

TUGAS AKHIR

PEMBANGUNAN PEMBANGKIT ALGORITMA GENETIK UNTUK PERMASALAHAN PERMUTASI

**Diajukan sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Strata-1
Teknik Industri**



Oleh

Nama : Wahyu Kresna Ristiano

No. Mahasiswa : 07 522 230

**JURUSAN TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2012

PERNYATAAN

Demi Allah, saya akui karya ini adalah hasil kerja saya sendiri kecuali nukilan dan ringkasan yang setiap satunya telah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah dalam karya tulis dan hak intelektual maka saya bersedia ijazah yang telah saya terima untuk ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 15 Januari 2012



resna Ristianto

NIM 07522230



LEMBAR PENGESAHAN PEMBIMBING

PEMBANGUNAN PEMBANGKIT ALGORITMA GENETIK

UNTUK PERMASALAHAN PERMUTASI

TUGAS AKHIR



Oleh

Nama : Wahyu Kresna Ristiano

No. Mahasiswa : 07 522 230

Yogyakarta, 15 Januari 2012

Pembimbing,

A handwritten signature in black ink, which appears to be 'Ridwan', is written over the word 'Pembimbing,'.

(Muhammad Ridwan Andi Purnomo, ST., M.Sc., Ph.D)

LEMBAR PENGESAHAN PENGUJI

**PEMBANGUNAN PEMBANGKIT ALGORITMA GENETIK
UNTUK PERMASALAHAN PERMUTASI**

TUGAS AKHIR

Oleh :

Nama : Wahyu Kresna Ristiano
No. Mahasiswa : 07522230

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Teknik Industri
Fakultas Teknologi Industri Universitas Islam Indonesia
Yogyakarta, Januari 2011

Tim Penguji

Muhammad Ridwan Andi Purnomo, ST., M.Sc., Ph.D
Ketua

Agus Mansur, ST., M.Eng.Sc.
Anggota I

Sri Indrawati, ST., M.Eng.
Anggota II

Mengetahui,
Ketua Jurusan Teknik Industri
Universitas Islam Indonesia

Drs. H. M. Ibnu Mastur, MSIE

10/2/2012

HALAMAN PERSEMBAHAN

To my beloved family and Ummina Lintang.



MOTTO

There is no wealth like knowledge, no poverty like ignorance.

Tidak ada kekayaan yang sebanding dengan ilmu,
tidak ada kemiskinan yang sebanding dengan kebodohan.

~Ali bin Abi Thalib.



KATA PENGANTAR

Assalamu'alaykum Warahmatullah Wabarakatuh

Alhamdulillah, puji dan syukur penulis panjatkan ke hadirat Allah 'Azza wa Jalla atas limpahan rahmat, taufik serta hidayahnya. Sholawat dan salam tercurah kepada Nabi Muhammad SAW beserta keluarga dan para sahabat, serta orang-orang yang bertaqwa, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "*Pembangunan Pembangkit Algoritma Genetik untuk Permasalahan Permutasi*" dapat diselesaikan dengan baik.

Tugas akhir ini disusun sebagai salah satu syarat yang harus dipenuhi oleh setiap mahasiswa Jurusan Teknik Industri untuk menyelesaikan studi Strata-1 pada Fakultas Teknologi Industri, Universitas Islam Indonesia. Dalam penyusunan laporan Tugas Akhir ini penulis mendapatkan banyak bantuan dan dukungan dari berbagai pihak, untuk itu penulis menyampaikan ucapan terimakasih dan penghargaan yang setinggi-tingginya kepada pihak-pihak yang telah memberikan dukungannya baik secara langsung maupun tidak langsung, dengan penuh rasa syukur penulis ucapkan terima kasih kepada:

1. Dekan Fakultas Teknologi Industri.
2. Ketua Jurusan Teknik Industri.
3. Bapak Muhammad Ridwan Andi Purnomo, ST., M.Sc., Ph.D selaku dosen pembimbing yang telah memberikan bimbingan, petunjuk, saran serta waktunya dalam pembuatan Tugas Akhir ini.
4. Kedua orang tua serta adik-adik tercinta yang telah memberikan doa dan dukungan baik secara material maupun immaterial.

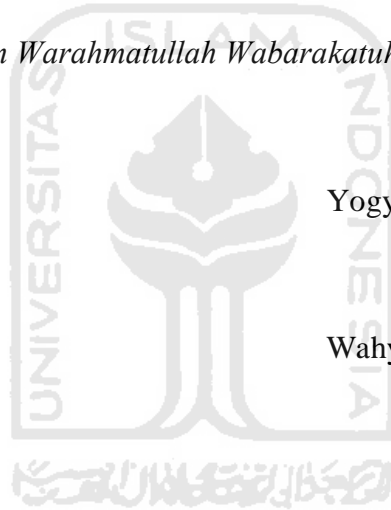
5. Teman-teman serta semua pihak yang tidak bisa disebutkan satu persatu, yang telah membantu dalam penyelesaian tugas akhir ini.

Semoga Allah *'Azza wa Jalla* melimpahkan rahmat dan hidayahnya kepada semua pihak yang telah membantu terselesaikannya penulisan laporan Tugas Akhir ini. Penulis menyadari bahwa dalam penyusunan laporan Tugas Akhir ini masih banyak terdapat kekeliruan dan kekurangan. Untuk itu penulis menyampaikan permohonan maaf sebelumnya serta sangat mengharapkan kritik dan saran yang membangun guna penyempurnaan di masa mendatang. Semoga laporan ini dapat bermanfaat bagi penulis dan semua pembaca.

Wassalamu'alaykum Warahmatullah Wabarakatuh

Yogyakarta, 15 Januari 2012

Wahyu Kresna Ristiano



ABSTRAK

Algoritma Genetik (AG) merupakan pencarian heuristik yang didasarkan pada mekanisme seleksi alami dan evolusi biologis. AG dapat digunakan untuk mencari solusi permasalahan dalam dunia nyata seperti permasalahan pemodelan dan optimasi. Namun, untuk dapat menggunakannya dalam permasalahan optimasi, diperlukan proses pemrograman komputer yang cukup rumit dan kebanyakan pembangkit AG yang ada bersifat komersial serta dibangun menggunakan platform luar sehingga membuat proses komputasinya menjadi sangat lama. Pada penelitian ini, akan dibangun pembangkit AG sumber terbuka yang dapat digunakan untuk menyelesaikan permasalahan optimasi khususnya pada permasalahan permutasi. Untuk mengimplementasikan pembangkit yang dibangun, dilakukan pengujian terhadap permasalahan travelling salesman problem (TSP) menggunakan operator crossover yang berbeda. Dari hasil perhitungan dan analisis diketahui bahwa pembangkit yang dibangun dapat menyelesaikan kasus TSP dengan baik.

Kata kunci: Algoritma Genetik, Travelling Salesman Problem (TSP), Combinatorial.



DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN	Error! Bookmark not defined.
LEMBAR PENGESAHAN PEMBIMBING	ii
LEMBAR PENGESAHAN PENGUJI	Error! Bookmark not defined.
HALAMAN PERSEMBAHAN	v
MOTTO	vi
KATA PENGANTAR	vii
ABSTRAK	ix
DAFTAR ISI	x
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan.....	3
BAB II KAJIAN PUSTAKA	
2.1 Kajian Deduktif.....	5
2.1.1 Pengertian AG	5
2.1.2 Terminologi AG	6
2.1.3 Proses AG	7
2.1.4 Kelebihan AG.....	13
2.1.5 Kekurangan Algoritma Genetik	14
2.1.6 <i>Visual Basic for Application (VBA)</i>	14
2.2 Kajian Induktif.....	15

BAB III METODOLOGI PENELITIAN

3.1	Lokasi dan Objek Penelitian.....	18
3.2	Data yang Dibutuhkan.....	18
3.3	Alat dan Bahan	19
3.4	Pembangunan Pembangkit Algoritma Genetik	19
3.5	Diagram Alir Penelitian	20

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

4.1	Pembangunan Pembangkit Algoritma Genetik	21
4.1.1	Inisialisasi Populasi	21
4.1.2	Seleksi.....	22
4.1.3	<i>Crossover</i>	22
4.1.4	Mutasi	24
4.2	Pengolahan Data	25
4.2.1	Pengujian Pembangkit	25

BAB V PEMBAHASAN

5.1	Analisis Pembangunan Pembangkit AG	28
5.2	Analisis Kasus	29

BAB VI KESIMPULAN DAN SARAN

6.1	Kesimpulan.....	31
6.2	Saran.....	31

DAFTAR PUSTAKA

LAMPIRAN

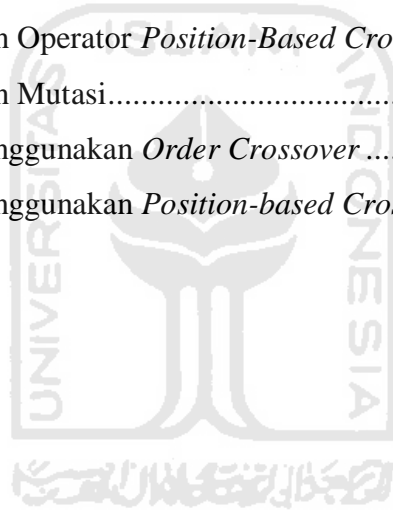
DAFTAR TABEL

Tabel 4.1 Koordinat Lokasi Kota.....	25
Tabel 4.2 Hasil Pengujian Menggunakan Operator <i>Order Crossover</i>	26
Tabel 4.3 Hasil Pengujian Menggunakan Operator <i>Position-Based Crossover</i>	27



DAFTAR GAMBAR

Gambar 2.1 Operator PMX.....	10
Gambar 2.2 Operator PMX (lanjutan)	11
Gambar 2.3 Operator OX.....	12
Gambar 2.4 Operator <i>Position-Based Crossover</i>	13
Gambar 3.1 Diagram Alir Penelitian	20
Gambar 4.1 Kode Bayangan Inisialisasi Populasi.....	21
Gambar 4.2 Kode Bayangan Seleksi	22
Gambar 4.3 Kode Bayangan Operator <i>Order Crossover</i>	23
Gambar 4.4 Kode Bayangan Operator <i>Position-Based Crossover</i>	23
Gambar 4.5 Kode Bayangan Operator <i>Position-Based Crossover</i> (lanjutan)	24
Gambar 4.6 Kode Bayangan Mutasi.....	24
Gambar 4.7 Grafik AG Menggunakan <i>Order Crossover</i>	26
Gambar 4.8 Grafik AG Menggunakan <i>Position-based Crossover</i>	27



BAB I

PENDAHULUAN

1.1 Latar Belakang

Optimasi merupakan proses membuat sesuatu menjadi lebih baik (Guo, et al., 2010). Dalam beberapa dekade terakhir, banyak pendekatan yang digunakan dalam pencarian nilai optimal seperti *tabu search*, *simulated annealing*, algoritma koloni semut, dan algoritma-algoritma evolusioner seperti *evolution strategies* serta Algoritma Genetik (AG) telah dikembangkan untuk memecahkan permasalahan optimasi.

AG merupakan pencarian heuristik dan teknik optimasi yang pada awalnya termotivasi oleh prinsip teori evolusi Darwin terhadap seleksi genetik (McCall, 2005). AG merupakan pendekatan yang fleksibel yang dapat diaplikasikan pada cakupan yang sangat luas dalam pembelajaran dan permasalahan optimasi. Salah satu permasalahan dalam penggunaan AG untuk optimasi adalah diperlukannya proses pemrograman komputer yang cukup rumit. Walaupun telah ada beberapa pembangkit AG (*AG generator*), namun kebanyakan pembangkit tersebut bersifat komersial, sebagai contohnya yaitu Generator (www.nli-ltd.com), perangkat lunak ini merupakan perangkat lunak yang komersil dan hanya menyediakan *demo version* bagi user yang ingin menggunakan perangkat lunak tersebut. Semua pembangkit AG yang ada menggunakan *Microsoft Excel* sebagai antarmuka untuk membangun model optimasinya. Namun, permasalahan yang masih dihadapi adalah semua pembangkit AG tersebut dibangun menggunakan platform luar sehingga proses komputasinya

menjadi sangat lama. Selain itu, dari sekian aplikasi dan sistem yang ada, belum terdapat aplikasi sumber terbuka (*open source*) untuk AG yang mampu mempermudah *user* dalam mendapatkan informasi yang diinginkannya.

Sejak satu dekade terakhir ini, komunitas *open source* mulai mengembangkan pembangkit AG, untuk mempermudah *user* dalam melakukan proses AG guna menyelesaikan permasalahan yang dihadapinya, namun pembangkit yang dibangun bersifat kasuistik dan tidak fleksibel seperti perangkat lunak yang seharusnya dibangun untuk digunakan untuk menyelesaikan permasalahan AG dengan kromosom representasi permutasi dan *string bit*. Berdasarkan permasalahan diatas, maka dalam penelitian ini akan dilakukan pembangunan pembangkit algoritma genetik sumber terbuka yang fleksibel digunakan untuk menyelesaikan permasalahan permutasi.

1.2 Rumusan Masalah

Dari latar belakang di atas, dapat dirumuskan pokok permasalahan yang akan diselesaikan dalam penelitian ini yaitu bagaimana pembangunan pembangkit algoritma genetik sumber terbuka untuk permasalahan permutasi?

1.3 Batasan Masalah

Agar penelitian ini lebih terarah, mudah dipahami dan topik yang dibahas tidak meluas, maka perlu dilakukan pembatasan lingkup penelitian. Adapun pembatasan lingkup penelitian ini adalah:

- a. Pembangkit yang akan dibangun difokuskan pada AG.

- b. Pengembangan pembangkit yang dibangun tidak termasuk dalam pembuatan *Graphical User Interface* (GUI)
- c. Pembangkit yang dibangun tidak termasuk kasus kromosom dengan representasi bilangan real dan string bit.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk membangun pembangkit algoritma genetik sumber terbuka untuk permasalahan permutasi.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

- a. Memberikan kemudahan bagi pengguna untuk menyelesaikan permasalahan optimasi yang dapat diselesaikan dengan algoritma genetik khususnya pada permasalahan permutasi.
- b. Dapat digunakan untuk mengembangkan pembangkit algoritma genetik sumber terbuka yang lebih murah daripada perangkat lunak komersil.
- c. Mampu memberikan tambahan wawasan dan dapat digunakan sebagai bahan rujukan penelitian berikutnya.

1.6 Sistematika Penulisan

Laporan tugas akhir ini disusun secara sistematis ke dalam beberapa bab, dan masing-masing bab akan diuraikan sebagai berikut:

BAB II KAJIAN LITERATUR

Bab ini berisi 2 sub bab yaitu kajian deduktif dan kajian induktif mengenai penjelasan teori penunjang yang digunakan sebagai landasan dalam menyelesaikan permasalahan yang ada dan menjawab rumusan masalah, terutama mengenai algoritma genetik, beserta prosedur yang ada di dalamnya seperti seleksi, *crossover* dan mutasi, serta teori mengenai *Visual Basic for Application* (VBA).

BAB III METODE PENELITIAN

Bab ini berisi tentang uraian metode pengumpulan data, diagram alir penelitian, dan cara pengolahan serta analisis data.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Bab ini berisi pengumpulan data-data yang akan diolah sesuai dengan penelitian yang dilakukan, serta pengolahan data untuk menyelesaikan kasus yang diangkat.

BAB V PEMBAHASAN

Bab ini berisikan pembahasan terhadap data-data yang telah di olah menggunakan landasan berupa teori-teori penunjang penelitian yang di sajikan pada BAB II.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisikan kesimpulan dari pembahasan yang telah dilakukan dengan menjawab rumusan masalah yang telah diajukan, serta saran yang diberikan untuk penelitian selanjutnya.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

KAJIAN PUSTAKA

2.1 Kajian Deduktif

Bab ini berisi tentang penjelasan teori penunjang yang digunakan sebagai landasan dalam menyelesaikan permasalahan yang ada dan menjawab rumusan masalah yang diajukan.

2.1.1 Pengertian AG

John Holland dari University of Michigan memulai karyanya pada AG pada awal tahun 1960. AG didefinisikan sebagai teknik pencarian stokastik yang didasarkan atas mekanisme seleksi alami dan evolusi biologis (Goldberg, 1989). Tidak ada jaminan bahwa AG dapat memberikan solusi yang optimal (Bunnag dan Sun, 2005), namun AG dapat memberikan solusi yang baik. AG dapat digunakan untuk mencari solusi permasalahan-permasalahan dalam dunia nyata seperti permasalahan pemodelan dan optimasi (McCall, 2005), keseimbangan lini perakitan (Rubinovitz dan Letivin, 1995), tata letak fasilitas pada industri pakaian (Martens, 2004), penjadwalan mata pelajaran (Wang, 2003), dan penjadwalan ukuran lot (Khouja, et al., 1998). Selain itu, Gen dan Cheng (1997) menyebutkan bahwa AG juga dapat menyelesaikan permasalahan seperti

optimasi permutasi, optimasi reliabilitas, *flow shop*, penjadwalan *job shop*, penjadwalan mesin dan transportasi.

2.1.2 Terminologi AG

Beberapa terminologi yang digunakan dalam AG adalah (Gen dan Cheng, 1997):

- a. *Chromosome* (kromosom) merupakan tempat penyimpanan informasi genetika.
- b. *Genotype* (struktur) merupakan kombinasi satu atau beberapa kromosom membentuk fungsi kerja suatu organisme. Interaksi sekumpulan kromosom disebut *genotype*. Di dalam AG struktur dianalogikan sebagai *genotype*.
- c. *Phenotype* (set parameter) merupakan interaksi di dalam struktur terjadi karena adanya proses transformasi kode-kode genetika. Modifikasi ini disebut *phenotype*. *Phenotype* tersebut merupakan representasi set parameter masalah yang sedang dihadapi. Representasi kode dapat berupa numerik atau non numerik.
- d. *Alleles* (*feature value*) merupakan suatu *feature* memiliki nilai *feature* tertentu yang disebut dengan *allele*.
- e. *Locus* (*positioning*) adalah letak gen dalam suatu kromosom disebut *locus*. Setiap *feature* memiliki urutan posisi di dalam *string*.
- f. *Genes* ialah suatu kromosom dibentuk oleh beberapa *gen*.

2.1.3 Proses AG

Dalam menyelesaikan sebuah permasalahan, algoritma genetik memelihara sebuah populasi dari individu (disebut juga dengan *strings* atau kromosom) dan secara probabilistik melakukan modifikasi populasi menggunakan operator genetik seperti seleksi, *crossover* dan mutasi, dengan maksud untuk mendapatkan solusi optimal sebuah permasalahan (Guo, et al., 2010).

a. Inisialisasi Populasi

Proses pencarian solusi yang optimal dengan algoritma genetik tidak dimulai dari suatu nilai awal melainkan dari sekumpulan nilai awal yang disebut populasi (Gen dan Cheng, 1997). Untuk menghindari terjadinya konvergensi dini, salah satu cara yang bisa digunakan ialah dengan menggunakan populasi yang besar (Hwang dan He, 2006). Algoritma genetik memproses sebuah populasi dari struktur data yang serupa dengan kromosom, setiap kromosom akan merepresentasikan sebuah individu, setiap individu merupakan solusi dari sebuah permasalahan yang biasa direpresentasikan dengan format *string* (Kamrani, et al., 2001). Selain itu, kromosom dapat direpresentasikan dengan menggunakan (Kusumadewi dan Purnomo, 2005):

- a. *String bit* : 10011,01101,11101, dst.
- b. Bilangan real : 65.65, -67.98, 562.88, dst.
- c. Elemen permutasi : E2, E10, E5, dst.
- d. Daftar aturan : R1, R2, R3, dst.
- e. Elemen program : pemrograman genetika
- f. Struktur lainnya.

b. Proses Evaluasi

Proses evaluasi *fitness* kromosom terdiri atas tiga langkah seperti berikut (Gen dan Cheng, 1997):

Langkah 1 Konversi kromosom *genotype* ke *phenotype*. Di sini berarti konversi barisan biner menjadi harga nyata relatif $x^k = (x_1^k, x_2^k)$, $k = 1, 2 \dots$ populasi.

Langkah 2 Evaluasi fungsi obyektif $f(x^k)$

Langkah 3 konversi harga fungsi obyektif menjadi ketahanan, untuk maksimal soal. Ketahanan = harga fungsi obyektif $eval(v_k) = f(x^k)$, $k = 1, 2 \dots$ populasi

c. Proses Seleksi

Seleksi merupakan proses pemilihan calon induk, dalam proses seleksi ini terdapat beberapa metode. Dalam kebanyakan praktek, pendekatan roda *roulette* digunakan sebagai prosedur seleksi. Susunan prosedur roda *roulette* dapat dijelaskan seperti (Gen dan Cheng, 1997):

a. Hitung nilai *fitness eval* (v_k) untuk masing-masing kromosom v_k

$$eval(v_k) = f(x), \quad k = 1, 2 \dots \text{populasi}$$

b. Hitung total *fitness* untuk populasi

$$F = eval(v_k) = f(x), \quad k = 1, 2 \dots \text{populasi}$$

c. Hitung probabilitas seleksi p^k untuk masing-masing kromosom v_k

$$P^k = eval(v_k)/F \quad k = 1, 2 \dots \text{populasi}$$

d. Hitung probabilitas kumulatif q_k untuk masing-masing kromosom v_k

$$q^k = p_j \quad k = 1, 2 \dots \text{populasi}$$

Proses seleksi dimulai dengan memutar *roulette* sejumlah populasi, satu kromosom akan terseleksi untuk sebuah populasi baru dengan langkah-langkah seperti:

Langkah 1 Bangkitkan bilangan random r pada kisaran $[0,1]$

Langkah 2 Jika $r \leq q_1$, kemudian pilih kromosom pertama v_1 ; sebaliknya, pilih kromosom ke- k , v_k ($2 \leq k \leq k$ populasi) sedemikian sehingga $q_{k-1} < r \leq q_k$.

Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang fit. Ada beberapa metode seleksi dari induk, antara lain (Gen dan Cheng, 1997):

1. *Rank-based fitness assignment.*
2. *Roulette wheel selection.*
3. *Stochastic universal sampling.*
4. *Local selection.*
5. *Truncation selection.*
6. *Tournament selection.*

d. Proses Rekombinasi (*Crossover*)

Operator rekombinasi melibatkan pertukaran material genetik di antara dua induk kromosom untuk menghasilkan kromosom anak (*offspring*). Penentuan jumlah kromosom induk yang diharapkan untuk melakukan persilangan dilakukan dengan cara menentukan probabilitas persilangan dari *popsize* kromosom induk (pc). Operator *crossover* representasi elemen permutasi yang biasa digunakan, yaitu

partial-mapped crossover, *order crossover*, dan *position-based crossover* (Gen dan Cheng, 1997).

1. *Partial-Mapped Crossover (PMX)*

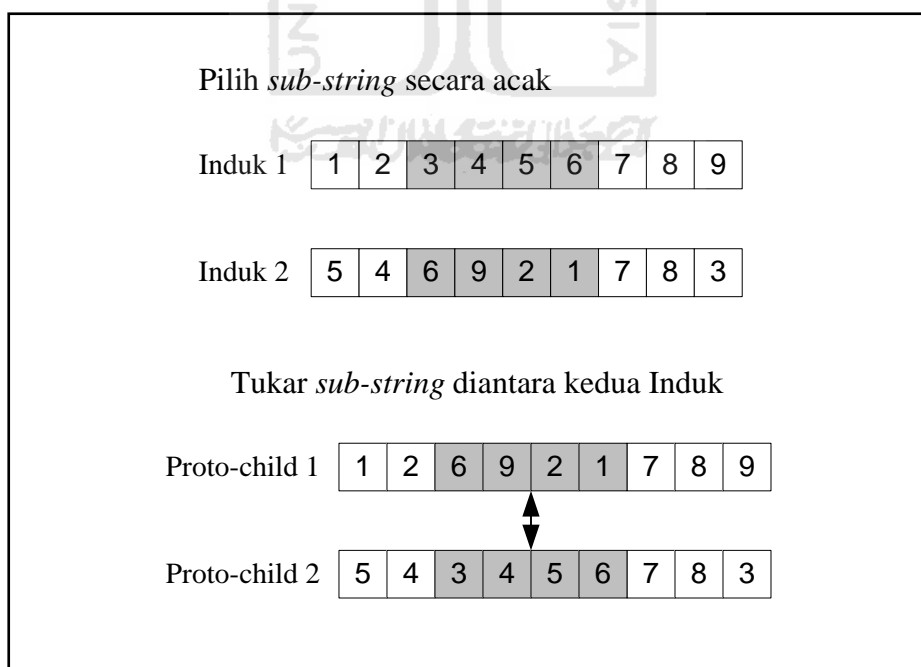
PMX merupakan penyilangan dua titik yang sederhana yang ditambahkan dengan beberapa prosedur perbaikan. Adapun prosedur langkah PMX dijelaskan pada langkah 1 sampai dengan langkah 4, serta pada Gambar 2.2 dan Gambar 2.3.

Langkah 1 Pilih 2 posisi pada kromosom secara acak. *Sub-string* yang berada di antara 2 posisi secara acak tadi dinamakan *mapping section*.

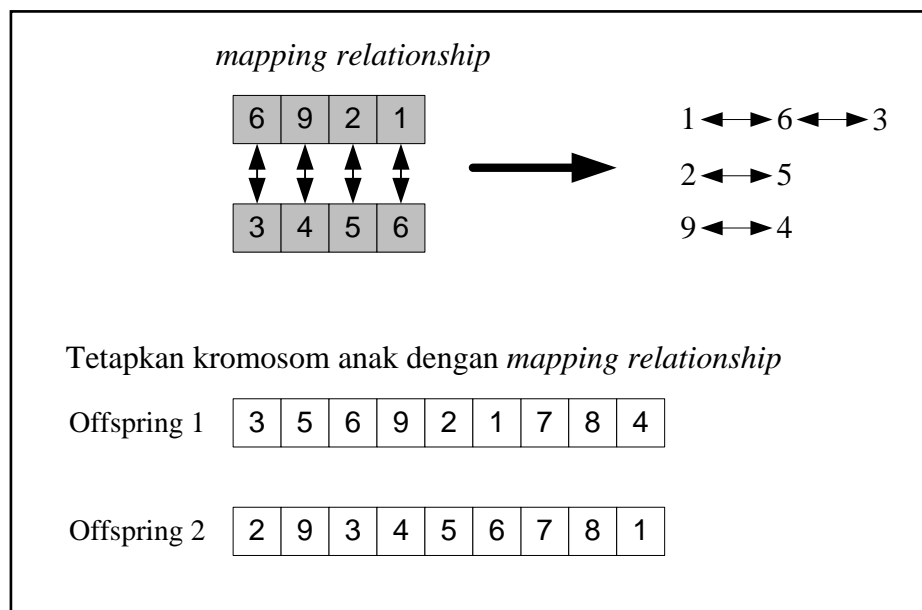
Langkah 2 Tukar dua *sub-string* di antara induk untuk menghasilkan *proto-child*.

Langkah 3 Tentukan *mapping relationship* di antara 2 *mapping section*.

Langkah 4 Tetapkan kromosom anak berdasarkan pada *mapping relationship*.



Gambar 2.1 Operator PMX



Gambar 2.2 Operator PMX (lanjutan)

2. Order Crossover (OX)

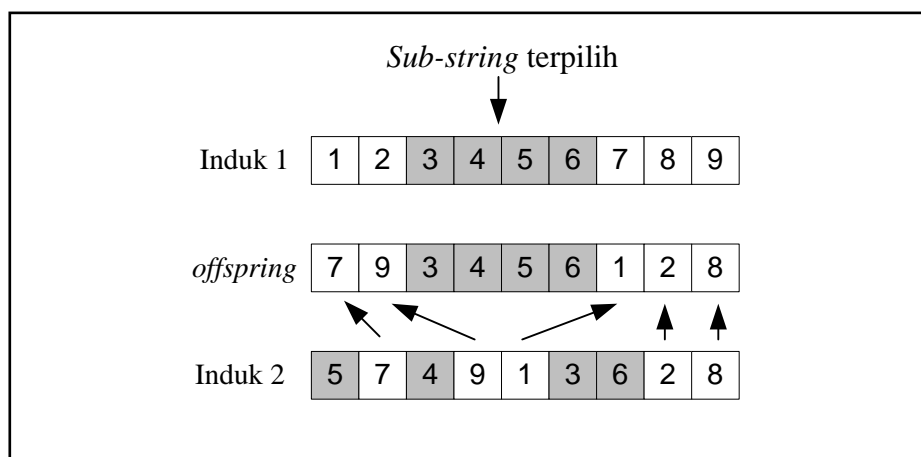
OX merupakan variasi dari operator PMX dengan beberapa perbedaan di dalam prosedur perbaikan. Adapun langkah-langkah OX adalah:

Langkah 1 Pilih *sub-string* dari satu induk secara acak.

Langkah 2 Hasilkan *proto-child* dengan menyalin *sub-string* ke dalam posisi yang sama dengan *sub-string*.

Langkah 3 Hapus kota yang telah ada pada *sub-string* dari induk yang ke-2. Urutan kota yang dihasilkan berisi kota yang dibutuhkan oleh *proto-child*.

Langkah 4 Letakkan kota pada posisi yang belum terisi dari arah kiri ke arah kanan menurut urutan rangkaian untuk menghasilkan kromosom anak.

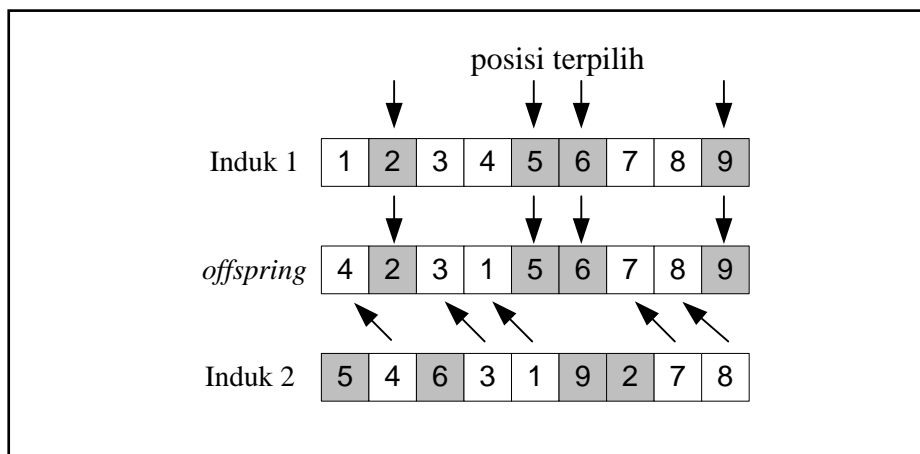


Gambar 2.3 Operator OX

3. *Position-Based Crossover*

Position-based crossover merupakan variasi dari operator OX dengan beberapa perbedaan di dalam pemilihan kota yang terpilih. Adapun langkah-langkah *position-based crossover* adalah:

- Langkah 1* Pilih kumpulan posisi dari satu induk secara acak.
- Langkah 2* Hasilkan sebuah *proto-child* dengan melakukan penyalinan kota pada posisi yang terpilih ke dalam posisi yang sama pada *proto-child*.
- Langkah 3* Hapus kota yang telah terpilih dari Induk 2. Hasil urutan kota memuat kota yang dibutuhkan oleh *proto-child*.
- Langkah 4* Letakkan kota pada posisi yang belum terisi dari arah kiri ke arah kanan menurut urutan rangkaian untuk menghasilkan kromosom anak.



Gambar 2.4 Operator *Position-Based Crossover*

e. Mutasi

Proses selanjutnya setelah rekombinasi adalah mutasi. Mutasi gen adalah proses penggantian gen dengan nilai inversinya. Proses ini dilakukan secara acak pada posisi gen tertentu pada individu-individu yang terpilih untuk dimutasikan. Banyaknya individu yang mengalami mutasi ditentukan oleh besarnya probabilitas mutasi. Probabilitas mutasi (P_m) didefinisikan sebagai prosentase dari jumlah total gen pada populasi yang mengalami mutasi. Probabilitas ini mengendalikan banyaknya gen baru yang akan dimunculkan untuk di evaluasi. Jika probabilitasnya terlalu kecil, banyak gen yang mungkin berguna tidak pernah dievaluasi. Namun jika probabilitasnya terlalu besar, maka akan terlalu banyak gangguan acak, sehingga anak akan kehilangan kemiripan dari induknya (Kusumadewi dan Purnomo, 2005).

2.1.4 Kelebihan AG

Kelebihan algoritma genetik terdapat pada beberapa hal, yaitu (Michalewicz, 1996):

- a. *Simple*, algoritma genetik mudah untuk dikembangkan dan divalidasi.

- b. *Efficient*, algoritma genetik bersifat paralel, menggunakan sumber daya yang berasal dari keseluruhan populasi dari setiap individu. Selama proses evolusi berlangsung, masing-masing individu melakukan pertukaran informasi menggunakan *crossover*.
- c. *Global optimum*, menggunakan populasi, stokastik *crossover*, dan mutasi menuju *global optimum* agar terhindar dari *local optimum*.
- d. *Domain Independent*, algoritma genetik merupakan metode parametrik, sesuai untuk digunakan dalam cakupan yang luas, karena tidak membutuhkan pengetahuan mengenai distribusi data dan kontinuitas.

2.1.5 Kekurangan Algoritma Genetik

Kekurangan Algoritma Genetik terdapat pada beberapa hal, (Berlianty dan Arifin, 2010):

- a. Algoritma genetik bekerja dengan bilangan acak pada kromosom awal, sehingga memungkinkan kromosom terbaik tidak terlibat dalam proses.
- b. Algoritma genetik menggunakan pembangkitan bilangan random dalam setiap pemilihan kromosom baik untuk induk, proses persilangan maupun mutasi.
- c. Solusi yang dihasilkan belum tentu merupakan solusi yang optimal, karena sangat dipengaruhi oleh bilangan acak yang dibangkitkan.

2.1.6 *Visual Basic for Application (VBA)*

Visual Basic mulai berkembang sejak bahasa generasi ketiga yang dinamakan *Basic (Beginners All Purpose Instruction Code)*. *Basic* merupakan bahasa yang sangat

cocok digunakan untuk *microcomputer* karena kecilnya kebutuhan *memory* dibandingkan dengan bahasa yang lain. Kemudian, sejak tahun 1980 hingga saat ini, *Microsoft* melanjutkan komitmen mereka untuk mengembangkan *Basic* mulai dalam sistem operasi DOS hingga sistem operasi *Windows*. Kini, *Visual Basic* merupakan bahasa pemrograman visual yang sangat sukses (Darlington, 2004).

Visual Basic for Application (VBA) merupakan cabang dari bahasa pemrograman *Visual Basic* yang dikembangkan untuk *Microsoft Office*. Dengan kata lain, VBA merupakan bagian dari aplikasi *Microsoft Office*, oleh karena itu VBA tidak dapat digunakan tanpa menggunakan aplikasi *Microsoft Office* (Darlington, 2004). Dengan VBA kita bisa melakukan banyak hal seperti melakukan peramalan dan penganggaran, menganalisis data ilmiah, menciptakan faktur dan formulir lainnya, mengembangkan grafik dari data, dan lain-lain (Walkenbach, 2007). Kode program ditulis dan disimpan di sebuah *module* VBA. *Module* VBA bersama-sama dengan *worksheet* dan *object* lain disimpan di dalam *Workbook Excel* dan dapat dilihat, ditulis serta disunting melalui *Visual Basic Editor* (VBE).

2.2 Kajian Induktif

Kajian induktif merupakan salah satu sub bab yang menyajikan penelitian-penelitian terdahulu yang mengacu pada algoritma genetik, adapun penelitian-penelitian terdahulu dengan topik penggunaan algoritma genetik dalam permasalahan elemen permutasi antara lain:

Albayrak dan Allahverdi (2011) meneliti mengenai pembangunan operator mutasi baru untuk menyelesaikan TSP menggunakan algoritma genetik. Operator mutasi baru disebut dengan *Greedy Sub Tour Mutation* (GTSM) digunakan untuk

meningkatkan kinerja dari algoritma genetik untuk mencari jarak terpendek di dalam TSP. Aplikasi dari operator mutasi baru ini memberikan hasil yang lebih efektif terhadap nilai *error* terbaik dan nilai *error* rata-rata.

Majumdar dan Bhunia (2011) telah melakukan penelitian mengenai varian dari *Asymmetric Travelling Salesman Problem* (ATSP) dimana waktu tempuh diantara kedua kota diwakili oleh nilai interval sebagai nilai tetap (deterministik) seperti pada ATSP klasik. Pada penelitian ini, algoritma genetik dengan nilai interval fungsi fitness diusulkan untuk menyelesaikan interval ATSP. Untuk menganalisa performansi serta efektivitas dari algoritma yang diusulkan dan operator genetik yang berbeda, penelitian ini melakukan studi komputasi mengenai algoritma yang diusulkan dengan mengujikannya terhadap studi kasus.

Liu (2010) melakukan penelitian mengenai pembangunan tiga pembangkit inialisasi solusi yaitu *nearest neighbor-type 1* (NN1), *nearest neighbor-type2* (NN2), dan *random generator* (RAN) dalam algoritma genetik untuk menyelesaikan topik permasalahan *probabilistic travelling salesman problem* (PTSP). Dari penelitian ini dapat kita ketahui bahwa NN1 relatif lebih baik dibandingkan dengan yang lain jika dilihat dari nilai rata-rata $E[t]$ dan dapat menghemat waktu komputasi sebesar 50.2%.

Zhao, et al., (2009) melakukan penelitian mengenai algoritma genetik yang digunakan untuk *travelling salesman problem* pada komoditas *pickup-and-delivery*. Pada algoritma yang diusulkan, Zhao mendesain konstruksi perjalanan heuristik yang baru yang digunakan untuk membangkitkan inialisasi populasi, dan mengusulkan *a novel-pheromone* berbasis operator *crossover* yang memanfaatkan informasi lokal dan global untuk dapat menghasilkan kromosom anak. Sebagai tambahannya, prosedur pencarian lokal ditambahkan pada algoritma genetik untuk mempercepat terjadinya

konvergen. Algoritma yang diusulkan ini telah diujikan kepada 500 pelanggan, hasil komputasi memperlihatkan bahwa algoritma yang diusulkan membuat lebih cepat dan lebih baik terjadinya konvergen daripada metode heuristik yang ada.

Wu, et al., (2007) telah melakukan penelitian mengenai penggunaan algoritma genetik untuk desain dan tata letak *cellular manufacturing*. Pada penelitian ini, *hierarchical genetic algorithms* (HGA) dikembangkan secara simultan untuk membentuk *cell* manufaktur dan menentukan pengelompokan tata letak dari sistem *cellular manufacturing*. Dari hasil analisis komputasi dapat diketahui bahwa struktur (skema seleksi baru) dan operator (group operator mutasi) yang ditemukan efektif digunakan untuk meningkatkan kualitas hasil dalam mempercepat terjadinya konvergen.

Zhang dan Ishikawa (2004) telah melakukan penelitian mengenai *hybrid* algoritma genetik yang digunakan untuk mencari solusi dalam optimisasi permasalahan kombinatorial. Dua poin penting dari penelitian ini terletak pada *elitism* dan pencarian tidak berulang (*non redundant search*). Hasilnya adalah penerapan *hybrid* algoritma genetik terbukti mampu menyelesaikan permasalahan kombinatorial 1,3 kali lebih cepat dibandingkan menggunakan standar algoritma genetik.

BAB III

METODOLOGI PENELITIAN

3.1 Lokasi dan Objek Penelitian

Penelitian ini dilakukan di Laboratorium Data Mining, Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia dengan objek penelitian yaitu Algoritma Genetik.

3.2 Data yang Dibutuhkan

Untuk menunjang kegiatan penelitian, diperlukan data-data sebagai berikut:

a. *Popsize*

Merupakan ukuran populasi, yaitu jumlah individu (kromosom) yang terdapat dalam populasi.

b. *Maxgen*

Merupakan jumlah maksimum generasi yang akan digunakan dalam melakukan proses algoritma genetik.

c. Peluang *Crossover* (*Pc*)

Peluang *crossover* menunjukkan rasio dari anak yang dihasilkan dalam setiap generasi dengan ukuran populasi.

d. Peluang Mutasi (P_m)

Peluang mutasi menunjukkan prosentasi jumlah total gen pada populasi yang akan mengalami mutasi.

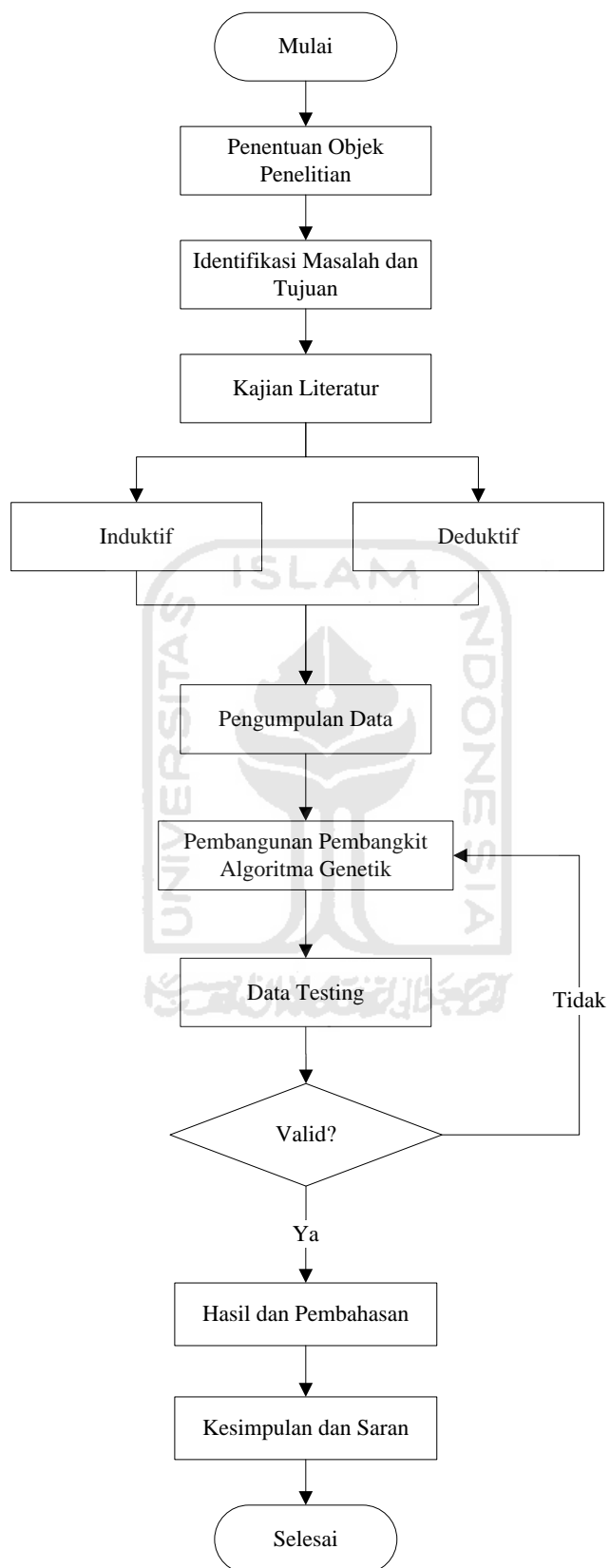
3.3 Alat dan Bahan

Alat dan bahan yang digunakan untuk pelaksanaan penelitian ini yaitu alat tulis, data penunjang penelitian, dan perangkat lunak *Microsoft Excel 2007* beserta *Visual Basic Editor*.

3.4 Pembangunan Pembangkit Algoritma Genetik

Pembangunan pembangkit ini dibangun dengan melakukan penyandian terhadap prosedur-prosedur yang terdapat pada algoritma genetik yang dapat digunakan untuk menyelesaikan studi kasus pada kromosom dengan representasi elemen permutasi. Pembangkit yang dibangun kemudian akan dilakukan uji stabilitas terhadap keseluruhan penyandian yang dilakukan terhadap prosedur inialisasi populasi, seleksi, *crossover*, dan mutasi.

3.5 Diagram Alir Penelitian



Gambar 3.1 Diagram Alir Penelitian

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pembangunan Pembangkit Algoritma Genetik

Pembangunan pembangkit algoritma genetik dilakukan dengan melakukan penyandian menggunakan bahasa pemrograman VBA yang ditulis di dalam VBE. Prosedur yang dibangun meliputi prosedur inialisasi populasi, seleksi, *crossover* dan mutasi.

4.1.1 Inialisasi Populasi

Inialisasi populasi diciptakan untuk membentuk sebuah populasi dari kumpulan individu yang merepresentasikan solusi. Ukuran populasi ditentukan oleh *user* dengan memberikan nama pada sebuah *range* dengan nama "*popsize*". Setelah ukuran populasi ditentukan, kemudian inialisasi terhadap kromosom dilakukan secara acak.

```
Memulai Proses Inialisasi Populasi:  
Membangkitkan sejumlah (popsize) individu secara acak  
Menghitung nilai fitness setiap individu ( $v_k$ )
```

Gambar 4.1 Kode Bayangan Inialisasi Populasi

4.1.2 Seleksi

Seleksi merupakan sebuah proses dalam algoritma genetik yang digunakan untuk memilih calon induk. Dalam penelitian ini, metode seleksi yang digunakan adalah *roulette wheel selection*. Setelah melakukan perhitungan nilai *fitness* dari setiap kromosom, kemudian menghitung nilai probabilitas seleksi dan probabilitas kumulatif, maka akan dibangkitkan bilangan random guna memilih kromosom induk dengan membandingkannya dengan nilai probabilitas *fitness* kumulatif.

```

Memulai Proses Seleksi:
Menghitung total fitness
Menghitung fitness relatif ( $p_k$ ) setiap individu
Menghitung fitness kumulatif ( $q_k$ )
Membangkitkan bilangan acak ( $r$ ) pada kisaran  $[0,1]$ , sejumlah


popsize


Jika nilai  $r \leq q_1$ , maka pilih individu  $v_1$ , sebaliknya, pilih
kromosom ke- $k$ ,  $v_k$  ( $2 \leq k \leq k$  populasi) sedemikian sehingga
 $q_{k-1} < r \leq q_k$ .

```

Gambar 4.2 Kode Bayangan Seleksi

4.1.3 Crossover

Crossover merupakan sebuah proses perkawinan silang antara 2 induk kromosom untuk menghasilkan kromosom anak. Probabilitas *crossover* ditentukan oleh *user* dengan memberi nama pada sebuah *range* dengan nama "*pc*". Ada 2 operator *crossover* yang dibangun, yaitu *order crossover* dan *position based crossover*. Dalam melakukan pemilihan terhadap operator *crossover* yang digunakan, *user* cukup

memberikan nama pada *range* dengan “*pilcross*” dan memberikan nilai “1” untuk *order crossover* dan nilai “2” untuk *position based crossover*.

a. *Order crossover*

Memulai Proses *Order Crossover*:

Membangkitkan bilangan acak pada kisaran $[0,1]$, sejumlah *popsize*.

Jika bilangan acak ke-*k* individu kurang dari peluang *crossover* (*pc*), pilih individu tersebut sebagai induk untuk *crossover*.

Memilih *sub-string* dari satu induk secara acak.

Menghasilkan *proto-child* dengan menyalin *sub-string* ke dalam posisi yang sama dengan *sub-string*.

Menghapus kota yang telah ada pada *sub-string* dari induk yang ke-2.

Meletakkan kota pada posisi yang belum terisi dari arah kiri ke arah kanan menurut urutan rangkaian untuk menghasilkan kromosom anak.

Gambar 4.3 Kode Bayangan Operator *Order Crossover*

b. *Position-based crossover*

Memulai Proses *Position-Based Crossover*:

Membangkitkan bilangan acak pada kisaran $[0,1]$, sejumlah *popsize*.

Jika bilangan acak ke-*k* individu kurang dari peluang *crossover* (*pc*), pilih individu tersebut sebagai induk untuk *crossover*.

Gambar 4.4 Kode Bayangan Operator *Position-Based Crossover*

Memilih kumpulan posisi (kota) dari satu induk secara acak.
 Menghasilkan sebuah *proto-child* dengan melakukan penyalinan kota pada posisi yang terpilih ke dalam posisi yang sama pada *proto-child*.
 Menghapus kota yang telah terpilih dari induk ke-2.
 Letakkan kota pada posisi yang belum terisi dari arah kiri ke arah kanan menurut urutan rangkaian untuk menghasilkan kromosom anak.

Gambar 4.5 Kode Bayangan Operator *Position-Based Crossover* (lanjutan)

4.1.4 Mutasi

Mutasi merupakan operator genetik lain selain *crossover*. Pada proses mutasi ini, akan dibangkitkan bilangan random yang kemudian akan dibandingkan dengan probabilitas mutasi. Apabila nilai dari bilangan random pada setiap kromosom yang dibangkitkan kurang dari besarnya nilai probabilitas mutasi, maka kromosom tersebut akan terkena mutasi. Besarnya nilai probabilitas mutasi ditentukan oleh user dengan memberi nama pada sebuah range dengan nama "*pm*".

Memulai Proses Mutasi:
 Membangkitkan bilangan acak pada kisaran $[0,1]$, sejumlah *popsize*.
 Jika bilangan acak ke-*k* individu kurang dari peluang mutasi (*pm*), maka individu tersebut akan terkena mutasi.
 Memilih dua kota pada satu individu secara acak
 Menukar kedua kota yang terpilih

Gambar 4.6 Kode Bayangan Mutasi

4.2 Pengolahan Data

4.2.1 Pengujian Pembangkit

Pengujian data dilakukan dengan menggunakan sebuah studi kasus dengan data numerik yang akan diselesaikan dengan pembangkit yang baru dibangun, selanjutnya akan diujikan menggunakan operator *order crossover* dan *position based crossover*. Studi kasus yang digunakan menggunakan permasalahan *travelling salesman problem* (TSP) yang diselesaikan dengan algoritma genetik dimana terdapat 7 kota dengan koordinat lokasi seperti pada Tabel 4.1

Tabel 4.1 Koordinat Lokasi Kota

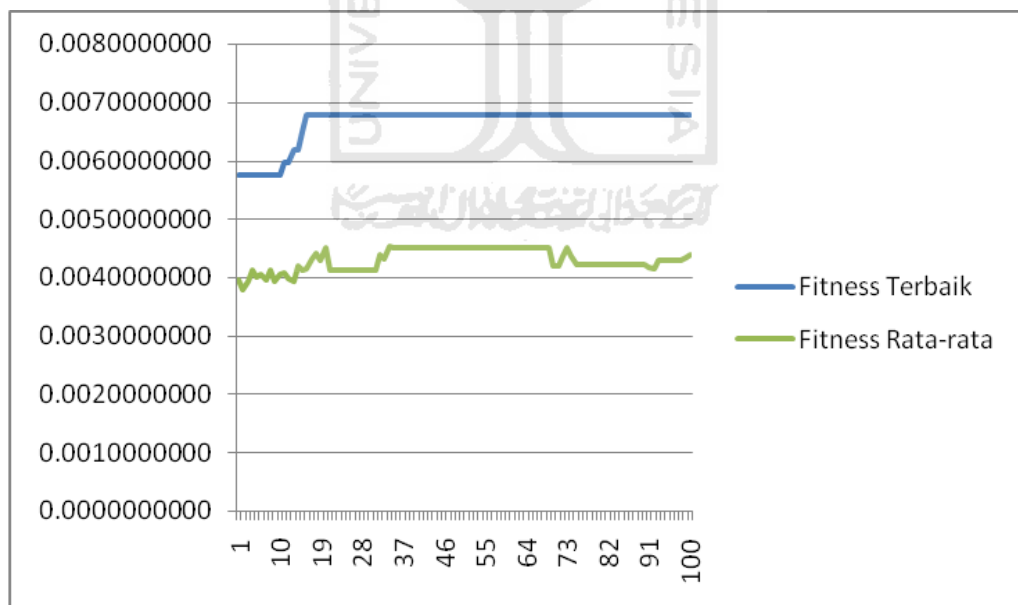
Kota	X	Y
1.	29	94
2.	59	73
3.	59	26
4.	73	69
5.	34	11
6.	24	87
7.	12	65

Pengujian ini dilakukan sebanyak 100 kali percobaan. Jarak antar kota dihitung menggunakan bentuk *Euclidean*, adapun parameter yang digunakan dalam studi kasus ini seperti ukuran populasi (*popsize*) = 20, maksimum generasi (*maxgen*) = 100, peluang *crossover* (*pc*) = 0.25, peluang mutasi (*pm*) = 0.1, fungsi *fitness* = 1/panjang jalur. Setelah melakukan proses komputasi, didapatkan hasil seperti pada Tabel 4.2, Tabel 4.3, Gambar 4.7 dan Gambar 4.8.

Tabel 4.2 Hasil Pengujian Menggunakan Operator *Order Crossover*

Jumlah	Urutan Kota							Jarak
91	3	1	2	4	0	5	6	147.398
7	6	5	0	4	2	3	1	153.742
2	0	6	5	4	2	1	3	160.915

Tabel 4.2 di atas memperlihatkan bahwa pembangkit yang dibangun menggunakan operator *order crossover* setelah diujikan sebanyak 100 kali percobaan, menghasilkan 91 urutan kota dengan jarak 147.398 dan menghasilkan 7 urutan kota dengan jarak 153.742 serta menghasilkan 2 urutan kota dengan jarak 160.915. Pada Gambar 4.7 di bawah ini menunjukkan hasil komputasi pembangkit yang dibangun menggunakan operator *order crossover* pada satu kali pengujian, terlihat bahwa pembangkit yang dibangun mulai menghasilkan nilai *fitness* yang konvergen pada generasi ke-17.

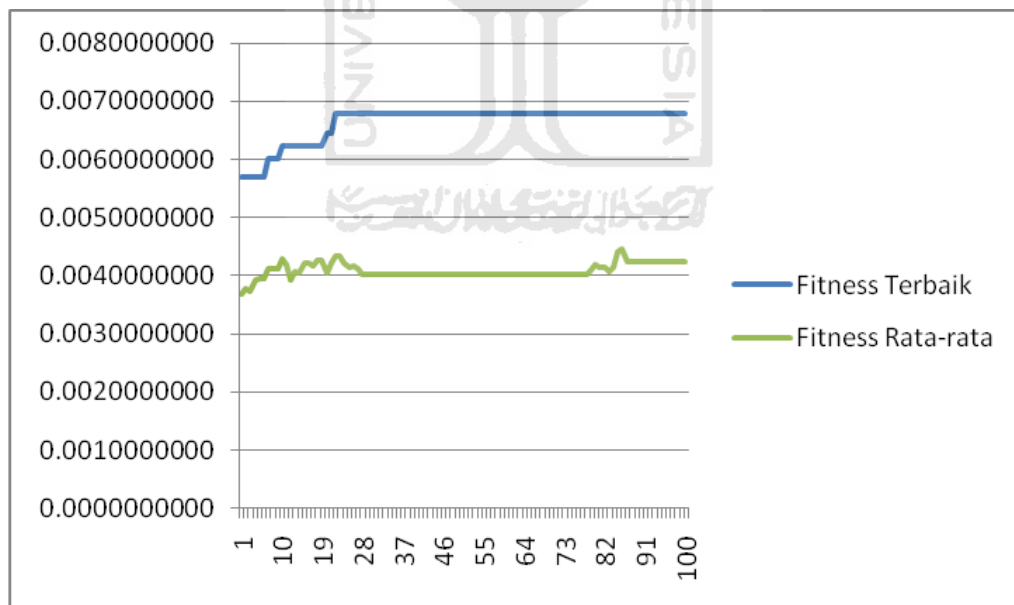
Gambar 4.7 Grafik AG Menggunakan *Order Crossover*

Tabel 4.3, menunjukkan bahwa pembangkit yang dibangun menggunakan operator *position-based crossover* setelah diujikan sebanyak 100 kali percobaan, menghasilkan 91 urutan kota dengan jarak 147.398 dan menghasilkan 9 urutan kota dengan jarak 153.742.

Tabel 4.3 Hasil Pengujian Menggunakan Operator *Position-Based Crossover*

Jumlah	Urutan Kota							Jarak
91	3	1	2	4	0	5	6	147.398
9	6	5	0	4	2	3	1	153.742

Gambar 4.8 di bawah ini menunjukkan hasil komputasi pembangkit yang dibangun menggunakan operator *position-based crossover* pada satu kali pengujian, terlihat bahwa pembangkit yang dibangun mulai menghasilkan nilai *fitness* yang konvergen pada generasi ke-23.



Gambar 4.8 Grafik AG Menggunakan *Position-based Crossover*

BAB V

PEMBAHASAN

5.1 Analisis Pembangunan Pembangkit AG

Pembangkit AG dibangun menggunakan bahasa pemrograman VBA yang ditulis dalam VBE. Pembangunan pembangkit menggunakan bahasa pemrograman VBA ini dilakukan karena VBA merupakan bagian dari aplikasi *Microsoft Office*, yang artinya pembangunan pembangkit yang dilakukan berada di dalam *platform*, sehingga proses komputasinya menjadi lebih cepat dibandingkan dengan aplikasi lainnya yang membangun pembangkit menggunakan *platform* luar.

Pembangkit yang dibangun mempunyai beberapa parameter-parameter di dalamnya. Agar pembangkit yang dibangun bisa digunakan, pengguna diharuskan untuk memberikan nama pada *range* parameter yang digunakan dengan nama-nama seperti: ukuran populasi diberi nama dengan "*popsize*", maksimum generasi diberi nama dengan "*maxgen*", peluang *crossover* diberi nama dengan "*pc*", peluang mutasi diberi nama dengan "*pm*", dan "*pilcross*" merupakan nama yang digunakan untuk memilih operator *crossover* yang diinginkan, nilai 1 untuk operator *order crossover* dan nilai 2 untuk operator *position based crossover*. Alasan yang mendasari mengapa *range* parameter harus diberi nama sesuai dengan yang telah ditentukan di atas adalah agar parameter yang digunakan dalam proses AG dapat digunakan dan diproses sesuai dengan keinginan pengguna.

Pembangkit yang dibangun mempunyai beberapa kelebihan dan kekurangan. Dibandingkan dengan pembangkit komersial yang ada, pembangkit yang dibangun jauh lebih murah, karena pembangkit yang dibangun bersifat *opensource*. Selain itu pembangkit ini mempunyai kelebihan mudah digunakan dan *user friendly*, karena pembangkit ini dibangun di dalam *Microsoft Excell* sebagai antar muka untuk membangun model optimasinya. Adapun kekurangan dari pembangkit yang dibangun yang dapat digunakan untuk pengembangan yang bisa dilakukan kedepan adalah perlunya pembangunan kromosom dengan representasi lainnya seperti *stringbit*, bilangan real, daftar aturan, dll. Selain itu, pengembangan pembangkit yang dapat dilakukan adalah perlunya penambahan prosedur *scaling* terhadap nilai *fitness* yang digunakan untuk mempertahankan posisi gen guna terhindar dari lokal optimum.

5.2 Analisis Kasus

Dalam eksperimen terdapat satu buah studi kasus *travelling salesman problem* (TSP) yang akan digunakan untuk menguji pembangkit yang dirancang. Dari hasil pengujian pembangkit yang dilakukan didapatkan hasil yaitu, untuk studi kasus pertama dengan menggunakan operator *order crossover* dimana terdapat tujuh koordinat lokasi kota, menggunakan ukuran populasi 20, maksimum generasi 100, peluang *crossover* 0.25, dan peluang mutasi 0.1, dari 100 kali percobaan, jarak terpendek yang dihasilkan ialah 147,398. Dari 100 kali percobaan yang telah dilakukan, ternyata jarak terpendek yang didapatkan tidaklah selalu sama, terbukti dari 100 kali percobaan terdapat 9 jarak terpendek yang berbeda, yaitu 153,742 sebanyak 7 kali dan 160,915 sebanyak 2 kali, dengan demikian stabilitas pembangkit yang dibangun sebesar 91%. Hal ini dapat terjadi karena dalam AG terdapat beberapa proses yang menggunakan nilai-nilai

probabilistik misalnya pada proses seleksi, proses *crossover*, dan proses mutasi yang memungkinkan hasil yang didapatkan AG bisa berbeda.

Pada studi kasus pertama yang diselesaikan dengan menggunakan operator *position-based crossover*, dengan menggunakan parameter yang sama yaitu *popsize* 20, *maxgen* 100, *pc* 0.25, dan *pm* 0.1, dari 100 kali percobaan, jarak terpendek yang dihasilkan ialah 147,398. Setelah melakukan pengujian sebanyak 100 kali percobaan, ternyata jarak terpendek yang didapatkan tidaklah selalu sama, terbukti dari 100 kali percobaan terdapat 9 jarak terpendek yang berbeda yaitu 153,742, dengan demikian stabilitas pembangkit yang dibangun sebesar 91%. Pada studi kasus TSP diatas yang diselesaikan dengan dua operator *crossover* yang berbeda, terlihat bahwa hasil jarak terpendek yang didapatkan tidaklah selalu sama, hal ini senada dengan apa yang dikemukakan oleh Bunnag dan Sun (2005) dan Gen dan Cheng (1997) bahwa tidak ada jaminan bahwa AG dapat memberikan solusi yang optimal, namun dapat memberikan solusi yang baik.

Gambar 4.7 dan Gambar 4.8 merupakan grafik nilai *fitness* yang dihasilkan AG dengan *maxgen* 100. Pada kedua gambar tersebut terlihat bahwa nilai *fitness* terbaik pada grafik yang dihasilkan naik perlahan hingga mencapai nilai *fitness* paling tinggi pada angka 0.006784, grafik yang dihasilkan tidak mungkin mengalami penurunan nilai, hal ini dikarenakan adanya proses penggantian antara satu kromosom terbaik dalam satu populasi menggantikan satu kromosom dengan nilai terburuk dalam satu populasi, sehingga grafik yang dihasilkan akan terus naik hingga tidak ada lagi nilai *fitness* yang terbaik lainnya.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil pengolahan dan analisis data yang telah dilakukan, dapat disimpulkan bahwa pembangkit AG yang dibangun dapat digunakan untuk menyelesaikan permasalahan permutasi dengan baik.

6.2 Saran

Adapun saran yang dapat diberikan guna tindakan perbaikan dan penelitian selanjutnya, yaitu:

- a. Dilakukan pembangunan pembangkit menggunakan representasi elemen yang berbeda seperti string bit, bilangan real, dan daftar aturan.
- b. Dilakukan pembangunan *user interface* guna memudahkan penggunaan pembangkit yang dibangun.

DAFTAR PUSTAKA

- Albayrak, M., & Allahverdi, N. (2011). Development a New Mutation Operator to Solve the Traveling Salesman Problem by Aid of Genetic Algorithms. *Expert Systems with Applications* 38 , 1313–1320.
- Berlianty, I., & Arifin, M. (2010). *Teknik-Teknik Optimasi Heuristik*. Yogyakarta: GRAHA ILMU.
- Bunnag, D., & Sun, M. (2005). Genetic Algorithm for Constrained Global Optimization in Continuous Variables. *Applied Mathematics and Computation* 171 , 604-636.
- Darlington, K. (2004). *VBA for Excel - Made Simple*. Burlington: Made Simple Books.
- Gen, M., & Cheng, R. (1997). *Genetic Algorithms & Engineering Design*. Canada: John Wiley & Sons, Inc.
- Goldberg, D. (1989). *Genetic Algorithms in Search, Otimization and Machine Learning*. Reading, MA: Addison-Wesley.
- Guo, P., Wang, X., & Han, Y. (2010). The Enhanced Genetic Algorithms for the Optimization Design. *3rd International Conference on Biomedical Engineering and Informatics* , 2990-2994.
- Hwang, S.-F., & He, R.-S. (2006). A Hybrid Real-Parameter Genetic Algorithm for Function Optimization. *Advance Engineering Informatics* 20 , 7-21.
- Kamrani, A., Rong, W., & Gonzalez, R. (2001). A Genetic Algorithm Methodology for Data Mining and Intelligent Knowledge Acquisition. *Computers & Industrial Engineering* 40 , 361-377.
- Khouja, M., Michalewicz, Z., & Wilmot, M. (1998). The Use of Genetic Algorithms to Solve the Economic Lot Size Scheduling Problem. *European Journal of Operational Research* 110 , 509-524.
- Kusumadewi, S., & Purnomo, H. (2005). *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik*. Yogyakarta: GRAHA ILMU.
- Liu, Y.-H. (2010). Different Initial Solution Generators in Genetic Algorithms for Solving the Probabilistic Traveling Salesman Problem. *Applied Mathematics and Computation* 216 , 125–137.
- Majumdar, J., & Bhunia, A. (2011). Genetic Algorithm for Asymmetric Traveling Salesman Problem with Imprecise Travel Times. *Journal of Computational and Applied Mathematics* 235 , 3063–3078.

- Martens, J. (2004). Two Genetic Algorithms to Solve a Layout Problem in the Fashion Industry. *European Journal of Operational Research* 154 , 304–322.
- McCall, J. (2005). Genetic Algorithms for Modelling and Optimisation. *Journal of Computational and Applied Mathematics* 184 , 205-222.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer.
- Rubinovitz, J., & Letivin, G. (1995). Genetic Algorithm for Assembly Line Balancing. *International Journal Production Economics* 41 , 343-354.
- Walkenbach, J. (2007). *Excel 2007 VBA Programming For Dummies*. Indianapolis: Wiley Publishing, Inc.
- Wang, Y.-Z. (2003). Using Genetic Algorithm Methods to Solve Course Scheduling Problems. *Expert Systems with Applications* 25 , 39–50.
- Wu, X., Chu, C.-H., Wang, Y., & Yan, W. (2007). A Genetic Algorithm for Cellular Manufacturing Design and Layout. *European Journal of Operational Research* 181 , 156–167.
- Zhang, H., & Ishikawa, M. (2004). A Solution to Combinatorial Optimization with Time-Varying Parameters by a Hybrid Genetic Algorithm. *International Congress Series* 1269 , 149-152.
- Zhao, F., Li, S., Sun, J., & Mei, D. (2009). Genetic Algorithm for the One-Commodity Pickup-and-Delivery Traveling Salesman Problem. *Computers & Industrial Engineering* 56 , 1642–1648.

LAMPIRAN

1.1 Penyandian Algoritma Genetik

1.1.1 Inisialisasi Populasi

```

Function InisialisasiTSP()
    i = 0
    j = 0
    k = 0

    ReDim pop(0 To (barisPopulasi - 1), 0 To (kolomKromosom - 1))
    ReDim tpop(0 To (barisPopulasi - 1), 0 To (kolomKromosom - 1))

    For i = 0 To (barisPopulasi - 1)
        For j = 0 To (kolomKromosom - 1)
            pop(i, j) = j
            ActiveSheet.Cells(i + 4, j + 6).Value = pop(i, j)
        Next
    Next

    Dim ixR1 As Integer
    Dim ixR2 As Integer
    Dim nR1 As Integer
    Dim nR2 As Integer

    For i = 0 To (barisPopulasi - 1)
        For j = 0 To (kolomKromosom - 1)
            ixR1 = Math.Round(Rnd() * (kolomKromosom - 1))
            ixR2 = Math.Round(Rnd() * (kolomKromosom - 1))

            nR1 = pop(i, ixR1)
            nR2 = pop(i, ixR2)
            pop(i, ixR1) = nR2
            pop(i, ixR2) = nR1
        Next
    Next

    For i = 0 To (barisPopulasi - 1)
        For j = 0 To (kolomKromosom - 1)
            ActiveSheet.Cells(i + 4, j + 6).Value = pop(i, j)
        Next
    Next

    Dim tFit As Double
    Dim dist As Double
    Dim x1 As Double
    Dim y1 As Double
    Dim x2 As Double
    Dim y2 As Double

    tFit = 0
    For i = 0 To (barisPopulasi - 1)
        dist = 0
        For j = 0 To (kolomKromosom - 1)

```

```

        If (j = (kolomKromosom - 1)) Then
            x1 = ActiveSheet.Cells(pop(i, 0) + 4, 3).Value
            y1 = ActiveSheet.Cells(pop(i, 0) + 4, 4).Value
            x2 = ActiveSheet.Cells(pop(i, j) + 4, 3).Value
            y2 = ActiveSheet.Cells(pop(i, j) + 4, 4).Value
            dist = dist + (((x1 - x2) * (x1 - x2)) + ((y1 -
y2) * (y1 - y2))) ^ (1 / 2))
        Else
            x1 = ActiveSheet.Cells(pop(i, j) + 4, 3).Value
            y1 = ActiveSheet.Cells(pop(i, j) + 4, 3).Value
            x2 = ActiveSheet.Cells(pop(i, j + 1) + 4,
3).Value
            y2 = ActiveSheet.Cells(pop(i, j + 1) + 4,
3).Value
            dist = dist + (((x1 - x2) * (x1 - x2)) + ((y1 -
y2) * (y1 - y2))) ^ (1 / 2))
        End If
    Next
    ActiveSheet.Cells(i + 4, (kolomKromosom - 1) + 11).Value
= dist
    ActiveSheet.Cells(i + 4, (kolomKromosom - 1) + 12).Value
= (1 / dist)
    tFit = tFit + (1 / dist)
Next

    Dim kFit As Double
    kFit = 0
    For i = 0 To (barisPopulasi - 1)
        kFit = kFit + (ActiveSheet.Cells(i + 4, (kolomKromosom - 1)
+ 12).Value / tFit)
        ActiveSheet.Cells(i + 4, (kolomKromosom - 1) + 13).Value =
kFit
        If (i = (barisPopulasi - 1)) Then
            ActiveSheet.Cells(i + 4, (kolomKromosom - 1) + 13).Value = "1"
        Next
    End Function

```


1.1.2 Seleksi

```

Sub Seleksi()
    Dim lselect As Integer
    lselect = 0
    Dim rr As Double

    For i = 0 To (barisPopulasi - 1)
        lselect = 0
        rr = Rnd

        ActiveSheet.Cells(i + 4, (kolomKromosom - 1) + 14).Value = rr
        For j = 0 To ((barisPopulasi - 1) - 1)
            If ((rr >= ActiveSheet.Cells(j + 4, (kolomKromosom - 1) +
13).Value)
                And (rr < ActiveSheet.Cells(j + 4 + 1, (kolomKromosom - 1) +
13).Value)) Then
                lselect = j + 1
            End If
        Next
        For j = 0 To (kolomKromosom - 1)
            tpop(i, j) = pop(lselect, j)
            ActiveSheet.Cells(i + (barisPopulasi - 1) + 17, j +
6).Value = tpop(i, j)
        Next
    Next
End Sub

```

1.1.3 Order Crossover

```

Sub CrossoverOrderCrossover()
    For i = 0 To (barisPopulasi - 1)
        For j = 0 To (kolomKromosom - 1)
            pop(i, j) = tpop(i, j)
            ActiveSheet.Cells(i + 4, j + 6).Value = pop(i,
j)
            ActiveSheet.Cells(i + (barisPopulasi - 1) + 17,
j + 6).Value = -1
        Next
    Next

    Dim lselect As Integer
    lselect = 0
    Dim rr As Double
    For i = 0 To (barisPopulasi - 1)
        rr = Rnd
        ActiveSheet.Cells(i + (barisPopulasi - 1) + 17,
(kolomKromosom - 1) + 14).Value = rr
        If (rr < pCross) Then
            lselect = lselect + 1
            For j = 0 To (kolomKromosom - 1)
                tpop(i, j) = pop(i, j)
                ActiveSheet.Cells(i + (barisPopulasi - 1) +
17, j + 6).Value = tpop(i, j)
            Next
        End If
    Next
End Sub

```

```

Next
Else
  For j = 0 To (kolomKromosom - 1)
    tpop(i, j) = -1
    ActiveSheet.Cells(i + (barisPopulasi - 1) +
17, j + 6).Value = tpop(i, j)
  Next
End If
Next

Dim i1 As Integer
Dim i2 As Integer
Dim nR1 As Integer
Dim nR2 As Integer
Dim ada As Boolean
Dim m As Integer

i1 = -1
i2 = -1
For i = 0 To (barisPopulasi - 1)
  If (tpop(i, 0) <> -1) Then
    If (i1 = -1) Then
      i1 = i
    ElseIf (i2 = -1) Then
      i2 = i
      rr = Rnd * 2
      ActiveSheet.Cells(i + 4, (kolomKromosom
- 1) + 14).Value = rr
      For j = 0 To (kolomKromosom - 1)
        tpop(i1, j) = -1
        tpop(i2, j) = -1
        ActiveSheet.Cells(i1 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i1, j)
        ActiveSheet.Cells(i2 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i2, j)
      Next
      For j = Math.Round(rr) To
Math.Round(rr) + 2
        nR1 = pop(i1, j)
        nR2 = pop(i2, j)
        tpop(i1, j) = nR2
        tpop(i2, j) = nR1
        ActiveSheet.Cells(i1 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i1, j)
        ActiveSheet.Cells(i2 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i2, j)
      Next
      For j = 0 To (kolomKromosom - 1)
        ada = False
        For k = 0 To (kolomKromosom - 1)
          If (pop(i2, j) = tpop(i2, k))
Then
            ada = True
          End If
        Next

```

```

If (ada = False) Then
    For k = 0 To (kolomKromosom -
1)
        If (tpop(i2, k) = -1) Then
            tpop(i2, k) = pop(i2,
j)
                Exit For
            End If
        Next
    End If
Next

For j = 0 To (kolomKromosom - 1)
    ada = False
    For k = 0 To (kolomKromosom - 1)
        If (pop(i1, j) = tpop(i1, k))
Then
            ada = True
        End If
    Next
    If (ada = False) Then
1)
        For k = 0 To (kolomKromosom -
j)
            If (tpop(i1, k) = -1) Then
                tpop(i1, k) = pop(i1,
                    Exit For
                End If
            Next
        End If
    Next
    For j = 0 To (kolomKromosom - 1)
        ActiveSheet.Cells(i1 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i1, j)
        ActiveSheet.Cells(i2 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i2, j)
    Next
    i1 = -1
    i2 = -1
    End If
End If
Next

For i = 0 To (barisPopulasi - 1)
    If (tpop(i, 0) = -1) Then
        For j = 0 To (kolomKromosom - 1)
            tpop(i, j) = pop(i, j)
            ActiveSheet.Cells(i + (barisPopulasi - 1) +
17, j + 6).Value = pop(i, j)
        Next
    End If
Next
End Sub

```

1.1.4 Position-Based Crossover

```

Sub CrossoverPositionBased()
    For i = 0 To (barisPopulasi - 1)

        For j = 0 To (kolomKromosom - 1)
            pop(i, j) = tpop(i, j)
            ActiveSheet.Cells(i + 4, j + 6).Value = pop(i,
j)
            ActiveSheet.Cells(i + (barisPopulasi - 1) + 17,
j + 6).Value = -1
        Next
    Next

    Dim lselect As Integer
    lselect = 0
    Dim rr As Double
    For i = 0 To (barisPopulasi - 1)
        rr = Rnd
        ActiveSheet.Cells(i + 4, (kolomKromosom - 1) +
14).Value = rr
        If (rr < pCross) Then
            lselect = lselect + 1
            For j = 0 To (kolomKromosom - 1)
                tpop(i, j) = pop(i, j)
                ActiveSheet.Cells(i + (barisPopulasi - 1) +
17, j + 6).Value = tpop(i, j)
            Next
        Else
            For j = 0 To (kolomKromosom - 1)
                tpop(i, j) = -1
                ActiveSheet.Cells(i + (barisPopulasi - 1) +
17, j + 6).Value = tpop(i, j)
            Next
        End If
    Next

    Dim i1 As Integer
    Dim i2 As Integer
    Dim nR1 As Integer
    Dim nR2 As Integer
    Dim ada As Boolean
    Dim m As Integer

    i1 = -1
    i2 = -1
    For i = 0 To (barisPopulasi - 1)
        If (tpop(i, 0) <> -1) Then
            If (i1 = -1) Then
                i1 = i
            ElseIf (i2 = -1) Then
                i2 = i
            End If
        End If
    Next

    rr1 = Rnd * 3
    rr2 = Rnd * 3
    rr3 = Rnd * 3

```

```

ActiveSheet.Cells(i + (barisPopulasi -
1) + 17,
(kolomKromosom - 1) +
14).Value = rr1 & " " & rr2 & " " & rr3

For j = 0 To (kolomKromosom - 1)
    tpop(i1, j) = -1
    tpop(i2, j) = -1
    ActiveSheet.Cells(i1 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i1, j)
    ActiveSheet.Cells(i2 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i2, j)
Next

nR1 = pop(i1, rr1)
nR2 = pop(i2, rr1)
tpop(i1, rr1) = nR2
tpop(i2, rr1) = nR1

ActiveSheet.Cells(i1 + (barisPopulasi -
1) + 17, rr1 + 6).Value = tpop(i1, rr1)
ActiveSheet.Cells(i1 + (barisPopulasi -
1) + 17, rr1 + 6).Value = tpop(i2, rr1)

nR1 = pop(i1, rr2)
nR2 = pop(i2, rr2)
tpop(i1, rr2) = nR2
tpop(i2, rr2) = nR1

ActiveSheet.Cells(i1 + (barisPopulasi -
1) + 17, rr2 + 6).Value = tpop(i1, rr2)
ActiveSheet.Cells(i1 + (barisPopulasi -
1) + 17, rr2 + 6).Value = tpop(i2, rr2)

nR1 = pop(i1, rr3)
nR2 = pop(i2, rr3)
tpop(i1, rr3) = nR2
tpop(i2, rr3) = nR1

ActiveSheet.Cells(i1 + (barisPopulasi -
1) + 17, rr3 + 6).Value = tpop(i1, rr3)
ActiveSheet.Cells(i1 + (barisPopulasi -
1) + 17, rr3 + 6).Value = tpop(i2, rr3)

For j = 0 To (kolomKromosom - 1)
    ada = False
    For k = 0 To (kolomKromosom - 1)
        If (pop(i2, j) = tpop(i2, k))
Then
            ada = True
        End If
    Next
    If (ada = False) Then
        For k = 0 To (kolomKromosom -
1)
            If (tpop(i2, k) = -1) Then
                tpop(i2, k) = pop(i2,
j)
            Exit For
        End If

```

```

        Next
    End If
Next

For j = 0 To (kolomKromosom - 1)
    ada = False
    For k = 0 To (kolomKromosom - 1)
        If (pop(i1, j) = tpop(i1, k))

Then
            ada = True
        End If
    Next
    If (ada = False) Then
        For k = 0 To (kolomKromosom -

1)
            If (tpop(i1, k) = -1) Then
                tpop(i1, k) = pop(i1,

j)
            Exit For
        End If
    Next
End If
Next
For j = 0 To (kolomKromosom - 1)
    ActiveSheet.Cells(i1 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i1, j)
    ActiveSheet.Cells(i2 +
(barisPopulasi - 1) + 17, j + 6).Value = tpop(i2, j)
Next
    i1 = -1
    i2 = -1
End If
End If
Next

For i = 0 To (barisPopulasi - 1)
    If (tpop(i, 0) = -1) Then
        For j = 0 To (kolomKromosom - 1)
            tpop(i, j) = pop(i, j)
            ActiveSheet.Cells(i + (barisPopulasi - 1) +
17, j + 6).Value = pop(i, j)
        Next
    End If
Next
Exit Sub
End Sub

```

1.1.5 Mutasi

```

Sub Mutasi()
    For i = 0 To (barisPopulasi - 1)
        For j = 0 To (kolomKromosom - 1)
            pop(i, j) = tpop(i, j)
            ActiveSheet.Cells(i + 4, j + 6).Value = pop(i,
j)
        Next
    Next
    Dim rr As Double
    Dim nR1 As Integer
    Dim nR2 As Integer

    For i = 0 To (barisPopulasi - 1)
        For j = 0 To (kolomKromosom - 1)
            rr = Rnd
            ActiveSheet.Cells(i + 4, j + (kolomKromosom -
1) + 14).Value = rr
            If (rr < pMutasi) Then
                If (j < (kolomKromosom - 1)) Then
                    nR1 = tpop(i, j)
                    nR2 = tpop(i, j + 1)
                    tpop(i, j) = nR2
                    tpop(i, j + 1) = nR1
                ElseIf (j = (kolomKromosom - 1)) Then
                    nR1 = tpop(i, j)
                    nR2 = tpop(i, 0)
                    tpop(i, j) = nR2
                    tpop(i, 0) = nR1
                End If
            End If
        Next
    Next
    For i = 0 To (barisPopulasi - 1)
        For j = 0 To (kolomKromosom - 1)
            ActiveSheet.Cells(i + (barisPopulasi - 1) + 17,
j + 6).Value = tpop(i, j)
        Next
    Next
End Sub

```

1.2 Hasil Pengujian Pembangkit

1.2.1 Menggunakan Operator *Order Crossover*

Pengujian ke-	Urutan Kota							Jarak
1	3	1	2	4	0	5	6	147.398
2	6	5	0	4	2	1	3	147.398
3	6	5	0	4	2	1	3	147.398
4	6	5	0	4	2	1	3	147.398
5	6	5	0	4	2	1	3	147.398
6	6	5	0	4	2	1	3	147.398
7	3	1	2	4	0	5	6	147.398
8	3	1	2	4	0	5	6	147.398
9	6	5	0	4	2	1	3	147.398
10	3	1	2	4	0	5	6	147.398
11	3	1	2	4	0	5	6	147.398
12	3	1	2	4	0	5	6	147.398
13	6	5	0	4	2	1	3	147.398
14	6	5	0	4	2	1	3	147.398
15	3	2	1	4	0	5	6	147.398
16	6	5	0	4	2	1	3	147.398
17	3	1	2	4	0	5	6	147.398
18	3	1	2	4	0	5	6	147.398
19	3	1	2	4	0	5	6	147.398
20	6	5	0	4	2	1	3	147.398
21	3	1	2	4	0	5	6	147.398
22	3	2	1	4	0	5	6	147.398
23	3	1	2	4	0	5	6	147.398
24	6	5	0	4	2	1	3	147.398
25	6	5	0	4	1	2	3	147.398
26	3	1	2	4	0	5	6	147.398
27	6	5	0	4	2	1	3	147.398
28	6	5	0	4	2	1	3	147.398
29	6	5	0	4	1	2	3	147.398
30	6	5	0	4	2	1	3	147.398
31	3	1	2	4	0	5	6	147.398
32	6	5	0	4	1	2	3	147.398
33	3	2	1	4	0	5	6	147.398
34	1	2	3	4	0	5	6	153.742
35	6	5	0	4	2	1	3	147.398
36	3	2	1	4	0	5	6	147.398
37	0	6	5	4	2	1	3	160.915
38	3	1	2	4	0	5	6	147.398

Pengujian ke-	Urutan Kota							Jarak
39	6	5	0	4	2	1	3	147.398
40	6	5	0	4	1	2	3	147.398
41	3	1	2	4	0	5	6	147.398
42	3	1	2	4	0	5	6	147.398
43	3	1	2	4	0	5	6	147.398
44	6	5	0	4	2	1	3	147.398
45	3	1	2	4	0	5	6	147.398
46	3	2	1	4	0	5	6	147.398
47	1	2	3	4	0	5	6	153.742
48	3	1	2	4	0	5	6	147.398
49	6	5	0	4	3	2	1	153.742
50	3	1	2	4	0	5	6	147.398
51	6	5	0	4	2	3	1	153.742
52	6	5	0	4	2	1	3	147.398
53	3	1	2	4	0	5	6	147.398
54	3	1	2	4	0	5	6	147.398
55	6	5	0	4	2	1	3	147.398
56	3	1	2	4	0	5	6	147.398
57	6	5	0	4	2	1	3	147.398
58	6	5	0	4	2	1	3	147.398
59	6	5	0	4	2	1	3	147.398
60	6	5	0	4	2	1	3	147.398
61	3	1	2	4	0	5	6	147.398
62	3	1	2	4	0	5	6	147.398
63	3	1	2	4	0	5	6	147.398
64	6	5	0	4	2	1	3	147.398
65	3	1	2	4	0	5	6	147.398
66	3	1	2	4	0	5	6	147.398
67	3	1	2	4	0	5	6	147.398
68	3	1	2	4	0	5	6	147.398
69	3	1	2	4	0	5	6	147.398
70	0	6	5	4	2	1	3	160.915
71	6	5	0	4	2	1	3	147.398
72	6	5	0	4	2	1	3	147.398
73	3	1	2	4	0	5	6	147.398
74	3	1	2	4	0	5	6	147.398
75	6	5	0	4	2	1	3	147.398
76	3	1	2	4	0	5	6	147.398
77	3	1	2	4	0	5	6	147.398
78	3	1	2	4	0	5	6	147.398
79	3	1	2	4	0	5	6	147.398
80	3	1	2	4	0	5	6	147.398

Pengujian ke-	Urutan Kota							Jarak
81	3	1	2	4	0	5	6	147.398
82	3	1	2	4	0	5	6	147.398
83	3	1	2	4	0	5	6	147.398
84	6	5	0	4	2	1	3	147.398
85	3	1	2	4	0	5	6	147.398
86	3	1	2	4	0	5	6	147.398
87	1	2	3	4	0	5	6	153.742
88	3	1	2	4	0	5	6	147.398
89	6	5	0	4	3	2	1	153.742
90	3	1	2	4	0	5	6	147.398
91	6	5	0	4	2	3	1	153.742
92	6	5	0	4	2	1	3	147.398
93	3	1	2	4	0	5	6	147.398
94	3	1	2	4	0	5	6	147.398
95	6	5	0	4	1	2	3	147.398
96	3	1	2	4	0	5	6	147.398
97	6	5	0	4	2	1	3	147.398
98	6	5	0	4	2	1	3	147.398
99	6	5	0	4	2	1	3	147.398
100	6	5	0	4	2	1	3	147.398

1.2.2 Menggunakan Operator *Position-Based Crossover*

Pengujian ke-	Urutan Kota							Jarak
1	3	1	2	4	0	5	6	147.398
2	3	1	2	4	0	5	6	147.398
3	6	5	0	4	2	3	1	153.742
4	6	5	0	4	2	1	3	147.398
5	6	5	0	4	2	1	3	147.398
6	3	1	2	4	0	5	6	147.398
7	6	5	0	4	2	1	3	147.398
8	3	1	2	4	0	5	6	147.398
9	3	1	2	4	0	5	6	147.398
10	6	5	0	4	2	1	3	147.398
11	3	1	2	4	0	5	6	147.398
12	3	1	2	4	0	5	6	147.398
13	3	1	2	4	0	5	6	147.398
14	6	5	0	4	2	1	3	147.398
15	3	1	2	4	0	5	6	147.398
16	1	2	3	4	0	5	6	153.742

Pengujian ke-	Urutan Kota							Jarak
17	6	5	0	4	2	1	3	147.398
18	3	1	2	4	0	5	6	147.398
19	6	5	0	4	2	1	3	147.398
20	3	1	2	4	0	5	6	147.398
21	3	1	2	4	0	5	6	147.398
22	3	1	2	4	0	5	6	147.398
23	6	5	0	4	2	3	1	153.742
24	6	5	0	4	2	1	3	147.398
25	3	1	2	4	0	5	6	147.398
26	3	1	2	4	0	5	6	147.398
27	3	1	2	4	0	5	6	147.398
28	3	1	2	4	0	5	6	147.398
29	6	5	0	4	1	2	3	147.398
30	6	5	0	4	2	1	3	147.398
31	3	1	2	4	0	5	6	147.398
32	6	5	0	4	2	1	3	147.398
33	3	1	2	4	0	5	6	147.398
34	6	5	0	4	2	1	3	147.398
35	3	1	2	4	0	5	6	147.398
36	3	1	2	4	0	5	6	147.398
37	3	1	2	4	0	5	6	147.398
38	6	5	0	4	2	1	3	147.398
39	3	1	2	4	0	5	6	147.398
40	3	1	2	4	0	5	6	147.398
41	3	1	2	4	0	5	6	147.398
42	3	1	2	4	0	5	6	147.398
43	6	5	0	4	2	3	1	153.742
44	6	5	0	4	2	1	3	147.398
45	6	5	0	4	2	1	3	147.398
46	3	1	2	4	0	5	6	147.398
47	6	5	0	4	1	2	3	147.398
48	3	1	2	4	0	5	6	147.398
49	3	1	2	4	0	5	6	147.398
50	6	5	0	4	1	2	3	147.398
51	3	1	2	4	0	5	6	147.398
52	3	1	2	4	0	5	6	147.398
53	3	2	1	4	0	5	6	147.398
54	6	5	0	4	2	1	3	147.398
55	3	1	2	4	0	5	6	147.398
56	1	2	3	4	0	5	6	153.742
57	6	5	0	4	2	1	3	147.398
58	3	1	2	4	0	5	6	147.398

Pengujian ke-	Urutan Kota							Jarak
59	6	5	0	4	2	1	3	147.398
60	3	1	2	4	0	5	6	147.398
61	3	1	2	4	0	5	6	147.398
62	3	1	2	4	0	5	6	147.398
63	6	5	0	4	2	3	1	153.742
64	6	5	0	4	2	1	3	147.398
65	6	5	0	4	2	1	3	147.398
66	3	1	2	4	0	5	6	147.398
67	3	1	2	4	0	5	6	147.398
68	3	1	2	4	0	5	6	147.398
69	6	5	0	4	2	1	3	147.398
70	3	1	2	4	0	5	6	147.398
71	3	1	2	4	0	5	6	147.398
72	3	1	2	4	0	5	6	147.398
73	6	5	0	4	2	1	3	147.398
74	3	1	2	4	0	5	6	147.398
75	1	2	3	4	0	5	6	153.742
76	6	5	0	4	2	1	3	147.398
77	3	1	2	4	0	5	6	147.398
78	6	5	0	4	2	1	3	147.398
79	3	1	2	4	0	5	6	147.398
80	3	1	2	4	0	5	6	147.398
81	3	1	2	4	0	5	6	147.398
82	3	1	2	4	0	5	6	147.398
83	6	5	0	4	2	3	1	153.742
84	6	5	0	4	2	1	3	147.398
85	6	5	0	4	2	1	3	147.398
86	3	1	2	4	0	5	6	147.398
87	6	5	0	4	2	1	3	147.398
88	3	1	2	4	0	5	6	147.398
89	3	1	2	4	0	5	6	147.398
90	6	5	0	4	2	1	3	147.398
91	3	1	2	4	0	5	6	147.398
92	3	1	2	4	0	5	6	147.398
93	3	1	2	4	0	5	6	147.398
94	6	5	0	4	2	1	3	147.398
95	3	1	2	4	0	5	6	147.398
96	1	2	3	4	0	5	6	153.742
97	6	5	0	4	2	1	3	147.398
98	3	1	2	4	0	5	6	147.398
99	6	5	0	4	2	1	3	147.398
100	3	1	2	4	0	5	6	147.398

1.3 Hasil Pengujian Pembangkit (Satu kali pengujian)

1.3.1 Menggunakan Operator *Order Crossover*

Generasi Ke-	Kromosom							Fitness Terbaik	Fitness Terburuk	Fitness Rata-rata
1	0	3	2	1	4	6	5	0.0057449700	0.0039787677	0.0039673829
2	0	3	2	1	4	6	5	0.0057449700	0.0037868021	0.0037925928
3	0	3	2	1	4	6	5	0.0057449700	0.0037668155	0.0039274605
4	0	3	2	1	4	6	5	0.0057449700	0.0039566829	0.0041374889
5	0	3	2	1	4	6	5	0.0057449700	0.0039566829	0.0040010256
6	0	3	2	1	4	6	5	0.0057449700	0.0038058182	0.0040548035
7	0	3	2	1	4	6	5	0.0057449700	0.0036489417	0.0039690899
8	0	3	2	1	4	6	5	0.0057449700	0.0040352141	0.0041192484
9	0	3	1	2	4	6	5	0.0057449700	0.0037553011	0.0039293500
10	0	3	2	1	4	6	5	0.0057449700	0.0040130162	0.0040448713
11	6	5	4	0	2	3	1	0.0059564887	0.0040130162	0.0040819967
12	6	5	4	0	2	3	1	0.0059564887	0.0037574958	0.0039942240
13	6	5	4	0	2	1	3	0.0061904107	0.0037574958	0.0039438405
14	6	5	4	0	2	1	3	0.0061904107	0.0039787677	0.0041934516
15	6	5	0	4	2	3	1	0.0065044032	0.0039302912	0.0041275730
16	6	5	0	4	2	1	3	0.0067843510	0.0039787677	0.0041477136
17	6	5	0	4	2	1	3	0.0067843510	0.0039683712	0.0043220356
18	6	5	0	4	1	2	3	0.0067843510	0.0039683712	0.0044036670
19	6	5	0	4	2	1	3	0.0067843510	0.0039683712	0.0042883311
20	6	5	0	4	2	1	3	0.0067843510	0.0039787677	0.0045074230
21	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
22	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
23	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
24	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
25	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
26	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
27	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
28	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
29	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
30	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
31	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041321813
32	6	5	0	4	2	1	3	0.0067843510	0.0040585571	0.0043891798
33	6	5	0	4	2	1	3	0.0067843510	0.0038501864	0.0043287695
34	6	5	0	4	2	1	3	0.0067843510	0.0039811090	0.0045332173
35	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
36	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
37	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
38	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394

Generasi Ke-	Kromosom							Fitness Terbaik	Fitness Terburuk	Fitness Rata-rata
39	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
40	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
41	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
42	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
43	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
44	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
45	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
46	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
47	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
48	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
49	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
50	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
51	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
52	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
53	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
54	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
55	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
56	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
57	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
58	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
59	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
60	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
61	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
62	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
63	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
64	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
65	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
66	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
67	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
68	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
69	6	5	0	4	2	1	3	0.0067843510	0.0039748767	0.0045143394
70	6	5	0	4	1	2	3	0.0067843510	0.0038474867	0.0041999000
71	6	5	0	4	1	2	3	0.0067843510	0.0037574958	0.0041903023
72	6	5	0	4	1	2	3	0.0067843510	0.0037574958	0.0043721809
73	6	5	0	4	1	2	3	0.0067843510	0.0039302912	0.0045119659
74	6	5	0	4	1	2	3	0.0067843510	0.0038501864	0.0043639777
75	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
76	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
77	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
78	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
79	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
80	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243

Generasi Ke-	Kromosom							Fitness Terbaik	Fitness Terburuk	Fitness Rata-rata
81	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
82	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
83	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
84	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
85	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
86	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
87	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
88	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
89	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
90	6	5	0	4	1	2	3	0.0067843510	0.0039811090	0.0042242243
91	6	5	0	4	1	2	3	0.0067843510	0.0036747776	0.0041736014
92	6	5	0	4	2	1	3	0.0067843510	0.0037668155	0.0041603874
93	6	5	0	4	1	2	3	0.0067843510	0.0039302912	0.0042979911
94	6	5	0	4	1	2	3	0.0067843510	0.0039302912	0.0042979911
95	6	5	0	4	1	2	3	0.0067843510	0.0039302912	0.0042979911
96	6	5	0	4	1	2	3	0.0067843510	0.0039302912	0.0042979911
97	6	5	0	4	1	2	3	0.0067843510	0.0039302912	0.0042979911
98	6	5	0	4	1	2	3	0.0067843510	0.0039302912	0.0042979911
99	6	5	0	4	2	1	3	0.0067843510	0.0040719009	0.0043497994
100	6	5	0	4	2	1	3	0.0067843510	0.0040719009	0.0043971922

1.3.2 Menggunakan Operator *Position-based Crossover*

Generasi Ke-	Kromosom							Fitness Terbaik	Fitness Terburuk	Fitness Rata-rata
1	3	2	1	0	5	4	6	0.0056920929	0.0034616197	0.0036725910
2	3	2	1	5	0	4	6	0.0056920929	0.0035139048	0.0037672240
3	3	2	1	0	5	4	6	0.0056920929	0.0035359484	0.0037127220
4	3	2	1	0	5	4	6	0.0056920929	0.0036621953	0.0039234965
5	3	2	1	0	5	4	6	0.0056920929	0.0037974982	0.0039547449
6	3	2	1	0	5	4	6	0.0056920929	0.0037974982	0.0039547449
7	4	5	6	0	1	3	2	0.0060120189	0.0039683712	0.0041278957
8	4	5	6	0	1	3	2	0.0060120189	0.0039683712	0.0041278957
9	4	5	6	0	1	3	2	0.0060120189	0.0039683712	0.0041278957
10	0	6	5	4	2	1	3	0.0062144618	0.0041912058	0.0042979741
11	0	5	6	4	2	1	3	0.0062144618	0.0037689139	0.0041812114
12	0	5	6	4	2	1	3	0.0062144618	0.0037633279	0.0039102011
13	0	5	6	4	2	1	3	0.0062144618	0.0038474867	0.0040561876
14	0	5	6	4	2	1	3	0.0062144618	0.0038474867	0.0040333968

Generasi Ke-	Kromosom							Fitness Terbaik	Fitness Terburuk	Fitness Rata-rata
15	0	5	6	4	1	2	3	0.0062144618	0.0042043230	0.0042103448
16	0	5	6	4	1	2	3	0.0062144618	0.0040688814	0.0042045744
17	0	5	6	4	1	2	3	0.0062144618	0.0039659232	0.0041638625
18	0	5	6	4	1	2	3	0.0062144618	0.0037574958	0.0042746079
19	0	5	6	4	1	2	3	0.0062144618	0.0037574958	0.0042746079
20	5	6	0	4	2	1	3	0.0064333868	0.0037458696	0.0040382092
21	5	6	0	4	2	1	3	0.0064333868	0.0040585571	0.0042139936
22	6	5	0	4	2	1	3	0.0067843510	0.0037689139	0.0043282690
23	6	5	0	4	2	1	3	0.0067843510	0.0040352141	0.0043372956
24	6	5	0	4	1	2	3	0.0067843510	0.0038474867	0.0042062335
25	6	5	0	4	2	1	3	0.0067843510	0.0037689139	0.0041394671
26	6	5	0	4	2	1	3	0.0067843510	0.0038474867	0.0041688263
27	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0041234734
28	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
29	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
30	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
31	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
32	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
33	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
34	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
35	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
36	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
37	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
38	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
39	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
40	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
41	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
42	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
43	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
44	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
45	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
46	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
47	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
48	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
49	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
50	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
51	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
52	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
53	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
54	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
55	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
56	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821

Generasi Ke-	Kromosom							Fitness Terbaik	Fitness Terburuk	Fitness Rata-rata
57	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
58	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
59	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
60	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
61	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
62	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
63	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
64	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
65	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
66	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
67	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
68	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
69	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
70	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
71	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
72	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
73	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
74	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
75	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
76	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
77	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
78	6	5	0	4	2	1	3	0.0067843510	0.0037574958	0.0040141821
79	3	2	1	4	0	5	6	0.0067843510	0.0038762217	0.0040981928
80	3	2	1	4	0	5	6	0.0067843510	0.0039566829	0.0041964273
81	3	2	1	4	0	5	6	0.0067843510	0.0040585571	0.0041386737
82	3	2	1	4	0	5	6	0.0067843510	0.0039302912	0.0041403396
83	3	2	1	4	0	5	6	0.0067843510	0.0038474867	0.0040566544
84	3	1	2	4	0	5	6	0.0067843510	0.0038501864	0.0041320663
85	3	2	1	4	0	5	6	0.0067843510	0.0040223085	0.0043991506
86	3	1	2	4	0	5	6	0.0067843510	0.0038501864	0.0044603051
87	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
88	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
89	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
90	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
91	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
92	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
93	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
94	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
95	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
96	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
97	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
98	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116

Generasi Ke-	Kromosom							Fitness Terbaik	Fitness Terburuk	Fitness Rata-rata
99	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116
100	3	1	2	4	0	5	6	0.0067843510	0.0037868021	0.0042507116

