pada sekolah tersebut kemudian dibagi dengan nilai sekolah rata-rata. Dari perhitungan tersebut akan dikurangi dengan nilai pinalti jika memang kromosom tersebut terkena pinalti.

Perhitungan *fitness* dilakukan pada semua kromosom yang meliputi kromosom awal, *offspring* hasil dari proses *crossover* dan *offspring* dari proses mutasi. Proses perhitungan nilai *fitness* dapat dilihat pada Source Code 5.4.

```
int totalindividu = jmlindividu + jmloffspringcross + jmloffspringmutasi;
int[,] kromosomakhir = new int[totalindividu, jumlahguru];
int[] tempnot = new int[jumlahsekolah];

for (int i = 0; i < totalindividu; i++)
{
    if (i < jmlindividu)
    {
        for (int j = 0; j < jumlahguru; j++)
        {
            kromosomakhir[i, j] = kromosom[i, j];
        }
    }
    else if (i >= jmlindividu && i < (jmlindividu + jmloffspringcross))
    {
        for (int j = 0; j < jumlahguru; j++)
        {
            kromosomakhir[i, j] = offspringcross[(i - jmlindividu), j];
        }
    }
    else
    {
        for (int j = 0; j < jumlahguru; j++)
        {
            kromosomakhir[i, j] = offspringmutasi[(i - jmlindividu -
jmloffspringcross), j];
        }
    }
}

//mencari nilai guru yang tidak terpilih pada sekolah yang terpilih
for (int i = 0; i < jumlahguru; i++)
{
    String temp_npsn = npsnguru[i];
    for (int j = 0; j < jumlahsekolah; j++)
    {
        if (temp_npsn == npsnnot[j])
        {
            tempnot[j] += nilaiguru[i];
        }
    }
}

//memberi nilai penalti fitness pada kromosom tertentu
cekpinalti = new int[totalindividu];

for (int i = 0; i < totalindividu; i++)
{
    cekpinalti[i] = 0;
    for (int j = 0; j < jumlahguru; j++)
    {
        int temp_npsn = kromosomakhir[i, j] - 1;
        if ((idajarguru[temp_npsn] == 7 && idajarguru[j] <= 6) ||
(idajarguru[temp_npsn] == 8 && idajarguru[j] <= 6) ||
(idajarguru[temp_npsn] <= 6 && idajarguru[j] > 6) ||
(idajarguru[temp_npsn] == 7 && idajarguru[j] == 8) ||
(idajarguru[temp_npsn] == 8 && idajarguru[j] == 7) )
        {
            richTextBox12.Text += "PENALTY ";
        }
    }
}
```

```
        cekpinalti[i]++;
    }
    else richTextBox12.Text += "pass ";
}
}

//menambahkan nilai guru dalam satu sekolah
for (int h = 0; h < totalindividu; h++)
{
    for (int i = 0; i < jumlahguru; i++)
    {
        int nilai = kromosomakhir[h, i] - 1;
        String temp_npsn = npsnguru[i];
        for (int j = 0; j < jumlahsekolah; j++)
        {
            if (temp_npsn == npsnnot[j])
            {
                tempnot1[h, j] += nilaiguru[nilai];
            }
        }
    }
}

//mencari nilai fitness
fitness = new double[totalindividu];
Double standardeviasi;

for (int i = 0; i < totalindividu; i++)
{
    int minimal = tempnot1[i, 0];
    int jumAverage = 0;
    int kuadratjumlahnilaisekolah = 0;
    int jumlahnilaisekolahkuadrat = 0;
    int jumlahnilaisekolah = 0;
    double ratarata = 0;
    double cv = 0;

    richTextBox13.Text += "\n acak " + i + "\n";

    for (int j = 0; j < jumlahsekolah; j++)
    {
        jumlahnilaisekolah += tempnot1[i, j];
        jumlahnilaisekolahkuadrat += tempnot1[i, j] ^ 2;
        jumAverage += tempnot1[i, j];
    }

    ratarata = jumAverage / jumlahsekolah;
    kuadratjumlahnilaisekolah = jumlahnilaisekolah ^ 2;
    standardeviasi = Math.Sqrt(((jumlahsekolah *
jumlahnilaisekolahkuadrat) - kuadratjumlahnilaisekolah) / jumlahsekolah -
(jumlahsekolah - 1));
    cv = standardeviasi / ratarata;
    richTextBox3.Text += "Cv " + i + " : " + cv + "\n";

    if (cekpinalti[i] == 0) { fitness[i] = 1-cv; }
    else { fitness[i] = (1-cv) - 1; }
    richTextBox13.Text += " Hasil : " + fitness[i] + "\n";
}
}
```

### Source Code 5.4 Proses Perhitungan *Fitness*

#### 5.1.5. Proses Seleksi

Proses seleksi pada penelitian ini menggunakan metode *elitism* dimana penyeleksian individu dilakukan berdasarkan nilai *fitness* pada individu tersebut. Semakin besar nilai *fitness* pada individu tersebut, maka semakin tinggi peluang individu tersebut untuk menjadi individu baru pada generasi berikutnya.

Jumlah individu yang akan diseleksi sama dengan jumlah individu pada populasi awal. Proses seleksi *elitism* dapat dilihat pada Source Code 5.5.

```
//seleksi
int[] temp_id = new int[totalindividu];
for (int i = 0; i < totalindividu; i++)
{
    temp_id[i] = i;
}

Array.Sort(fitness, temp_id);

for (int i = 0; i < totalindividu; i++)
{
    richTextBox14.Text += fitness[i] + " : " + temp_id[i] + "\n";
}

int x = 0;
for (int i = totalindividu; i > 0; i--)
{
    if (x < jmlindividu)
    {
        for (int j = 0; j < jumlahguru; j++)
        {
            richTextBox15.Text += kromosomakhir[temp_id[i - 1], j] + " ";
            kromosom[x, j] = kromosomakhir[temp_id[i - 1], j];
        }
    }
    x++;
    richTextBox15.Text += "\n";
}
```

Source Code 5.5 Proses Seleksi *Eltism*

## 5.2. Implementasi Antarmuka Pengguna

Pada sistem ini implementasi antarmuka pengguna terdiri dari form untuk halaman utama, dimana dari halaman ini terdapat 3 tab menu yaitu menu Data Guru, Data Sekolah dan Mutasi.

### 5.2.1. Implementasi Halaman Data Guru

Halaman menu data guru merupakan tampilan antarmuka yang pertama kali muncul saat menjalankan program ini. Pada halaman ini pengguna dapat memasukkan data guru yang baru atau memperbarui data guru yang sudah ada. Pengguna juga dapat menghapus data guru tersebut dengan menekan button hapus. Untuk mendapatkan nilai tiap guru maka pengguna dapat menekan button nilai. Implementasi halaman data guru dapat dilihat pada Gambar 5.1.

The screenshot shows a web application window titled "Optimasi Mutasi Guru". It has three tabs: "Data Guru", "Data Sekolah", and "Mutasi". The "Data Guru" tab is active, displaying a form for entering teacher information. The form includes fields for NAMA, NIP, JENIS KELAMIN (with radio buttons for LAKI-LAKI and PEREMPUAN), PANGKAT, UNIT KERJA (dropdown menu), and STATUS (dropdown menu). There are "Insert" and "Update" buttons at the bottom of the form. Below the form is a table with columns: nip, nama, jenis\_kelamin, pangkat, nama\_sekolah, content\_ajar, Ubah, Hapus, and Nilai. The table contains several rows of data, with the first row highlighted in blue.

nip	nama	jenis_kelamin	pangkat	nama_sekolah	content_ajar	Ubah	Hapus	Nilai
123401923	okyprima	LAKI-LAKI	4C	SD MUHAMMADIYA...	GURU KELAS 4	Ubah	Hapus	Nilai
1	Guru 1	LAKI-LAKI	2C	SDN 1 LATENG	GURU KELAS 1	Ubah	Hapus	Nilai
8	Guru 8	LAKI-LAKI	4B	SDN 1 LATENG	GURU AGAMA	Ubah	Hapus	Nilai
7	Guru 7	PEREMPUAN	4E	SDN 1 LATENG	GURU PENJASK...	Ubah	Hapus	Nilai
6	Guru 6	PEREMPUAN	2B	SDN 1 LATENG	GURU KELAS 6	Ubah	Hapus	Nilai

**Gambar 5.1** Implementasi Halaman Data Guru

#### 5.2.1.1. Implementasi Halaman Penilaian Kinerja Guru

Halaman Penilaian Kinerja Guru adalah halaman yang digunakan pengguna untuk memasukkan data-data yang diperlukan untuk mendapatkan nilai seorang guru. Halaman ini akan muncul jika pengguna menekan button nilai pada salah satu guru di halaman data guru. Pada halaman ini terdapat 2 tab yaitu SKP dan PPK yang dapat dilihat pada Gambar 5.2 dan Gambar 5.3.

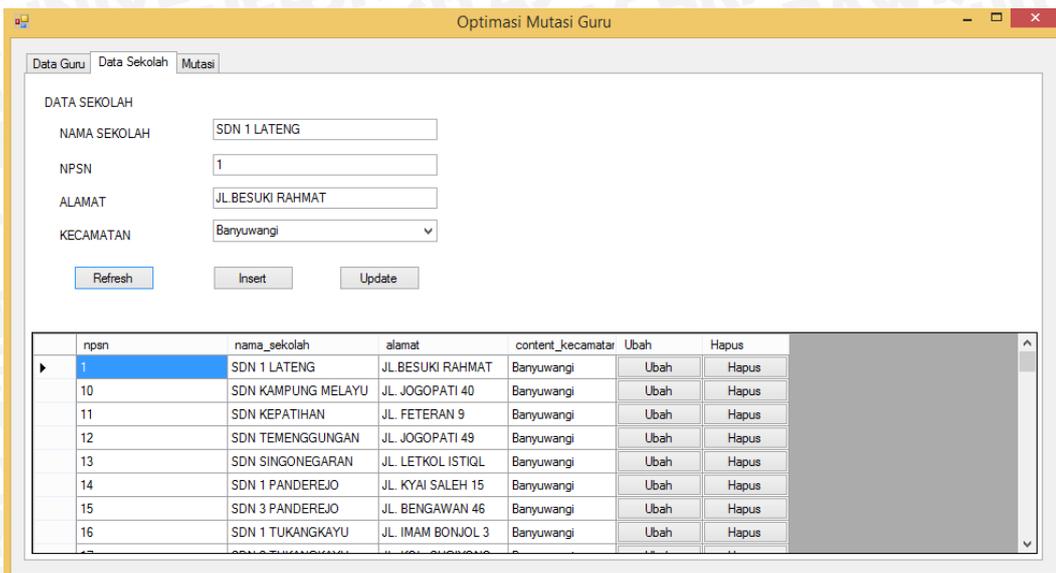
no	pertanyaan	hasil	Action
1	Merencanakan dan melaksanakan pembelajaran	84	Action
10	Menjadi anggota kegiatan kepramukaan, sebagai Pengurus aktif	90	Action
11	Menjadi tim penilai angka kredit	37	Action
2	Menjadi Kepala Sekolah/Madrasah per Tahun	92	Action
3	Menyusun kurikulum pada satuan pendidikannya	73	Action
4	Membimbing siswa dalam kegiatan ekstrakurikuler	90	Action
5	Mengikuti diklat fungsional lamanya antara 30 s.d 80 jam	68	Action
6	Lokakarya atau kegiatan bersama (seperti KKG/MGMP/KKKS/MKKS)	63	Action
7	Membuat artikel/buku/training/duk dalam bentuk dan format...	60	Action

**Gambar 5.2** Implementasi Halaman Penilaian Kinerja Guru SKP

**Gambar 5.3** Implementasi Halaman Penilaian Kinerja Guru PPK

### 5.2.2. Implementasi Halaman Data Sekolah

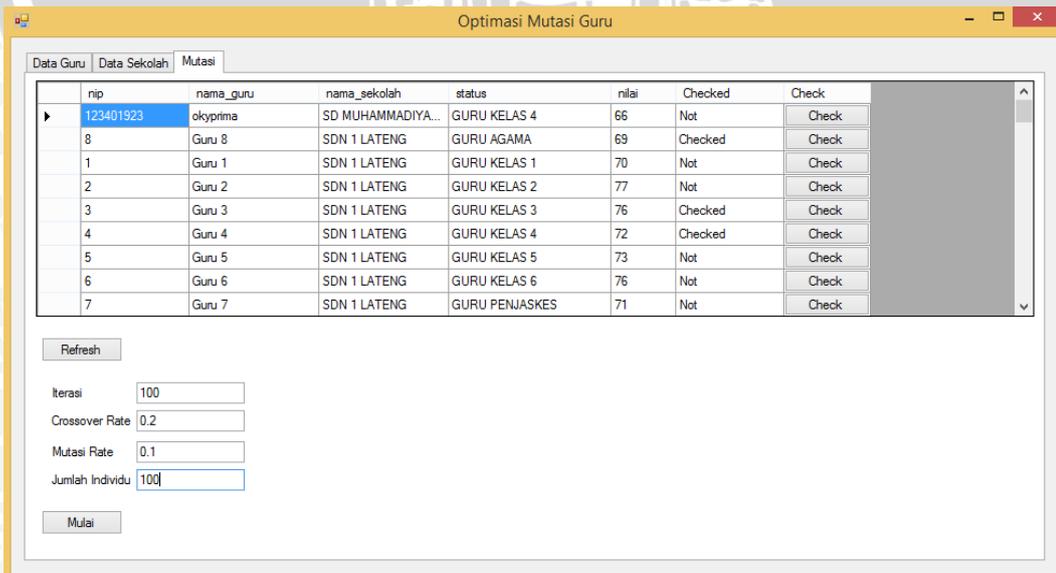
Halaman data sekolah digunakan pengguna untuk memproses segala sesuatu tentang sekolah. Pengguna dapat memasukkan data sekolah baru, memperbarui dan menghapus data sekolah yang sudah ada sebelumnya. Implementasi halaman data sekolah ditunjukkan pada Gambar 5.4.



Gambar 5.4 Implementasi Halaman Data Sekolah

### 5.2.3. Implementasi Halaman Mutasi Guru

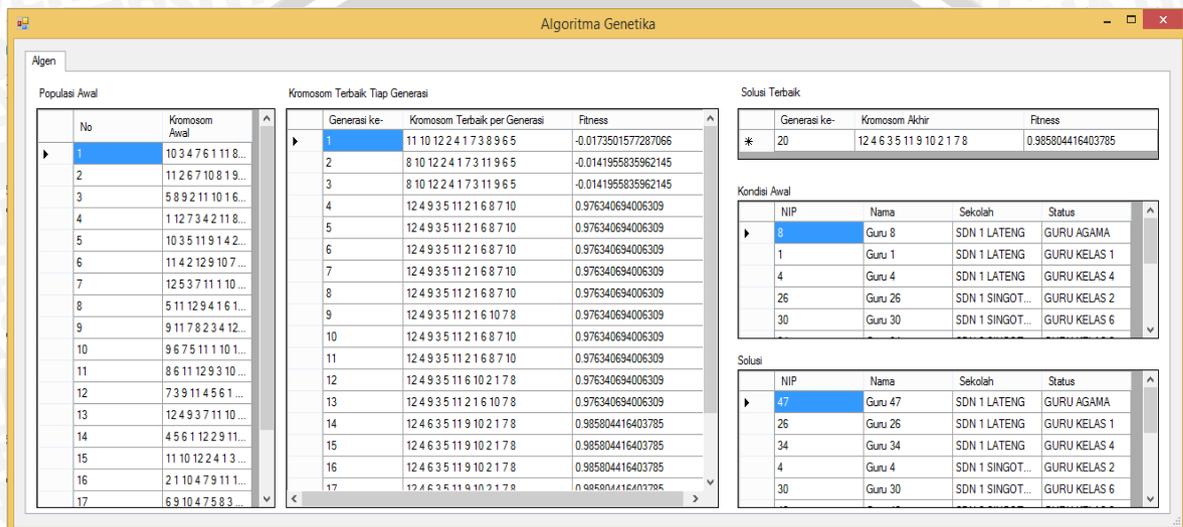
Halaman mutasi digunakan untuk memilih guru mana saja yang akan dimutasi. Dengan menekan button check maka guru tersebut akan terpilih untuk dimutasi. Terdapat juga beberapa *field* yang merupakan parameter awal untuk menjalankan metode algoritma genetika. Gambar 5.5 merupakan implementasi halaman mutasi.



Gambar 5.5 Implementasi Halaman Mutasi

### 5.2.4. Implementasi Halaman Proses Algoritma Genetika

Halaman proses algoritma genetika berisi proses dan menampilkan solusi yang paling optimal dengan metode algoritma genetika. Pada halaman ini, pengguna mendapatkan solusi bagaimana penempatan guru yang akan dimutasi agar mendapatkan solusi yang optimal. Gambar 5.6 merupakan implementasi halaman proses algoritma genetika.



Gambar 5.6 Implementasi Halaman Proses Algoritma Genetika

## BAB VI

### PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas mengenai hasil uji coba yang telah dilakukan dalam optimasi pemerataan mutasi guru dengan menggunakan algoritma genetika. Pengujian dilakukan untuk mengetahui parameter algoritma genetika yang paling optimal agar dapat menghasilkan *fitness* terbaik. Pada bab ini terdapat 3 macam pengujian yang akan dilakukan yaitu pengujian ukuran generasi, pengujian ukuran populasi dan pengujian kombinasi *crossover rate* dan *mutation rate*.

#### 6.1 Hasil Pengujian Sistem

Hasil pengujian sistem dilakukan untuk mengetahui kromosom terbaik dengan *fitness* tertinggi yang dihasilkan dari proses algoritma genetika. Dalam uji coba ini akan dilakukan pengujian terhadap 12 orang guru dari 4 sekolah yang berbeda. Guru yang terpilih pada pengujian ini ditunjukkan pada Tabel 6.1.

**Tabel 6.1** Guru yang Dimutasi Pada Pengujian

NIP	Nama	Sekolah	Status
8	Guru 8	SDN 1 LATENG	GURU AGAMA
1	Guru 1	SDN 1 LATENG	GURU KELAS 1
4	Guru 4	SDN 1 LATENG	GURU KELAS 4
26	Guru 26	SDN 1 SINGOTRUNAN	GURU KELAS 2
30	Guru 30	SDN 1 SINGOTRUNAN	GURU KELAS 6
34	Guru 34	SDN 2 SINGOTRUNAN	GURU KELAS 2
36	Guru 36	SDN 2 SINGOTRUNAN	GURU KELAS 4
16	Guru 16	SDN 3 LATENG	GURU AGAMA
11	Guru 11	SDN 3 LATENG	GURU KELAS 3
15	Guru 15	SDN 3 LATENG	GURU PENJASKES
42	Guru 42	SDN 3 SINGOTRUNAN	GURU KELAS 2
47	Guru 47	SDN 3 SINGOTRUNAN	GURU PENJASKES

Dari Tabel 6.1, kemudian dilakukan inisialisasi parameter algoritma genetika dengan banyak populasi 20 kromosom, banyak generasi 1, nilai *crossover rate* 0.5 dan *mutation rate* 0.5. Hasil pengujian sistem yang menghasilkan 20 kromosom terbaik pada generasi satu ditunjukkan pada Tabel 6.2.

**Tabel 6.2** Hasil Pengujian Sistem

No	Kromosom	Fitness
1	9 5 4 1 2 10 3 6 12 8 7 11	-0.079241998
2	7 5 4 9 2 3 11 1 10 12 8 6	-0.079304764
3	1 8 10 7 2 5 12 4 6 3 11 9	-0.079304764
4	7 6 11 12 5 2 8 9 3 10 1 4	-0.079304764
5	7 6 11 12 5 2 8 9 3 10 4 1	-0.079304764
6	7 4 10 1 5 9 6 8 12 3 2 11	-0.079304764
7	3 7 12 8 10 1 2 5 6 9 11 4	-0.079304764
8	7 11 1 5 6 8 9 3 4 12 2 10	-0.079304764
9	7 4 10 2 5 9 6 8 12 3 1 11	-0.07936748
10	6 9 5 2 11 1 12 4 3 10 8 7	-0.07936748
11	3 10 7 6 11 12 5 2 8 9 4 1	-0.07936748
12	1 8 10 7 2 9 11 12 4 6 3 5	-0.07936748
13	2 1 3 5 12 8 9 11 7 10 4 6	-0.07936748
14	3 10 1 2 12 9 4 5 6 8 11 7	-0.07936748
15	10 4 1 8 9 2 12 3 5 7 6 11	-0.07936748
16	7 9 11 2 1 12 4 10 6 8 3 5	-0.07936748
17	7 4 10 2 5 9 6 8 12 3 1 11	-0.07936748
18	6 1 12 4 9 2 11 5 7 10 3 8	-0.07936748
19	2 1 3 5 12 8 9 11 7 10 4 6	-0.07936748
20	9 7 11 2 1 12 4 10 6 8 3 5	-0.07936748
21	10 4 1 8 9 2 12 3 5 7 6 11	-0.07936748
22	10 9 1 8 4 2 12 3 5 7 6 11	-0.07936748
23	7 4 10 2 1 9 6 8 12 3 5 11	-0.07936748
24	7 6 11 12 5 2 4 9 3 10 8 1	-0.07936748
25	7 4 5 2 11 1 9 12 10 8 3 6	-0.07936748

## 6.2 Hasil dan Analisa Pengujian Parameter Algoritma Genetika

Pada subbab ini akan dilakukan uji coba untuk mengetahui berapa *fitness* tertinggi dari kromosom terbaik pada setiap percobaan parameter algoritma genetika. Parameter tersebut diantara lain banyak populasi, banyak generasi, *crossover rate* dan *mutation rate*. Hasil pengujian ini kemudian dianalisa untuk diambil suatu kesimpulan dari penelitian ini.

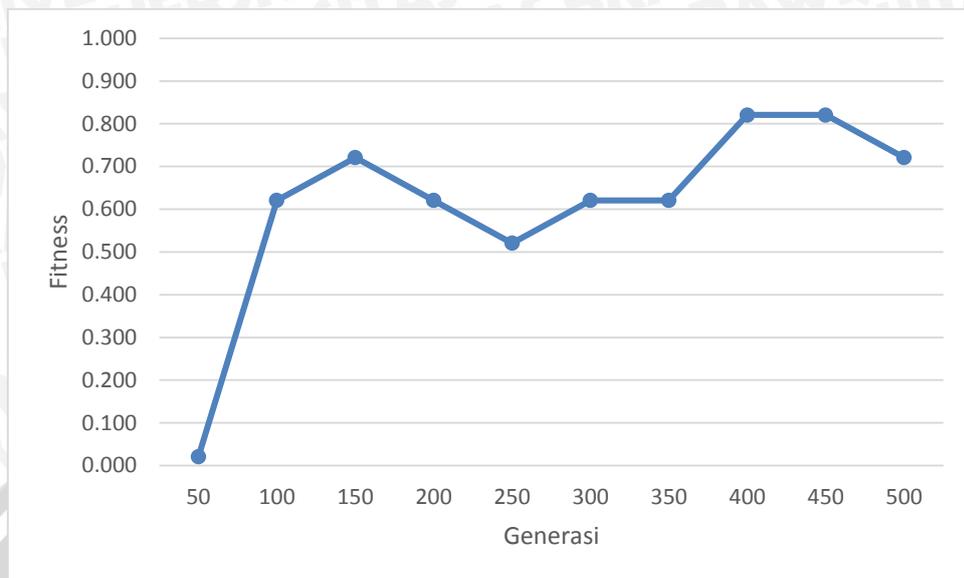
### 6.2.1 Hasil dan Analisa Uji Coba Banyaknya Generasi

Pada proses pengujian ini bertujuan untuk mengetahui jumlah generasi yang optimal untuk menghasilkan rata-rata *fitness* terbaik pada penelitian ini. Data yang digunakan dalam pengujian adalah data pada Tabel 6.1. Jumlah populasi yang digunakan sebanyak 25 individu dengan banyak generasi kelipatan 50 mulai dari 50 sampai dengan 500 generasi. Nilai *crossover rate* dan mutasi *rate* adalah masing-masing 0.5. Setiap generasi dilakukan pengujian sebanyak 10 kali percobaan dan dihitung rata-rata nilai *fitness*-nya. Dari uji coba ini nantinya akan diketahui pada generasi keberapakah diperoleh nilai yang optimal untuk pemecahan masalah pada penelitian ini. Hasil uji coba generasi ditampilkan pada Tabel 6.3.

**Tabel 6.3** Hasil Percobaan Banyak Generasi

Banyak	Nilai fitness percobaan ke-										Rata-Rata
Generasi	1	2	3	4	5	6	7	8	9	10	Fitness
50	-0.079	-0.079	-0.079	-0.079	-0.079	-0.079	-0.079	-0.079	-0.079	0.921	0.021
100	-0.079	0.921	-0.079	-0.079	0.921	0.921	0.921	0.921	0.921	0.921	0.621
150	-0.079	-0.079	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.721
200	0.921	0.921	0.921	-0.079	0.921	0.921	-0.079	0.921	-0.079	0.921	0.621
250	0.921	-0.079	0.921	0.921	-0.079	-0.079	0.921	0.921	0.921	-0.079	0.521
300	-0.079	-0.079	0.921	0.921	0.921	0.921	0.921	-0.079	0.921	0.921	0.621
350	0.921	0.921	-0.079	-0.079	0.921	0.921	0.921	0.921	-0.079	0.921	0.621
400	0.921	0.921	0.921	0.921	0.921	-0.079	0.921	0.921	0.921	0.921	0.821
450	0.921	0.921	0.921	0.921	0.921	0.921	0.921	-0.079	0.921	0.921	0.821
500	0.921	0.921	0.921	0.921	0.921	0.921	-0.079	0.921	-0.079	0.921	0.721

Berdasarkan hasil uji coba pada Tabel 6.3 dapat diketahui bahwa setiap kali uji coba dilakukan dengan generasi yang berbeda, dalam pengujian ini dilakukan sebanyak 50 generasi dan kelipatannya. Dapat diketahui juga pada pengujian tersebut, setiap generasi tertentu menghasilkan nilai rata-rata *fitness* yang berbeda-beda. Dari data tersebut dapat dibuat sebuah grafik untuk melihat perbedaan dari hasil pengujian banyak generasi terhadap nilai *fitness* yang ditunjukkan pada grafik Gambar 6.1.



**Gambar 6.1** Grafik Uji Coba Banyak Generasi

Dari grafik pada Gambar 6.1 dapat dilihat bahwa banyaknya generasi berpengaruh pada nilai *fitness*-nya. Nilai *fitness* terendah terdapat pada 50 generasi dikarenakan dengan jumlah generasi seperti itu masih kurang untuk memproses menggunakan algoritma genetika secara optimal dikarenakan area pencarian algoritma genetika masih sempit [WAY-13]. Dari grafik Gambar 6.1 dapat diketahui bahwa semakin banyak generasi tidak selalu semakin baik nilai *fitness*-nya, tetapi waktu yang dibutuhkan untuk memproses tersebut akan semakin lama. Dapat dilihat pula bahwa generasi yang mempunyai nilai *fitness* terbaik adalah dengan jumlah 400 generasi, karena pada generasi setelah 400 tidak didapatkan nilai *fitness* yang lebih baik. Dengan kondisi seperti ini maka generasi yang paling optimal adalah 400 karena tidak dijumpai lagi solusi yang lebih baik [WAY-12].

### 6.2.2 Hasil dan Analisa Uji Coba Ukuran Populasi

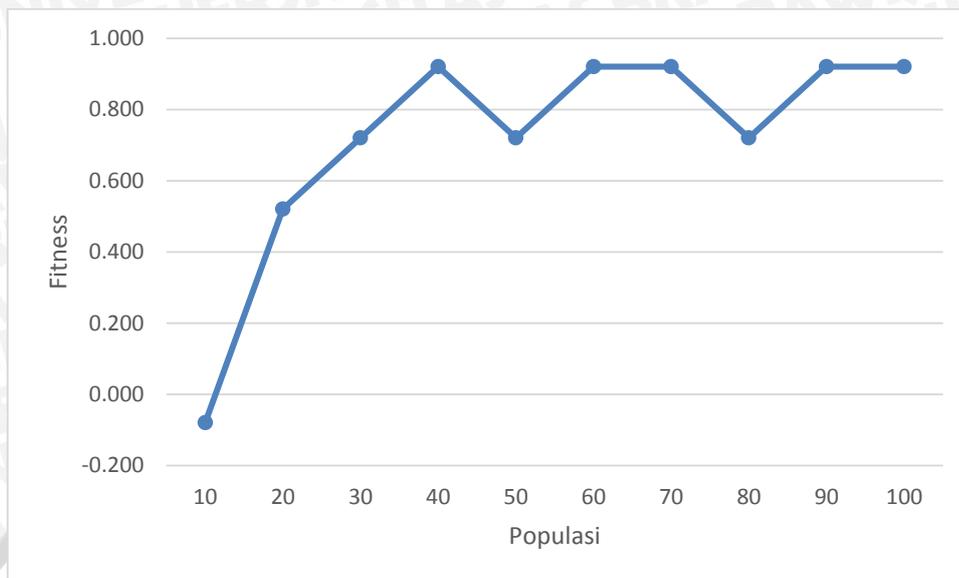
Pada proses pengujian ini bertujuan untuk mengetahui jumlah populasi awal yang optimal untuk menghasilkan rata-rata *fitness* terbaik pada penelitian ini. Data yang digunakan dalam pengujian adalah data pada Tabel 6.1. Jumlah generasi yang digunakan sebanyak 400 generasi dengan banyak populasi kelipatan 10 mulai dari 10 sampai dengan 100 populasi. Nilai *crossover rate* dan mutasi *rate* adalah masing-masing 0.5. Setiap ukuran populasi dilakukan pengujian sebanyak 5 kali

percobaan dan dihitung rata-rata nilai *fitness*-nya. Dari uji coba ini nantinya akan diketahui pada jumlah populasi keberapakah diperoleh nilai yang optimal untuk pemecahan masalah pada penelitian ini. Hasil uji coba ukuran populasi ditampilkan pada Tabel 6.4.

**Tabel 6.4** Hasil Percobaan Ukuran Populasi

Banyak Populasi	Nilai fitness percobaan ke-					Rata-Rata Fitness
	1	2	3	4	5	
10	-0.079	-0.079	-0.079	-0.079	-0.079	-0.079
20	0.921	-0.079	0.921	0.921	-0.079	0.521
30	-0.079	0.921	0.921	0.921	0.921	0.721
40	0.921	0.921	0.921	0.921	0.921	0.921
50	-0.079	0.921	0.921	0.921	0.921	0.721
60	0.921	0.921	0.921	0.921	0.921	0.921
70	0.921	0.921	0.921	0.921	0.921	0.921
80	0.921	0.921	-0.079	0.921	0.921	0.721
90	0.921	0.921	0.921	0.921	0.921	0.921
100	0.921	0.921	0.921	0.921	0.921	0.921

Berdasarkan hasil uji coba pada Tabel 6.4 dapat diketahui bahwa setiap kali uji coba dilakukan dengan populasi yang berbeda, dalam pengujian ini dilakukan sebanyak 10 populasi dan kelipatannya. Dapat diketahui juga pada pengujian tersebut, setiap populasi tertentu menghasilkan nilai rata-rata *fitness* yang berbeda-beda. Dari data tersebut dapat dibuat sebuah grafik untuk melihat perbedaan dari hasil pengujian ukuran populasi terhadap nilai *fitness* yang ditunjukkan pada grafik Gambar 6.2.



**Gambar 6.2** Grafik Uji Coba Ukuran Populasi

Dari grafik pada Gambar 6.2 dapat dilihat bahwa banyaknya populasi berpengaruh pada nilai *fitness*-nya. Nilai *fitness* terendah terdapat pada 10 populasi dikarenakan dengan jumlah generasi seperti itu masih kurang untuk memproses menggunakan algoritma genetika secara optimal dikarenakan area pencarian algoritma genetika masih sempit [WAY-13]. Dari grafik Gambar 6.2 dapat diketahui bahwa semakin banyak populasi tidak selalu semakin baik nilai *fitness*-nya, tetapi waktu yang dibutuhkan untuk memproses tersebut akan semakin lama. Dapat dilihat pula bahwa ukuran populasi yang mempunyai nilai *fitness* terbaik adalah dengan jumlah populasi 40, 60, 70, 90 dan 100. Dari hasil tersebut maka dapat disimpulkan bahwa jumlah populasi 40 yang paling optimal karena untuk populasi 60, 70, 90 dan 100 memerlukan waktu yang lebih lama untuk menjalankan program dan memiliki nilai *fitness* yang sama atau tidak lebih baik. Dengan kondisi ini dapat dikatakan konvergensi tercapai, karena sulit untuk mendapatkan solusi yang lebih baik [WAY-13].

### 6.2.3 Hasil dan Analisa Uji Coba Kombinasi Cr dan Mr

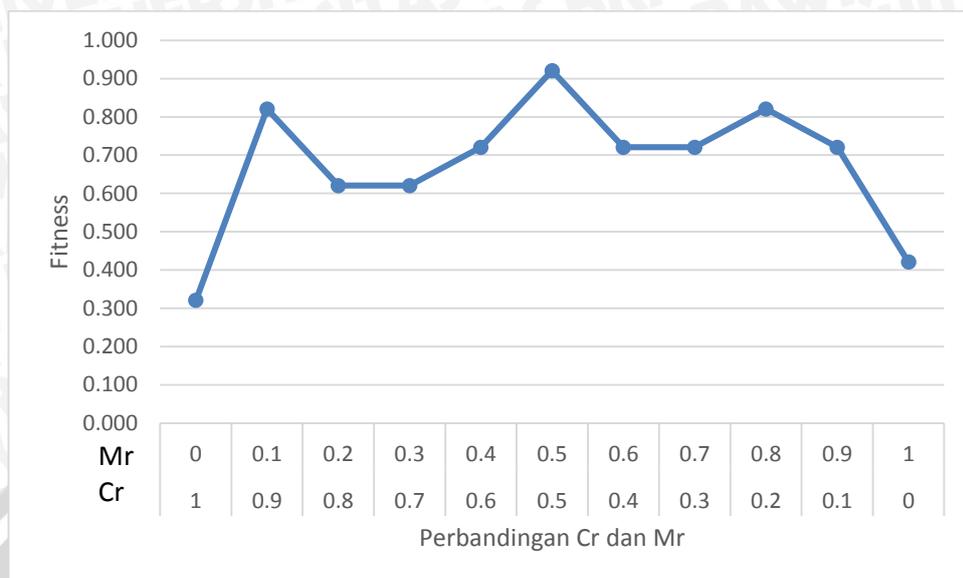
Pada proses pengujian ini bertujuan untuk mengetahui kombinasi *crossover rate* dan *mutation rate* yang optimal untuk menghasilkan rata-rata *fitness* terbaik pada penelitian ini. Data yang digunakan dalam pengujian adalah data pada Tabel

6.1. Jumlah generasi yang digunakan sebanyak 400 generasi didapatkan dari hasil pengujian generasi paling optimal, banyak populasi 40 didapatkan dari pengujian ukuran populasi paling optimal. Nilai *crossover rate* dan mutasi *rate* dalam pengujian ini adalah nilai antara 0 sampai dengan 1 dengan kelipatan 0.1 tiap pengujian. Setiap kombinasi *crossover rate* dan *mutation rate* dilakukan pengujian sebanyak 10 kali percobaan dan dihitung rata-rata nilai *fitness*-nya. Dari uji coba ini nantinya akan diketahui nilai *crossover rate* dan *mutation rate* yang optimal untuk pemecahan masalah pada penelitian ini. Hasil uji coba kombinasi *crossover rate* dan *mutation rate* ditampilkan pada Tabel 6.4.

**Tabel 6.5** Hasil Percobaan Kombinasi *Crossover Rate* dan *Mutation Rate*

Kombinasi		Nilai fitness percobaan ke-										Rata-Rata Fitness
Cr	Mr	1	2	3	4	5	6	7	8	9	10	
1	0	-0.079	-0.079	0.921	-0.079	-0.079	0.921	0.921	-0.079	0.921	-0.079	0.321
0.9	0.1	0.921	0.921	0.921	-0.079	0.921	0.921	0.921	0.921	0.921	0.921	0.821
0.8	0.2	-0.079	0.921	0.921	0.921	0.921	0.921	0.921	0.921	-0.079	-0.079	0.621
0.7	0.3	0.921	0.921	0.921	-0.079	0.921	0.921	-0.079	0.921	0.921	-0.079	0.621
0.6	0.4	0.921	0.921	-0.079	0.921	0.921	-0.079	0.921	0.921	0.921	0.921	0.721
0.5	0.5	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921
0.4	0.6	0.921	0.921	0.921	-0.079	0.921	0.921	0.921	0.921	-0.079	0.921	0.721
0.3	0.7	0.921	-0.079	0.921	0.921	0.921	0.921	0.921	0.921	0.921	-0.079	0.721
0.2	0.8	0.921	-0.079	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.821
0.1	0.9	0.921	-0.079	0.921	0.921	0.921	0.921	0.921	0.921	-0.079	0.921	0.721
0	1	-0.079	0.921	0.921	0.921	-0.079	-0.079	0.921	-0.079	0.921	-0.079	0.421

Berdasarkan hasil uji coba pada Tabel 6.5 dapat diketahui bahwa setiap kali uji coba dilakukan dengan populasi yang berbeda. Dapat diketahui juga pada pengujian tersebut, setiap kombinasi *crossover rate* dan *mutation rate* tertentu menghasilkan nilai rata-rata *fitness* yang berbeda-beda. Dari data tersebut dapat dibuat sebuah grafik untuk melihat perbedaan dari hasil pengujian kombinasi *crossover rate* dan *mutation rate* terhadap nilai *fitness* yang ditunjukkan pada grafik Gambar 6.3.

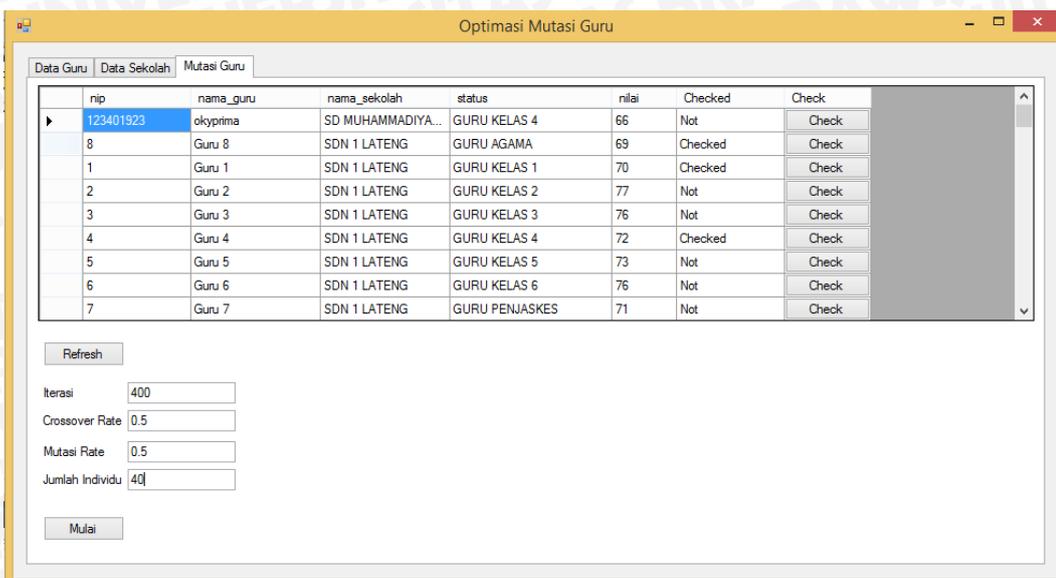


**Gambar 6.3** Grafik Uji Coba Kombinasi *Crossover Rate* dan *Mutation Rate*

Dari grafik pada Gambar 6.3 dapat dilihat bahwa kombinasi *crossover rate* dan *mutation rate* berpengaruh pada nilai *fitness*-nya. Nilai *fitness* terendah terdapat pada kombinasi Mr 0 dan Cr 1 dengan nilai *fitness* sebesar 0.321 dan nilai *fitness* terbesar terdapat pada kombinasi Mr 0.5 dan Cr 0.5 dengan nilai *fitness* sebesar 0.921. Dengan menggunakan nilai Cr rendah, pengujian ini akan sangat bergantung pada Mr dimana cenderung tidak dapat belajar dari generasi sebelumnya. Di sisi lain, jika menggunakan nilai Cr yang tinggi maka akan cenderung kehilangan kemampuannya untuk mempertahankan keberagaman populasi [WAY-14]. Jadi dapat disimpulkan kombinasi *crossover rate* dan *mutation rate* paling optimal secara berturut-turut adalah 0.5 dan 0.5.

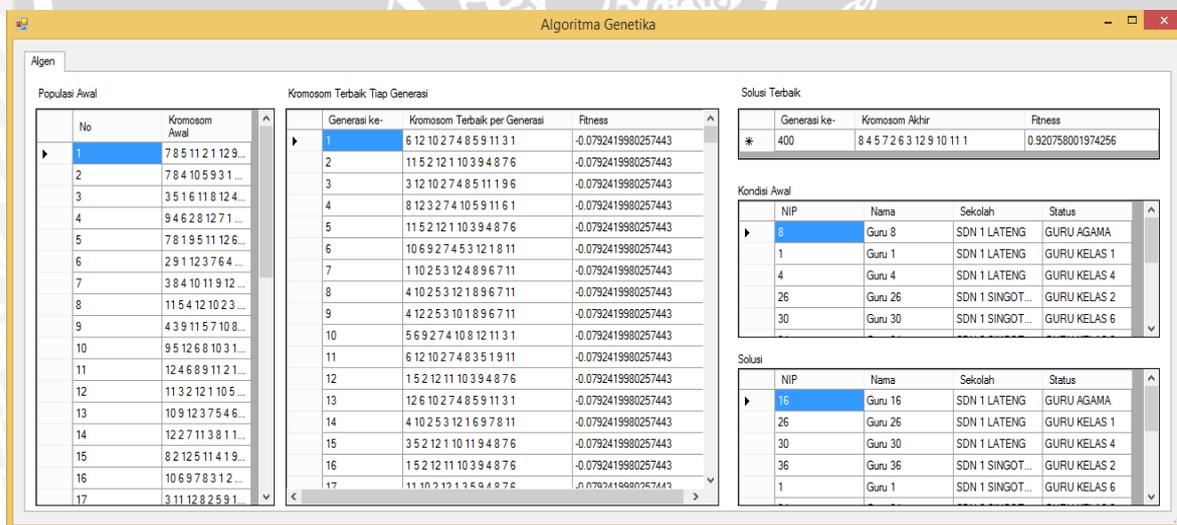
#### 6.2.4 Hasil dan Analisa Uji Coba Menggunakan Parameter Terbaik

Pada proses pengujian bertujuan untuk mengetahui solusi yang dihasilkan oleh sistem dengan menggunakan parameter yang paling optimal. Parameter optimal didapatkan dari hasil pengujian sebelumnya. Parameter yang digunakan yaitu jumlah generasi 400, ukuran populasi 40, *crossover rate* dan *mutation rate* berturut-turut 0.5 dan 0.5. Data yang digunakan dalam pengujian ini adalah data pada Tabel 6.1. *Input* parameter yang paling optimal pada pengujian ini akan ditunjukkan pada Gambar 6.4.



Gambar 6.4 Input Parameter Terbaik

Dari *input* parameter tersebut kemudian dilakukan proses perhitungan menggunakan algoritma genetika dalam sistem. Hasil keluaran dari sistem menggunakan parameter terbaik dapat dilihat pada Gambar 6.5.



Gambar 6.5 Output Sistem Menggunakan Parameter Terbaik

Dari Gambar 6.5 dapat dilihat bahwa didapatkan solusi terbaik pada kromosom [8 4 5 7 2 6 3 12 9 10 11 1] dengan nilai *fitness* sebesar 0.920758001974256. Hasil keluaran ini tidak menghasilkan penalti sehingga guru dapat dimutasikan ke posisi yang semestinya.



## BAB VII

### PENUTUP

#### 7.1. Kesimpulan

Kesimpulan yang didapat dari hasil uji coba yang telah dilakukan adalah sebagai berikut :

1. Faktor yang mempengaruhi seorang guru SD akan dimutasi atau tidak adalah kebutuhan. Pada penelitian ini yang dimaksud kebutuhan adalah atas dasar pemerataan jumlah kualitas kinerja guru SD di tiap sekolah. Kinerja guru didapatkan dari hasil penilaian SKP yang terdiri dari 11 aspek dan penilaian perilaku kerja yang terdiri dari 7 aspek.
2. Representasi kromosom dengan cara permutasi dengan proses reproduksi *one cut point crossover* dan *reciprocal exchange mutation* yang digunakan dalam penelitian ini mampu menyelesaikan permasalahan optimasi dalam pemerataan guru SD. Algoritma Genetika mampu menentukan penempatan guru yang akan dimutasikan sehingga menghasilkan kualitas guru optimal di tiap sekolah.
3. Dari hasil pengujian, pada uji coba banyaknya generasi didapatkan generasi yang paling optimal adalah 400 dengan rata-rata nilai *fitness* 0.821. Untuk ukuran populasi yang paling optimal adalah 40 dengan rata-rata nilai *fitness* 0.921. Sedangkan untuk kombinasi *crossover rate* dan *mutation rate* yang paling optimal adalah 0.5 dan 0.5 dengan rata-rata nilai *fitness* adalah 0.921.

#### 7.2. Saran

Saran yang dapat diberikan setelah menyelesaikan penelitian ini adalah dengan menambahkan parameter lagi sebagai penentuan seorang guru akan dimutasi, tidak hanya terpaku pada nilai kualitas guru tersebut. Untuk pengembangan selanjutnya, perlu dilakukan penggabungan (hybrid) algoritma genetika dengan algoritma yang lainnya. Teknik ini terbukti dapat meningkatkan akurasi dan efisiensi untuk pencarian solusi optimum [WAY-13]. Serta perlu

dilakukan untuk menggunakan metode lain pada reproduksi agar menghasilkan solusi yang lebih bervariasi.



**DAFTAR PUSTAKA**

- [SRI-03] Sri Kusumadewi. 2003. Artificial Intelligence (Teknik dan Aplikasinya). Graha Ilmu. Yogyakarta
- [SUY-11] Suyanto. 2011. Artificial Intelligence (Searching, Reasoning, Planning, Learning). Informatika. Bandung
- [FAC-10] Fachrudin Affandi. Penerapan Algoritma Genetika Untuk Masalah Penjadwalan *Job Shop* pada Lingkungan Industri Pakaian. Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS). Surabaya.
- [SIT-10] Siti Alvi Sholikatin. 2010. Laporan Kerja Praktek Rancang Bangun Aplikasi Inventaris Obat di Klinik Vira Medika Ledug. Sistem Informasi, STMIK AMIKOM. Purwokerto.
- [WAY-13] Wayan Firdaus Mahmudy. 2013. Modul Matakuliah Algoritma Evolusi. Program Teknologi Informasi dan Ilmu Komputer (PTIIK). Universitas Brawijaya. Malang
- [IGD-09] I gede Garry A.S, Dien Azizun, Yudo Nugroho. 2009. Database. Sekolah Tinggi Ilmu Ekonomi Indonesia. Surabaya
- [MAS-13] Masgilang Prayudi. 2013. Hubungan Persepsi Siswa Tentang Profesionalisme Guru IPS dengan Prestasi Belajar Pada Siswa Kelas VII Semester Ganjil Mata Pelajaran IPS. Prodi Pendidikan Geografi. Universitas Lampung. Lampung.
- [JEN-10] Jeniusman Ahmad Hutagalung. 2010. Peran Mutasi Dalam Meningkatkan Prestasi Kerja di Perusahaan Daerah Pasar. Prodi Magister Studi Pembangunan, Universitas Sumatera Utara. Medan.
- [HER-13] Herman. 2013. Guru, Kunci Utama Masalah Pendidikan di Indonesia. <http://www.beritasatu.com/pendidikan/155912-guru-kunci-utama-masalah-pendidikan-di-indonesia.html> [23 Mei 2014]
- [DHA-07] Dhany Rahadian Basuki. 2007. Optimasi Penempatan Pegawai Dengan Algoritma Genetika. Sistem Informasi. STIKOM. Surabaya.
- [BKN-13] Badan Kepegawaian Negara. 2013. Ketentuan Pelaksanaan Peraturan Pemerintah Nomor 46 Tahun 2011 Tentang Penilaian Prestasi Kerja Pegawai Negeri Sipil. Jakarta.

- [MON-14] Monica Intan Pratiwi. 2014. Implementasi Algoritma Genetika Pada Optimasi Biaya Pemenuhan Kebutuhan Gizi. Program. Prodi Teknik Informatika. Universitas Brawijaya. Malang.
- [WAY-12] Wayan F. Mahmudy, Romeo M. Marian, Lee H. S. Luong. 2012. 'Solving part type selection and loading problem in flexible manufacturing system using real coded genetic algorithms – Part II: optimization', *International Conference on Control, Automation and Robotics*, Singapore, 12-14 September, World Academy of Science, Engineering and Technology, pp. 706-710.
- [WAY-14] Wayan F. Mahmudy, Romeo M. Marian, and Lee H. S. Luong. 2014, 'Hybrid genetic algorithms for part type selection and machine loading problems with alternative production plans in flexible manufacturing system', *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 8, no. 1, pp. 80-93.

