

**OPTIMASI TRAVELING SALESMAN PROBLEM
DENGAN ALGORITMA GENETIKA
MENGUNAKAN OPERATOR PARTIALLY MATCHED CROSSOVER**

SKRIPSI

Oleh:
WAHYU PRADANA STYA BUDI
NIM. 09610060



**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2013**

**OPTIMASI TRAVELING SALESMAN PROBLEM
DENGAN ALGORITMA GENETIKA
MENGUNAKAN OPERATOR PARTIALLY MATCHED CROSSOVER**

SKRIPSI

Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Sains (S.Si)

Oleh:
WAHYU PRADANA STYA BUDI
NIM. 09610060

**JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2013**

**OPTIMASI TRAVELING SALESMAN PROBLEM
DENGAN ALGORITMA GENETIKA
MENGUNAKAN OPERATOR PARTIALLY MATCHED CROSSOVER**

SKRIPSI

Oleh:
WAHYU PRADANA STYA BUDI
NIM. 09610060

Telah Diperiksadan Disetujui untuk Diuji:
Tanggal: 10 September 2013

Pembimbing I,

Pembimbing II,

Ari Kusumastuti, S.Si, M.Pd
NIP. 19770521 200501 2 004

Abdul Azis, M.Si
NIP. 19760318 200604 1 002

Mengetahui,
Ketua Jurusan Matematika

Abdussakir, M.Pd
NIP. 19751006 200312 1 001

**OPTIMASI TRAVELING SALESMAN PROBLEM
DENGAN ALGORITMA GENETIKA
MENGUNAKAN OPERATOR PARTIALLY MATCHED CROSSOVER**

SKRIPSI

**Oleh:
WAHYU PRADANA STYA BUDI
NIM. 09610060**

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Sains (S.Si)
Tanggal: 16 September 2013

Penguji Utama : Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001 _____

Ketua Penguji : Abdussakir, M.Pd
NIP. 19751006 200312 1 001 _____

Sekretaris Penguji : Ari Kusumastuti, S.Si., M.Pd
NIP. 19770521 200501 2 004 _____

Anggota Penguji : Abdul Azis, M.Si
NIP. 19760318 200604 1 002 _____

Mengesahkan,
Ketua Jurusan Matematika

Abdussakir, M.Pd
NIP. 19751006 200312 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Wahyu Pradana Stya Budi
NIM : 09610060
Jurusan : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Optimasi *Traveling Salesman Problem* Dengan Algoritma
Genetika Menggunakan Operator *Partially Matched Crossover*

menyatakan dengan sebenarnya bahwa tugas akhir/skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila di kemudian hari terbukti atau dapat dibuktikan tugas akhir/skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, September 2013

Yang membuat pernyataan,

Wahyu Pradana Stya Budi

NIM. 09610060

MOTTO

*Raihlah ilmu, dan untuk meraih ilmu belajarlah untuk
tenang dan sabar.*

(Khalifah 'Umar)



Persembahan



Karya ini penulis persembahkan kepada orang tua:
Herman Budiono & Sulikah

Dan adik-adik:
Dwiky Dian Febrianto, Putri Oktaviani, dan
Shifa Biyandra Febrianti

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Syukur alhamdulillah penulis haturkan ke hadirat Allah SWT yang telah melimpahkan Rahmat dan Hidayah-Nya, sehingga penulis dapat menyelesaikan studi di Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan tugas akhir/skripsi ini yang berjudul "*Optimasi Traveling Salesman Problem dengan Algoritma Genetika Menggunakan Operator Partially Matched Crossover*" dengan baik.

Selanjutnya penulis haturkan ucapan terima kasih seiring do'a dan harapan *jaza kumullah ahsanal jaza'* kepada semua pihak yang telah membantu terselesaikannya skripsi ini. Ucapan terima kasih ini penulis sampaikan kepada:

1. Prof. Dr. H. Mudjia Rahardjo, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Hj. Bayyinatul M., drh., M.Siselaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Abdussakir, M.Pd selaku Ketua Jurusan Matematika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Ari Kusumastuti, S.Si, M.Pd dan Abdul Azis, M.Si selaku dosen pembimbing tugas akhir/skripsi ini yang telah banyak memberikan arahan dan pengalaman yang berharga.
5. Segenap sivitas akademika Jurusan Matematika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.

6. Ayah dan Ibu serta adik-adik penulis yang selalu memberikan do'a dan motivasi yang tiada henti kepada penulis.
7. Teman-teman Jurusan Matematika tahun angkatan 2009, khususnya Pangestuti Prima Darajat terima kasih atas dukungannya serta telah memberikan kenangan yang indah dan pengalaman yang tidak terlupakan.
8. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang ikut membantu dalam menyelesaikan tugas akhir/skripsi ini baik berupa materil maupun moril.

Penulis berharap semoga tugas akhir/skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. Semoga Allah SWT senantiasa membalas kebaikan kita semuanya. *Amin Ya Rabbal Alamin.*

Wassalamu 'alaikum Wr. Wb.

Malang, September 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL	
HALAMAN PENGAJUAN	
HALAMAN PERSETUJUAN	
HALAMAN PENGESAHAN	
HALAMAN PERNYATAAN KEASLIAN TULISAN	
HALAMAN MOTTO	
HALAMAN PERSEMBAHAN	
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
ملخص	xvi
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian	6
1.4 Batasan Masalah.....	6
1.5 Manfaat Penelitian	6
1.6 Metode Penelitian	7
1.7 Sistematika Penulisan.....	8
BAB II KAJIAN TEORI	
2.1 Optimasi dengan Algoritma Genetika	9
2.2 <i>Traveling Salesman Problem</i>	10
2.2.1 Aplikasi TSP	12
2.3 Algoritma Genetika	15
2.3.1 Istilah-istilah dalam Algoritma Genetika	17
2.3.2 Parameter dalam Algoritma Genetika	19
2.3.3 Proses Pengkodean (<i>Encoding</i>).....	20
2.3.4 Proses Pembangkitan Populasi (Inisialisasi).....	21
2.3.5 Proses Seleksi.....	22
2.3.6 Proses <i>Crossover</i>	25
2.3.7 Proses Mutasi	26
2.4 Kajian Optimasi dalam Al-Qur'an	29
2.5 Kajian Genetika (Keturunan) dalam Al-Qur'an	31
BAB III PEMBAHASAN	
3.1 <i>Traveling Salesman Problem</i>	34
3.2 Algoritma Genetika	37
3.2.1 Inisialisasi Populasi.....	37

3.2.2 Proses Seleksi.....	44
3.2.3 Proses <i>Crossover</i> Menggunakan <i>Partially Matched Crossover</i>	52
3.2.4 Proses Mutasi	58
3.3 Algoritma Genetika Generasi Kedua.....	60
3.3.1 Inisialisasi Populasi.....	60
3.3.2 Proses Seleksi.....	61
3.3.3 Proses <i>Crossover</i> Menggunakan <i>Partially Matched Crossover</i>	69
3.3.4 Proses Mutasi	75
3.4 Integrasi Algoritma Genetika dengan Islam.....	76
BAB IV PENUTUP	
4.1 Kesimpulan.....	79
4.2 Saran.....	80

**DAFTAR PUSTAKA
LAMPIRAN-LAMPIRAN**

DAFTAR GAMBAR

Gambar 2.1	Rute $A \rightarrow B \rightarrow C \rightarrow E \rightarrow D \rightarrow A$ sebagai Rute Terpendek.....	10
Gambar 2.2	<i>Knight Tour's Problem</i> oleh Euler.....	11
Gambar 2.3	Visualisasi Gen, <i>Allele</i> , Kromosom, dan Populasi pada Algoritma Genetika.....	18
Gambar 2.4	<i>Permutation Encoding</i> pada TSP.....	21
Gambar 2.5	Contoh Probabilitas Terpilihnya Suatu Kromosom dalam Roda <i>Roulette</i>	24
Gambar 3.1	Representasi Matriks Jarak Antar <i>Node</i> dalam Digraf.....	36
Gambar 3.2	Contoh Rute TSP.....	37
Gambar 3.3	Kromosom 1 (Z_1).....	38
Gambar 3.4	Kromosom 2 (Z_2).....	39
Gambar 3.5	Kromosom 3 (Z_3).....	39
Gambar 3.6	Kromosom 4 (Z_4).....	40
Gambar 3.7	Kromosom 5 (Z_5).....	40
Gambar 3.8	Kromosom 6 (Z_6).....	41
Gambar 3.9	Kromosom 7 (Z_7).....	41
Gambar 3.10	Kromosom 8 (Z_8).....	42
Gambar 3.11	Kromosom 9 (Z_9).....	42
Gambar 3.12	Kromosom 10 (Z_{10}).....	43
Gambar 3.13	Probabilitas Kumulatif dalam <i>Roulette Wheel</i>	48
Gambar 3.14	Rute Terpendek.....	60
Gambar 3.15	Probabilitas Kumulatif dalam <i>Roulette Wheel</i>	65
Gambar 4.1	Solusi Rute Terpendek.....	79

DAFTAR TABEL

Tabel 3.1	Contoh Permasalahan TSP 5 Node	35
Tabel 3.2	Matriks Jarak Antar Node	35
Tabel 3.3	Perbandingan Populasi Sebelum dan Sesudah Proses Seleksi.....	52
Tabel 3.4	Perbandingan Populasi Sebelum dan Sesudah Proses <i>Crossover</i> ..	58
Tabel 3.5	Perbandingan Populasi Sebelum dan Sesudah Proses Mutasi	59
Tabel 3.6	Perbandingan Populasi Sebelum dan Sesudah Proses Seleksi.....	69
Tabel 3.3	Perbandingan Populasi Sebelum dan Sesudah Proses <i>Crossover</i> ..	75
Tabel 3.4	Perbandingan Populasi Sebelum dan Sesudah Proses Mutasi	76



ABSTRAK

Budi, Wahyu Pradana Sty. 2013. **Optimasi *Traveling Salesman Problem* Dengan Algoritma Genetika Menggunakan Operator *Partially Matched Crossover***. Skripsi. Jurusan Matematika Fakultas Sains dan Teknologi. Universitas Islam Negeri Maulana Malik Ibrahim Malang.
Pembimbing: (I) Ari Kusumastuti, S.Si., M.Pd.
(II) Abdul Aziz, M.Si.

Traveling Salesman Problem (TSP) adalah suatu permasalahan untuk menemukan lintasan dari seorang salesman yang berawal dari sebuah lokasi asal, mengunjungi sebuah himpunan kota dan kembali lagi ke lokasi asal yang mana total dari jarak yang ditempuh adalah minimum dan setiap kota dilewati tepat hanya satu kali. Algoritma genetika adalah salah satu metode yang terbaik untuk menyelesaikan kasus TSP.

Penelitian ini difokuskan pada menemukan lintasan/rute terpendek menggunakan Algoritma Genetika. Asumsinya bahwa berawal dari suatu titik (*node*) awal kemudian melewati semua himpunan kota dan kembali lagi ke awal tanpa ada yang terlewat dua kali.

Langkah-langkah dalam Algoritma Genetika meliputi: (1) Inisialisasi Populasi, (2) Seleksi, (3) *Crossover* (kawin silang), dan (4) Mutasi. Metode yang digunakan untuk membangkitkan populasi awal adalah dengan Algoritma *Random Generator*. Inti dari Algoritma *Random Generator* adalah pembangkitan bilangan acak untuk nilai setiap gen sesuai dengan representasi kromosom. Jumlah bilangan acak yang dibangkitkan sebanyak gen pada setiap kromosom. Metode yang digunakan untuk proses seleksi adalah *Roulette Wheel Selection*. Metode ini menggunakan roda *roulette* sebagai alat untuk menyeleksi kromosom. Semakin besar probabilitas kumulatif suatu kromosom, semakin besar pula probabilitas terpilihnya kromosom tersebut untuk menjadi *parent*. Proses selanjutnya adalah *crossover* dengan menggunakan PMX (*Partially Matched Crossover*). Prosedur dalam menggunakan operator PMX adalah sebagai berikut: pertama tentukan dua posisi pada kromosom dengan aturan acak, *substring* yang berada dalam dua posisi ini dinamakan daerah pemetaan, kemudian tukar *substring* antar *parent* untuk menghasilkan keturunan, lalu tentukan hubungan pemetaan di antara dua daerah pemetaan, dan yang terakhir tentukan kromosom keturunan mengacu pada hubungan pemetaan. Proses mutasi menggunakan *Swapping Mutation*, yaitu dengan menukar posisi sebuah gen dengan gen sesudahnya.

Kesimpulan yang didapatkan setelah menyelesaikan TSP dengan Algoritma Genetika menggunakan PMX didapatkan solusi terbaik/rute terpendeknya adalah $1 \rightarrow 7 \rightarrow 2 \rightarrow 6 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 1$ dengan jarak 158. Saran untuk penelitian selanjutnya adalah mengembangkan operator *crossover* untuk PMX agar menghasilkan solusi yang lebih baik.

Kata kunci: *Traveling Salesman Problem*, Algoritma Genetika, PMX

ABSTRACT

Budi, Wahyu PradanaStya. 2013. **Genetic Algorithm For The Traveling Salesman Problem Optimization Using Partially Matched Crossover Operator**. Thesis. Mathematics Department, Faculty of Science and Technology. Maulana Malik Ibrahim Malang State Islamic University. Promotor: (I) Ari Kusumastuti, S.Si., M.Pd. (II) Abdul Aziz, M.Si.

Keywords: Traveling Salesman Problem, Genetic Algorithm, PMX

Traveling Salesman Problem (TSP) is a problem to determine the path of a salesman who came from a home location, visiting a set of cities and back to the home location where the total distance traveled is minimum and each city passed right only one time. Genetic algorithm is one of the best methods to solve the TSP case.

This study focused on finding the shortest route/path using Genetic Algorithms. Assumption is that started from a point (node) then passes all initial set of the city and back again to the beginning without any point passed twice.

Steps in Genetic Algorithms include: (1) Initialize population, (2) Selection, (3) Crossover, and (4) Mutation. The method used to generate the initial population of the algorithm is the Random Number Generator. The essence of the algorithms Random Number Generator is generating random numbers to the value of each gene according to the representation of chromosomes. The number of random numbers generated as many genes on each chromosome. The method used for the selection process is the Roulette Wheel Selection. This method uses the roulette wheel as a tool for selecting chromosomes. The greater the cumulative probability of a chromosome, the greater the probability that elected to become the parent chromosome. The next process is to use a crossover PMX (Partially Matched Crossover). The procedure of using PMX operator is as follows: first define two positions on the chromosome with random rules, substring that these two positions are in the area called mapping, then the substring swap between parent to produce offspring, and define the mapping relationships between the two regions mapping, and the latter refers to the offspring chromosomes specify the mapping relationship. Mutation process using Swapping Mutation, ie by swapping the position of a gene by gene afterwards.

The conclusion obtained after completing the TSP with Genetic Algorithm using PMX obtained the best solution / shortest route is $1 \rightarrow 7 \rightarrow 2 \rightarrow 6 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 1$ with the total distance is 158.

Suggestions for further research is to develop for PMX crossover operator to produce a better solution.

ملخص

بودي. واهيوبرادانا سيتيا. ٣ ٠١ ٢. الخوارزمية الجينية لهذه المشكلة الأمثل السفر عن طريق المشغل بائع كروس المتطابقة جزئيا. أطروحة. قسم الرياضيات. كلية العلوم والتكنولوجيا. جامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج.
المشرف: (١) أري كوسومستوتي, بكالوريوس في العلوم ماجستير في التربية
(٢) عبد العزيز, ماجستير في العلوم

الكلمات الرئيسية: السفر مشكلة بائع، الخوارزميات الجينية، المتطابقة جزئيا كروس.

السفر مشكلة بائع مشكلة لإيجاد مسار من بائع الذي جاء من موقع المنزل، وزيارة مجموعة من المدن ومرة أخرى إلى الموقع الأصلي حيث المسافة الإجمالية سافر هو الحد الأدنى وكل مدينة مرت وقت واحد الحق فقط. الخوارزمية الجينية هي واحدة من أفضل الطرق لحل مشكلة حالات السفر بائع. وركزت هذه الدراسة على إيجاد مسار /أقصر طريق استخدام الخوارزميات الجينية. افتراض أن نشأت من نقطة) عقدة (ثم تذهب من خلال كل مجموعة أولية من المدينة والعودة مرة أخرى إلى بداية والتي بدونها تجاوزت مرتين.

الخطوات في الخوارزميات الجينية ما يلي (1): تهيئة السكان، (2) اختيار، (3) كروس (هجن)، و (4) الطفرات. الطريقة المستخدمة لتوليد السكان الأولي للخوارزمية هو عدد عشوائية المولدات. جوهر خوارزميات عشوائية عدد المولدات يولد أرقام عشوائية لقيمة كل الجينات وفقا للتمثيل الكروموسومات. إنشاء عدد من الأرقام العشوائية العديد من الجينات على كل كروموسوم. الطريقة المستخدمة لعملية الاختيار هو اختيار عجلة الروليت. يستخدم هذا الأسلوب عجلة الروليت كأداة لاختيار الكروموسومات. كلما زاد الاحتمال التراكمي للكروموسوم، وزيادة احتمال أن انتخب لتصبح كروموسوم الأم. عملية القادم هو استخدام كروس PMX (كروس المتطابقة جزئيا). الإجراء استخدام المشغل PMX هي كما يلي: أولاً تعريف موقعين على كروموسوم مع قواعد عشوائية، فرعية أن هذين الموقعين في منطقة تسمى رسم الخرائط، ثم تبادل فرعية بين الوالدين لإنتاج النسل، وتحديد العلاقات ورسم الخرائط بين رسم الخرائط منطقتين، و يشير هذا الأخير إلى الكروموسومات ذرية تحديد العلاقة رسم الخرائط. مبادلة الطفرات باستخدام عملية الطفرة، أي عن طريق مبادلة موقف الجين عن طريق الجينات بعد ذلك.

الاستنتاجات التي تم الحصول عليها بعد الانتهاء من ملعقة شاي مع الخوارزميات الجينية التي تم الحصول عليها باستخدام PMX أفضل حل /أقصر الطرق هو $١ \rightarrow ٧ \rightarrow ٢ \rightarrow ٦ \rightarrow ٥ \rightarrow ٣ \rightarrow ٤$ مع مسافة ١٥٨. اقتراحات لمزيد من البحث هو تطوير ل PMX المشغل كروس لإنتاج أفضل حل.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Optimasi merupakan proses untuk menemukan nilai minimum atau maksimum dari sebuah fungsi, peluang, atau pencarian nilai lainnya dalam berbagai kasus. Optimasi sangat berguna dalam berbagai bidang antara lain pada bidang ekonomi, arsitektur, jaringan komputer, telekomunikasi, transportasi, perdagangan dan lain-lain (Moengin, 2011:1).

Allah berfirman dalam Q.S Al-Furqan ayat 67:

وَالَّذِينَ إِذَا أَنْفَقُوا لَمْ يُسْرِفُوا وَلَمْ يَقْتُرُوا وَكَانَ بَيْنَ ذَلِكَ قَوَامًا ﴿٦٧﴾

Artinya: "Dan orang-orang yang apabila membelanjakan (harta), mereka tidak berlebihan, dan tidak (pula) kikir, dan adalah (pembelanjaan itu) di tengah-tengah antara yang demikian."

Pada ayat di atas dengan jelas disebutkan bahwa apabila manusia atau orang yang beriman ingin membelanjakan sesuatu, maka ketika membelanjakan tersebut dia tidak boleh terlalu boros, dan juga tidak boleh terlalu kikir (Mohan, 2008). Seperti halnya dengan teori optimasi yang tujuannya adalah untuk mendapatkan hasil yang optimal baik itu minimum maupun maksimum.

Secara umum metode optimasi dapat dibagi menjadi dua bagian yaitu optimasi kombinatorial dan pemrograman matematika. Optimasi kombinatorial adalah bagian dari optimasi yang berhubungan dengan riset operasi, teori algoritma, dan teori kompleksitas komputasi. Ini memiliki aplikasi penting dalam beberapa bidang termasuk kecerdasan buatan, matematika, dan rekayasa

perangkat lunak. Beberapa masalah umum yang melibatkan optimasi kombinatorial adalah *Minimum Spanning Tree Problem* dan *Traveling Salesman Problem* (Moengin, 2011:2-3).

Traveling Salesman Problem (TSP) adalah salah satu bentuk masalah optimasi yang kompleks. Pada dasarnya bertujuan untuk menemukan rute dari perjalanan *salesman* yang dimulai dari lokasi (kota) awal, mengunjungi beberapa kota yang telah ditentukan kemudian kembali ke kota asal sedemikian rupa sehingga jarak total yang ditempuh minimum dan setiap kota dikunjungi tepat satu kali (Gutin dan Punnen, 2004:1).

Secara garis besar, TSP dibedakan menjadi dua kelompok yaitu *Symmetric Traveling Salesman Problem* (STSP) dan *Asymmetric Traveling Salesman Problem* (ATSP). Misal $V = \{v_1, v_2, \dots, v_n\}$ adalah suatu himpunan kota dan $A = \{(r, s) : r, s \in V\}$ adalah himpunan pasangan kota (r, s) dalam A maka TSP dikatakan simetris (STSP) jika $d_{rs} = d_{sr}$. STSP adalah permasalahan untuk menemukan rute perjalanan terpendek yang mengunjungi setiap kota tepat satu kali. Sedangkan dikatakan ATSP jika paling tidak ada satu pasangan kota yang memiliki jarak yang berbeda satu sama lain atau disimbolkan $d_{rs} \neq d_{sr}$. Dalam kasus ini, d_{rs} adalah jarak antara r dan s , begitu juga sebaliknya d_{sr} adalah jarak antara s dan r (Davendra, 2010:1-2).

Traveling Salesman Problem dapat diselesaikan dengan banyak metode antara lain *Hill Climbing Method*, *Dynamic Programming*, *Tabu Search*, *Ant Colony System*, *Simmulated Annealing*, Algoritma Genetika dan lain-lain. Saat ini,

metode yang banyak digunakan untuk menyelesaikan TSP adalah Algoritma Genetika karena metode ini sangat efisien dan tergolong masih baru.

Algoritma Genetika pertama kali ditemukan oleh John Holland sekitar tahun 1960-an. Tujuan dari Holland mengembangkan algoritma ini adalah bukan untuk mendesain suatu algoritma yang dapat memecahkan suatu masalah, namun lebih mengarah pada studi mengenai fenomena adaptasi yang terjadi di alam dan mencoba menerapkan mekanisme adaptasi alam tersebut ke dalam sistem komputer (Wati, 2011).

Algoritma Genetika terinspirasi dari teori Charles Darwin, "*Survival of the Fittest*". Di alam, individu dalam suatu populasi bersaing satu sama lain untuk memperebutkan sumber daya yang ada seperti makanan, tempat tinggal dan sebagainya. Pada spesies yang sama pun, individu bersaing menarik pasangan untuk bereproduksi (Sivanandam dan Deepa, 2008:15). Oleh karena itu, individu-individu yang kalah bersaing akan terseleksi dengan sendirinya. Sedangkan individu-individu yang menang akan menghasilkan keturunan yang baik dan kuat. Begitu seterusnya setiap keturunan baru akan menghasilkan keturunan baru yang lebih baik dan kuat.

Algoritma Genetika sangat berbeda dengan metode optimasi klasik. Menurut Sivanandam dan Deepa (2008) beberapa perbedaan antara Algoritma Genetika dengan metode-metode optimasi klasik adalah: (1) menggunakan pengkodean parameter, bukan menggunakan parameter itu sendiri (2) bekerja pada populasi, tidak hanya tunggal (3) hanya menggunakan nilai dari fungsi untuk

optimasi, bukan turunan dari fungsi atau yang lainnya dan (4) menggunakan fungsi transisi.

Saat ini pemecahan masalah menggunakan Algoritma Genetika tidak hanya di bidang sains, namun sudah berkembang ke bidang-bidang lainnya seperti bidang teknik, pertanian, industri, dan lain-lain.

Algoritma Genetika memiliki tiga alat utama yaitu: *selection* (seleksi), *crossover* (kawin silang), dan *mutation* (mutasi). Ketiga alat tersebut berisi metode yang berbeda-beda disebut operator. *Crossover* memiliki peran penting dalam Algoritma Genetika. Banyak sekali operator dalam *crossover* yang telah ditemukan. Yang pertama kali adalah *Partially Matched Crossover* (PMX) atau disebut juga *Partially Mapped Crossover* dan ditemukan oleh Goldberg dan Lingle pada tahun 1985. Operator tersebut untuk pertama kalinya dapat menyelesaikan 33 titik (kota) pada TSP (Ahmed, 2010:97).

Prosedur dalam menggunakan PMX adalah sebagai berikut: pertama tentukan dua posisi pada kromosom dengan aturan acak, *substring* yang berada dalam dua posisi ini dinamakan daerah pemetaan, kemudian tukar *substring* antar induk untuk menghasilkan keturunan, lalu tentukan hubungan pemetaan di antara dua daerah pemetaan, dan yang terakhir tentukan kromosom keturunan mengacu pada hubungan pemetaan (Coley, 1999:63).

Kumar, dkk. (2012) pada jurnalnya yang berjudul *A Comparative Analysis of PMX, CX, and OX Crossover Operators for Solving Travelling Salesman Problem* mengatakan bahwa dari ketiga operator *crossover* yang dicoba yaitu PMX, CX (*Cyclic Crossover*), dan OX (*Ordered Crossover*), PMX merupakan

yang terbaik. Dalam penelitiannya mereka melakukan percobaan dua kali yaitu dengan 25 kota dan 30 kota. Dari kedua percobaan tersebut, jarak minimum yang dihasilkan PMX lebih baik daripada CX dan OX.

Firman Allah SWT dalam Q.S Al-Furqan ayat 62:

وَهُوَ الَّذِي جَعَلَ اللَّيْلَ وَالنَّهَارَ خِلْفَةً لِّمَنۢ أَرَادَ أَن يَذَّكَّرَ أَوْ أَرَادَ شُكُورًا ﴿٦٢﴾

Artinya: “Dan Dia (pula) yang menjadikan malam dan siang silih berganti bagi orang yang ingin mengambil pelajaran atau orang yang ingin bersyukur.”

Pada ayat tersebut disampaikan bahwa ciri-ciri seorang Muslim yang diharapkan adalah pribadi yang menghargai waktu. Seorang Muslim tidak patut menunggu dimotivasi oleh orang lain untuk mengelola waktunya, sebab sudah merupakan kewajiban bagi setiap Muslim. Ajaran Islam menganggap pemahaman terhadap hakikat menghargai waktu sebagai salah satu indikasi keimanan dan bukti ketaqwaan (Gamal, 2012).

Oleh karena itu penulis tertarik untuk membahas dan menganalisis tentang Algoritma Genetika untuk menyelesaikan TSP. Dari paparan di atas, maka penulis ingin menganalisis permasalahan tersebut dalam skripsi ini dengan judul “Optimasi Traveling Salesman Problem Dengan Algoritma Genetika Menggunakan Operator Partially Matched Crossover”.

1.2 Rumusan Masalah

Dari latar belakang di atas, maka dalam skripsi ini difokuskan pada permasalahan tentang bagaimana hasil optimasi pada kasus TSP dengan Algoritma Genetika menggunakan operator PMX?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penulisan skripsi ini adalah untuk mengetahui hasil optimasi kasus TSP dengan Algoritma Genetika menggunakan operator PMX.

1.4 Batasan Masalah

Agar permasalahan yang akan diteliti lebih terfokus, maka penulis membuat beberapa batasan yaitu:

1. Pada proses inialisasi populasi dibangkitkan sebanyak sepuluh kromosom menggunakan Algoritma *Random Generator*.
2. Teknik untuk seleksi menggunakan *Roulette Wheel Selection*.
3. Teknik untuk proses *crossover* menggunakan operator PMX.
4. Teknik untuk proses mutasi menggunakan *Swapping Mutation*.
5. Data yang digunakan adalah data dari jurnal yang berjudul *Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator* oleh Zakir H. Ahmed (2010).
6. Mengacu pada jurnal Zakir H. Ahmed (2010), probabilitas *crossover* yang digunakan adalah 1 (100%) dan probabilitas mutasinya adalah 0,01 (1%).

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan beberapa manfaat antara lain:

1. Dapat memahami konsep dasar mengenai TSP dan Algoritma Genetika.

2. Dapat menyelesaikan TSP dengan Algoritma Genetika, khususnya menggunakan operator PMX.
3. Dapat digunakan sebagai pembanding antara metode-metode yang digunakan dalam menyelesaikan TSP, khususnya dalam membandingkan jenis operator yang digunakan.

1.6 Metode Penelitian

Metode penelitian untuk optimasi TSP menggunakan Algoritma Genetika dengan operator PMX dilakukan langkah-langkah berikut:

1. Mengambil dan mengidentifikasi data jarak pada jurnal yang berjudul *Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator* oleh Zakir H. Ahmed (2010).
2. Melakukan tahapan-tahapan Algoritma Genetika untuk menyelesaikan TSP (mencari rute terpendek). Langkah-langkah Algoritma Genetika sebagai berikut:
 - a. Inisialisasi atau mendefinisikan kromosom. Pada proses ini akan dibangkitkan sebanyak sepuluh kromosom.
 - b. Proses seleksi kromosom. Proses penyeleksian kromosom menggunakan teknik *Roulette Wheel Selection*. Dari proses ini ditentukan kromosom-kromosom yang akan dijadikan *parent*.
 - c. Proses *crossover* dengan operator PMX. Setelah mendapatkan *parent* dalam proses seleksi, selanjutnya *parent-parent* tersebut di-*crossover* untuk mendapatkan keturunan baru (*offspring*) yang lebih baik.

d. Proses mutasi untuk meningkatkan variasi populasi dengan teknik *Swapping Mutation*.

3. Membuat kesimpulan dan interpretasi hasil TSP dengan Algoritma Genetika menggunakan operator PMX.

1.7 Sistematika Penulisan

Penulis membagi skripsi ini dalam empat bab, sebagai berikut:

Bab I Pendahuluan

Memaparkan latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian dan sistematika penulisan.

Bab II Kajian Teori

Memaparkan teori-teori yang mendukung dalam skripsi ini yaitu teori tentang optimasi; *Traveling Salesman Problem* (TSP); Algoritma Genetika; proses-proses dalam Algoritma Genetika; serta kajian optimasi dan genetika (keturunan) dalam Al-Qur'an.

Bab III Pembahasan

Menganalisis dan membahas bagaimana penyelesaian TSP menggunakan Algoritma Genetika dengan operator PMX.

Bab IV Penutup

Memaparkan hasil dari pembahasan berupa kesimpulan dan saran.

BAB II

KAJIAN TEORI

2.1 Optimasi dengan Algoritma Genetika

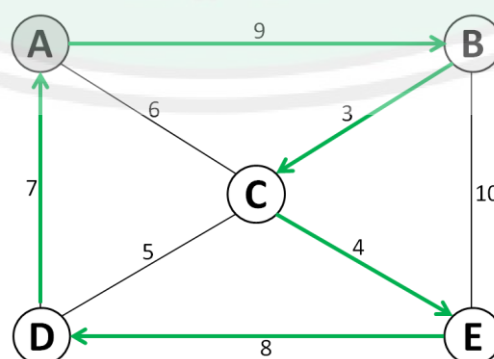
Optimasi (*optimization*) adalah aktivitas untuk mendapatkan hasil terbaik di bawah keadaan yang diberikan. Tujuan akhir dari semua aktivitas tersebut adalah meminimumkan usaha (*effort*) atau memaksimalkan manfaat yang diinginkan. Karena usaha yang diperlukan atau manfaat (*benefit*) yang diinginkan dapat dinyatakan sebagai fungsi dari variabel keputusan, maka optimasi dapat didefinisikan sebagai proses untuk menemukan kondisi yang memberikan nilai minimum atau maksimum dari sebuah fungsi, peluang, maupun pencarian nilai lainnya dalam berbagai kasus (Moengin, 2011:1).

Banyak metode optimasi yang telah dikembangkan untuk menyelesaikan masalah-masalah optimasi. Metode optimasi secara umum dapat dibagi menjadi dua bagian yaitu optimasi kombinatorial dan pemrograman matematika. Optimasi kombinatorial adalah topik dalam ilmu komputer teoritis dan matematika terapan yang berfungsi untuk mencari solusi dengan biaya yang terkecil sedangkan teknik pemrograman matematika adalah metode pencarian titik optimum dan keduanya menjadi bagian dari riset operasi. Ini memiliki aplikasi penting dalam beberapa bidang termasuk kecerdasan buatan, matematika, dan rekayasa perangkat lunak. Beberapa masalah umum yang melibatkan optimasi kombinatorial adalah *Minimum Spanning Tree Problem* dan *Traveling Salesman Problem* (Moengin, 2011:2-3)

2.2 Traveling Salesman Problem

Traveling Salesman Problem (TSP) adalah suatu permasalahan untuk menemukan lintasan dari seorang *salesman* yang berawal dari sebuah lokasi asal, mengunjungi sebuah himpunan kota dan kembali lagi ke lokasi asal yang mana total dari jarak yang ditempuh adalah minimum dan setiap kota dilewati tepat hanya satu kali (Gutin dan Punnen, 2004:1).

TSP termasuk sebagai masalah optimasi kombinatorial. Dalam bentuk yang biasa dari TSP, sebuah peta dengan beberapa kota di dalamnya diberikan kepada seorang *salesman* yang harus mengunjungi semua kota tepat satu kali dan membuat sebuah rute dengan jarak yang paling pendek diantara semua kemungkinan rute yang ada. Data yang digunakan terdiri dari graf komplit berbobot, dan tujuannya adalah untuk menemukan sebuah lintasan Hamilton, yaitu lintasan yang melewati seluruh titik dari graf tersebut dengan total bobot yang paling minimum. Dalam konteks TSP, lintasan Hamilton biasa disebut perjalanan atau rute. Sebagai contoh, diberikan peta seperti pada Gambar 2.1, rute terpendeknya adalah (A,B,C,E,D,A) dengan total jarak 31 (Greco, 2008:1).



Gambar 2.1: Rute $A \rightarrow B \rightarrow C \rightarrow E \rightarrow D \rightarrow A$ sebagai Rute Terpendek.

Sejarah TSP bisa ditelusur misalnya dari Euler yang mempelajari *Knight Tour's Problem* (1759), Kirkman yang mempelajari grafik polihedron (1856) maupun Hamilton yang membuat game Icosian (1856) yang bertujuan mencari jalur sirkuit berbasis grafik polyhedron yang memenuhi kondisi tertentu. Istilah TSP sendiri berasal dari buku yang diterbitkan oleh seorang veteran *salesman* sekitar tahun 1930-an di Jerman, meski dalam buku ini masalah TSP lebih dibahas dari aspek bisnis dan belum diformulasikan secara matematis (Setiawan, 2010).

42	57	44	9	40	21	46	7
55	10	41	58	45	8	39	20
12	43	56	61	22	59	6	47
63	54	11	30	25	28	19	38
32	13	62	27	60	23	48	5
53	64	31	24	29	26	37	18
14	33	2	51	16	35	4	49
1	52	15	34	3	50	17	36

Gambar 2.2: *Knight Tour's Problem* oleh Euler

Secara umum, TSP dibedakan menjadi dua bentuk yaitu *Symmetric TSP* (TSP Simetris) dan *Asymmetric TSP* (TSP Asimetris). Pada bentuk simetris yang biasanya dikenal sebagai STSP, hanya ada satu jalan antara dua kota sehingga jarak antara kota A ke kota B sama dengan jarak antara kota B ke kota A atau dapat ditulis $d_{AB} = d_{BA}$, dengan d_{AB} adalah jarak dari A ke B dan d_{BA} adalah jarak dari B ke A. Sedangkan pada ATSP (TSP Asimetris) memungkinkan adanya dua jalan atau jarak antara dua kota. Sehingga jarak dari kota A ke kota B tidak sama dengan jarak dari kota B ke kota A atau dapat ditulis $d_{AB} \neq d_{BA}$. Oleh karena

itu, jumlah kemungkinan rute pada ATSP dan STSP dengan n kota berturut-turut adalah $(n-1)!$ dan $\frac{(n-1)!}{2}$ (Greco, 2008:1-2).

Ada beberapa pendekatan dalam menyelesaikan TSP. Dewasa ini, menyelesaikan kasus TSP merupakan hal yang menarik. Hampir setiap ada pendekatan baru untuk menyelesaikan masalah optimasi, selalu diaplikasikan untuk menyelesaikan TSP. Teknik pertama untuk menyelesaikan TSP adalah dengan metode-metode klasik. Metode-metode tersebut terdiri dari metode eksak dan heuristik. Metode heuristik seperti *Cutting Planes* dan *Branch and Bound* hanya dapat menyelesaikan masalah-masalah kecil. Sedangkan metode *Markov Chain*, *Simulated Annealing* dan *Tabu Search* bagus untuk masalah yang besar. Di samping itu, beberapa algoritma yang berdasar pada prinsip *Greedy* seperti *Nearest Neighbor* dan *Spanning Tree* dapat menjadi metode yang efisien (Greco, 2008:2).

2.2.1 Aplikasi TSP

TSP yang pada awalnya muncul sebagai bagian masalah logistik dan transportasi ini telah berkembang dan dimanfaatkan di berbagai sektor (Setiawan, 2010). Berikut disajikan sejumlah contoh aplikasinya di beberapa bidang:

1. Logistik dan *Supply Chain*

Logistik dan *Supply Chain* secara umum mengandung pengertian sebagai suatu sistem untuk mengelola orang, sumberdaya, aktivitas, teknologi dan sebagainya yang dibutuhkan guna mengantarkan barang dan jasa dari titik asal (pemasok/*supplier*) ke titik tujuan (pelanggan) sehingga tepat waktu, tepat

jenis, tepat jumlah, tepat mutu, tepat lokasi, tepat sumber daya dan sebagainya. Pemakaian TSP dalam logistik antara lain sangat vital bagi perusahaan pengiriman barang dan jasa, industri *travel* dan biro perjalanan atau divisi yang bertanggung jawab atas pasokan, distribusi dan pemasaran bahan baku dan bahan jadi suatu perusahaan. Masalahnya bukan hanya menentukan rute terpendek, tapi juga menentukan jenis barang tertentu yang bisa disimpan di beberapa gudang tertentu berbasis *konstrain* tertentu (misal kapasitas persediaan, jam buka dan tutup, dan sebagainya). Untuk prosedur pengiriman barang yang rutin, pemilihan rute yang tepat akan menghemat biaya dalam jumlah besar dalam waktu yang lama.

2. Transportasi

Salah satu pionir TSP dalam bidang transportasi adalah Merrill Flood, seorang matematikawan Amerika, yang merumuskan rute bis sekolah di New Jersey sekitar tahun 1940an. Flood yang pernah menjabat sebagai wakil presiden *Institute of Industrial Engineer* tahun 1962-1965 ini juga mengembangkan pendekatan analisis sistem yang inovatif dan *benefit-cost analysis* di bidang kebijakan publik. Sekitar tahun yang sama, penggunaan TSP untuk memecahkan masalah transportasi peralatan pertanian (yang dipakai untuk menguji tanah) dikaji di tempat yang berbeda oleh Mahalanobis (di Bengal) dan Jessen (di Iowa). Salah satu modifikasi TSP yang sudah “*advanced*” diteliti oleh Zheng, Zhang dan Liu yang menggunakan *Ant Colony Algorithm* (yang masih relatif baru) untuk mendapatkan solusi optimum guna menentukan interval keberangkatan bis kota dengan

mengakomodasi faktor kepadatan penumpang. Lebih jauh, implementasi TSP juga telah dikaji untuk diterapkan di *Google Map*.

3. Manufaktur

Pemanfaatan TSP di bidang manufaktur misalnya dalam menentukan jalur penge-drill-an untuk membuat lubang pada *printed circuit board*. Untuk produksi secara masal, pemakaian jalur yang lebih pendek dapat meningkatkan kecepatan sekaligus menurunkan biaya produksi secara signifikan, dan TSP merupakan perangkat yang baik untuk mencapai efisiensi produksi yang maksimum.

4. Pengurutan *Gene Chromosome (Genome)*, termasuk DNA

Peneliti di bidang ini, misalnya di *National Institute of Health*, Amerika, menggunakan bantuan *Concorde's TSP solver* untuk menyusun peta radiasi hibrida sebagai upaya pengurutan genom. TSP menyediakan mekanisme untuk mengintegrasikan peta-peta lokal (“kota”) ke dalam sebuah peta radiasi hibrid tunggal, di mana “jarak” dinyatakan sebagai ukuran kecenderungan sebuah peta lokal akan bergabung dengan peta lokal yang lain. Penggunaan TSP juga diadopsi oleh sebuah grup penelitian di Prancis untuk membangun peta genom tikus. Sementara grup penelitian lainnya menggunakan *Concorde* untuk mengkalkulasi urutan DNA di sebuah proyek penelitian *Genetic Engineering*.

5. Bidang lainnya

Beberapa contoh lain penerapan TSP misalnya penjadwalan pengiriman koran atau pos, pengambilan uang koin di sejumlah telepon umum,

pemasangan jaringan telekomunikasi, penentuan urutan bintang oleh satelit interferometer dan sebagainya.

2.3 Algoritma Genetika

Algoritma Genetika terinspirasi dari prinsip-prinsip genetika dan seleksi alam (teori evolusi Darwin) yang pertama kali ditemukan di Universitas Michigan, Amerika Serikat, oleh John Holland pada tahun 1975 melalui sebuah penelitian dan kemudian dipopulerkan oleh salah satu muridnya, David Golberg. Konsep dasar dari Algoritma Genetika sebenarnya dirancang untuk mensimulasikan proses-proses dalam sistem alam yang diperlukan untuk evolusi, khususnya teori evolusi alam yang dicetuskan oleh Charles Darwin, yaitu "*Survival of the Fittest*". Menurut teori ini, dalam suatu populasi, makhluk yang kuat akan mendominasi makhluk yang lebih lemah dalam persaingan mereka untuk memperebutkan sumber daya alam yang langka (Sutojo, dkk., 2011:403).

Menurut Sutojo, dkk (2011) Algoritma Genetika adalah teknik pencarian heuristik yang didasarkan pada gagasan evolusi seleksi alam dan genetika. Algoritma ini memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Dalam proses evolusi, individu secara terus menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. "Hanya individu yang kuat yang mampu bertahan". Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Proses perkembangbiakan ini didasarkan pada analogi struktur genetika dan

perilaku kromosom dalam populasi individu dengan menggunakan dasar sebagai berikut:

1. Individu dalam populasi bersaing untuk sumber daya alam dan pasangannya.
2. Mereka yang paling sukses di setiap kompetisi akan menghasilkan keturunan yang lebih baik daripada individu-individu yang berkinerja buruk.
3. Gen dari individu yang baik akan menyebar ke seluruh populasi sehingga dua orang tua (*parent*) yang baik kadang-kadang akan menghasilkan keturunan yang lebih baik dari orangtuanya.
4. Setiap ada pergantian generasi terbaru ini biasanya lebih cocok dengan lingkungan mereka. Dengan kata lain, generasi baru ini bisa menyesuaikan dengan keadaan lingkungannya.

Ciri-ciri permasalahan yang membutuhkan Algoritma Genetika menurut Sutojo, dkk (2011) antara lain:

1. Ruang pencarian sangat besar, kompleks atau kurang dipahami.
2. Tidak ada pengetahuan yang memadai untuk menyederhanakan ruang pencarian yang sangat besar menjadi ruang pencarian yang lebih sempit.
3. Tidak ada analisis matematis yang bias menangani ketika metode konvensional gagal menyelesaikan masalah yang dihadapi.
4. Solusi yang dihasilkan tidak harus optimal, asal sudah memenuhi kriteria sudah bisa diterima.

5. Membutuhkan solusi “*real-time*”, yaitu solusi yang bisa didapatkan dengan cepat sehingga dapat diimplementasikan untuk permasalahan yang mempunyai perubahan yang cepat.

2.3.1 Istilah-istilah dalam Algoritma Genetika

Sutojo, dkk (2011) memaparkan istilah atau definisi yang digunakan dalam Algoritma Genetika yang perlu diketahui sebagai berikut:

1. Gen (*genotype*)

Gen adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Dalam Algoritma Genetika, gen bisa bernilai biner, *float*, *integer*, maupun karakter (*string*).

2. *Allele*

Allele adalah nilai dari suatu gen, bisa berupa biner, *float*, *integer*, maupun karakter (*string*).

3. Kromosom

Kromosom adalah gabungan dari gen-gen yang membentuk arti tertentu. Ada beberapa macam bentuk kromosom, yaitu kromosom biner, kromosom *float*, kromosom *string*, kromosom kombinatorial. Dalam TSP, yang digunakan adalah kromosom kombinatorial, yaitu kromosom yang disusun dari gen-gen yang dinilai berdasarkan urutannya.

4. Individu

Individu adalah kumpulan gen, bisa dikatakan sama dengan kromosom. Individu menyatakan salah satu kemungkinan solusi dari suatu permasalahan. Dalam TSP individu menyatakan jalur yang ditempuh.

5. Populasi

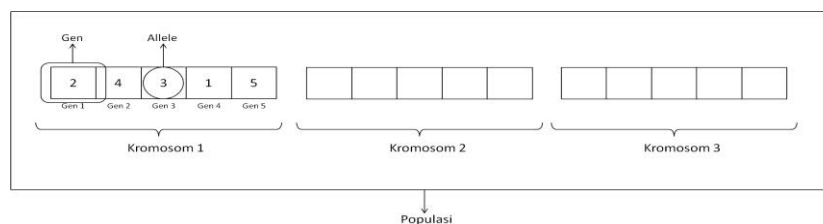
Populasi merupakan sekumpulan individu atau kromosom yang akan diproses secara bersama-sama dalam satu siklus proses evolusi.

6. Generasi

Generasi menyatakan satu satuan siklus proses evolusi. Satu siklus proses evolusi dimulai dari pembangkitan populasi awal hingga proses mutasi selesai.

7. Nilai *fitness*.

Nilai *fitness* merupakan sebuah nilai yang menyatakan seberapa baik nilai dari suatu individu. Nilai inilah yang dijadikan acuan untuk mencapai nilai optimal. Algoritma Genetika bertujuan untuk mencari individu yang mempunyai nilai *fitness* yang paling optimal. Dalam TSP, karena TSP bertujuan untuk meminimumkan jarak, maka nilai *fitness*-nya adalah inversi dari jarak.



Gambar 2.3: Visualisasi Gen, *Allele*, Kromosom, dan Populasi pada Algoritma Genetika

2.3.2 Parameter dalam Algoritma Genetika

Makna parameter di sini adalah parameter kontrol algoritma genetika, yaitu ukuran populasi, probabilitas *crossover* (pc), dan probabilitas mutasi (pm). Nilai parameter ini ditentukan juga berdasarkan permasalahan yang akan dipecahkan.

Probabilitas *crossover* menyatakan seberapa sering proses *crossover* akan terjadi di antara dua kromosom *parent*. Jika tidak terjadi *crossover*, keturunan (*offspring*) merupakan salinan mutlak dari *parent*. Jika terjadi *crossover*, keturunan dibuat dari bagian-bagian kromosom *parent*. Jika probabilitas *crossover* 100%, maka keseluruhan keturunan dibuat dengan *crossover*. Jika probabilitas *crossover* 0%, maka seluruh generasi baru dibuat dari salinan kromosom-kromosom dari populasi lama, tetapi ini tidak berarti bahwa generasi baru sama dengan yang lama karena adanya penekanan selektif. Meskipun *crossover* bertujuan untuk mendapatkan kromosom yang memiliki bagian baik dari *parent* atau bahkan menjadi lebih baik dari *parent*-nya, ada baiknya juga jika membiarkan beberapa bagian dari populasi untuk bertahan ke generasi berikutnya. Probabilitas *crossover* sebaiknya cukup tinggi, yaitu antara 80% sampai 95% untuk memberikan hasil yang baik (Saputro dan Yento, 2004: 64).

Sebuah individu yang mengarah pada solusi optimal akan diperoleh melalui proses *crossover* dengan catatan bahwa *crossover* hanya bisa dilakukan jika sebuah bilangan acak r dalam interval $[0,1]$ yang dibangkitkan nilainya lebih kecil dari parameter probabilitas *crossover* tertentu pc , dengan kata lain $r < pc$ (Sutojo, dkk., 2011:418).

Probabilitas mutasi menyatakan seberapa sering bagian-bagian kromosom akan dimutasikan. Jika tidak ada mutasi, keturunan diambil atau disalin langsung setelah *crossover* tanpa perubahan. Jika mutasi dilakukan, bagian-bagian kromosom diubah. Jika probabilitas mutasi 100%, semua kromosom diubah. Jika probabilitas mutasi 0%, tidak ada yang diubah. Probabilitas mutasi dalam algoritma genetika seharusnya diberi nilai yang kecil yaitu antara 0,5% sampai 1% (Saputro dan Yento, 2004: 64).

2.3.3 Proses Pengkodean (*Encoding*)

Proses pengkodean atau *encoding* adalah salah satu proses yang sulit dalam Algoritma Genetika. Hal ini disebabkan karena proses *encoding* untuk setiap permasalahan berbeda karena tidak semua teknik *encoding* cocok untuk setiap permasalahan. Proses *encoding* menghasilkan *string* yang kemudian disebut kromosom. *String* terdiri dari sekumpulan bit yang dikenal sebagai gen (Lukas, dkk., 2005: 1-2).

Ada bermacam-macam teknik *encoding* yang dapat dilakukan dalam Algoritma Genetika antara lain: *binary encoding*, *permutation encoding*, *value encoding*, dan *tree encoding*. Teknik *encoding* yang digunakan pada TSP adalah *permutation encoding*. Pada *permutation encoding*, kromosom-kromosom adalah kumpulan angka yang mewakili posisi dalam sebuah rangkaian. Pada TSP, kromosom mewakili urutan kota yang dikunjungi *salesman*. Jadi apabila ada satu kromosom berbentuk sebagai berikut $P_1 = (X_1, X_2, X_3, \dots, X_n)$ berarti *salesman*

bergerak dari kota bernomor X_1 ke X_2 dan seterusnya hingga kota ke- X_n dan kembali lagi ke X_1 (Lukas, dkk., 2005:2).

Kromosom A	2	4	7	8	1	3	5	6	9
Kromosom B	5	1	7	6	2	3	9	4	8

Gambar 2.4: *Permutation Encoding* pada TSP

Gambar 2.4 di atas menjelaskan bahwa kromosom A mempunyai lintasan perjalanan yang dimulai dari 2 dan berakhir pula di 2. Demikian pula untuk kromosom B yang berawal dari 5 dan berakhir pula di 5. Atau dapat ditulis sebagai berikut:

$$\text{Kromosom A} = (2,4,7,8,1,3,5,6,9)$$

$$\text{Kromosom B} = (5,1,7,6,2,3,9,4,8)$$

2.3.4 Proses Pembangkitan Populasi (Inisialisasi)

Membangkitkan populasi awal adalah proses membangkitkan sejumlah kromosom secara acak atau melalui prosedur tertentu. Ukuran untuk populasi tergantung pada masalah yang akan diselesaikan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian dilakukan pembangkitan populasi awal. Syarat-syarat yang harus dipenuhi untuk menunjukkan suatu solusi harus benar-benar diperhatikan dalam pembangkitan setiap kromosomnya (Yulyantari, 2011).

Teknik dalam pembangkitan populasi awal ada beberapa cara, diantaranya adalah Algoritma *Random Generator*, pendekatan tertentu, dan permutasi gen. Semua teknik tersebut dapat diberlakukan pada kasus TSP termasuk Algoritma *Random Generator*.

Inti dari Algoritma *Random Generator* adalah melibatkan pembangkitan bilangan random untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan. Contoh penggunaan Algoritma *Random Generator* dalam kasus TSP menurut Yulyantari (2011) adalah sebagai berikut:

Sebuah kromosom dengan sembilan kota direpresentasikan sebagai berikut:

$$\text{Kromosom } A = (X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9)$$

Selanjutnya dibangkitkan bilangan acak sebanyak jumlah kota (gen) dalam kromosom. Misal didapatkan bilangan acak tersebut sebagai berikut:

$$[0.23 \ 0.82 \ 0.45 \ 0.74 \ 0.87 \ 0.11 \ 0.56 \ 0.69 \ 0.78]$$

dimana posisi i dalam *list* menunjukkan kota ke- i . Nilai acak dalam posisi i menentukan urutan didatanginya kota i dalam lintasan TSP. Dengan nilai-nilai acak di atas, dapat dilihat bahwa nilai 0.11 adalah nilai yang paling kecil, sehingga kota ke-6 menempati urutan pertama. Kemudian 0.23 adalah nilai acak terkecil kedua, sehingga kota ke-1 menempati urutan kedua dan seterusnya. Sehingga dengan demikian, dari nilai-nilai acak di atas dapat ditentukan lintasan yang terjadi adalah:

$$\text{Kromosom } A = (X_6, X_1, X_3, X_7, X_8, X_4, X_9, X_5, X_2)$$

2.3.5 Proses Seleksi

Proses seleksi adalah proses yang memegang peranan penting dalam Algoritma Genetika. Proses seleksi ini digunakan agar hanya kromosom-kromosom yang berkualitas yang dapat melanjutkan peranannya dalam proses

Algoritma Genetika berikutnya. Ada bermacam-macam teknik untuk melakukan proses seleksi pada suatu permasalahan tergantung pada permasalahan apa yang akan diselesaikan. Di antara teknik-teknik untuk seleksi adalah *Roulette Wheel Selection*, *Rank Base Selection*, dan *Steady State Selection* (Lukas, dkk., 2005:2).

Teknik yang digunakan pada proses seleksi ini adalah teknik *Roulette Wheel Selection*. Teknik seleksi ini merupakan yang paling sederhana. Teknik ini juga biasa disebut *Stochastic Sampling with Replacement*.

Langkah pertama yang dilakukan dalam seleksi adalah pencarian nilai *fitness*. Nilai *fitness* ini yang nantinya akan digunakan pada tahap-tahap seleksi berikutnya. Algoritma Genetika lebih banyak digunakan untuk masalah maksimasi. Untuk masalah maksimasi, fungsi *fitness* adalah sama dengan fungsi objektifnya. Tetapi untuk masalah minimasi, fungsi *fitness* didefinisikan sebagai $F(x) = \frac{1}{f(x)}$, dimana $f(x)$ adalah fungsi objektif. Karena TSP termasuk masalah minimasi, maka digunakan fungsi *fitness* tersebut dimana $f(x)$ merupakan biaya (jarak) dari rute yang direpresentasikan oleh sebuah kromosom (Ahmed, 2010:98).

Tahap-tahap perhitungan *fitness* adalah sebagai berikut:

1. Mencari jarak tempuh tiap kromosom (mZ_i), $\forall i = 1, 2, 3, \dots$
2. Mencari nilai *fitness* tiap kromosom (f_i)

$$f_i = \frac{1}{mZ_i}, \quad \forall i = 1, 2, 3, \dots$$

3. Mencari total *fitness* $\left(\sum_{i=1}^n f_i \right)$

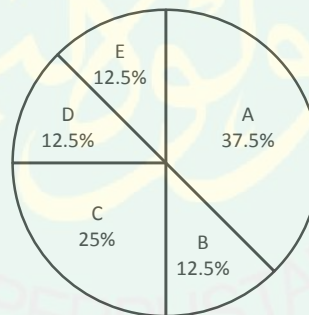
4. Mencari probabilitas masing-masing kromosom (p_i)

$$p_i = \frac{f_i}{\sum_{i=1}^n f_i}, \quad \forall i = 1, 2, 3, \dots$$

5. Mencari probabilitas kumulatif tiap kromosom (q_i)

$$q_i = \sum_{k=1}^i p_k, \quad \forall i = 1, 2, 3, \dots$$

Selanjutnya pemilihan sebuah rute yang menghasilkan populasi berikutnya dilakukan dengan cara mengambil sebanyak n -buah bilangan acak r dengan $0 < r < 1$ dan membandingkan bilangan acak tersebut dengan probabilitas kumulatif *fitness* tiap rute (Sarwadi, 2004:4).



Gambar 2.5: Contoh Probabilitas Terpilihnya Suatu Kromosom dalam Roda *Roulette*

Untuk mempertahankan jumlah kromosom tetap pada satu generasi maka perlu dibangkitkan kromosom baru yang merupakan hasil penyilangan dari kromosom yang hidup. Untuk itu selanjutnya dilakukan proses *crossover*.

2.3.6 Proses *Crossover*

Proses kawin silang atau yang lebih dikenal dengan nama proses *crossover* (disebut juga pindah silang atau rekombinasi) adalah menyilangkan dua kromosom sehingga membentuk kromosom baru yang harapannya lebih baik dari pada *parent*-nya. Tidak semua kromosom pada suatu populasi akan mengalami proses *crossover* didasarkan pada probabilitas *crossover* yang telah ditentukan terlebih dahulu. Ada beberapa teknik *crossover* yang dapat digunakan untuk menyelesaikan TSP. Salah satunya adalah PMX (Lukas, dkk., 2005:2).

Prosedur dalam menggunakan PMX adalah sebagai berikut: pertama tentukan dua posisi pada kromosom dengan aturan acak, *substring* yang berada dalam dua posisi ini dinamakan daerah pemetaan, kemudian tukar *substring* antar *parent* untuk menghasilkan keturunan, lalu tentukan hubungan pemetaan di antara dua daerah pemetaan, dan yang terakhir tentukan kromosom keturunan mengacu pada hubungan pemetaan (Coley, 1999:63).

Contoh proses *Partially Matched Crossover* adalah sebagai berikut:

1. Sejajarkan dua *parent* yang telah didapatkan dari proses seleksi. Misalnya

$$Parent\ 1 = (1, 5, 7, 3, 6, 4, 2, 1) \text{ dan } Parent\ 2 = (7, 6, 2, 4, 3, 5, 1, 7).$$

<i>Parent 1</i>	1	5	7	3	6	4	2	1
-----------------	---	---	---	---	---	---	---	---

<i>Parent 2</i>	7	6	2	4	3	5	1	7
-----------------	---	---	---	---	---	---	---	---

2. Pilih secara acak *substring* dari kedua *parent*.

<i>Parent 1</i>	1	5	7	3	6	4	2	1
-----------------	---	---	---	---	---	---	---	---

<i>Parent 2</i>	7	6	2	4	3	5	1	7
-----------------	---	---	---	---	---	---	---	---

Bagian yang berwarna biru adalah *substring* masing-masing *parent* yang berfungsi sebagai daerah pemetaan.

3. Tukar posisi *substring* pada kedua *parent*. Sehingga *substring* pada *Parent 1* berganti menjadi *substring Parent 2* dan sebaliknya.

<i>Offspring 1</i>			2	4	3			
			↓	↓	↓			
<i>Offspring 2</i>			7	3	6			

4. Gambarkan pemetaan tiap gen pada *substring* antara *Parent 1* dan *Parent 2*. Sehingga pemetaan yang terjadi adalah: $2 \leftrightarrow 7$, $4 \leftrightarrow 3$, dan $3 \leftrightarrow 6$. Tanda panah menunjukkan pemetaan yang terjadi, misalnya $2 \leftrightarrow 7$ maka 2 dapat dipetakan ke 7 dan sebaliknya 7 juga dapat dipetakan ke 2.
5. Isikan bagian-bagian gen yang kosong sesuai dengan pemetaan dari kedua *substring*. Gen yang tidak memiliki pemetaan tidak perlu diubah.

<i>Offspring 1</i>	1	5	2	4	3	6	7	1
<i>Offspring 2</i>	2	4	7	3	6	5	1	7

6. Kesimpulannya, anak yang dihasilkan dari *crossover* antara *Parent 1* dan *Parent 2* di atas adalah:

$$\text{Offspring 1} = (1, 5, 2, 4, 3, 6, 7, 1)$$

$$\text{Offspring 2} = (2, 4, 7, 3, 6, 5, 1, 7)$$

2.3.7 Proses Mutasi

Proses mutasi dilakukan setelah proses *crossover* dengan cara memilih kromosom yang akan dimutasi secara acak kemudian menentukan titik mutasi

pada kromosom tersebut secara acak pula. Banyaknya kromosom yang akan mengalami mutasi dihitung berdasarkan probabilitas mutasi yang telah ditentukan terlebih dahulu.

Ada bermacam-macam teknik mutasi yang dapat digunakan untuk menyelesaikan suatu masalah dengan Algoritma Genetika. Seperti pada teknik *crossover*, teknik mutasi juga dirancang untuk digunakan pada suatu masalah yang spesifik sehingga tidak setiap teknik mutasi dapat diterapkan pada suatu masalah yang akan diselesaikan. Selain itu, teknik mutasi yang digunakan juga harus sesuai dengan teknik *encoding* yang digunakan untuk menyelesaikan permasalahan tersebut (Lukas, dkk., 2005:2-3).

Beberapa teknik mutasi yang dapat digunakan dalam penyelesaian TSP adalah *Swapping Mutation*, *Inversion Mutation*, *Insertion Mutation*, dan *Reciprocal Mutation*. Teknik mutasi yang digunakan dalam skripsi ini adalah *Swapping Mutation*.

Pada teknik *Swapping Mutation*, proses mutasi dilakukan dengan cara menukar gen yang dipilih secara acak dengan gen sesudahnya. Jika gen tersebut berada di akhir kromosom, maka ditukar dengan gen yang pertama.

Langkah pertama yang harus dilakukan adalah menghitung panjang total gen yang ada dalam satu populasi (Fitrah, 2006:4).

$$\text{panjang total gen} = \text{jumlah gen dalam satu kromosom} \times \text{jumlah kromosom}$$

Misalkan ada enam kromosom sebagai berikut:

$$\text{Kromosom 1} = (D, B, E, C)$$

$$\text{Kromosom 2} = (B, D, C, E)$$

$$\text{Kromosom 3} = (C, D, E, B)$$

$$\text{Kromosom 4} = (E, C, B, D)$$

$$\text{Kromosom 5} = (E, B, C, D)$$

$$\text{Kromosom 6} = (C, D, B, E)$$

Maka panjang total gen adalah $4 \times 6 = 24$.

Untuk memilih posisi gen yang mengalami mutasi dilakukan dengan membangkitkan bilangan bulat acak antara 1 sampai panjang total gen yaitu 24. Misalkan ditentukan probabilitas mutasi sebesar 0,2 atau 20%, maka jumlah gen yang akan dimutasi adalah $20\% \times 24 = 4,8 \approx 5$ (Fitrah, 2006:4).

Langkah selanjutnya adalah membangkitkan bilangan bulat secara acak antara 1 sampai 24. Misalkan didapatkan bilangan bulat acak tersebut adalah 3,7,10,20,24. Maka proses mutasinya adalah sebagai berikut:

1. Gen ke-3 ditukar dengan gen sesudahnya yaitu gen ke-4
2. Gen ke-7 ditukar dengan gen ke-8
3. Gen ke- 10 ditukar dengan gen ke-11
4. Gen ke-20 ditukar dengan gen ke-17, dan
5. Gen ke 24 ditukar dengan gen ke-21

Sehingga kromosom-kromosom yang telah dimutasi berubah menjadi:

Kromosom 1 = (D, B, C, E)

Kromosom 2 = (B, D, E, C)

Kromosom 3 = (C, E, D, B)

Kromosom 4 = (E, C, B, D)

Kromosom 5 = (D, B, C, E)

Kromosom 6 = (E, D, B, C)

Setelah semua proses Algoritma Genetika dilakukan mulai dari inisialisasi populasi hingga proses terakhir yaitu mutasi. Maka proses Algoritma Genetika untuk satu generasi telah selesai. Kriteria berhenti yaitu apabila setelah dalam beberapa generasi berturut-turut diperoleh nilai *fitness* yang terendah tidak berubah (Fitrah, 2006:4).

2.4 Kajian Optimasi dalam Al-Qur'an

Allah berfirman dalam Q.S Al-Furqan ayat 67:

وَالَّذِينَ إِذَا أَنْفَقُوا لَمْ يُسْرِفُوا وَلَمْ يَقْتُرُوا وَكَانَ بَيْنَ ذَلِكَ قَوَامًا

Artinya: “Dan orang-orang yang apabila membelanjakan (harta), mereka tidak berlebihan, dan tidak (pula) kikir, dan adalah (pembelanjaan itu) di tengah-tengah antara yang demikian.”

Pada ayat di atas dengan jelas disebutkan bahwa apabila manusia atau orang yang beriman yang ingin membelanjakan sesuatu, maka ketika membelanjakan tersebut dia tidak boleh terlalu boros, dan juga tidak boleh terlalu kikir (Mohan, 2008). Seperti halnya dengan teori optimasi yang tujuannya adalah untuk mendapatkan hasil yang optimal baik itu minimum maupun maksimum.

Menurut tafsir Al-Maraghi, kata *Al-Israf* berarti melampaui batas dalam mengeluarkan nafkah karena melihat saingannya yang mempunyai harta. *At-*

Taqtir berarti kikir. *Qawaman* berarti pertengahan atau sederhana. Sedangkan maksud dari ayat di atas adalah bahwa agar orang-orang tidak berlaku *mubadzir* dalam mengeluarkan nafkah, maka tidak tidak mengeluarkannya lebih dari kebutuhan, tidak pula kikir terhadap diri mereka dan keluarga mereka, sehingga mengabaikan kewajiban terhadap mereka, tetapi mereka mengeluarkannya secara adil dan pertengahan, dan sebaik-baik perkara adalah yang paling pertengahan (Al-Maraghi, 1993:63-71).

Ayat lain yang mengajarkan untuk tidak berlebih-lebihan salah satunya adalah penggalan Q.S Al-A'raf ayat 31 sebagai berikut:

﴿... إِنَّهُ لَا يُحِبُّ الْمُسْرِفِينَ﴾

Artinya: "...Sesungguhnya Allah tidak menyukai orang-orang yang berlebih-lebihan."

Ayat Al-Qur'an di atas menjelaskan sebuah pelajaran yang sangat penting. Secara sangat bijak dan dalam bentuk perintah, Allah mengajarkan hidup sederhana sekaligus secara tegas melarang hidup berlebihan. Allah meminta manusia untuk bersyukur dan berterima kasih atas berbagai karunia yang diberikan kepada manusia. Hidup boros dan berlebihan merupakan tindakan yang berlawanan dengan rasa syukur. Sedangkan hidup hemat dan sederhana adalah wujud penghormatan atasNya. Hidup hemat dan sederhana adalah wujud rasa syukur yang bersifat *maknawiyah*. Itu merupakan bentuk penghormatan terhadap rahmat Allah yang tersimpan dalam karunia dan kebaikanNya, penyebab keberkahan dan ditambahkannya nikmat, sumber kesehatan jasmani layaknya kenikmatan yang tersembunyi dalam karunia yang tampaknya tidak nikmat.

Karena hidup boros dan berlebihan berlawanan dengan hikmah-hikmah di atas, maka hal tersebut memberikan dampak-dampak yang sebaliknya (Nursi, 2011).

Menurut Tafsir Al-Maraghi, berlebih-lebihan artinya melampaui batas. Adapun garis-garis batasnya antara lain: (1) Batas *thabi'i* atau naluri, seperti lapar, kenyang, haus dan hilangnya dahaga. (2) Batas ekonomis, yaitu apabila pembelanjaan seseorang menurut ukuran tertentu dari pemasukannya. Yakni ukuran yang tidak menghabiskan seluruh hasil usahanya. (3) Batas *syara'*, karena pemberi *syara'* telah mengharamkan beberapa jenis makanan dan minuman. Bagi wanita diharamkan minum pada bejana-bejana yang dibuat dari emas dan perak karena hal itu dianggap berlebih-lebihan yang terlarang (Al-Maraghi, 1993:237).

2.5 Kajian Genetika (Keturunan) dalam Al-Qur'an

Salah satu ayat dalam Al-Qur'an yang membahas mengenai genetika (keturunan) adalah pada Q.S Ali-Imron ayat 33-34 yang berbunyi:

﴿ إِنَّ اللَّهَ اصْطَفَىٰ آدَمَ وَنُوحًا وَآلَ إِبْرَاهِيمَ وَآلَ عِمْرَانَ عَلَى الْعَالَمِينَ ﴿٣٣﴾ ذُرِّيَّةً
عَصُوبًا مِنْ غَضٍّ ۗ وَاللَّهُ سَمِيعٌ عَلِيمٌ ﴿٣٤﴾

Artinya: 33. Sesungguhnya Allah telah memilih Adam, Nuh, keluarga Ibrahim dan keluarga 'Imran melebihi segala umat (di masa mereka masing-masing), 34. (sebagai) satu keturunan yang sebagiannya (turunan) dari yang lain. dan Allah Maha mendengar lagi Maha mengetahui.

Ayat yang pertama menerangkan bahwa Nabi Adam sebagai bapak manusia, Allah SWT telah memilih Adam as dan keluarga Ibrahim as, serta keluarga Imran, dan menjadikan mereka manusia pilihan di masanya masing-masing, lagi pula diberikan kepada mereka *nubuwwah* dan *risalah*. Sedangkan

ayat yang kedua mengatakan bahwa dari keturunan Adam, lahirlah para Nabi dan Rasul. Rasul yang kedua adalah Nuh as, sebagai bapak manusia yang telah menyelamatkan dia dan keluarganya dari bencana yang dahsyat dalam satu bahtera. Keturunannya banyak yang menjadi Nabi dan Rasul. Kemudian ada Ibrahim as sebagai Nabi dan Rasul. Sesudah Ibrahim as datanglah berturut-turut beberapa orang Nabi dan Rasul yang berasal dari keturunannya, seperti Ismail as, Ishak as, Yakub as dan *Asbat* (anak cucu Bani Israil). Di antara keturunan Nabi Ibrahim yang terkemuka adalah: keluarga Ismail, Imran yaitu 'Isa dan ibunya Maryam binti keturunan Yakub. Kemudian kenabian itu ditutup dengan seorang putra dari keturunan Nabi Ismail as yaitu Muhammad SAW (Ihsan, 2012).

Menurut Ibnu Katsir, tafsir kedua ayat tersebut adalah bahwa Allah SWT memberitahukan bahwa Allah telah memilih beberapa keluarga atas keluarga lainnya di belahan bumi ini. Allah memilih Adam yang telah diciptakan-Nya sendiri dan ditiupkan ruh-Nya kepada-Nya, serta memerintahkan para malaikat bersujud kepadanya. Selanjutnya Allah memilih Nuh as dan menjadikannya Rasul pertama yang diutus kepada penduduk bumi ini, pada saat manusia menyembah berhala dan menyekutukan-Nya. Selain itu Allah memilih keluarga Ibrahim yang di antara keturunannya adalah Nabi Muhammad SAW, manusia paling mulia penutup para Nabi (Ghoffar, 2004).

Dari penjelasan tersebut dapat dilihat bahwa ada hubungan antara Q.S Ali-Imran ayat 33-34 dengan Algoritma Genetika. Q.S Ali-Imran 33-34 menjelaskan bahwa keturunan-keturunan yang baik dihasilkan dari orang tua yang baik pula. Berawal dari Nabi Adam dan berakhir pada Nabi Muhammad SAW, keturunan

dari para Nabi dan Rasul pasti baik. Sesuai dengan konsep Algoritma Genetika yang mengatakan bahwa keturunan yang baik akan menghasilkan keturunan yang baik pula, sehingga generasi yang berikutnya akan menjadi generasi yang lebih baik lagi.



BAB III

PEMBAHASAN

3.1 *Traveling Salesman Problem*

Menurut Davendra (2010), TSP secara matematika dapat dideskripsikan sebagai berikut:

$$\min \sum d_{ij} x_{ij} \quad (3.1)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (3.2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, 2, \dots, n \quad (3.3)$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1, 2, \dots, n \quad i \neq j \quad (3.4)$$

Dimana d_{ij} adalah jarak dari *node* i ke *node* j ; variabel $x_{ij} = 1$ berarti rute yang dilewati yaitu dari *node* i ke *node* j ; $x_{ij} = 0$ berarti rute yang tidak terpilih. Fungsi 3.1 merupakan fungsi tujuan yang berarti total jarak minimum yang didapatkan dari rute optimum; fungsi 3.2 berarti bahwa hanya dapat pergi dari *node* i satu kali saja, misalkan berada pada *node* i maka harus pergi ke *node* j yang belum pernah dikunjungi; kemudian fungsi 3.3 berarti bahwa hanya dapat mengunjungi *node* j satu kali saja (Davendra, 2010:26).

Pertama-tama yang dibutuhkan adalah mendefinisikan sebuah permasalahan untuk kemudian diselesaikan menggunakan Algoritma Genetika. dalam TSP, permasalahan berupa data jarak antar *node* (kota) yang digambarkan seperti di bawah ini:

Node	1	2	3	4	5
1	-	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
2	<i>e</i>	-	<i>f</i>	<i>g</i>	<i>h</i>
3	<i>i</i>	<i>j</i>	-	<i>k</i>	<i>l</i>
4	<i>m</i>	<i>n</i>	<i>o</i>	-	<i>p</i>
5	<i>q</i>	<i>r</i>	<i>s</i>	<i>t</i>	-

Tabel 3.1: Contoh Permasalahan TSP 5 Node

Tabel 3.1 di atas merupakan permasalahan TSP dengan 5 *node*, dimana *a, b, c, ..., t* adalah jarak antar *node*. Sebagai contoh, jarak dari *node* 1 ke *node* 2 adalah *a*, jarak dari *node* 5 ke *node* 2 adalah *r*, dan seterusnya. Dalam skripsi ini, data yang digunakan merupakan matriks jarak pada jurnal yang berjudul *Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator* oleh Zakir H. Ahmed (2010).

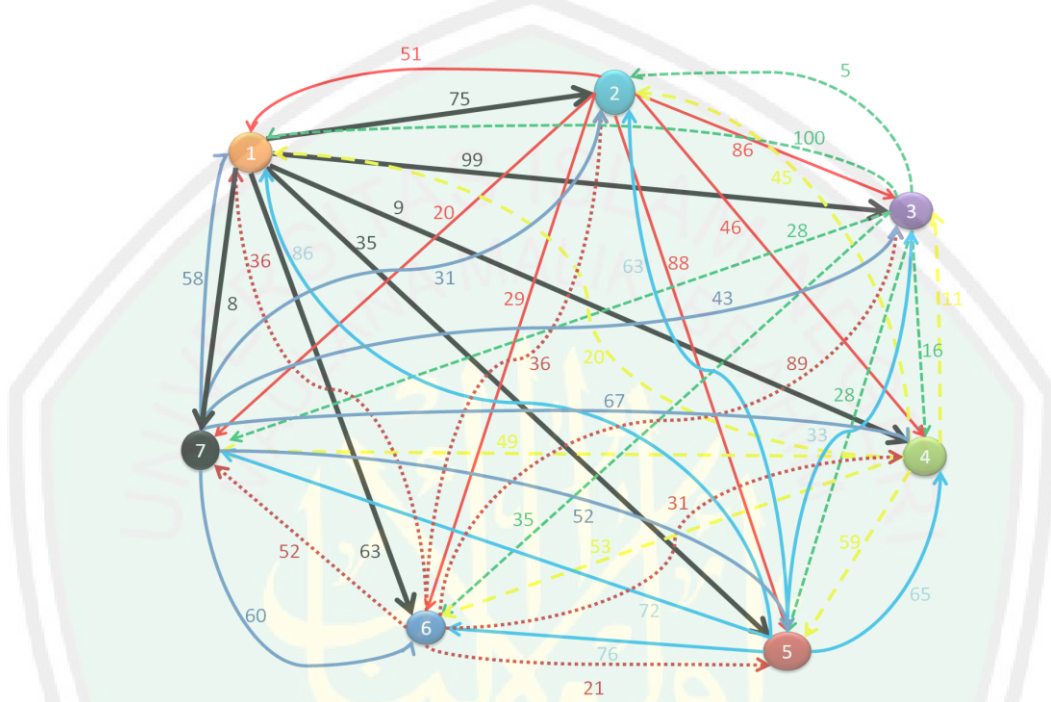
Node	1	2	3	4	5	6	7
1	0	75	99	9	35	63	8
2	51	0	86	46	88	29	20
3	100	5	0	16	28	35	28
4	20	45	11	0	59	53	49
5	86	63	33	65	0	76	72
6	36	53	89	31	21	0	52
7	58	31	43	67	52	60	0

Tabel 3.2: Matriks Jarak antar Node

Tabel 3.2 di atas menggambarkan sebuah peta dengan tujuh *node*. Antara dua *node* terdapat dua jalan yang saling berbeda dengan jarak yang berbeda pula. Jika *d* adalah jarak maka dapat ditulis:

$$d_{ij} \neq d_{ji}, \forall i, j = 1, 2, 3, \dots, 7 \text{ dan } i \neq j.$$

Data jarak di atas dapat direpresentasikan ke dalam bentuk digraf karena termasuk graf berarah. Gambar 3.1 di bawah ini merupakan representasi matriks jarak pada Tabel 3.2 jika digambarkan dalam bentuk digraf (graf berarah).



Gambar 3.1: Representasi Matriks Jarak antar *Node* dalam *Digraph*

Tujuan TSP adalah untuk menemukan rute terpendek dari sebuah *node* awal hingga kembali ke *node* awal lagi dengan syarat semua *node* hanya dilewati tepat satu kali. Contoh rutanya adalah sebagai berikut:

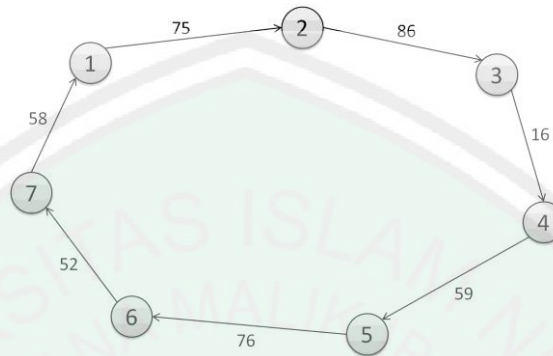
*Node*1 → *Node*2 → *Node*3 → *Node*4 → *Node*5 → *Node*6 → *Node*7 → *Node*1

Rute perjalanan tersebut diawali dari *Node*1 dan berakhir pada *Node*1 pula. Total jarak yang ditempuh adalah:

$$\begin{aligned} d_{12} + d_{23} + d_{34} + d_{45} + d_{56} + d_{67} + d_{71} \\ = 75 + 86 + 16 + 59 + 76 + 52 + 58 = 422 \end{aligned}$$

Terlihat dalam rute tersebut tidak ada *node* yang terlewati dua kali.

Gambar rute tersebut adalah sebagai berikut:



Gambar 3.2: Contoh Rute TSP

3.2 Algoritma Genetika

3.2.1 Inisialisasi Populasi

Langkah pertama yang harus dilakukan adalah inisialisasi populasi atau membangkitkan populasi awal. Dalam skripsi ini populasi awal terdiri dari sepuluh kromosom. Penulis membatasi hanya sepuluh kromosom agar ruang pencarian solusi tidak terlalu luas. Kromosom-kromosom tersebut dibangkitkan menggunakan Algoritma *Random Generator*. Setiap kromosom disimbolkan dengan (Z_i) untuk setiap $i = 1, 2, 3, \dots, 10$.

Inti dari Algoritma *Random Generator* adalah melibatkan pembangkitan bilangan acak untuk nilai setiap gen sesuai dengan representasi kromosom. Jumlah bilangan acak yang dibangkitkan adalah sebanyak gen pada setiap kromosom. Karena gen merupakan representasi dari *node*, maka dalam satu kromosom terdapat 7 (tujuh) gen. Sehingga bilangan acak yang dibangkitkan adalah sebanyak tujuh bilangan untuk setiap kromosom.

Untuk membangkitkan bilangan acak tersebut pada *Matlab* digunakan perintah `rand(1,7)`.

1. Kromosom 1 (Z_1)

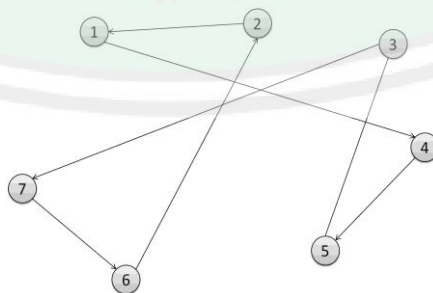
Bilangan acak yang didapatkan adalah:

[0.8235 0.6948 0.3171 0.9502 0.0344 0.4387 0.3816]

Sesuai dengan Algoritma *Random Generator*, posisi i dalam *list* menunjukkan *node* ke- i . Bilangan acak dalam posisi i menentukan urutan didatanginya kota i dalam lintasan. Dari bilangan-bilangan acak di atas, terlihat bahwa 0,0344 merupakan bilangan yang paling kecil. Sehingga *node* ke-5 menempati urutan pertama, 0,3171 merupakan bilangan terkecil kedua sehingga *node* ke-3 menempati urutan kedua, 0,3816 merupakan bilangan terkecil ketiga sehingga *node* ke-7 menempati urutan ketiga, begitu seterusnya sehingga didapatkan lintasan/rute sebagai berikut:

*Node*5 → *Node*3 → *Node*7 → *Node*6 → *Node*2 → *Node*1 → *Node*4 → *Node*5

Atau dapat ditulis: $Z_1 = (5, 3, 7, 6, 2, 1, 4)$



Gambar 3.3: Kromosom 1 (Z_1)

2. Kromosom 2 (Z_2)

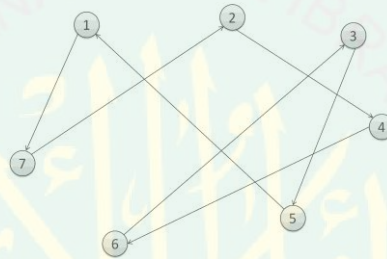
Bilangan acak yang didapatkan adalah:

[0.1707 0.2277 0.4357 0.3111 0.9234 0.4302 0.1848]

Dengan cara yang sama, didapatkan rute:

$Node1 \rightarrow Node7 \rightarrow Node2 \rightarrow Node4 \rightarrow Node6 \rightarrow Node3 \rightarrow Node5 \rightarrow Node1$

Atau dapat ditulis $Z_2 = (1, 7, 2, 4, 6, 3, 5)$.



Gambar 3.4: Kromosom 2 (Z_2)

3. Kromosom 3 (Z_3)

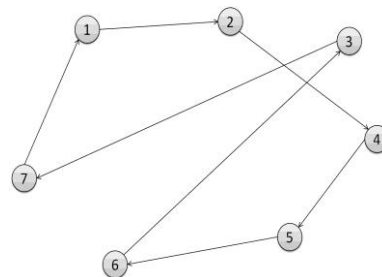
Bilangan acak yang didapatkan adalah:

[0.9049 0.9797 0.4389 0.1111 0.2581 0.4087 0.5949]

Dengan cara yang sama, didapatkan rute:

$Node4 \rightarrow Node5 \rightarrow Node6 \rightarrow Node3 \rightarrow Node7 \rightarrow Node1 \rightarrow Node2 \rightarrow Node4$

Atau dapat ditulis $Z_3 = (4, 5, 6, 3, 7, 1, 2)$



Gambar 3.5: Kromosom 3 (Z_3)

4. Kromosom 4 (Z_4)

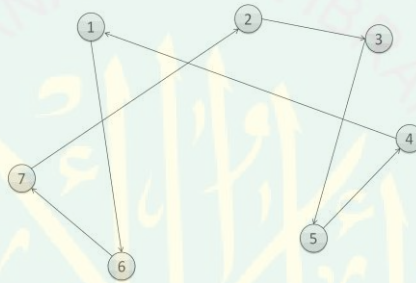
Bilangan acak yang didapatkan adalah:

[0.2622 0.6028 0.7112 0.2217 0.1174 0.2967 0.3188]

Dengan cara yang sama, didapatkan rute:

Node 5 → *Node 4* → *Node 1* → *Node 6* → *Node 7* → *Node 2* → *Node 3* → *Node 5*

Atau dapat ditulis $Z_4 = (5, 4, 1, 6, 7, 2, 3)$



Gambar 3.6: Kromosom 4 (Z_4)

5. Kromosom 5 (Z_5)

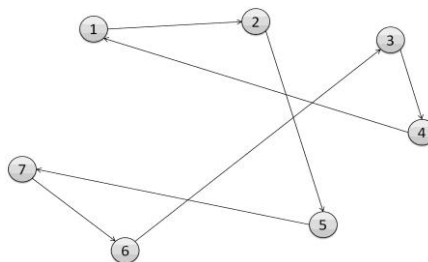
Bilangan acak yang didapatkan adalah:

[0.4242 0.5079 0.0855 0.2625 0.8010 0.0292 0.9289]

Dengan cara yang sama, didapatkan rute:

Node 6 → *Node 3* → *Node 4* → *Node 1* → *Node 2* → *Node 5* → *Node 7* → *Node 6*

Atau dapat ditulis $Z_5 = (6, 3, 4, 1, 2, 5, 7)$



Gambar 3.7: Kromosom 5 (Z_5)

6. Kromosom 6 (Z_6)

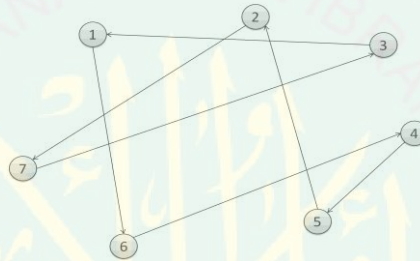
Bilangan acak yang didapatkan adalah:

[0.7303 0.4886 0.5785 0.2373 0.4588 0.9631 0.5468]

Dengan cara yang sama, didapatkan rute:

$Node4 \rightarrow Node5 \rightarrow Node2 \rightarrow Node7 \rightarrow Node3 \rightarrow Node1 \rightarrow Node6 \rightarrow Node4$

Atau dapat ditulis $Z_6 = (4,5,2,7,3,1,6)$



Gambar 3.8: Kromosom 6 (Z_6)

7. Kromosom 7 (Z_7)

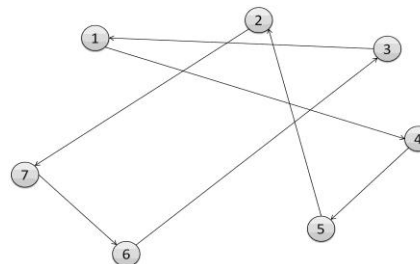
Bilangan acak yang didapatkan adalah:

[0.5211 0.2316 0.4889 0.6241 0.6791 0.3955 0.3674]

Dengan cara yang sama, didapatkan rute:

$Node2 \rightarrow Node7 \rightarrow Node6 \rightarrow Node3 \rightarrow Node1 \rightarrow Node4 \rightarrow Node5 \rightarrow Node2$

Atau dapat ditulis $Z_7 = (2,7,6,3,1,4,5)$



Gambar 3.9: Kromosom 7 (Z_7)

8. Kromosom 8 (Z_8)

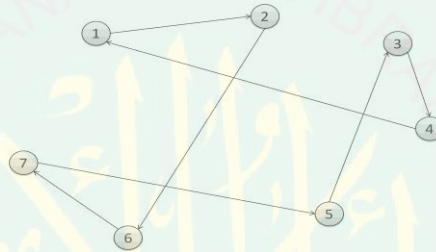
Bilangan acak yang didapatkan adalah:

[0.9880 0.0377 0.8852 0.9133 0.7962 0.0987 0.2619]

Dengan cara yang sama, didapatkan rute:

$Node 2 \rightarrow Node 6 \rightarrow Node 7 \rightarrow Node 5 \rightarrow Node 3 \rightarrow Node 4 \rightarrow Node 1 \rightarrow Node 2$

Atau dapat ditulis $Z_8 = (2, 6, 7, 5, 3, 4, 1)$



Gambar 3.10: Kromosom 8 (Z_8)

9. Kromosom 9 (Z_9)

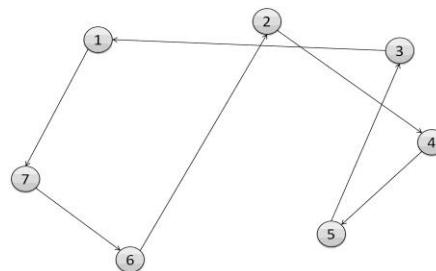
Bilangan acak yang didapatkan adalah:

[0.3354 0.6797 0.1366 0.7212 0.1068 0.6538 0.4942]

Dengan cara yang sama, didapatkan rute:

$Node 5 \rightarrow Node 3 \rightarrow Node 1 \rightarrow Node 7 \rightarrow Node 6 \rightarrow Node 2 \rightarrow Node 4 \rightarrow Node 5$

Atau dapat ditulis $Z_9 = (5, 3, 1, 7, 6, 2, 4)$



Gambar 3.11: Kromosom 9 (Z_9)

10. Kromosom 10 (Z_{10})

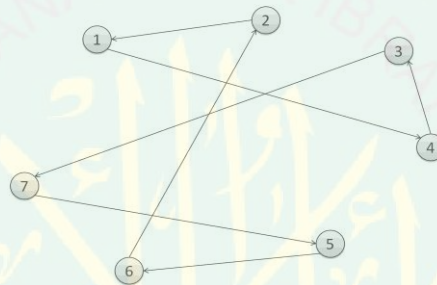
Bilangan acak yang didapatkan adalah:

[0.7791 0.7150 0.9037 0.8909 0.3342 0.6987 0.1978]

Dengan cara yang sama, didapatkan rute:

$Node7 \rightarrow Node5 \rightarrow Node6 \rightarrow Node2 \rightarrow Node1 \rightarrow Node4 \rightarrow Node3 \rightarrow Node7$

Atau dapat ditulis $Z_{10} = (7, 5, 6, 2, 1, 4, 3)$



Gambar 3.12: Kromosom 10 (Z_{10})

Setelah melakukan proses inialisai populasi menggunakan Algoritma *Random Generator*, didapatkan sepuluh kromosom sebagai populasi awal yang akan digunakan untuk langkah-langkah berikutnya dalam Algoritma Genetika. Sepuluh kromosom awal tersebut adalah:

$$Z_1 = (5, 3, 7, 6, 2, 1, 4)$$

$$Z_2 = (1, 7, 2, 4, 6, 3, 5)$$

$$Z_3 = (4, 5, 6, 3, 7, 1, 2)$$

$$Z_4 = (5, 4, 1, 6, 7, 2, 3)$$

$$Z_5 = (6, 3, 4, 1, 2, 5, 7)$$

$$Z_6 = (4, 5, 2, 7, 3, 1, 6)$$

$$Z_7 = (2, 7, 6, 3, 1, 4, 5)$$

$$Z_8 = (2, 6, 7, 5, 3, 4, 1)$$

$$Z_9 = (5, 3, 1, 7, 6, 2, 4)$$

$$Z_{10} = (7, 5, 6, 2, 1, 4, 3)$$

Setelah populasi awal tersebut didapatkan, maka dapat dilakukan proses yang selanjutnya yaitu proses seleksi.

3.2.2 Proses Seleksi

Setelah mendapatkan populasi awal, selanjutnya dilakukan proses seleksi. Seleksi digunakan untuk memilih kromosom-kromosom mana saja yang akan dipilih sebagai *parent* (orang tua) untuk proses *crossover* dan mutasi. Semakin tinggi nilai *fitness* suatu kromosom maka semakin besar kemungkinannya untuk terpilih. Pertama-tama akan dicari jarak tempuh tiap kromosom, nilai *fitness*, total *fitness*, probabilitas *fitness*, dan probabilitas kumulatif *fitness*.

1. Mencari jarak tempuh tiap kromosom (mZ_i)

$$\begin{aligned} mZ_1 &= 33 + 28 + 60 + 53 + 51 + 9 + 59 \\ &= 293 \end{aligned}$$

$$\begin{aligned} mZ_2 &= 8 + 31 + 46 + 53 + 89 + 28 + 86 \\ &= 341 \end{aligned}$$

$$\begin{aligned} mZ_3 &= 59 + 76 + 89 + 28 + 58 + 75 + 46 \\ &= 431 \end{aligned}$$

$$\begin{aligned} mZ_4 &= 65 + 20 + 63 + 52 + 31 + 86 + 28 \\ &= 345 \end{aligned}$$

$$\begin{aligned} mZ_5 &= 89 + 16 + 20 + 75 + 88 + 72 + 60 \\ &= 420 \end{aligned}$$

$$\begin{aligned} mZ_6 &= 59 + 63 + 20 + 43 + 100 + 63 + 31 \\ &= 379 \end{aligned}$$

$$\begin{aligned} mZ_7 &= 20 + 60 + 89 + 100 + 9 + 59 + 63 \\ &= 400 \end{aligned}$$

$$\begin{aligned} mZ_8 &= 29 + 52 + 52 + 33 + 16 + 20 + 75 \\ &= 277 \end{aligned}$$

$$\begin{aligned} mZ_9 &= 33 + 100 + 8 + 60 + 53 + 46 + 59 \\ &= 359 \end{aligned}$$

$$\begin{aligned} mZ_{10} &= 52 + 76 + 53 + 51 + 9 + 11 + 28 \\ &= 280 \end{aligned}$$

2. Mencari nilai *fitness* tiap kromosom (f_i)

$$f_i = \frac{1}{mZ_i}$$

$$f_1 = \frac{1}{mZ_1} = \frac{1}{293} = 0,0034130$$

$$f_2 = \frac{1}{mZ_2} = \frac{1}{341} = 0,0029326$$

$$f_3 = \frac{1}{mZ_3} = \frac{1}{431} = 0,0023202$$

$$f_4 = \frac{1}{mZ_4} = \frac{1}{345} = 0,0028986$$

$$f_5 = \frac{1}{mZ_5} = \frac{1}{420} = 0,0023810$$

$$f_6 = \frac{1}{mZ_6} = \frac{1}{379} = 0,0026385$$

$$f_7 = \frac{1}{mZ_7} = \frac{1}{400} = 0,0025000$$

$$f_8 = \frac{1}{mZ_8} = \frac{1}{277} = 0,0036101$$

$$f_9 = \frac{1}{mZ_9} = \frac{1}{359} = 0,0027855$$

$$f_{10} = \frac{1}{mZ_{10}} = \frac{1}{280} = 0,0035714$$

3. Mencari total *fitness* $\sum_{i=1}^{10} f_i$.

$$\begin{aligned} \sum_{i=1}^{10} f_i &= f_1 + f_2 + f_3 + f_4 + f_5 + f_6 + f_7 + f_8 + f_9 + f_{10} \\ &= 0,0034130 + 0,0029326 + 0,0023202 + 0,0028986 + \dots \\ &\quad \dots + 0,0023810 + 0,0026385 + 0,0025000 + 0,0036101 + \dots \\ &\quad \dots + 0,0027855 + 0,0035714 \\ &= 0,0290509 \end{aligned}$$

4. Mencari probabilitas *fitness* tiap kromosom (p_i)

$$p_i = \frac{f_i}{\sum_{i=1}^{10} f_i}$$

$$p_1 = \frac{0,0034130}{0,0290509} = 0,117483$$

$$p_2 = \frac{0,0029326}{0,0290509} = 0,100947$$

$$p_3 = \frac{0,0023202}{0,0290509} = 0,079867$$

$$p_4 = \frac{0,0028986}{0,0290509} = 0,099777$$

$$p_5 = \frac{0,0023810}{0,0290509} = 0,081960$$

$$p_6 = \frac{0,0026385}{0,0290509} = 0,090823$$

$$p_7 = \frac{0,0025000}{0,0290509} = 0,086056$$

$$p_8 = \frac{0,0036101}{0,0290509} = 0,124268$$

$$p_9 = \frac{0,0027855}{0,0290509} = 0,095883$$

$$p_{10} = \frac{0,0035714}{0,0290509} = 0,122936$$

Dari proses perhitungan di atas dapat dilihat bahwa kromosom Z_8 memiliki jarak tempuh paling kecil sehingga mempunyai probabilitas terpilih sebagai *parent* lebih besar daripada kromosom lainnya. Setelah itu digunakan teknik *Roulette Wheel Selection* untuk melakukan proses seleksi. Tetapi sebelum itu, terlebih dahulu dihitung probabilitas kumulatifnya.

$$c_1 = 0,117483$$

$$c_2 = 0,117483 + 0,100947 = 0,218430$$

$$c_3 = 0,218430 + 0,079867 = 0,298297$$

$$c_4 = 0,298297 + 0,099777 = 0,398074$$

$$c_5 = 0,398074 + 0,081960 = 0,480034$$

$$c_6 = 0,480034 + 0,090823 = 0,570857$$

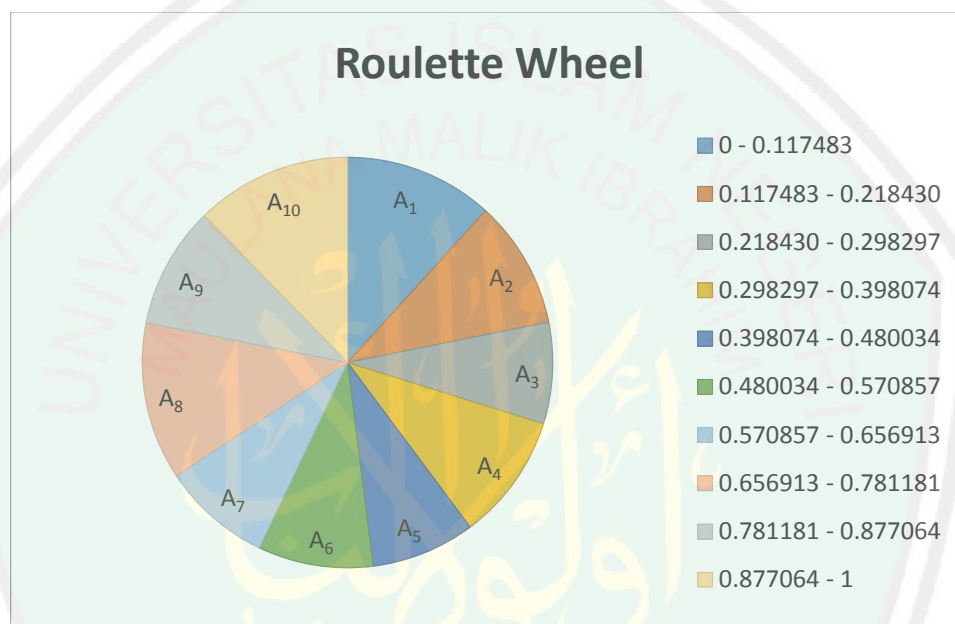
$$c_7 = 0,570857 + 0,086056 = 0,656913$$

$$c_8 = 0,656913 + 0,124268 = 0,781181$$

$$c_9 = 0,781181 + 0,095883 = 0,877064$$

$$c_{10} = 0,877604 + 0,122936 = 1$$

Probabilitas kumulatif tersebut jika digambarkan dengan *Roulette Wheel* adalah sebagai berikut:



Gambar 3.13: Probabilitas Kumulatif dalam *Roulette Wheel*

Keterangan:

A_i dengan $i = 1, 2, 3, \dots, 10$ merupakan interval dari c_i ke c_{i+1} .

Langkah-langkah seleksi menggunakan metode *Roulette Wheel* adalah sebagai berikut:

1. Bangkitkan bilangan acak R antara 0 sampai 1 sebanyak kromosom dalam satu populasi.
2. Jika $R_k < c_i$ untuk setiap k dan $i = 1, 2, 3, \dots, 10$, maka kromosom ke- k dipilih sebagai *parent*. Selain itu pilih kromosom ke- k sebagai *parent*

dengan syarat $c_{i-1} < R_k < c_i$. Dengan kata lain, jika R_k berada di dalam A_i maka kromosom ke- k digantikan kromosom ke- i .

Bilangan acak R dibangkitkan sebanyak jumlah kromosom dalam satu populasi yaitu 10 bilangan. Perintah untuk membangkitkan sepuluh bilangan acak pada Matlab adalah `rand(1,10)`. Hasilnya adalah sebagai berikut:

[0.0760 0.2399 0.1233 0.1839 0.2400 0.4173 0.0497 0.9027 0.9448 0.4909]

1. $R_1 = 0,0760$

Nilai R_1 berada di dalam A_1 , sehingga Kromosom 1 tetap menjadi Kromosom 1.

$$Z_1 = Z_1 = (5, 3, 7, 6, 2, 1, 4)$$

2. $R_2 = 0,2399$

Nilai R_2 berada di dalam A_3 , sehingga Kromosom 2 digantikan oleh Kromosom 3.

$$Z_2 = Z_3 = (4, 5, 6, 3, 7, 1, 2)$$

3. $R_3 = 0,1233$

Nilai R_3 berada di dalam A_2 , sehingga Kromosom 3 digantikan oleh Kromosom 2.

$$Z_3 = Z_2 = (1, 7, 2, 4, 6, 3, 5)$$

4. $R_4 = 0,1839$

Nilai R_4 berada di dalam A_2 , sehingga Kromosom 4 digantikan oleh Kromosom 2.

$$Z_4 = Z_2 = (1, 7, 2, 4, 6, 3, 5)$$

5. $R_5 = 0,2400$

Nilai R_5 berada di dalam A_3 , sehingga Kromosom 5 digantikan oleh Kromosom 3.

$$Z_5 = Z_3 = (4, 5, 6, 3, 7, 1, 2)$$

6. $R_6 = 0,4173$

Nilai R_6 berada di dalam A_5 , sehingga Kromosom 6 digantikan oleh Kromosom 5.

$$Z_6 = Z_5 = (6, 3, 4, 1, 2, 5, 7)$$

7. $R_7 = 0,0497$

Nilai R_7 berada di dalam A_1 , sehingga Kromosom 7 digantikan oleh Kromosom 1.

$$Z_7 = Z_1 = (5, 3, 7, 6, 2, 1, 4)$$

8. $R_8 = 0,9027$

Nilai R_8 berada di dalam A_{10} , sehingga Kromosom 8 digantikan oleh Kromosom 10.

$$Z_8 = Z_{10} = (7, 5, 6, 2, 1, 4, 3)$$

9. $R_9 = 0,9448$

Nilai R_9 berada di dalam A_{10} , sehingga Kromosom 9 tetap menjadi Kromosom 10.

$$Z_9 = Z_{10} = (7, 5, 6, 2, 1, 4, 3)$$

10. $R_{10} = 0,4909$

Nilai R_{10} berada di dalam A_6 , sehingga Kromosom 10 tetap menjadi Kromosom 6.

$$Z_{10} = Z_6 = (4, 5, 2, 7, 3, 1, 6)$$

Dengan demikian, maka populasi baru yang terbentuk setelah dilakukan proses seleksi adalah:

$$Z_1 = (5, 3, 7, 6, 2, 1, 4)$$

$$Z_2 = (4, 5, 6, 3, 7, 1, 2)$$

$$Z_3 = (1, 7, 2, 4, 6, 3, 5)$$

$$Z_4 = (1, 7, 2, 4, 6, 3, 5)$$

$$Z_5 = (4, 5, 6, 3, 7, 1, 2)$$

$$Z_6 = (6, 3, 4, 1, 2, 5, 7)$$

$$Z_7 = (5, 3, 7, 6, 2, 1, 4)$$

$$Z_8 = (7, 5, 6, 2, 1, 4, 3)$$

$$Z_9 = (7, 5, 6, 2, 1, 4, 3)$$

$$Z_{10} = (4, 5, 2, 7, 3, 1, 6)$$

Perbedaan populasi sebelum proses seleksi dengan populasi sesudah proses seleksi disajikan dalam Tabel 3.3 berikut ini:

Sebelum Proses Seleksi	Jarak	Sesudah Proses Seleksi	Jarak
$Z_1 = (5, 3, 7, 6, 2, 1, 4)$	293	$Z_1 = (5, 3, 7, 6, 2, 1, 4)$	293
$Z_2 = (1, 7, 2, 4, 6, 3, 5)$	341	$Z_2 = (4, 5, 6, 3, 7, 1, 2)$	431
$Z_3 = (4, 5, 6, 3, 7, 1, 2)$	431	$Z_3 = (1, 7, 2, 4, 6, 3, 5)$	341
$Z_4 = (5, 4, 1, 6, 7, 2, 3)$	345	$Z_4 = (1, 7, 2, 4, 6, 3, 5)$	341
$Z_5 = (6, 3, 4, 1, 2, 5, 7)$	420	$Z_5 = (4, 5, 6, 3, 7, 1, 2)$	431
$Z_6 = (4, 5, 2, 7, 3, 1, 6)$	379	$Z_6 = (6, 3, 4, 1, 2, 5, 7)$	420
$Z_7 = (2, 7, 6, 3, 1, 4, 5)$	400	$Z_7 = (5, 3, 7, 6, 2, 1, 4)$	293
$Z_8 = (2, 6, 7, 5, 3, 4, 1)$	277	$Z_8 = (7, 5, 6, 2, 1, 4, 3)$	280
$Z_9 = (5, 3, 1, 7, 6, 2, 4)$	359	$Z_9 = (7, 5, 6, 2, 1, 4, 3)$	280
$Z_{10} = (7, 5, 6, 2, 1, 4, 3)$	280	$Z_{10} = (4, 5, 2, 7, 3, 1, 6)$	379

Tabel 3.3: Perbandingan Populasi Sebelum dan Sesudah Proses Seleksi

Setelah melakukan proses seleksi menggunakan Metode *Roulette Wheel*, terlihat ada beberapa kromosom pada populasi awal yang terseleksi yaitu Kromosom 4 (Z_4), Kromosom 7 (Z_7), Kromosom 8 (Z_8) dan Kromosom 9 (Z_9). Populasi baru yang terbentuk akan digunakan untuk proses selanjutnya yaitu proses *crossover*.

3.2.3 Proses *Crossover* Menggunakan *Partially Matched Crossover*

Probabilitas *crossover* (pc) yang digunakan adalah 1 (100%), sehingga seluruh kromosom dalam populasi mengalami *crossover*. Setelah semua

kromosom mengalami *crossover*, selanjutnya dipilih kromosom-kromosom anak yang memiliki jarak paling kecil.

Crossover $Z_1 \times Z_2$

1. *Parent* yang akan di-*crossover* adalah $Z_1 = (5, 3, 7, 6, 2, 1, 4)$ dan

$$Z_2 = (4, 5, 6, 3, 7, 1, 2).$$

Z_1	5	3	7	6	2	1	4
-------	---	---	---	---	---	---	---

Z_2	4	5	6	3	7	1	2
-------	---	---	---	---	---	---	---

2. Membuat *substring* secara acak dari kedua *parent*.

Z_1	5	3	7	6	2	1	4
-------	---	---	---	---	---	---	---

Z_2	4	5	6	3	7	1	2
-------	---	---	---	---	---	---	---

3. Menukar posisi *substring*.

Z_1							2
							↑ ↓
Z_2							4

4. Pemetaan yang terjadi adalah:

$$2 \leftrightarrow 4$$

5. Isikan bagian-bagian gen sesuai pemetaan.

Z_1	5	3	7	6	4	1	2
-------	---	---	---	---	---	---	---

Z_2	2	5	6	3	7	1	4
-------	---	---	---	---	---	---	---

6. Kromosom anak yang dihasilkan dari *crossover* Z_1 dan Z_2 di atas adalah:

$$Z_1' = (5, 3, 7, 6, 4, 1, 2)$$

$$Z_2' = (2, 5, 6, 3, 7, 1, 4)$$

Crossover $Z_3 \times Z_4$

Crossover antara Z_3 dan Z_4 menghasilkan anak yang sama dengan induknya karena $Z_3 = Z_4$.

$$Z_3' = (1, 7, 2, 4, 6, 3, 5)$$

$$Z_4' = (1, 7, 2, 4, 6, 3, 5)$$

Crossover $Z_5 \times Z_6$

1. *Parent* yang akan di-crossover adalah $Z_5 = (4, 5, 6, 3, 7, 1, 2)$ dan

$$Z_6 = (6, 3, 4, 1, 2, 5, 7).$$

Z_5	4	5	6	3	7	1	2
-------	---	---	---	---	---	---	---

Z_6	6	3	4	1	2	5	7
-------	---	---	---	---	---	---	---

2. Membuat *substring* dari kedua *parent* secara acak.

Z_5	4	5	6	3	7	1	2
-------	---	---	---	---	---	---	---

Z_6	6	3	4	1	2	5	7
-------	---	---	---	---	---	---	---

3. Menukar posisi *substring*.

Z_5		3	4	1	2	5	7
		↓	↓	↓	↓	↓	↓
Z_6		5	6	3	7	1	2

4. Pemetaan yang terjadi adalah:

$$3 \leftrightarrow 5, 4 \leftrightarrow 6, 1 \leftrightarrow 3, 2 \leftrightarrow 7, 5 \leftrightarrow 1 \text{ dan } 7 \leftrightarrow 2.$$

5. Isikan bagian-bagian gen sesuai pemetaan.

Z_5	6	3	4	1	2	5	7
-------	---	---	---	---	---	---	---

Z_6	4	5	6	3	7	1	2
-------	---	---	---	---	---	---	---

6. Anak yang dihasilkan dari *crossover* Z_5 dan Z_6 di atas adalah:

$$Z_5' = (6, 3, 4, 1, 2, 5, 7)$$

$$Z_6' = (4, 5, 6, 3, 7, 1, 2)$$

Crossover $Z_7 \times Z_8$

1. *Parent* yang akan di-*crossover* adalah $Z_7 = (5, 3, 7, 6, 2, 1, 4)$ dan

$$Z_8 = (7, 5, 6, 2, 1, 4, 3).$$

Z_7	5	3	7	6	2	1	4
-------	---	---	---	---	---	---	---

Z_8	7	5	6	2	1	4	3
-------	---	---	---	---	---	---	---

2. Membuat *substring* dari kedua *parent* secara acak.

Z_7	5	3	7	6	2	1	4
-------	---	---	---	---	---	---	---

Z_8	7	5	6	2	1	4	3
-------	---	---	---	---	---	---	---

3. Menukar posisi *substring*.

Z_7	7	5	6	2	1		
-------	---	---	---	---	---	--	--

\updownarrow \updownarrow \updownarrow \updownarrow \updownarrow

Z_8	5	3	7	6	2		
-------	---	---	---	---	---	--	--

4. Pemetaan yang terjadi adalah:

$$5 \leftrightarrow 7, 3 \leftrightarrow 5, 7 \leftrightarrow 6, 6 \leftrightarrow 2 \text{ dan } 2 \leftrightarrow 1$$

5. Isikan bagian-bagian gen sesuai pemetaan.

Z_7	7	5	6	2	1	3	4
-------	---	---	---	---	---	---	---

Z_8	5	3	7	6	2	4	1
-------	---	---	---	---	---	---	---

6. Anak yang dihasilkan dari *crossover* Z_7 dan Z_8 di atas adalah:

$$Z_7' = (7, 5, 6, 2, 1, 3, 4)$$

$$Z_8' = (5, 3, 7, 6, 2, 4, 1)$$

Crossover $Z_9 \times Z_{10}$

1. *Parent* yang akan di-*crossover* adalah $Z_9 = (7, 5, 6, 2, 1, 4, 3)$ dan

$$Z_{10} = (4, 5, 2, 7, 3, 1, 6).$$

Z_9	7	5	6	2	1	4	3
-------	---	---	---	---	---	---	---

Z_{10}	4	5	2	7	3	1	6
----------	---	---	---	---	---	---	---

2. Membuat *substring* dari kedua *parent* secara acak.

Z_9	7	5	6	2	1	4	3
-------	---	---	---	---	---	---	---

Z_{10}	4	5	2	7	3	1	6
----------	---	---	---	---	---	---	---

3. Menukar posisi *substring*.

Z_9			2	7			
-------	--	--	---	---	--	--	--



Z_{10}			6	2			
----------	--	--	---	---	--	--	--

4. Pemetaan yang terjadi adalah:

$$2 \leftrightarrow 6 \text{ dan } 7 \leftrightarrow 2.$$

5. Isikan bagian-bagian gen sesuai pemetaan.

Z_9	6	5	2	7	1	4	3
-------	---	---	---	---	---	---	---

Z_{10}	4	5	6	2	3	1	7
----------	---	---	---	---	---	---	---

6. Anak yang dihasilkan dari *crossover* Z_9 dan Z_{10} di atas adalah:

$$Z_9' = (6, 5, 2, 7, 1, 4, 3)$$

$$Z_{10}' = (4, 5, 6, 2, 3, 1, 7)$$

Proses *crossover* seluruhnya disajikan pada Lampiran.

Setelah dilakukan proses *crossover*, kromosom-kromosom dalam populasi berubah sesuai hasil *crossover*. Dari seluruh kemungkinan *crossover* yang telah dilakukan, diambil kromosom-kromosom anak yang memiliki jarak paling kecil dari masing-masing kromosom *parent*. Kromosom-kromosom tersebut adalah sebagai berikut:

Sebelum Proses <i>Crossover</i>	Jarak	Sesudah Proses <i>Crossover</i>	Jarak
$Z_1 = (1, 7, 2, 4, 6, 3, 5)$	341	$Z_1' = (1, 7, 2, 6, 3, 5, 4)$	270
$Z_2 = (7, 5, 6, 2, 1, 4, 3)$	280	$Z_2' = (3, 7, 2, 4, 6, 1, 5)$	262
$Z_3 = (7, 5, 6, 2, 1, 4, 3)$	280	$Z_3' = (1, 4, 2, 7, 3, 6, 5)$	259
$Z_4 = (4, 5, 2, 7, 3, 1, 6)$	379	$Z_4' = (1, 4, 2, 7, 3, 6, 5)$	259
$Z_5 = (5, 3, 1, 7, 6, 2, 4)$	359	$Z_5' = (1, 7, 2, 4, 5, 3, 6)$	248
$Z_6 = (1, 7, 2, 4, 6, 3, 5)$	341	$Z_6' = (6, 7, 5, 2, 1, 4, 3)$	273
$Z_7 = (6, 3, 4, 1, 2, 5, 7)$	420	$Z_7' = (5, 3, 2, 6, 7, 1, 4)$	245
$Z_8 = (7, 5, 6, 2, 1, 4, 3)$	280	$Z_8' = (5, 3, 7, 2, 1, 4, 6)$	226
$Z_9 = (5, 3, 1, 7, 6, 2, 4)$	359	$Z_9' = (5, 3, 7, 2, 1, 4, 6)$	226
$Z_{10} = (7, 5, 6, 2, 1, 4, 3)$	280	$Z_{10}' = (5, 3, 2, 7, 4, 1, 6)$	229

Tabel 3.4: Perbandingan Populasi Sebelum dan Sesudah Proses *Crossover*

Setelah dilakukan proses *crossover*, populasi baru yang didapatkan memiliki kromosom-kromosom yang jarak tempuhnya lebih kecil. Hal ini menunjukkan bahwa proses *crossover* sangat berpengaruh pada Algoritma Genetika. Setelah didapatkan populasi dari proses *crossover*, selanjutnya dapat dilakukan proses mutasi.

3.2.4 Proses Mutasi

Probabilitas mutasi (pm) yang digunakan adalah 0,01 (1%). Langkah-langkah dalam proses mutasi adalah sebagai berikut:

1. Banyaknya gen dalam satu kromosom adalah 7, dan jumlah kromosom dalam populasi adalah 10. Maka panjang total gen adalah $7 \times 10 = 70$.

2. Jumlah gen yang akan dimutasi adalah $0,01 \times 70 = 0,7 \approx 1$. Dengan demikian akan dibangkitkan sebanyak 1 (satu) bilangan bulat acak antara 1 sampai 70.
3. Pembangkitan bilangan bulat acak dilakukan dengan perintah `randint(1,1,[1,70])` pada Matlab. Bilangan acak yang dihasilkan adalah:

[5]

4. Proses mutasi:

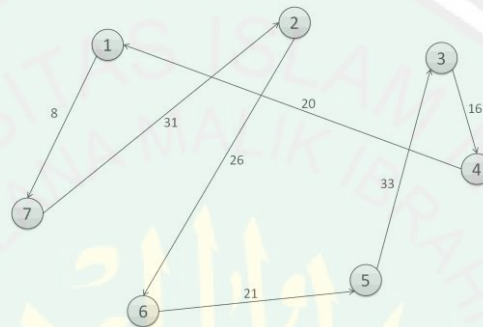
Bilangan bulat acak yang keluar adalah 5, sehingga dilakukan mutasi pada gen ke-5 dengan cara menukar gen tersebut dengan gen sesudahnya yaitu gen ke-6. Gen ke-5 terdapat pada kromosom Z_1' dengan nilai 3 bertukar posisi dengan gen sesudahnya yang bernilai 5.

Sebelum Proses Mutasi	Jarak	Sesudah Proses Mutasi	Jarak
$Z_1' = (1, 7, 2, 6, 3, 5, 4)$	270	$Z_1' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_2' = (3, 7, 2, 4, 6, 1, 5)$	262	$Z_2' = (3, 7, 2, 4, 6, 1, 5)$	262
$Z_3' = (1, 4, 2, 7, 3, 6, 5)$	259	$Z_3' = (1, 4, 2, 7, 3, 6, 5)$	259
$Z_4' = (1, 4, 2, 7, 3, 6, 5)$	259	$Z_4' = (1, 4, 2, 7, 3, 6, 5)$	259
$Z_5' = (1, 7, 2, 4, 5, 3, 6)$	248	$Z_5' = (1, 7, 2, 4, 5, 3, 6)$	248
$Z_6' = (6, 7, 5, 2, 1, 4, 3)$	273	$Z_6' = (6, 7, 5, 2, 1, 4, 3)$	273
$Z_7' = (5, 3, 2, 6, 7, 1, 4)$	245	$Z_7' = (5, 3, 2, 6, 7, 1, 4)$	245
$Z_8' = (5, 3, 7, 2, 1, 4, 6)$	226	$Z_8' = (5, 3, 7, 2, 1, 4, 6)$	226
$Z_9' = (5, 3, 7, 2, 1, 4, 6)$	226	$Z_9' = (5, 3, 7, 2, 1, 4, 6)$	226
$Z_{10}' = (5, 3, 2, 7, 4, 1, 6)$	229	$Z_{10}' = (5, 3, 2, 7, 4, 1, 6)$	229

Tabel 3.5: Perbandingan Populasi Sebelum dan Sesudah Proses Mutasi

Proses Algoritma Genetika untuk satu generasi telah selesai dilakukan. Kromosom yang memiliki total jarak terpendek adalah kromosom anak 1 (Z_1') dengan jarak 158. Jadi, rute yang paling pendek adalah rute:

Node1 → *Node7* → *Node2* → *Node6* → *Node5* → *Node3* → *Node4* → *Node1*



Gambar 3.14: Rute Terpendek

Untuk memastikan apakah rute tersebut sudah minimum, maka perlu dilakukan proses Algoritma Genetika untuk generasi kedua. Jika rute yang didapatkan memiliki jarak yang sama atau lebih besar, maka rute terpendeknya tetap seperti pada Gambar 3.14. Sedangkan jika rute yang didapatkan memiliki jarak yang lebih kecil, maka dilakukan lagi proses Algoritma Genetika untuk generasi yang selanjutnya hingga didapatkan rute dengan jarak yang sama atau lebih besar dari generasi sebelumnya.

3.3 Algoritma Genetika Generasi Kedua

3.3.1 Inisialisasi Populasi

Populasi awal yang digunakan pada generasi kedua adalah populasi akhir setelah proses mutasi pada generasi pertama. Sehingga populasi awal yang didapatkan adalah:

$$Z_1' = (1, 7, 2, 6, 5, 3, 4)$$

$$Z_2' = (3, 7, 2, 4, 6, 1, 5)$$

$$Z_3' = (1, 4, 2, 7, 3, 6, 5)$$

$$Z_4' = (1, 4, 2, 7, 3, 6, 5)$$

$$Z_5' = (1, 7, 2, 4, 5, 3, 6)$$

$$Z_6' = (6, 7, 5, 2, 1, 4, 3)$$

$$Z_7' = (5, 3, 2, 6, 7, 1, 4)$$

$$Z_8' = (5, 3, 7, 2, 1, 4, 6)$$

$$Z_9' = (5, 3, 7, 2, 1, 4, 6)$$

$$Z_{10}' = (5, 3, 2, 7, 4, 1, 6)$$

3.3.2 Proses Seleksi

Setelah didapatkan populasi awal yang baru, maka proses seleksi dapat dilakukan. Seperti sebelumnya, pertama-tama akan dicari jarak tempuh tiap kromosom, nilai *fitness*, total *fitness*, probabilitas *fitness*, dan probabilitas kumulatif *fitness*.

1. Mencari jarak tempuh tiap kromosom (mZ_i')

$$\begin{aligned} mZ_1' &= 8 + 31 + 29 + 21 + 33 + 16 + 20 \\ &= 158 \end{aligned}$$

$$\begin{aligned} mZ_2' &= 28 + 31 + 46 + 53 + 36 + 35 + 33 \\ &= 262 \end{aligned}$$

$$\begin{aligned} mZ_3' &= 9 + 45 + 20 + 43 + 35 + 21 + 86 \\ &= 259 \end{aligned}$$

$$\begin{aligned} mZ_4' &= 9 + 45 + 20 + 43 + 35 + 21 + 86 \\ &= 259 \end{aligned}$$

$$\begin{aligned} mZ_5' &= 8 + 31 + 46 + 59 + 33 + 35 + 36 \\ &= 248 \end{aligned}$$

$$\begin{aligned} mZ_6' &= 52 + 52 + 63 + 51 + 9 + 11 + 35 \\ &= 273 \end{aligned}$$

$$\begin{aligned} mZ_7' &= 33 + 5 + 29 + 52 + 58 + 9 + 59 \\ &= 245 \end{aligned}$$

$$\begin{aligned} mZ_8' &= 33 + 28 + 31 + 51 + 9 + 53 + 21 \\ &= 226 \end{aligned}$$

$$\begin{aligned} mZ_9' &= 33 + 28 + 31 + 51 + 9 + 53 + 21 \\ &= 226 \end{aligned}$$

$$\begin{aligned} mZ_{10}' &= 33 + 5 + 20 + 67 + 20 + 63 + 21 \\ &= 229 \end{aligned}$$

2. Mencari nilai *fitness* tiap kromosom (f_i)

$$f_i' = \frac{1}{mZ_i'}$$

$$f_1' = \frac{1}{mZ_1'} = \frac{1}{158} = 0,0063291$$

$$f_2' = \frac{1}{mZ_2'} = \frac{1}{262} = 0,0038168$$

$$f_3' = \frac{1}{mZ_3'} = \frac{1}{259} = 0,0038610$$

$$f_4' = \frac{1}{mZ_4'} = \frac{1}{259} = 0,0038610$$

$$f_5' = \frac{1}{mZ_5'} = \frac{1}{248} = 0,0040323$$

$$f_6' = \frac{1}{mZ_6'} = \frac{1}{273} = 0,0036630$$

$$f_7' = \frac{1}{mZ_7'} = \frac{1}{245} = 0,0040816$$

$$f_8' = \frac{1}{mZ_8'} = \frac{1}{226} = 0,0044248$$

$$f_9' = \frac{1}{mZ_9'} = \frac{1}{226} = 0,0044248$$

$$f_{10}' = \frac{1}{mZ_{10}'} = \frac{1}{229} = 0,0043668$$

3. Mencari total *fitness* $\sum_{i=1}^{10} f_i'$.

$$\begin{aligned} \sum_{i=1}^{10} f_i' &= f_1' + f_2' + f_3' + f_4' + f_5' + f_6' + f_7' + f_8' + f_9' + f_{10}' \\ &= 0,0063291 + 0,0038168 + 0,0038610 + 0,0038610 + \dots \\ &\quad \dots + 0,0040323 + 0,0036630 + 0,0040816 + 0,0044248 + \dots \\ &\quad \dots + 0,0044248 + 0,0043668 \\ &= 0,0428612 \end{aligned}$$

4. Mencari probabilitas *fitness* tiap kromosom (p_i')

$$p_i' = \frac{f_i'}{\sum_{i=1}^{10} f_i'}$$

$$p_1' = \frac{0,0063291}{0,0428612} = 0,147665$$

$$p_2' = \frac{0,0038168}{0,0428612} = 0,089050$$

$$p_3' = \frac{0,0038610}{0,0428612} = 0,090081$$

$$p_4' = \frac{0,0038610}{0,0428612} = 0,090081$$

$$p_5' = \frac{0,0040323}{0,0428612} = 0,094078$$

$$p_6' = \frac{0,0036630}{0,0428612} = 0,085462$$

$$p_7' = \frac{0,0040816}{0,0428612} = 0,0095228$$

$$p_8' = \frac{0,0044248}{0,0428612} = 0,103236$$

$$p_9' = \frac{0,0044248}{0,0428612} = 0,103236$$

$$p_{10}' = \frac{0,0043668}{0,0428612} = 0,101882$$

Dari proses perhitungan di atas dapat dilihat bahwa kromosom Z_1' memiliki jarak tempuh paling kecil sehingga mempunyai probabilitas terpilih sebagai *parent* lebih besar daripada kromosom lainnya. Setelah itu digunakan teknik *Roulette Wheel Selection* untuk melakukan proses seleksi. Tetapi sebelum itu, terlebih dahulu dihitung probabilitas kumulatifnya.

$$c_1' = 0,147665$$

$$c_2' = 0,147665 + 0,089050 = 0,236715$$

$$c_3' = 0,236715 + 0,090081 = 0,326796$$

$$c_4' = 0,326796 + 0,090081 = 0,416878$$

$$c_5' = 0,416878 + 0,094078 = 0,510956$$

$$c_6' = 0,510956 + 0,085462 = 0,596418$$

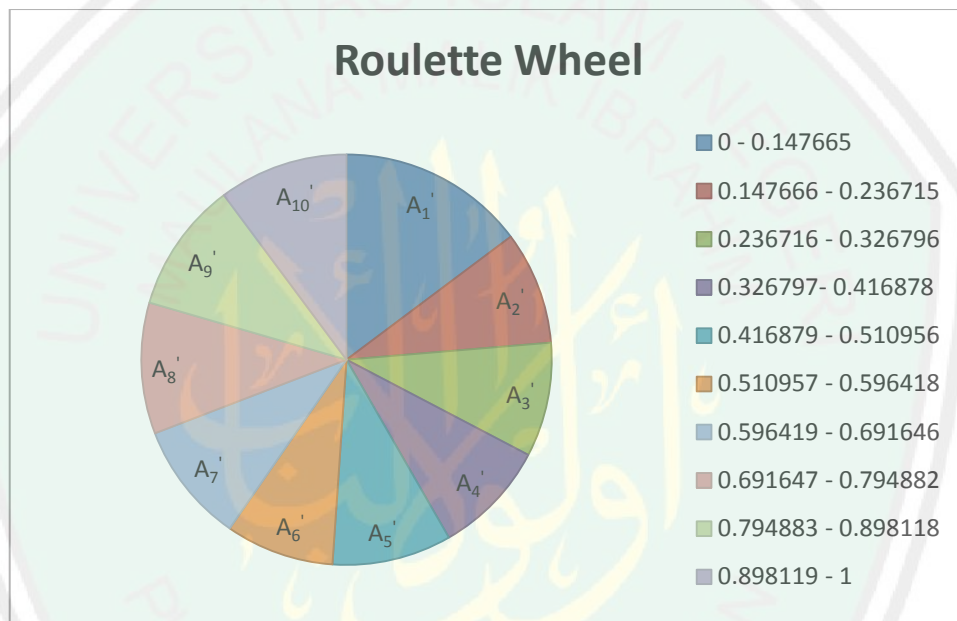
$$c_7' = 0,596418 + 0,095228 = 0,691646$$

$$c_8' = 0,691646 + 0,103236 = 0,794882$$

$$c_9' = 0,794882 + 0,103236 = 0,898118$$

$$c_{10}' = 0,898118 + 0,101882 = 1$$

Probabilitas kumulatif tersebut jika digambarkan dengan *Roulette Wheel* adalah sebagai berikut:



Gambar 3.15: Probabilitas Kumulatif dalam *Roulette Wheel*

Keterangan:

A_i' dengan $i = 1, 2, 3, \dots, 10$ merupakan interval dari c_i' ke c_{i+1}' .

Selanjutnya langkah-langkah seleksi menggunakan metode *Roulette Wheel* adalah sebagai berikut:

1. Bangkitkan bilangan acak $R2$ antara 0 sampai 1 sebanyak kromosom dalam satu populasi.
2. Jika $R2_k < c_i'$ untuk setiap k dan $i = 1, 2, 3, \dots, 10$, maka kromosom ke- k dipilih sebagai *parent*. Selain itu pilih kromosom ke- k sebagai *parent*

dengan syarat $c_{i-1}' < R2_k < c_i'$. Dengan kata lain, jika $R2_k$ berada di dalam A_i' maka kromosom ke- k digantikan kromosom ke- i .

Bilangan acak $R2$ dibangkitkan sebanyak jumlah kromosom dalam satu populasi yaitu 10 bilangan. Seperti pada generasi pertama, perintah untuk membangkitkan sepuluh bilangan acak pada Matlab adalah `rand(1,10)`. Hasilnya adalah sebagai berikut:

[0.0759 0.0540 0.5308 0.7792 0.9340 0.1299 0.5688 0.4694 0.0119 0.3371]

1. $R2_1 = 0,0759$

Nilai $R2_1$ berada di dalam A_1' , sehingga Kromosom 1 tetap menjadi Kromosom 1.

$$Z_1' = Z_1 = (1, 7, 2, 6, 5, 3, 4)$$

2. $R2_2 = 0,0540$

Nilai $R2_2$ berada di dalam A_1' , sehingga Kromosom 2 digantikan oleh Kromosom 1.

$$Z_2' = Z_1 = (1, 7, 2, 6, 5, 3, 4)$$

3. $R2_3 = 0,5308$

Nilai $R2_3$ berada di dalam A_6' , sehingga Kromosom 3 digantikan oleh Kromosom 6.

$$Z_3' = Z_6 = (6, 7, 5, 2, 1, 4, 3)$$

4. $R2_4 = 0,7792$

Nilai $R2_4$ berada di dalam A_8' , sehingga Kromosom 4 digantikan oleh Kromosom 8.

$$Z_4' = Z_8' = (5, 3, 7, 2, 1, 4, 6)$$

5. $R2_5 = 0,9340$

Nilai $R2_5$ berada di dalam A_{10}' , sehingga Kromosom 5 digantikan oleh Kromosom 10.

$$Z_5' = Z_{10}' = (5, 3, 2, 7, 4, 1, 6)$$

6. $R2_6 = 0,1299$

Nilai $R2_6$ berada di dalam A_1' , sehingga Kromosom 6 digantikan oleh Kromosom 1.

$$Z_6' = Z_1' = (1, 7, 2, 6, 5, 3, 4)$$

7. $R2_7 = 0,5688$

Nilai $R2_7$ berada di dalam A_6' , sehingga Kromosom 7 digantikan oleh Kromosom 6.

$$Z_7' = Z_6' = (6, 7, 5, 2, 1, 4, 3)$$

8. $R2_8 = 0,4694$

Nilai $R2_8$ berada di dalam A_5' , sehingga Kromosom 8 digantikan oleh Kromosom 5.

$$Z_8' = Z_5' = (1, 7, 2, 4, 5, 3, 6)$$

9. $R2_9 = 0,0119$

Nilai $R2_9$ berada di dalam A_1' , sehingga Kromosom 9 tetap menjadi Kromosom 1.

$$Z_9' = Z_1' = (1, 7, 2, 6, 5, 3, 4)$$

$$10. R2_{10} = 0,3371$$

Nilai $R2_{10}$ berada di dalam A_4' , sehingga Kromosom 10 tetap menjadi Kromosom 4.

$$Z_{10}' = Z_4' = (1, 4, 2, 7, 3, 6, 5)$$

Dengan demikian, maka populasi baru yang terbentuk adalah:

$$Z_1' = (1, 7, 2, 6, 5, 3, 4)$$

$$Z_2' = (1, 7, 2, 6, 5, 3, 4)$$

$$Z_3' = (6, 7, 5, 2, 1, 4, 3)$$

$$Z_4' = (5, 3, 7, 2, 1, 4, 6)$$

$$Z_5' = (5, 3, 2, 7, 4, 1, 6)$$

$$Z_6' = (1, 7, 2, 6, 5, 3, 4)$$

$$Z_7' = (6, 7, 5, 2, 1, 4, 3)$$

$$Z_8' = (1, 7, 2, 4, 5, 3, 6)$$

$$Z_9' = (1, 7, 2, 6, 5, 3, 4)$$

$$Z_{10}' = (1, 4, 2, 7, 3, 6, 5)$$

Perbedaan populasi sebelum proses seleksi dengan populasi sesudah proses seleksi disajikan pada Tabel 3.6 berikut ini:

Sebelum Proses Seleksi	Jarak	Sesudah Proses Seleksi	Jarak
$Z_1' = (1, 7, 2, 6, 5, 3, 4)$	158	$Z_1' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_2' = (3, 7, 2, 4, 6, 1, 5)$	262	$Z_2' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_3' = (1, 4, 2, 7, 3, 6, 5)$	259	$Z_3' = (6, 7, 5, 2, 1, 4, 3)$	273
$Z_4' = (1, 4, 2, 7, 3, 6, 5)$	259	$Z_4' = (5, 3, 7, 2, 1, 4, 6)$	226
$Z_5' = (1, 7, 2, 4, 5, 3, 6)$	248	$Z_5' = (5, 3, 2, 7, 4, 1, 6)$	229
$Z_6' = (6, 7, 5, 2, 1, 4, 3)$	273	$Z_6' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_7' = (5, 3, 2, 6, 7, 1, 4)$	245	$Z_7' = (6, 7, 5, 2, 1, 4, 3)$	273
$Z_8' = (5, 3, 7, 2, 1, 4, 6)$	226	$Z_8' = (1, 7, 2, 4, 5, 3, 6)$	248
$Z_9' = (5, 3, 7, 2, 1, 4, 6)$	226	$Z_9' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_{10}' = (5, 3, 2, 7, 4, 1, 6)$	229	$Z_{10}' = (1, 4, 2, 7, 3, 6, 5)$	259

Tabel 3.6: Perbandingan Populasi Sebelum dan Sesudah Proses Seleksi

Setelah melakukan proses seleksi menggunakan Metode *Roulette Wheel*, terlihat ada beberapa kromosom yang terseleksi yaitu Kromosom 2 (Z_2'), Kromosom 3 (Z_3'), Kromosom 7 (Z_7') dan Kromosom 9 (Z_9'). Populasi baru yang terbentuk akan digunakan untuk proses selanjutnya yaitu proses *crossover*.

3.3.3 Proses *Crossover* Menggunakan *Partially Matched Crossover*

Probabilitas *crossover* (pc) yang digunakan tetap sama seperti pada generasi pertama yaitu 1 (100%), sehingga seluruh kromosom dalam populasi mengalami *crossover*.

Crossover $Z_1' \times Z_2'$

Crossover antara Z_1' dan Z_2' menghasilkan anak yang sama dengan induknya karena $Z_1' = Z_2'$.

$$Z_1'' = (1, 7, 2, 6, 5, 3, 4)$$

$$Z_2'' = (1, 7, 2, 6, 5, 3, 4)$$

Crossover $Z_3' \times Z_4'$

1. Parent yang akan di-crossover adalah $Z_3' = (6, 7, 5, 2, 1, 4, 3)$ dan

$$Z_4' = (5, 3, 7, 2, 1, 4, 6)$$

Z_3'	6	7	5	2	1	4	3
--------	---	---	---	---	---	---	---

Z_4'	5	3	7	2	1	4	6
--------	---	---	---	---	---	---	---

2. Membuat *substring* secara acak dari kedua *parent*.

Z_3'	6	7	5	2	1	4	3
--------	---	---	---	---	---	---	---

Z_4'	5	3	7	2	1	4	6
--------	---	---	---	---	---	---	---

3. Menukar posisi *substring*.

Z_3'				2	1	4	6
--------	--	--	--	---	---	---	---

\downarrow \downarrow \downarrow \downarrow

Z_4'				2	1	4	3
--------	--	--	--	---	---	---	---

4. Pemetaan yang terjadi adalah:

$$2 \leftrightarrow 2, 1 \leftrightarrow 1, 4 \leftrightarrow 4 \text{ dan } 6 \leftrightarrow 3.$$

5. Isikan bagian-bagian gen sesuai pemetaan.

Z_3''	6	7	5	2	1	4	6
---------	---	---	---	---	---	---	---

Z_4''	5	6	7	2	1	4	3
---------	---	---	---	---	---	---	---

6. Anak yang dihasilkan dari *crossover* Z_3' dan Z_4' di atas adalah:

$$Z_3'' = (6, 7, 5, 2, 1, 4, 6)$$

$$Z_4'' = (5, 6, 7, 2, 1, 4, 3)$$

Crossover $Z_5' \times Z_6'$

1. *Parent* yang akan di-*crossover* adalah $Z_5' = (5, 3, 2, 7, 4, 1, 6)$ dan

$$Z_6' = (1, 7, 2, 6, 5, 3, 4).$$

Z_5'	5	3	2	7	4	1	6
--------	---	---	---	---	---	---	---

Z_6'	1	7	2	6	5	3	4
--------	---	---	---	---	---	---	---

2. Membuat *substring* dari kedua *parent* secara acak.

Z_5'	5	3	2	7	4	1	6
--------	---	---	---	---	---	---	---

Z_6'	1	7	2	6	5	3	4
--------	---	---	---	---	---	---	---

3. Menukar posisi *substring*.

Z_5'	1	7	2				
--------	---	---	---	--	--	--	--



Z_6'	5	3	2				
--------	---	---	---	--	--	--	--

4. Pemetaan yang terjadi adalah:

$$1 \leftrightarrow 5, 7 \leftrightarrow 3, \text{ dan } 2 \leftrightarrow 2.$$

5. Isikan bagian-bagian gen sesuai pemetaan.

Z_5''	1	7	2	3	4	5	6
---------	---	---	---	---	---	---	---

Z_6''	5	3	2	6	1	7	4
---------	---	---	---	---	---	---	---

6. Anak yang dihasilkan dari *crossover* Z_5' dan Z_6' di atas adalah:

$$Z_5'' = (1, 7, 2, 3, 4, 5, 6)$$

$$Z_6'' = (5, 3, 2, 6, 1, 7, 4)$$

Crossover $Z_7' \times Z_8'$

1. *Parent* yang akan di-*crossover* adalah $Z_7' = (6, 7, 5, 2, 1, 4, 3)$ dan

$$Z_8' = (1, 7, 2, 4, 5, 3, 6).$$

Z_7'	6	7	5	2	1	4	3
--------	---	---	---	---	---	---	---

Z_8'	1	7	2	4	5	3	6
--------	---	---	---	---	---	---	---

2. Membuat *substring* dari kedua *parent* secara acak.

Z_7'	6	7	5	2	1	4	3
--------	---	---	---	---	---	---	---

Z_8'	1	7	2	4	5	3	6
--------	---	---	---	---	---	---	---

3. Menukar posisi *substring*.

Z_7'	1	7					
--------	---	---	--	--	--	--	--

↕ ↕

Z_8'	6	7					
--------	---	---	--	--	--	--	--

4. Pemetaan yang terjadi adalah:

$$1 \leftrightarrow 6 \text{ dan } 7 \leftrightarrow 7.$$

5. Isikan bagian-bagian gen sesuai pemetaan.

Z_7''	1	7	5	2	6	4	3
---------	---	---	---	---	---	---	---

Z_8''	6	7	2	4	5	3	1
---------	---	---	---	---	---	---	---

6. Anak yang dihasilkan dari *crossover* Z_7' dan Z_8' di atas adalah:

$$Z_7'' = (1, 7, 5, 2, 6, 4, 3)$$

$$Z_8'' = (6, 7, 2, 4, 5, 3, 1)$$

Crossover $Z_9' \times Z_{10}'$

1. *Parent* yang akan di-*crossover* adalah $Z_9' = (1, 7, 2, 6, 5, 3, 4)$ dan

$$Z_{10}' = (1, 4, 2, 7, 3, 6, 5).$$

Z_9'	1	7	2	6	5	3	4
--------	---	---	---	---	---	---	---

Z_{10}'	1	4	2	7	3	6	5
-----------	---	---	---	---	---	---	---

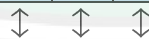
2. Membuat *substring* dari kedua *parent* secara acak.

Z_9'	1	7	2	6	5	3	4
--------	---	---	---	---	---	---	---

Z_{10}'	1	4	2	7	3	6	5
-----------	---	---	---	---	---	---	---

3. Menukar posisi *substring*.

Z_9'			2	7	3		
--------	--	--	---	---	---	--	--



Z_{10}'			2	6	5		
-----------	--	--	---	---	---	--	--

4. Pemetaan yang terjadi adalah:

$$2 \leftrightarrow 2, 7 \leftrightarrow 6 \text{ dan } 3 \leftrightarrow 5.$$

5. Isikan bagian-bagian gen sesuai pemetaan.

Z_9''	1	6	2	7	3	5	4
---------	---	---	---	---	---	---	---

Z_{10}''	1	4	2	6	5	7	3
------------	---	---	---	---	---	---	---

6. Anak yang dihasilkan dari *crossover* Z_9' dan Z_{10}' di atas adalah:

$$Z_9'' = (1, 6, 2, 7, 3, 5, 4)$$

$$Z_{10}'' = (1, 4, 2, 6, 5, 7, 3)$$

Proses *crossover* untuk generasi kedua selengkapnya disajikan pada Lampiran. Setelah dilakukan proses *crossover*, kromosom-kromosom dalam populasi berubah sesuai hasil *crossover*. Dari seluruh kemungkinan *crossover* yang telah dilakukan, diambil kromosom-kromosom anak yang memiliki jarak yang paling kecil dari masing-masing kromosom. Kromosom-kromosom tersebut adalah sebagai berikut:

Sebelum Proses <i>Crossover</i>	Jarak	Sesudah Proses <i>Crossover</i>	Jarak
$Z_1' = (1, 7, 2, 6, 5, 3, 4)$	158	$Z_1'' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_2' = (1, 7, 2, 6, 5, 3, 4)$	158	$Z_2'' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_3' = (6, 7, 5, 2, 1, 4, 3)$	273	$Z_3'' = (5, 3, 2, 7, 1, 4, 6)$	199
$Z_4' = (5, 3, 7, 2, 1, 4, 6)$	226	$Z_4'' = (5, 3, 2, 7, 1, 4, 6)$	199
$Z_5' = (5, 3, 2, 7, 4, 1, 6)$	229	$Z_5'' = (5, 3, 2, 7, 1, 4, 6)$	199
$Z_6' = (1, 7, 2, 6, 5, 3, 4)$	158	$Z_6'' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_7' = (6, 7, 5, 2, 1, 4, 3)$	273	$Z_7'' = (5, 3, 2, 6, 1, 4, 7)$	213
$Z_8' = (1, 7, 2, 4, 5, 3, 6)$	248	$Z_8'' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_9' = (1, 7, 2, 6, 5, 3, 4)$	158	$Z_9'' = (1, 7, 2, 6, 5, 3, 4)$	158
$Z_{10}' = (1, 4, 2, 7, 3, 6, 5)$	259	$Z_{10}'' = (1, 7, 2, 6, 5, 3, 4)$	158

Tabel 3.7: Perbandingan Populasi Sebelum dan Sesudah Proses *Crossover*

Setelah dilakukan proses *crossover* untuk generasi kedua, populasi baru yang didapatkan jarak tempuhnya semakin kecil. Setelah didapatkan populasi dari

proses *crossover* yang telah dilakukan, kemudian selanjutnya dapat dilakukan proses mutasi.

3.3.4 Proses Mutasi

Probabilitas mutasi (pm) yang digunakan tetap seperti pada generasi pertama yaitu 0,01 (1%). Langkah-langkah dalam proses mutasi adalah sebagai berikut:

1. Banyaknya gen dalam satu kromosom adalah 7, dan jumlah kromosom dalam populasi adalah 10. Maka panjang total gen adalah $7 \times 10 = 70$.
2. Jumlah gen yang akan dimutasi adalah $0,01 \times 70 = 0,7 \approx 1$. Dengan demikian akan dibangkitkan sebanyak 1 (satu) bilangan bulat acak antara 1 sampai 70.
3. Pembangkitan bilangan bulat acak dilakukan dengan perintah `randint(1,1,[1,70])` pada Matlab. Bilangan acak yang dihasilkan adalah:

[45]

4. Proses mutasi:

Bilangan bulat acak yang keluar adalah 45, sehingga dilakukan mutasi pada gen ke-45 dengan cara menukar gen tersebut dengan gen sesudahnya. Pada tabel berikut ini terlihat bahwa gen ke-45 terdapat pada kromosom Z_7 . Gen ke-45 yang bernilai 2 bertukar posisi dengan gen setelahnya yang bernilai 6.

Sebelum Proses Mutasi	Jarak	Sesudah Proses Mutasi	Jarak
$Z_1 = (1, 7, 2, 6, 5, 3, 4)$	158	$Z_1 = (1, 7, 2, 6, 5, 3, 4)$	158

	158		158
	199		199
	199		199
	199		199
	158		158
	213		282
	158		158
	158		158
	158		158

Tabel 3.7: Perbandingan Populasi Sebelum dan Sesudah Proses Mutasi

Dari tabel di atas dapat dilihat bahwa jarak tempuh yang paling minimum adalah 158. Kromosom-kromosom yang memiliki jarak 158 tersebut adalah Z_1 ", Z_2 ", Z_6 ", Z_8 " Z_9 ", dan Z_{10} ". Semua kromosom tersebut memiliki rute yang sama yaitu

$Node1 \rightarrow Node7 \rightarrow Node2 \rightarrow Node6 \rightarrow Node5 \rightarrow Node3 \rightarrow Node4 \rightarrow Node1$

Hasil optimasi ini merupakan hasil yang sama dengan proses Algoritma Genetika pada generasi pertama. Sehingga dapat disimpulkan bahwa 158 merupakan total jarak minimumnya.

3.4 Integrasi Algoritma Genetika dengan Islam

Allah SWT berfirman dalam Q.S. Al-Hajj ayat 5 sebagai berikut:

يَأْتِيهَا النَّاسُ إِنْ كُنْتُمْ فِي رَيْبٍ مِّنَ الْبَعْثِ فَإِنَّا خَلَقْنَاكُمْ مِّن تُّرَابٍ ... ﴿٥﴾

Artinya: Hai manusia, jika kamu dalam keraguan tentang kebangkitan (dari kubur), maka (ketahuilah) sesungguhnya Kami telah menjadikan kamu dari tanah,...

Penjelasan penggalan ayat tersebut adalah bahwa jika pada manusia dalam keraguan mengenai kedatangan pembangkitan, maka diperhatikan awal kejadiannya, agar keraguan itu hilang dan manusia mengetahui bahwa Allah SWT menciptakannya untuk kali yang pertama, dan berkuasa pula untuk menciptakan mereka kembali. Pengungkapan keraguan “padahal mereka yakin bahwa pembangkitan itu tidak akan terjadi” menunjukkan bahwa sekalipun mereka telah mencapai puncak kesombongan dan penentangan yang mungkin lahir dari hati mereka adalah keraguan tentang tidak mungkin terjadinya pembangkitan itu sama sekali tidak akan pernah terlintas dalam benak seorang yang berakal (Al-Maraghi, 1993: 145).

Penggalan ayat tersebut memiliki makna yang hampir sama dengan proses inialisasi populasi dalam Algoritma Genetika. Dalam proses Algoritma Genetika yang dibangkitkan adalah kromosom/individu untuk mendapatkan sebuah populasi awal. Sedangkan pada ayat tersebut Allah SWT membangkitkan manusia-manusia yang telah meninggal.

Allah SWT berfirman dalam Q.S. An-Nur ayat 26:

أَخْيَشَاتُ الْخَيْثِثِ وَالْخَيْثُوثِ وَالْخَيْثِثَاتُ لِلطَّيِّبِينَ وَالطَّيِّبَاتُ لِلطَّيِّبِينَ وَالطَّيِّبَاتُ لِلطَّيِّبَاتِ الْمُبْرَّاتِ وَالطَّيِّبَاتُ الْمُبْرَّاتُ لِهِنَّمْ مَغْفِرَةٌ وَرِزْقٌ كَرِيمٌ ﴿٢٦﴾

Artinya: Wanita-wanita yang keji adalah untuk laki-laki yang keji, dan laki-laki yang keji adalah buat wanita-wanita yang keji (pula), dan wanita-wanita yang baik adalah untuk laki-laki yang baik dan laki-laki yang baik

adalah untuk wanita-wanita yang baik (pula). Mereka (yang dituduh) itu bersih dari apa yang dituduhkan oleh mereka (yang menuduh itu). Bagi mereka ampunan dan rezeki yang mulia (surga).

Ayat di atas bermakna tentang kesamaan akhlak antara suami istri. Wanita yang baik adalah lelaki yang baik, dan wanita yang keji adalah bagi lelaki yang keji pula. Rasulullah SAW adalah orang terbaik di antara para lelaki yang baik, maka sudah tentu menurut logika yang sehat dan adat yang tersebar di tengah-tengah makhluk Aisyah Ash-Shiddiqah pun merupakan wanita terbaik di antara para wanita yang baik (Al-Maraghi, 1993: 163).

Ayat di atas memiliki makna yang hampir sama dengan proses seleksi pada Algoritma Genetika. Pada proses seleksi dalam Algoritma Genetika, kemungkinan kromosom/individu yang terpilih adalah yang memiliki nilai *fitness* paling besar atau yang memiliki jarak tempuh paling minimum karena suatu kromosom dikatakan baik atau layak adalah jika memiliki jarak tempuh yang minimum. Jika kromosom yang baik di-*crossover* dengan kromosom yang baik, maka kemungkinan anak yang dihasilkan akan menjadi baik pula.

BAB IV

PENUTUP

4.1 Kesimpulan

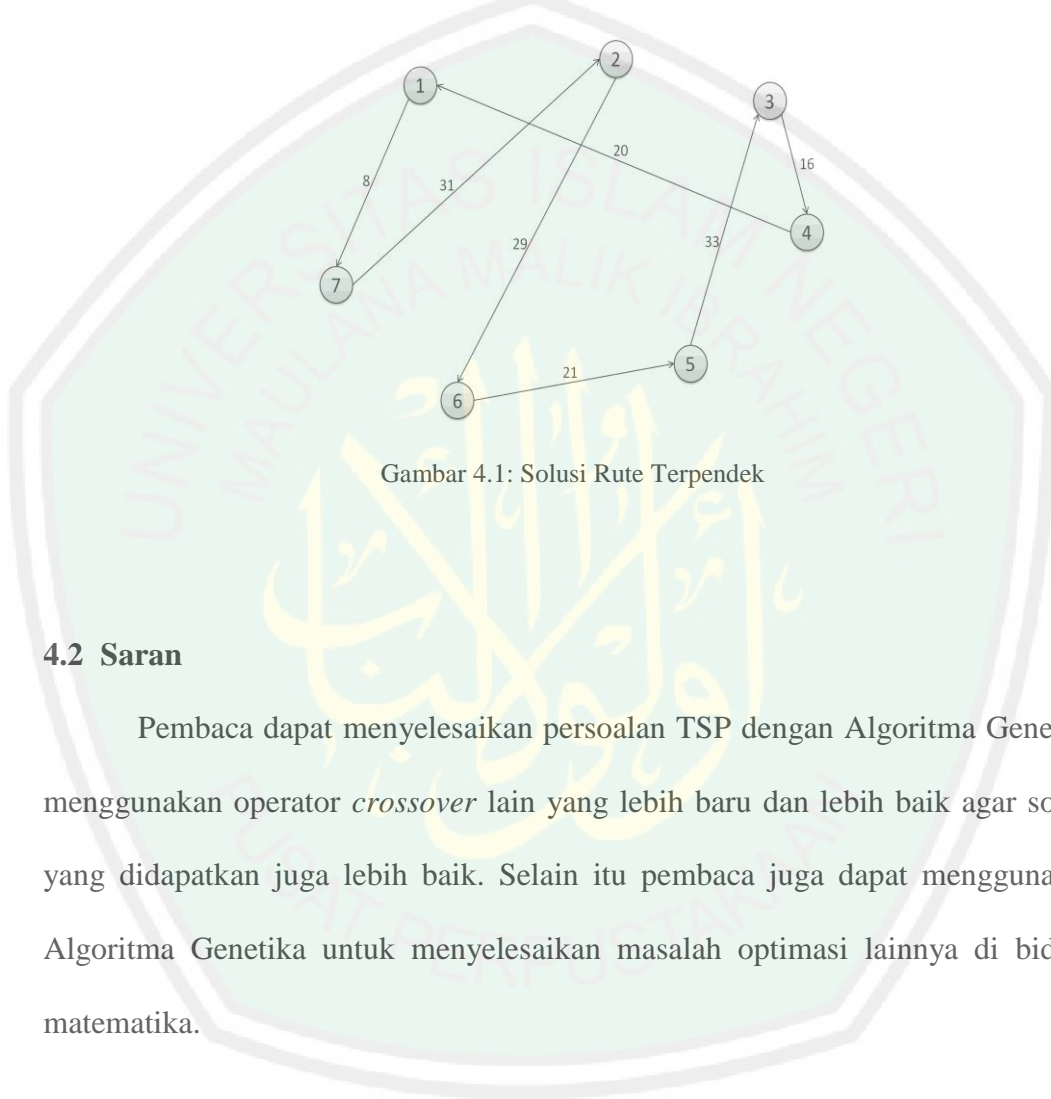
Hasil optimasi TSP menggunakan Algoritma Genetika dengan operator *crossover* PMX menunjukkan solusi optimal pada generasi kedua. Pada generasi pertama, didapatkan populasi awal dengan jarak yang besar. Setelah dilakukan proses seleksi, kromosom-kromosom dengan jarak yang besar terseleksi dan digantikan dengan kromosom yang terpilih dengan menggunakan *Roulette Wheel*. Selanjutnya pada saat proses *crossover* dengan menggunakan operator PMX, terjadi banyak sekali perubahan terhadap kromosom-kromosom tersebut. Proses *crossover* tersebut menghasilkan populasi baru dengan kromosom-kromosom yang memiliki jarak lebih minimum. Selanjutnya pada proses mutasi didapatkan jarak terpendeknya adalah 158 dengan rute $1 \rightarrow 7 \rightarrow 2 \rightarrow 6 \rightarrow 5 \rightarrow 3 \rightarrow 4 \rightarrow 1$. Kromosom yang mewakili rute tersebut adalah Z_1' .

Pada generasi kedua, populasi awal yang terbentuk merupakan populasi akhir yang terjadi pada generasi pertama. Hampir sama dengan generasi pertama, populasi menjadi lebih baik setelah dilakukan proses seleksi, *crossover* dan mutasi. Setelah proses mutasi dilakukan, terdapat enam kromosom yang memiliki jarak yang paling minimum. Jarak minimum tersebut memiliki nilai yang sama dengan generasi pertama yaitu 158.

Oleh karena itu dapat disimpulkan bahwa jarak paling minimum atau jarak terpendek pada kasus TSP ini adalah 158. Rute yang didapatkan adalah:

Node 1 → *Node 7* → *Node 2* → *Node 6* → *Node 5* → *Node 3* → *Node 4*
 → *Node 1*

Gambar rute tersebut adalah sebagai berikut:



Gambar 4.1: Solusi Rute Terpendek

4.2 Saran

Pembaca dapat menyelesaikan persoalan TSP dengan Algoritma Genetika menggunakan operator *crossover* lain yang lebih baru dan lebih baik agar solusi yang didapatkan juga lebih baik. Selain itu pembaca juga dapat menggunakan Algoritma Genetika untuk menyelesaikan masalah optimasi lainnya di bidang matematika.

DAFTAR PUSTAKA

- Ahmed, Z.H.. 2010. *Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Operator*. International Journal of Biometrics and Bioformatics (IJBB). Volume 3: Issue 6.
- Al-Maraghi, A.M.. 1993. *Terjemahan Tafsir Al-Maraghi Jilid 8*. Semarang: CV Toha Putra.
- Al-Maraghi, A.M.. 1993. *Terjemahan Tafsir Al-Maraghi Jilid 17*. Semarang: CV Toha Putra.
- Al-Maraghi, A.M.. 1993. *Terjemahan Tafsir Al-Maraghi Jilid 18*. Semarang: CV Toha Putra.
- Al-Maraghi, A.M.. 1993. *Terjemahan Tafsir Al-Maraghi Jilid 19*. Semarang: CV Toha Putra.
- Coley, D.. 1999. *An Introduction to Genetic Algorithms for Scientists and Engineers*. Singapore: World Scientific Publishing.
- Davendra, D (ed).. 2010. *Traveling Salesman Problem: Theory and Applications*. Rijeka: InTech.
- Faradian, M.I.. 2007. *Perbandingan Penggunaan Algoritma Genetika dengan Algoritma Konvensional pada Traveling Salesman Problem*. Makalah. Institut Teknologi Bandung.
- Fitrah, Aulia., Zaky, Achmad., dan Fitrasani. 2006. *Persoalan Algoritma Genetika pada Persoalan Pedagang Keliling (TSP)*. Bandung: Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.
- Gamal, M.. 2013. *Mempelajari Manajemen Waktu Islami*.
<http://kotasantri.com/pelangi/risalah/2013/06/19/mempelajari-manajemen-waktu-islami> (diakses tanggal 20 Juli 2013).
- Ghoffar, A., Mu'thi, A., dan Al-Atsari, A.I.. 2004. *Tafsir Ibnu Katsir jilid 2*. Bogor: Pustaka Imam Asy-Syafi'i.
- Greco, Frederio (ed). 2008. *Traveling Salesman Problem*. Rijeka: InTech.
- Gutin, G. dan Punnen, A.. 2004. *The Traveling Salesman Problem and Its Variations*. Dordrecht: Kluwer Academic Publishers.

- Ihsan, M.. *Tafsir Surat: ALI-IMRAN*.
http://users6.nofeehost.com/alquranonline/Alquran_Tafsir.asp?pageno=2&SuratKe=3 (diakses tanggal 22 November 2012).
- Kumar, N., Karambir, dan Kumar, R.. 2012. *A Comparative Analysis of PMX, CX and OX Crossover operators for solving Travelling Salesman Problem*. International Journal of Latest Research in Science and Technology Vol.1, Issue 2 :Page No.98-101. ISSN:2278-5299.
- Lawler, E.L., Lenstra, J.K., Rinnoy Kan, A.H.G., & Shmoys, D.B.. 1985. *The Traveling Salesman Problem: A guided Tour of Combinatorial Optimization*. Chicester: John Willey and Sons Ltd.
- Lukas, S., Anwar, T., & Yuliani, W.. 2005. *Penerapan Algoritma Genetika untuk Traveling Salesman Problem dengan Menggunakan Metode Order Crossover dan Insertion Mutation*. Seminar Nasional Aplikasi Teknologi Informasi 2005. ISBN: 979-756-061-6.
- Maredia, A.. 2010. *History, Analysis, and Implementation of Traveling Salesman Problem (TSP) and Related Problems*. Artikel. Department of Computer and Mathematical Sciences University of Houston-Downtown.
- Moengin, P.. 2011. *Metode Optimasi*. Bandung: Muara Indah.
- Mohan, H.. 2008. *Kajian Tafsir Surah Al-Furqan (25):63-77*.
<http://thenafi.wordpress.com/2008/06/13/kajian-tafsir-surah-al-furqan-25-63-77/> (diakses tanggal 21 November 2012).
- Nursi, B.S.. 2011. *Risalah Hidup Hemat dan Sederhana*.
http://tegarrezavie.multiply.com/journal/item/13?&show_interstitial=1&u=%2Fjournal%2Fitem (diakses pada tanggal 20 November 2012).
- Saiyed, A.R.. 2012. *The Traveling Salesman Problem*. Terre Haute: Indiana State University.
- Saputro, N. & Yento. 2004. *Pemakaian Algoritma Genetik untuk Penjadwalan Job Shop Dinamis Non Deterministik*. Jurnal Teknik Industri vol. 6, no. 1, juni 2004: 61 – 70. Bandung: Universitas Katolik Parahyangan.
- Sarwadi. 2004. *Algoritma Genetika untuk Penyelesaian Masalah Vehicle Routing Problem*. Jurnal Matematika dan Komputer. ISSN : 1410-8518.
- Setiawan, A.A.R.. 2010. *Konsep Dasar Traveling Salesman Problem (TSP) dan Aplikasinya untuk Logistik, Transportasi Hingga Pengurutan Genom-DNA*.

<http://blog.sivitas.lipi.go.id/blog.cgi?cetakisiblog&1148506932&&1036006740&&1285194786&arie014&> (diakses pada tanggal 15 November 2012)

Sivanandam, S.N. dan Deepa, S.N.. 2008. *Introduction to Genetic Algorithms*. New York: Springer.

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

Ucoluk, G.. 2011. *Genetic Algorithm Solution of the TSP Avoiding Special Crossover and Mutation*. Department of Computer Engineering Middle East Technical University, Turkey.

Wati, A.W.. 2011. *Penerapan Algoritma Genetika dalam Optimasi Model dan Simulasi dari Suatu Sistem*. Jurnal Teknik Industri. ISSN: 1411-6340.

Yulyantari. 2011. *Artificial Intelligence: Media Pembelajaran Online Mata Kuliah AI*. <http://www.yulyantari.com/tutorial/media.php?mod=materi&bab=5> (diakses pada tanggal 19 November 2012)

Lampiran

Proses *Crossover* pada Generasi Pertama

<i>Crossover</i>		Anak		Jarak
Z ₁	5 3 7 6 2 1 4	Z ₁ '	5 3 7 6 4 1 2	mZ ₁ '=335
Z ₂	4 5 6 3 7 1 2	Z ₂ '	2 5 6 3 7 1 4	mZ ₂ '=393
Z ₁	5 3 7 6 2 1 4	Z ₁ '	1 7 2 6 3 5 4	mZ ₁ '=270
Z ₃	1 7 2 4 6 3 5	Z ₃ '	5 3 7 4 6 2 1	mZ ₃ '=320
Z ₁	5 3 7 6 2 1 4	Z ₁ '	5 7 2 4 6 1 3	mZ ₁ '=365
Z ₄	1 7 2 4 6 3 5	Z ₄ '	1 3 7 6 2 4 5	mZ ₄ '=431
Z ₁	5 3 7 6 2 1 4	Z ₁ '	5 6 4 3 7 1 2	mZ ₁ '=367
Z ₅	4 5 6 3 7 1 2	Z ₅ '	7 5 3 6 2 1 4	mZ ₅ '=282
Z ₁	5 3 7 6 2 1 4	Z ₁ '	6 3 4 1 2 5 7	mZ ₁ '=420
Z ₆	6 3 4 1 2 5 7	Z ₆ '	5 3 7 6 2 1 4	mZ ₆ '=293
Z ₁	5 3 7 6 2 1 4	Z ₁ '	5 3 7 6 2 1 4	mZ ₁ '=293
Z ₇	5 3 7 6 2 1 4	Z ₇ '	5 3 7 6 2 1 4	mZ ₇ '=293
Z ₁	5 3 7 6 2 1 4	Z ₁ '	5 3 7 6 1 4 2	mZ ₁ '=299
Z ₈	7 5 6 2 1 4 3	Z ₈ '	7 5 6 4 2 1 3	mZ ₈ '=382
Z ₁	5 3 7 6 2 1 4	Z ₁ '	7 5 3 6 2 1 4	mZ ₁ '=382
Z ₉	7 5 6 2 1 4 3	Z ₉ '	5 3 6 2 1 4 7	mZ ₉ '=282
Z ₁	5 3 7 6 2 1 4	Z ₁ '	5 2 4 7 3 1 6	mZ ₁ '=282
Z ₁₀	4 5 2 7 3 1 6	Z ₁₀ '	7 5 3 6 2 1 4	mZ ₁₀ '=385
Z ₂	4 5 6 3 7 1 2	Z ₂ '	5 3 7 4 6 1 2	mZ ₂ '=380
Z ₁	5 3 7 4 6 1 2	Z ₁ '	4 5 6 7 2 1 3	mZ ₁ '=384
Z ₂	4 5 6 3 7 1 2	Z ₂ '	3 7 2 4 6 1 5	mZ ₂ '=262
Z ₃	1 7 2 4 6 3 5	Z ₃ '	1 5 6 3 7 4 2	mZ ₃ '=391
Z ₂	4 5 6 3 7 1 2	Z ₂ '	1 2 7 4 6 3 5	mZ ₂ '=398
Z ₄	1 7 2 4 6 3 5	Z ₄ '	4 6 5 3 7 1 2	mZ ₄ '=314

Z_2	4	5	6	3	7	1	2	Z_2'	4	5	6	3	7	1	2	$mZ_2' = 431$
Z_5	4	5	6	3	7	1	2	Z_5'	4	5	6	3	7	1	2	$mZ_5' = 431$
Z_2	4	5	6	3	7	1	2	Z_2'	6	3	4	5	7	1	2	$mZ_2' = 398$
Z_6	6	3	4	1	2	5	7	Z_6'	4	5	6	1	2	3	7	$mZ_6' = 427$
Z_2	4	5	6	3	7	1	2	Z_2'	4	5	6	1	2	3	7	$mZ_2' = 427$
Z_7	5	3	7	6	2	1	4	Z_7'	5	3	2	6	7	1	4	$mZ_7' = 245$
Z_2	4	5	6	3	7	1	2	Z_2'	7	5	6	3	4	1	2	$mZ_2' = 348$
Z_8	7	5	6	2	1	4	3	Z_8'	4	5	6	2	1	7	3	$mZ_8' = 306$
Z_2	4	5	6	3	7	1	2	Z_2'	7	5	6	2	1	4	3	$mZ_2' = 280$
Z_9	7	5	6	2	1	4	3	Z_9'	4	5	6	3	7	1	2	$mZ_9' = 314$
Z_2	4	5	6	3	7	1	2	Z_2'	4	5	2	7	3	1	6	$mZ_2' = 379$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	4	5	6	3	7	1	2	$mZ_{10}' = 314$
Z_3	1	7	2	4	6	3	5	Z_3'	5	3	7	4	6	2	1	$mZ_3' = 320$
Z_1	5	3	7	4	6	1	2	Z_1'	1	7	2	4	6	5	3	$mZ_1' = 292$
Z_3	1	7	2	4	6	3	5	Z_3'	1	5	6	3	7	4	2	$mZ_3' = 391$
Z_2	4	5	6	3	7	1	2	Z_2'	3	7	2	4	6	1	5	$mZ_2' = 361$
Z_3	1	7	2	4	6	3	5	Z_3'	1	7	2	4	6	3	5	$mZ_3' = 341$
Z_4	1	7	2	4	6	3	5	Z_4'	1	7	2	4	6	3	5	$mZ_4' = 341$
Z_3	1	7	2	4	6	3	5	Z_3'	4	5	6	3	2	1	7	$mZ_3' = 355$
Z_5	4	5	6	3	7	1	2	Z_5'	1	7	2	4	5	3	6	$mZ_5' = 248$
Z_3	1	7	2	4	6	3	5	Z_3'	6	3	4	2	1	7	5	$mZ_3' = 337$
Z_6	6	3	4	1	2	5	7	Z_6'	1	7	2	6	4	5	3	$mZ_6' = 291$
Z_3	1	7	2	4	6	3	5	Z_3'	3	7	6	4	2	1	5	$mZ_3' = 283$
Z_7	5	3	7	6	2	1	4	Z_7'	5	1	7	2	6	3	4	$mZ_7' = 318$
Z_3	1	7	2	4	6	3	5	Z_3'	7	5	2	4	6	3	1	$mZ_3' = 411$
Z_8	7	5	6	2	1	4	3	Z_8'	1	7	6	2	5	4	3	$mZ_8' = 385$
Z_3	1	7	2	4	6	3	5	Z_3'	6	7	5	2	1	4	3	$mZ_3' = 273$
Z_9	7	5	6	2	1	4	3	Z_9'	7	2	1	4	6	3	5	$mZ_9' = 333$

Z_3	1	7	2	4	6	3	5	Z_3'	1	4	2	7	3	6	5	$mZ_3' = 259$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	7	5	2	4	6	1	3	$mZ_{10}' = 377$
Z_4	1	7	2	4	6	3	5	Z_4'	5	3	7	4	6	2	1	$mZ_4' = 320$
Z_1	5	3	7	4	6	1	2	Z_1'	1	7	2	4	6	5	3	$mZ_1' = 292$
Z_4	1	7	2	4	6	3	5	Z_4'	1	5	6	3	7	4	2	$mZ_4' = 391$
Z_2	4	5	6	3	7	1	2	Z_2'	3	7	2	4	6	1	5	$mZ_2' = 262$
Z_4	1	7	2	4	6	3	5	Z_4'	1	7	2	4	6	3	5	$mZ_4' = 341$
Z_3	1	7	2	4	6	3	5	Z_3'	1	7	2	4	6	3	5	$mZ_3' = 341$
Z_4	1	7	2	4	6	3	5	Z_4'	4	5	6	3	2	1	7	$mZ_4' = 355$
Z_5	4	5	6	3	7	1	2	Z_5'	1	7	2	4	5	3	6	$mZ_5' = 248$
Z_4	1	7	2	4	6	3	5	Z_4'	6	3	4	2	1	7	5	$mZ_4' = 337$
Z_6	6	3	4	1	2	5	7	Z_6'	1	7	2	6	4	5	3	$mZ_6' = 291$
Z_4	1	7	2	4	6	3	5	Z_4'	3	7	6	4	2	1	5	$mZ_4' = 283$
Z_7	5	3	7	6	2	1	4	Z_7'	5	1	7	2	6	3	4	$mZ_7' = 318$
Z_4	1	7	2	4	6	3	5	Z_4'	7	5	2	4	6	3	1	$mZ_4' = 411$
Z_8	7	5	6	2	1	4	3	Z_8'	1	7	6	2	5	4	3	$mZ_8' = 385$
Z_4	1	7	2	4	6	3	5	Z_4'	6	7	5	2	1	4	3	$mZ_4' = 273$
Z_9	7	5	6	2	1	4	3	Z_9'	7	2	1	4	6	3	5	$mZ_9' = 333$
Z_4	1	7	2	4	6	3	5	Z_4'	1	4	2	7	3	6	5	$mZ_4' = 259$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	7	5	2	4	6	1	3	$mZ_{10}' = 377$
Z_5	4	5	6	3	7	1	2	Z_5'	5	3	7	4	6	1	2	$mZ_5' = 380$
Z_1	5	3	7	4	6	1	2	Z_1'	4	5	6	3	7	1	2	$mZ_1' = 431$
Z_5	4	5	6	3	7	1	2	Z_5'	4	5	6	3	7	1	2	$mZ_5' = 431$
Z_2	4	5	6	3	7	1	2	Z_2'	4	5	6	3	7	1	2	$mZ_2' = 431$
Z_5	4	5	6	3	7	1	2	Z_5'	1	2	7	4	6	3	5	$mZ_5' = 418$
Z_3	1	7	2	4	6	3	5	Z_3'	4	6	5	3	7	1	2	$mZ_3' = 431$

Z_5	4	5	6	3	7	1	2	Z_5'	1	7	2	4	5	3	6	$mZ_5' = 248$
Z_4	1	7	2	4	6	3	5	Z_4'	4	5	6	3	2	1	7	$mZ_4' = 355$
Z_5	4	5	6	3	7	1	2	Z_5'	6	3	4	5	7	1	2	$mZ_5' = 398$
Z_6	6	3	4	1	2	5	7	Z_6'	4	5	6	1	2	3	7	$mZ_6' = 427$
Z_5	4	5	6	3	7	1	2	Z_5'	4	5	6	3	2	1	7	$mZ_5' = 355$
Z_7	5	3	7	6	2	1	4	Z_7'	5	3	2	6	7	1	4	$mZ_7' = 245$
Z_5	4	5	6	3	7	1	2	Z_5'	7	5	6	3	4	1	2	$mZ_5' = 348$
Z_8	7	5	6	2	1	4	3	Z_8'	4	5	6	2	1	7	3	$mZ_8' = 306$
Z_5	4	5	6	3	7	1	2	Z_5'	7	5	6	2	1	4	3	$mZ_5' = 280$
Z_9	7	5	6	2	1	4	3	Z_9'	4	5	6	3	7	1	2	$mZ_9' = 431$
Z_5	4	5	6	3	7	1	2	Z_5'	4	5	2	7	3	1	6	$mZ_5' = 379$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	4	5	6	3	7	1	2	$mZ_{10}' = 431$
Z_6	6	3	4	1	2	5	7	Z_6'	5	3	7	1	2	6	4	$mZ_6' = 313$
Z_1	5	3	7	4	6	1	2	Z_1'	6	3	4	7	5	1	2	$mZ_1' = 396$
Z_6	6	3	4	1	2	5	7	Z_6'	4	5	6	3	7	1	2	$mZ_6' = 431$
Z_2	4	5	6	3	7	1	2	Z_2'	6	3	4	1	2	3	7	$mZ_2' = 374$
Z_6	6	3	4	1	2	5	7	Z_6'	2	7	1	4	6	3	5	$mZ_6' = 320$
Z_3	1	7	2	4	6	3	5	Z_3'	4	5	6	1	2	5	7	$mZ_3' = 473$
Z_6	6	3	4	1	2	5	7	Z_6'	1	7	2	4	6	5	3	$mZ_6' = 292$
Z_4	1	7	2	4	6	3	5	Z_4'	6	3	4	1	2	7	5	$mZ_4' = 348$
Z_6	6	3	4	1	2	5	7	Z_6'	4	5	6	1	2	3	7	$mZ_6' = 427$
Z_5	4	5	6	3	7	1	2	Z_5'	6	3	4	5	7	1	2	$mZ_5' = 398$
Z_6	6	3	4	1	2	5	7	Z_6'	6	3	4	5	2	1	7	$mZ_6' = 346$
Z_7	5	3	7	6	2	1	4	Z_7'	1	3	7	6	2	5	4	$mZ_7' = 413$
Z_6	6	3	4	1	2	5	7	Z_6'	7	5	6	1	2	3	4	$mZ_6' = 390$
Z_8	7	5	6	2	1	4	3	Z_8'	6	3	4	2	1	7	5	$mZ_8' = 337$
Z_6	6	3	4	1	2	5	7	Z_6'	6	7	5	2	1	4	3	$mZ_6' = 273$
Z_9	7	5	6	2	1	4	3	Z_9'	3	4	6	1	2	5	7	$mZ_9' = 383$

Z_6	6	3	4	1	2	5	7	Z_6'	6	4	2	7	3	5	1	$mZ_6' = 316$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	3	5	4	1	2	7	6	$mZ_{10}' = 357$
Z_7	5	3	7	6	2	1	4	Z_7'	5	3	7	6	2	1	4	$mZ_7' = 293$
Z_1	5	3	7	4	6	1	2	Z_1'	5	3	7	4	6	1	2	$mZ_1' = 380$
Z_7	5	3	7	6	2	1	4	Z_7'	2	5	6	3	7	1	4	$mZ_7' = 393$
Z_2	4	5	6	3	7	1	2	Z_2'	4	3	7	6	2	1	5	$mZ_2' = 303$
Z_7	5	3	7	6	2	1	4	Z_7'	2	1	7	4	6	3	5	$mZ_7' = 359$
Z_3	1	7	2	4	6	3	5	Z_3'	3	7	5	6	2	1	4	$mZ_3' = 280$
Z_7	5	3	7	6	2	1	4	Z_7'	1	7	2	4	3	5	6	$mZ_7' = 236$
Z_4	1	7	2	4	6	3	5	Z_4'	5	3	7	6	4	2	1	$mZ_4' = 283$
Z_7	5	3	7	6	2	1	4	Z_7'	4	5	6	7	2	1	3	$mZ_7' = 384$
Z_5	4	5	6	3	7	1	2	Z_5'	5	3	7	4	6	1	2	$mZ_5' = 380$
Z_7	5	3	7	6	2	1	4	Z_7'	1	3	7	6	2	5	4	$mZ_7' = 413$
Z_6	6	3	4	1	2	5	7	Z_6'	6	3	4	5	2	1	7	$mZ_6' = 346$
Z_7	5	3	7	6	2	1	4	Z_7'	7	5	3	6	2	1	4	$mZ_7' = 282$
Z_8	7	5	6	2	1	4	3	Z_8'	5	3	6	2	1	4	7	$mZ_8' = 282$
Z_7	5	3	7	6	2	1	4	Z_7'	5	6	7	2	1	4	3	$mZ_7' = 258$
Z_9	7	5	6	2	1	4	3	Z_9'	7	5	3	6	2	1	4	$mZ_9' = 282$
Z_7	5	3	7	6	2	1	4	Z_7'	5	6	2	7	3	1	4	$mZ_7' = 360$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	4	5	7	6	2	1	3	$mZ_{10}' = 410$
Z_8	7	5	6	2	1	4	3	Z_8'	5	3	7	2	1	4	6	$mZ_8' = 226$
Z_1	5	3	7	4	6	1	2	Z_1'	7	5	6	4	3	1	2	$mZ_1' = 365$
Z_8	7	5	6	2	1	4	3	Z_8'	1	5	6	3	7	4	2	$mZ_8' = 391$
Z_2	4	5	6	3	7	1	2	Z_2'	4	5	6	2	1	7	3	$mZ_2' = 306$
Z_8	7	5	6	2	1	4	3	Z_8'	7	2	1	4	6	3	5	$mZ_8' = 333$
Z_3	1	7	2	4	6	3	5	Z_3'	6	7	5	2	1	4	3	$mZ_3' = 273$

Z_8	7	5	6	2	1	4	3	Z_8'	1	7	2	4	5	6	3	$mZ_8' = 409$
Z_4	1	7	2	4	6	3	5	Z_4'	7	5	6	2	4	3	1	$mZ_4' = 346$
Z_8	7	5	6	2	1	4	3	Z_8'	4	5	6	2	1	7	3	$mZ_8' = 306$
Z_5	4	5	6	3	7	1	2	Z_5'	7	5	6	3	4	1	2	$mZ_5' = 348$
Z_8	7	5	6	2	1	4	3	Z_8'	7	4	6	1	2	5	3	$mZ_8' = 380$
Z_6	6	3	4	1	2	5	7	Z_6'	6	3	5	2	1	4	7	$mZ_6' = 349$
Z_8	7	5	6	2	1	4	3	Z_8'	5	3	6	2	1	4	7	$mZ_8' = 282$
Z_7	5	3	7	6	2	1	4	Z_7'	7	5	3	6	2	1	4	$mZ_7' = 282$
Z_8	7	5	6	2	1	4	3	Z_8'	7	5	6	2	1	4	3	$mZ_8' = 280$
Z_9	7	5	6	2	1	4	3	Z_9'	7	5	6	2	1	4	3	$mZ_9' = 280$
Z_8	7	5	6	2	1	4	3	Z_8'	6	5	2	7	3	4	1	$mZ_8' = 246$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	4	5	6	2	1	3	7	$mZ_{10}' = 433$
Z_9	7	5	6	2	1	4	3	Z_9'	5	3	7	2	1	4	6	$mZ_9' = 226$
Z_1	5	3	7	4	6	1	2	Z_1'	7	5	6	4	3	1	2	$mZ_1' = 365$
Z_9	7	5	6	2	1	4	3	Z_9'	1	5	6	3	7	4	2	$mZ_9' = 391$
Z_2	4	5	6	3	7	1	2	Z_2'	4	5	6	2	1	7	3	$mZ_2' = 306$
Z_9	7	5	6	2	1	4	3	Z_9'	7	2	1	4	6	3	5	$mZ_9' = 338$
Z_3	1	7	2	4	6	3	5	Z_3'	6	7	5	2	1	4	3	$mZ_3' = 273$
Z_9	7	5	6	2	1	4	3	Z_9'	1	7	2	4	5	6	3	$mZ_9' = 409$
Z_4	1	7	2	4	6	3	5	Z_4'	7	5	6	2	4	3	1	$mZ_4' = 346$
Z_9	7	5	6	2	1	4	3	Z_9'	4	5	6	2	1	7	3	$mZ_9' = 306$
Z_5	4	5	6	3	7	1	2	Z_5'	7	5	6	3	4	1	2	$mZ_5' = 348$
Z_9	7	5	6	2	1	4	3	Z_9'	7	4	6	1	2	5	3	$mZ_9' = 380$
Z_6	6	3	4	1	2	5	7	Z_6'	6	3	5	2	1	4	7	$mZ_6' = 349$
Z_9	7	5	6	2	1	4	3	Z_9'	5	3	6	2	1	4	7	$mZ_9' = 282$
Z_7	5	3	7	6	2	1	4	Z_7'	7	5	3	6	2	1	4	$mZ_7' = 282$
Z_9	7	5	6	2	1	4	3	Z_9'	7	5	6	2	1	4	3	$mZ_9' = 280$
Z_8	7	5	6	2	1	4	3	Z_8'	7	5	6	2	1	4	3	$mZ_8' = 280$

Z_9	7	5	6	2	1	4	3	Z_9'	6	5	2	7	3	4	1	$mZ_9' = 246$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	4	5	6	2	1	3	7	$mZ_{10}' = 433$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	5	3	7	2	4	1	6	$mZ_{10}' = 264$
Z_1	5	3	7	4	6	1	2	Z_1'	4	5	2	3	6	1	7	$mZ_1' = 354$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	4	5	6	3	7	1	2	$mZ_{10}' = 431$
Z_2	4	5	6	3	7	1	2	Z_2'	4	5	2	7	3	1	6	$mZ_2' = 379$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	7	1	2	4	6	3	5	$mZ_{10}' = 421$
Z_3	1	7	2	4	6	3	5	Z_3'	5	4	2	7	3	1	6	$mZ_3' = 357$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	1	7	2	4	3	5	6	$mZ_{10}' = 236$
Z_4	1	7	2	4	6	3	5	Z_4'	4	5	2	7	6	3	1	$mZ_4' = 400$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	4	5	6	7	3	1	2	$mZ_{10}' = 451$
Z_5	4	5	6	3	7	1	2	Z_5'	4	5	2	3	7	1	6	$mZ_5' = 388$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	4	1	3	7	2	5	6	$mZ_{10}' = 373$
Z_6	6	3	4	1	2	5	7	Z_6'	6	2	4	5	3	1	7	$mZ_6' = 389$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	5	3	2	7	4	1	6	$mZ_{10}' = 229$
Z_7	5	3	7	6	2	1	4	Z_7'	4	5	7	6	2	1	3	$mZ_7' = 411$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	6	5	7	2	1	4	3	$mZ_{10}' = 230$
Z_8	7	5	6	2	1	4	3	Z_8'	2	5	4	7	3	1	6	$mZ_8' = 461$
Z_{10}	4	5	2	7	3	1	6	Z_{10}'	4	5	6	2	1	3	7	$mZ_{10}' = 433$
Z_9	7	5	6	2	1	4	3	Z_9'	6	5	2	7	3	4	1	$mZ_9' = 246$

Lampiran

Proses *Crossover* pada Generasi Kedua

Crossover		Anak		Jarak
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	1 7 2 6 5 3 4	mZ ₁ " =158
Z ₂ '	1 7 2 6 5 3 4	Z ₂ "	1 7 2 6 5 3 4	mZ ₂ " =158
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	6 7 5 1 2 3 4	mZ ₁ " =420
Z ₃ '	6 7 5 2 1 4 3	Z ₃ "	1 7 2 5 6 4 3	mZ ₃ " =345
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	5 3 7 2 1 6 4	mZ ₁ " =296
Z ₄ '	5 3 7 2 1 4 6	Z ₄ "	1 7 2 6 5 4 2	mZ ₄ " =250
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	3 5 2 7 4 1 6	mZ ₁ " =350
Z ₅ '	5 3 2 7 4 1 6	Z ₅ "	7 1 2 6 5 3 4	mZ ₅ " =281
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	1 7 2 6 5 3 4	mZ ₁ " =158
Z ₆ '	1 7 2 6 5 3 4	Z ₆ "	1 7 2 6 5 3 4	mZ ₆ " =158
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	6 7 5 1 2 3 4	mZ ₁ " =420
Z ₇ '	6 7 5 2 1 4 3	Z ₇ "	1 7 2 5 6 4 3	mZ ₇ " =345
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	1 7 2 6 5 3 4	mZ ₁ " =158
Z ₈ '	1 7 2 4 5 3 6	Z ₈ "	1 7 2 4 5 3 6	mZ ₈ " =248
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	1 7 2 6 5 3 4	mZ ₁ " =158
Z ₉ '	1 7 2 6 5 3 4	Z ₉ "	1 7 2 6 5 3 4	mZ ₉ " =158
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	1 4 2 7 3 6 5	mZ ₁ " =259
Z ₁₀ '	1 4 2 7 3 6 5	Z ₁₀ "	1 7 2 6 5 3 4	mZ ₁₀ " =158
Z ₂ '	1 7 2 6 5 3 4	Z ₂ "	1 7 2 6 5 3 4	mZ ₂ " =158
Z ₁ '	1 7 2 6 5 3 4	Z ₁ "	1 7 2 6 5 3 4	mZ ₁ " =158
Z ₂ '	1 7 2 6 5 3 4	Z ₂ "	6 7 5 2 1 3 4	mZ ₂ " =386
Z ₃ '	6 7 5 2 1 4 3	Z ₃ "	1 7 2 6 5 4 3	mZ ₃ " =265
Z ₂ '	1 7 2 6 5 3 4	Z ₂ "	5 7 3 2 1 4 6	mZ ₂ " =254
Z ₄ '	5 3 7 2 1 4 6	Z ₄ "	1 2 7 6 5 3 4	mZ ₄ " =245

Z_2'	1	7	2	6	5	3	4	Z_2''	5	3	2	7	1	6	4	$mZ_2'' = 269$
Z_5'	5	3	2	7	4	1	6	Z_5''	1	7	2	6	4	5	3	$mZ_5'' = 291$
Z_2'	1	7	2	6	5	3	4	Z_2''	1	7	2	6	5	3	4	$mZ_2'' = 158$
Z_6'	1	7	2	6	5	3	4	Z_6''	1	7	2	6	5	3	4	$mZ_6'' = 158$
Z_2'	1	7	2	6	5	3	4	Z_2''	5	7	2	6	1	4	3	$mZ_2'' = 216$
Z_7'	6	7	5	2	1	4	3	Z_7''	6	7	1	2	5	3	4	$mZ_7'' = 375$
Z_2'	1	7	2	6	5	3	4	Z_2''	1	7	2	6	5	3	4	$mZ_2'' = 158$
Z_8'	1	7	2	4	5	3	6	Z_8''	1	7	2	4	5	3	6	$mZ_8'' = 248$
Z_2'	1	7	2	6	5	3	4	Z_2''	1	7	2	6	5	3	4	$mZ_2'' = 158$
Z_9'	1	7	2	6	5	3	4	Z_9''	1	7	2	6	5	3	4	$mZ_9'' = 158$
Z_2'	1	7	2	6	5	3	4	Z_2''	1	6	2	7	3	5	4	$mZ_2'' = 292$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	4	2	6	5	7	3	$mZ_{10}'' = 319$
Z_3'	6	7	5	2	1	4	3	Z_3''	1	7	2	5	6	4	3	$mZ_3'' = 345$
Z_1'	1	7	2	6	5	3	4	Z_1''	6	7	5	1	2	3	4	$mZ_1'' = 420$
Z_3'	6	7	5	2	1	4	3	Z_3''	1	7	2	6	5	4	3	$mZ_3'' = 265$
Z_2'	1	7	2	6	5	3	4	Z_2''	6	7	5	2	1	3	4	$mZ_2'' = 386$
Z_3'	6	7	5	2	1	4	3	Z_3''	3	7	5	2	1	4	6	$mZ_3'' = 345$
Z_4'	5	3	7	2	1	4	6	Z_4''	5	6	7	2	1	4	3	$mZ_4'' = 258$
Z_3'	6	7	5	2	1	4	3	Z_3''	5	3	2	7	1	4	6	$mZ_3'' = 178$
Z_5'	5	3	2	7	4	1	6	Z_5''	6	7	5	2	4	1	3	$mZ_5'' = 367$
Z_3'	6	7	5	2	1	4	3	Z_3''	1	7	2	5	6	4	3	$mZ_3'' = 345$
Z_6'	1	7	2	6	5	3	4	Z_6''	6	7	5	1	2	3	4	$mZ_6'' = 420$
Z_3'	6	7	5	2	1	4	3	Z_3''	6	7	5	2	1	4	3	$mZ_3'' = 273$
Z_7'	6	7	5	2	1	4	3	Z_7''	6	7	5	2	1	4	3	$mZ_7'' = 273$
Z_3'	6	7	5	2	1	4	3	Z_3''	1	7	5	2	6	4	3	$mZ_3'' = 294$
Z_8'	1	7	2	4	5	3	6	Z_8''	6	7	2	4	5	3	1	$mZ_8'' = 384$
Z_3'	6	7	5	2	1	4	3	Z_3''	2	7	1	6	5	3	4	$mZ_3'' = 256$
Z_9'	1	7	2	6	5	3	4	Z_9''	5	7	6	2	1	4	3	$mZ_9'' = 284$

Z_3'	6	7	5	2	1	4	3	Z_3''	6	5	2	7	3	4	1	$mZ_3'' = 246$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	3	4	5	2	1	6	7	$mZ_{10}'' = 347$
Z_4'	5	3	7	2	1	4	6	Z_4''	1	7	2	3	5	4	6	$mZ_4'' = 307$
Z_1'	1	7	2	6	5	3	4	Z_1''	5	3	7	6	1	2	4	$mZ_1'' = 337$
Z_4'	5	3	7	2	1	4	6	Z_4''	1	7	2	6	5	4	3	$mZ_4'' = 345$
Z_2'	1	7	2	6	5	3	4	Z_2''	5	3	7	2	1	6	4	$mZ_2'' = 296$
Z_4'	5	3	7	2	1	4	6	Z_4''	5	6	7	2	1	4	3	$mZ_4'' = 258$
Z_3'	6	5	7	2	1	4	3	Z_3''	3	5	7	2	1	4	6	$mZ_3'' = 333$
Z_4'	5	3	7	2	1	4	6	Z_4''	5	3	2	7	1	4	6	$mZ_4'' = 199$
Z_5'	5	3	2	7	4	1	6	Z_5''	5	3	7	2	4	1	6	$mZ_5'' = 242$
Z_4'	5	3	7	2	1	4	6	Z_4''	1	7	2	3	5	4	6	$mZ_4'' = 307$
Z_6'	1	7	2	6	5	3	4	Z_6''	5	3	7	6	1	2	4	$mZ_6'' = 337$
Z_4'	5	3	7	2	1	4	6	Z_4''	5	3	7	2	1	4	6	$mZ_4'' = 226$
Z_7'	6	7	5	2	1	4	3	Z_7''	6	7	5	2	1	4	3	$mZ_7'' = 273$
Z_4'	5	3	7	2	1	4	6	Z_4''	1	7	3	2	5	4	6	$mZ_4'' = 298$
Z_8'	1	7	2	6	5	3	4	Z_8''	5	3	2	6	1	7	4	$mZ_8'' = 237$
Z_4'	5	3	7	2	1	4	6	Z_4''	1	2	7	6	5	3	4	$mZ_4'' = 245$
Z_9'	1	7	2	6	5	3	4	Z_9''	5	7	3	2	1	4	6	$mZ_9'' = 254$
Z_4'	5	3	7	2	1	4	6	Z_4''	5	1	2	7	3	4	6	$mZ_4'' = 314$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	3	4	7	2	1	6	5	$mZ_{10}'' = 264$
Z_5'	5	3	2	7	4	1	6	Z_5''	1	7	2	3	4	5	6	$mZ_5'' = 312$
Z_1'	1	7	2	6	5	3	4	Z_1''	5	3	2	6	1	7	4	$mZ_1'' = 237$
Z_5'	5	3	2	7	4	1	6	Z_5''	4	7	2	6	5	1	3	$mZ_5'' = 331$
Z_2'	1	7	2	6	5	3	4	Z_2''	1	3	2	7	4	6	5	$mZ_2'' = 351$
Z_5'	5	3	2	7	4	1	6	Z_5''	5	6	7	2	1	4	3	$mZ_5'' = 258$
Z_3'	6	7	5	2	1	4	3	Z_3''	3	2	5	7	4	1	6	$mZ_3'' = 404$

Z_5'	5	3	2	7	4	1	6	Z_5''	5	3	7	2	4	1	6	$mZ_5'' = 242$
Z_4'	5	3	7	2	1	4	6	Z_4''	5	3	2	7	1	4	6	$mZ_4'' = 199$
Z_5'	5	3	2	7	4	1	6	Z_5''	1	7	2	3	4	5	6	$mZ_5'' = 312$
Z_6'	1	7	2	6	5	3	4	Z_6''	5	3	2	6	1	7	4	$mZ_6'' = 237$
Z_5'	5	3	2	7	4	1	6	Z_5''	5	3	2	7	1	4	6	$mZ_5'' = 199$
Z_7'	6	7	5	2	1	4	3	Z_7''	6	7	5	2	4	1	3	$mZ_7'' = 367$
Z_5'	5	3	2	7	4	1	6	Z_5''	1	7	2	3	4	5	6	$mZ_5'' = 312$
Z_8'	1	7	2	4	5	3	6	Z_8''	5	3	2	4	1	7	6	$mZ_8'' = 193$
Z_5'	5	3	2	7	4	1	6	Z_5''	7	1	2	6	5	3	4	$mZ_5'' = 281$
Z_9'	1	7	2	6	5	3	4	Z_9''	3	5	2	7	4	1	6	$mZ_9'' = 350$
Z_5'	5	3	2	7	4	1	6	Z_5''	5	4	2	7	3	1	6	$mZ_5'' = 357$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	3	2	7	4	6	5	$mZ_{10}'' = 351$
Z_6'	1	7	2	6	5	3	4	Z_6''	1	7	2	6	5	3	4	$mZ_6'' = 158$
Z_1'	1	7	2	6	5	3	4	Z_1''	1	7	2	6	5	3	4	$mZ_1'' = 158$
Z_6'	1	7	2	6	5	3	4	Z_6''	1	7	2	6	5	3	4	$mZ_6'' = 158$
Z_2'	1	7	2	6	5	3	4	Z_2''	1	7	2	6	5	3	4	$mZ_2'' = 158$
Z_6'	1	7	2	6	5	3	4	Z_6''	5	7	6	2	1	4	3	$mZ_6'' = 284$
Z_3'	6	7	5	2	1	4	3	Z_3''	2	7	1	6	5	3	4	$mZ_3'' = 256$
Z_6'	1	7	2	6	5	3	4	Z_6''	5	3	7	2	1	6	4	$mZ_6'' = 296$
Z_4'	5	3	7	2	1	4	6	Z_4''	1	7	2	6	5	4	3	$mZ_4'' = 265$
Z_6'	1	7	2	6	5	3	4	Z_6''	5	3	2	6	1	7	4	$mZ_6'' = 237$
Z_5'	5	3	2	7	4	1	6	Z_5''	1	7	2	3	4	5	6	$mZ_5'' = 312$
Z_6'	1	7	2	6	5	3	4	Z_6''	5	7	2	6	1	4	3	$mZ_6'' = 216$
Z_7'	6	7	5	2	1	4	3	Z_7''	6	7	1	2	5	3	4	$mZ_7'' = 375$
Z_6'	1	7	2	6	5	3	4	Z_6''	1	7	2	6	5	3	4	$mZ_6'' = 158$
Z_8'	1	7	2	4	5	3	6	Z_8''	1	7	2	4	5	3	6	$mZ_8'' = 248$
Z_6'	1	7	2	6	5	3	4	Z_6''	1	7	2	6	5	3	4	$mZ_6'' = 158$
Z_9'	1	7	2	6	5	3	4	Z_9''	1	7	2	6	5	3	4	$mZ_9'' = 158$

Z_6'	1	7	2	6	5	3	4	Z_6''	1	6	2	7	3	5	4	$mZ_6'' = 292$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	4	2	6	5	7	3	$mZ_{10}'' = 319$
Z_7'	6	7	5	2	1	4	3	Z_7''	1	7	2	5	6	4	3	$mZ_7'' = 345$
Z_1'	1	7	2	6	5	3	4	Z_1''	6	7	5	1	2	3	4	$mZ_1'' = 420$
Z_7'	6	7	5	2	1	4	3	Z_7''	1	7	2	6	5	4	3	$mZ_7'' = 256$
Z_2'	1	7	2	6	5	3	4	Z_2''	6	7	5	2	1	3	4	$mZ_2'' = 386$
Z_7'	6	7	5	2	1	4	3	Z_7''	6	7	5	2	1	4	3	$mZ_7'' = 273$
Z_3'	6	7	5	2	1	4	3	Z_3''	6	7	5	2	1	4	3	$mZ_3'' = 273$
Z_7'	6	7	5	2	1	4	3	Z_7''	5	3	7	2	1	4	6	$mZ_7'' = 226$
Z_4'	5	3	7	2	1	4	6	Z_4''	6	7	5	2	1	4	3	$mZ_4'' = 273$
Z_7'	6	7	5	2	1	4	3	Z_7''	5	3	2	6	1	4	7	$mZ_7'' = 213$
Z_5'	5	3	2	7	4	1	6	Z_5''	6	7	5	3	4	1	2	$mZ_5'' = 277$
Z_7'	6	7	5	2	1	4	3	Z_7''	6	7	1	2	5	3	4	$mZ_7'' = 375$
Z_6'	1	7	2	6	5	3	4	Z_6''	5	7	2	6	1	4	3	$mZ_6'' = 216$
Z_7'	6	7	5	2	1	4	3	Z_7''	1	7	5	2	6	4	3	$mZ_7'' = 294$
Z_8'	1	7	2	4	5	3	6	Z_8''	6	7	2	4	5	3	1	$mZ_8'' = 384$
Z_7'	6	7	5	2	1	4	3	Z_7''	2	7	1	6	5	3	4	$mZ_7'' = 256$
Z_9'	1	7	2	6	5	3	4	Z_9''	5	7	6	2	1	4	3	$mZ_9'' = 284$
Z_7'	6	7	5	2	1	4	3	Z_7''	6	5	2	7	3	4	1	$mZ_7'' = 246$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	3	4	5	2	1	6	7	$mZ_{10}'' = 347$
Z_8'	1	7	2	4	5	3	6	Z_8''	1	7	2	4	5	3	6	$mZ_8'' = 248$
Z_1'	1	7	2	6	5	3	4	Z_1''	1	7	2	6	5	3	4	$mZ_1'' = 158$
Z_8'	1	7	2	4	5	3	6	Z_8''	1	7	2	6	5	3	4	$mZ_8'' = 158$
Z_2'	1	7	2	6	5	3	4	Z_2''	1	7	2	4	5	3	6	$mZ_2'' = 248$
Z_8'	1	7	2	4	5	3	6	Z_8''	5	7	6	2	1	4	3	$mZ_8'' = 284$
Z_3'	6	7	5	2	1	4	3	Z_3''	2	7	1	4	5	3	6	$mZ_3'' = 267$

Z_8'	1	7	2	4	5	3	6	Z_8''	5	3	7	2	1	4	6	$mZ_8'' = 226$
Z_4'	5	3	7	2	1	4	6	Z_4''	1	7	2	4	5	3	6	$mZ_4'' = 248$
Z_8'	1	7	2	4	5	3	6	Z_8''	5	3	2	4	1	7	6	$mZ_8'' = 193$
Z_5'	5	3	2	7	4	1	6	Z_5''	1	7	2	3	4	5	6	$mZ_5'' = 312$
Z_8'	1	7	2	4	5	3	6	Z_8''	1	7	2	4	5	3	6	$mZ_8'' = 248$
Z_6'	1	7	2	6	5	3	4	Z_6''	1	7	2	4	5	3	6	$mZ_6'' = 248$
Z_8'	1	7	2	4	5	3	6	Z_8''	6	7	2	4	5	3	1	$mZ_8'' = 384$
Z_7'	6	7	5	2	1	4	3	Z_7''	1	7	5	2	6	4	3	$mZ_7'' = 294$
Z_8'	1	7	2	4	5	3	6	Z_8''	1	7	2	6	5	3	4	$mZ_8'' = 158$
Z_9'	1	7	2	6	5	3	4	Z_9''	1	7	2	4	5	3	6	$mZ_9'' = 248$
Z_8'	1	7	2	4	5	3	6	Z_8''	1	4	2	7	3	5	6	$mZ_8'' = 257$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	7	2	4	5	6	3	$mZ_{10}'' = 409$
Z_9'	1	7	2	6	5	3	4	Z_9''	1	7	2	6	5	3	4	$mZ_9'' = 158$
Z_1'	1	7	2	6	5	3	4	Z_1''	1	7	2	6	5	3	4	$mZ_1'' = 158$
Z_9'	1	7	2	6	5	3	4	Z_9''	1	7	2	6	5	3	4	$mZ_9'' = 158$
Z_2'	1	7	2	6	5	3	4	Z_2''	1	7	2	6	5	3	4	$mZ_2'' = 158$
Z_9'	1	7	2	6	5	3	4	Z_9''	5	7	6	2	1	4	3	$mZ_9'' = 284$
Z_3'	6	7	5	2	1	4	3	Z_3''	2	7	1	6	5	3	4	$mZ_3'' = 256$
Z_9'	1	7	2	6	5	3	4	Z_9''	5	3	7	2	1	6	4	$mZ_9'' = 296$
Z_4'	5	3	7	2	1	4	6	Z_4''	1	7	2	6	5	4	3	$mZ_4'' = 256$
Z_9'	1	7	2	6	5	3	4	Z_9''	5	3	2	6	1	7	4	$mZ_9'' = 237$
Z_5'	5	3	2	7	4	1	6	Z_5''	1	7	2	3	4	5	6	$mZ_5'' = 312$
Z_9'	1	7	2	6	5	3	4	Z_9''	1	7	2	6	5	3	4	$mZ_9'' = 158$
Z_6'	1	7	2	6	5	3	4	Z_6''	1	7	2	6	5	3	4	$mZ_6'' = 158$
Z_9'	1	7	2	6	5	3	4	Z_9''	6	7	2	1	5	3	4	$mZ_9'' = 271$
Z_7'	6	7	5	2	1	4	3	Z_7''	1	7	5	2	6	4	3	$mZ_7'' = 294$
Z_9'	1	7	2	6	5	3	4	Z_9''	1	7	2	4	5	3	6	$mZ_9'' = 248$
Z_8'	1	7	2	4	5	3	6	Z_8''	1	7	2	6	5	3	4	$mZ_8'' = 158$

Z_9'	1	7	2	6	5	3	4	Z_9''	1	6	2	7	3	5	4	$mZ_9'' = 292$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	4	2	6	5	7	3	$mZ_{10}'' = 319$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	7	2	4	3	6	5	$mZ_{10}'' = 238$
Z_1'	1	7	2	6	5	3	4	Z_1''	1	4	2	6	5	3	7	$mZ_1'' = 223$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	7	2	6	5	4	3	$mZ_{10}'' = 158$
Z_2'	1	7	2	6	5	3	4	Z_2''	1	4	2	7	3	5	6	$mZ_2'' = 257$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	5	6	7	2	1	4	3	$mZ_{10}'' = 258$
Z_3'	6	7	5	2	1	4	3	Z_3''	4	2	1	7	3	6	5	$mZ_3'' = 268$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	5	3	7	2	4	6	1	$mZ_{10}'' = 262$
Z_4'	5	3	7	2	1	4	6	Z_4''	1	4	2	7	5	3	6	$mZ_4'' = 230$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	5	3	2	7	4	6	1	$mZ_{10}'' = 249$
Z_5'	5	3	2	7	4	1	6	Z_5''	1	4	2	7	3	5	6	$mZ_5'' = 257$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	4	2	7	5	3	6	$mZ_{10}'' = 230$
Z_6'	1	7	2	6	5	3	4	Z_6''	1	7	2	5	3	6	4	$mZ_6'' = 246$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	6	7	2	4	3	1	5	$mZ_{10}'' = 351$
Z_7'	6	7	5	2	1	4	3	Z_7''	1	4	5	2	6	7	3	$mZ_7'' = 355$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	7	2	4	5	3	6	$mZ_{10}'' = 248$
Z_8'	1	7	2	4	5	3	6	Z_8''	1	4	2	7	3	6	5	$mZ_8'' = 259$
Z_{10}'	1	4	2	7	3	6	5	Z_{10}''	1	4	2	6	5	7	3	$mZ_{10}'' = 319$
Z_9'	1	7	2	6	5	3	4	Z_9''	1	6	2	7	3	5	4	$mZ_9'' = 292$