



**TUGAS AKHIR - KM184801**

**OPTIMASI *VEHICLE ROUTING PROBLEM WITH TIME WINDOWS* (VRPTW) PADA DISTRIBUSI KUE MENGGUNAKAN ALGORITMA GENETIKA**

**DINAH RAZAN ANSHORI**  
**0611144000032**

Dosen Pembimbing  
Drs. Daryono Budi Utomo, M.Si  
Dra. Nuri Wahyuningsih, M.Kes

Departemen Matematika  
Fakultas Matematika, Komputasi, dan Sains Data  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019



**TUGAS AKHIR - KM184801**

**OPTIMASI *VEHICLE ROUTING PROBLEM WITH TIME WINDOWS* (VRPTW) PADA DISTRIBUSI KUE MENGGUNAKAN ALGORITMA GENETIKA**

**DINAH RAZAN ANSHORI  
0611144000032**

**Dosen Pembimbing  
Drs. Daryono Budi Utomo, M.Si  
Dra. Nuri Wahyuningsih, M.Kes**

**Departemen Matematika  
Fakultas Matematika, Komputasi, dan Sains Data  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019**





FINAL PROJECT - KM184801

***OPTIMIZATION OF VEHICLE ROUTING  
PROBLEM WITH TIME WINDOWS (VRPTW) ON  
CAKE DISTRIBUTION USING GENETIC  
ALGORITHM***

DINAH RAZAN ANSHORI  
0611144000032

Supervisors

Drs. Daryono Budi Utomo, M.Si.  
Dra. Nuri Wahyuningsih, M.Kes.

Department Of Mathematics  
Faculty Of Mathematics, Computation, and Data Science  
Sepuluh Nopember Institute of Technology  
Surabaya 2019



**LEMBAR PENGESAHAN**

**OPTIMASI VEHICLE ROUTING PROBLEM WITH  
TIME WINDOWS (VRPTW) DENGAN  
MENGUNAKAN ALGORITMA GENETIKA**

**OPTIMIZATION OF VEHICLE ROUTING PROBLEM  
WITH TIME WINDOWS (VRPTW) USING GENETIC  
ALGORITHM**

**TUGAS AKHIR**

Diajukan untuk memenuhi salah satu syarat  
Untuk memperoleh gelar Sarjana Matematika  
Pada bidang studi Matematika Terapan  
Program Studi S-1 Departemen Matematika  
Fakultas Matematika, Komputasi, dan Sains Data  
Institut Teknologi Sepuluh Nopember Surabaya

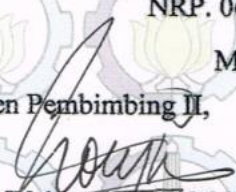
Oleh :

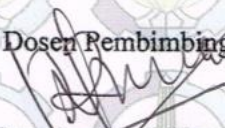
**DINAH RAZAN ANSHORI**  
NRP. 0611144000032

Menyetujui,


Dosen Pembimbing II,

Dosen Pembimbing I,

  
Dra. Nuri Wahyuningsih, M.Kes  
NIP. 19650220-198903 2 002

  
Drs. Daryono Budi Utomo, M.Si  
NIP. 19590419 198603 1 004

Mengesahkan,  
Ketua Departemen Matematika  
EMKSD-ITS

  
Dr. Anam Mukhlash, S.Si, MT  
NIP. 19700831 199403 1 003  
Surabaya, 23 Juli 2019

**OPTIMASI VEHICLE ROUTING PROBLEM WITH TIME  
WINDOWS (VRPTW) PADA DISTRIBUSI KUE  
MENGUNAKAN ALGORITMA GENETIKA**

**Nama** : Dinah Razan Anshori  
**NRP** : 0611144000032  
**Departemen** : Matematika  
**Dosen Pembimbing** : 1. Drs. Daryono Budi Utomo, M.Si  
2. Dra. Nuri Wahyuningsih, M.Kes

**ABSTRAK**

Dalam pendistribusian suatu produk perlu mempertimbangkan beberapa faktor antara lain waktu, jarak tempuh, biaya transportasi, serta rute yang akan dilalui dari satu tempat ke tempat yang lainnya. Pada tugas akhir ini, pendistribusian kue industri rumah tangga “Matoh Tenan” ke beberapa toko kue menggunakan konsep *Vehicle Routing Problem* (VRP) yang bertujuan untuk meminimalkan jarak, dengan memperhatikan *time window* setiap toko kue yang berbeda-beda. Untuk mengatasi permasalahan tersebut diperlukan suatu metode yang dapat mengoptimalkan jarak tempuh dari suatu tempat ke tempat lain dengan memperhatikan *time window*. Algoritma genetika menawarkan suatu solusi pemecahan masalah yang terbaik untuk permasalahan VRP. Dari hasil pengujian ukuran populasi terbaik adalah 150 populasi, sedangkan untuk kombinasi probabilitas *crossover* dan mutasi adalah 0,2 dan 0,7. Dari nilai-nilai parameter ini didapatkan rute optimal yaitu dari lokasi titik awal rute kendaraan 1: 9 → 3 → 5 → 6 → 10 → 4 → 12, rute kendaraan 2: 1 → 18 → 24 → 11 → 20 → 23 → 13, rute kendaraan 3: 26 → 7 → 14 → 25 → 21 → 19, rute kendaraan 4: 2 → 8 → 22 → 17 → 16 → 15 dengan nilai *fitness* sebesar 0,0160 dan jarak tempuh dari total perjalanan oleh 4 kendaraan yaitu 62,31 km.

**Kata Kunci** : distribusi, rute optimal, VRP, time window, algoritma genetika





**OPTIMIZATION OF VEHICLE ROUTING PROBLEM WITH  
TIME WINDOWS (VRPTW) ON CAKE DISTRIBUTION  
USING GENETIC ALGORITHM**

**Name of Student** : Dinah Razan Anshori  
**NRP** : 0611144000032  
**Department** : Mathematics  
**Supervisor** : 1. Drs. Daryono Budi Utomo, M.Si  
2. Dra. Nuri Wahyuningsih, M.Kes

**ABSTRACT**

*In the distribution of a product it is necessary to consider several factors including time, distance traveled, transportation costs, and routes to be traveled from one place to another. In this final project the cake distribution of home industry "Matoh Tenan" to a several of cake shops using the concept of Vehicle Routing Problem (VRP) which aims to minimize the distance, by taking notice of the different windows of each problem. To overcome this problem, we need a method that can optimize the distance from one place to another by taking notice to the time window. Genetic algorithms offer the best solution for VRP problems. From the results of testing the best population size is 150 populations, while for combinations of crossover and mutation probabilities are 0.2 and 0.7. From these parameter values the optimal route is obtained from the location of the starting point of the vehicle route 1: 9 → 3 → 5 → 6 → 10 → 4 → 12, vehicle route 2: 1 → 18 → 24 → 11 → 20 → 23 → 13, vehicle route 3: 26 → 7 → 14 → 25 → 21 → 19, vehicle route 4: 2 → 8 → 22 → 17 → 16 → 15 with a fitness value of 0.0160 and the distance traveled from the total trip by 4 vehicles, which is 62.31 km.*

*Keywords* : distribution, optimal route, VRP, time window, genetic algorithm



## **KATA PENGANTAR**

Assalamualaikum Wr. Wb

Alhamdulillahirabbilalamin, segala puji dan syukur bagi Allah SWT yang telah memberikan rahmat, kasih sayang, dan petunjuk-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul :

### **OPTIMASI *VEHICLE ROUTING PROBLEM WITH TIME WINDOWS* (VRPTW) PADA DISTRIBUSI KUE MENGGUNAKAN ALGORITMA GENETIKA**

Sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FMKSD Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dapat terselesaikan dengan baik berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih kepada :

1. Bapak Dr. Imam Mukhlash, S.Si, MT, selaku Kepala Departemen Matematika ITS
2. Bapak Drs. Daryono Budi Utomo, M.Si dan Ibu Dra. Nuri Wahyuningsih, M.Kes. selaku dosen pembimbing yang telah memberikan bimbingan, arahan dan motivasi kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
3. Ibu Valeriana Lukitosari, S.Si, M.T, dan Bapak Subchan Ph.D selaku dosen penguji yang telah memberikan saran demi perbaikan Tugas Akhir.

4. Bapak Drs. Sentot Didik Surjanto, M.Si selaku dosen wali yang telah memberikan dukungan dan motivasi selama perkuliahan hingga selesainya Tugas Akhir ini.
5. Bapak Ibu dosen serta seluruh staf Tenaga Kependidikan Departemen Matematika Institut Teknologi Sepuluh Nopember.
6. Ayah, Ibu, Najih, Rafan dan seluruh keluarga tercinta yang senantiasa memberikan dukungan secara moral materiil serta doa yang tak terhingga untuk kesuksesan penulis.
7. Sahabat penulis Juli, Eva, Ekik, Dian, Binuri, Mida dan Dwita terima kasih karena selalu memberikan semangat, dukungan, dan memberikan doa terbaik untuk penulis serta teman-teman lainnya yang tidak dapat disebutkan satu persatu oleh penulis, terima kasih atas dukungan dan doa yang diberikan.
8. Teman-teman mahasiswa Matematika ITS angkatan 2014 yang telah memberikan masa-masa berkesan bagi penulis selama menjadi bagian dari mereka.
9. Semua pihak yang tidak bisa penulis sebutkan satu-persatu, terima kasih telah memberikan saran, dukungan, motivasi dan membantu sampai terselesaikannya Tugas Akhir ini.

Penulis juga menyadari bahwa dalam Tugas Akhir ini masih terdapat kekurangan. Oleh sebab itu, kritik dan saran yang bersifat membangun sangat penulis harapkan demi kesempurnaan Tugas Akhir ini. Akhirnya, penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi banyak pihak.

Surabaya, Juli 2019

Penulis

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>TITLE PAGE</b> .....	iii
<b>LEMBAR PENGESAHAN</b> .....	v
<b>ABSTRAK</b> .....	vii
<b>ABSTRACT</b> .....	ix
<b>KATA PENGANTAR</b> .....	xi
<b>DAFTAR ISI</b> .....	xiii
<b>DAFTAR TABEL</b> .....	xv
<b>DAFTAR GAMBAR</b> .....	xvii
<b>DAFTAR LAMPIRAN</b> .....	xix
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	4
1.6 Sistematika Penulisan Tugas Akhir .....	5
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Penelitian Terdahulu .....	7
2.2 <i>Vehicle Routing Problem</i> .....	8
2.3 <i>Vehicle Routing Problem With Time Windows</i> .....	8
2.4 Algoritma Genetika .....	15
2.4.1 Representasi Kromosom .....	16
2.4.2 Inisialisasi.....	17
2.4.3 Fitness .....	18
2.4.4 Seleksi .....	18
2.4.5 <i>Crossover</i> .....	20
2.4.6 Mutasi.....	21

2.4.7 Evaluasi .....	22
<b>BAB III METODE PENELITIAN</b>	
3.1 Studi Literatur .....	25
3.2 Pengumpulan Data .....	25
3.3 Analisa dan Perancangan Sistem .....	25
3.4 Implementasi Perangkat Lunak .....	26
3.5 Evaluasi Hasil Uji Coba .....	26
3.6 Pengambilan Kesimpulan.....	26
3.7 Penulisan Laporan.....	26
<b>BAB IV ANALISIS DAN PEMBAHASAN</b>	
4.1 Proses Pengambilan Data.....	29
4.2 Pengolahan Data .....	29
4.2.1 Representasi Kromosom .....	30
4.2.2 Inisialisasi Populasi.....	31
4.2.3 Nilai <i>Fitness</i> .....	32
4.2.4 Seleksi .....	32
4.2.5 <i>Crossover</i> .....	34
4.2.6 Mutasi.....	36
4.2.7 Penggantian Populasi (Evaluasi) .....	37
4.3 Implementasi Program dan Analisa Data .....	39
4.3.1 Analisa Uji Coba Ukuran Populasi .....	46
4.3.2 Analisa Uji Coba Kombinasi Probabilitas <i>Crossover</i> dan Probabilitas Mutasi .....	47
4.4 Beberapa Contoh Rute.....	48
4.5 Rute Optimal .....	52
<b>BAB V PENUTUP</b>	
5.1 Kesimpulan .....	55
5.2 Saran .....	56
<b>DAFTAR PUSTAKA</b> .....	57
<b>LAMPIRAN</b> .....	59
<b>BIODATA PENULIS</b> .....	83

## DAFTAR TABEL

	Halaman
Tabel 2.1	Contoh Individu ..... 17
Tabel 4.1	Biaya Distribusi ..... 29
Tabel 4.2	Individu Hasil Inisialisasi Populasi Awal ..... 31
Tabel 4.3	Nilai <i>Fitness</i> , $p_i$ dan $q_i$ tiap individu ..... 33
Tabel 4.4	Hasil Seleksi Untuk Mendapatkan <i>Parent</i> ..... 34
Tabel 4.5	Kromosom Hasil <i>Crossover</i> ..... 36
Tabel 4.6	Pembangkitan Bilangan Acak ( $r$ ) ..... 37
Tabel 4.7	Populasi Hasil Proses <i>Crossover</i> dan Mutasi ..... 38
Tabel 4.8	Populasi Gabungan ..... 38





## DAFTAR GAMBAR

	Halaman
Gambar 2.1	Rute Kendaraan Dengan Demand Dan <i>Time Windows</i> Masing-Masing Pelanggan ..... 13
Gambar 2.2	Rute Kendaraan Yang Terkena Pinalti..... 14
Gambar 2.3	Bentuk Kromosom Pada Representasi Permutasi..... 16
Gambar 2.4	Seleksi <i>Stochastic Universal Sampling</i> ..... 20
Gambar 2.5	Proses <i>Crossover</i> Dengan Metode Order <i>Crossover</i> ..... 21
Gambar 2.6	Proses Mutasi Dengan <i>Exchange Mutation</i> ..... 22
Gambar 2.7	Diagram Alir Algoritma Genetika..... 23
Gambar 3.1	Diagram Alir Pengerjaan Tugas Akhir..... 27
Gambar 4.1	Representasi Bentuk Kromosom..... 30
Gambar 4.2	Pemetaan Individu pada Seleksi <i>Stochastic Universal Sampling</i> ..... 33
Gambar 4.3	Proses Mutasi Dengan Menggunakan Metode <i>Exchange Mutation</i> ..... 37
Gambar 4.4	Grafik Uji Coba Populasi ..... 46
Gambar 4.5	Grafik Uji Coba Kombinasi Probabilitas <i>Crossover</i> dan Probabilitas Mutasi ..... 47
Gambar 4.6	Rute Optimal ..... 53



## DAFTAR LAMPIRAN

	Halaman
Lampiran A	Data Lokasi Toko Kue, <i>Time Windows</i> , dan <i>Demand</i> ..... 59
Lampiran B	Matriks Jarak..... 61
Lampiran C	Matriks Waktu..... 63
Lampiran D	Tabel Perhitungan <i>Fitness</i> ..... 65
Lampiran E	<i>Source Code</i> VRPTW ..... 69
Lampiran F	<i>Source Code</i> Proses <i>Crossover</i> ..... 74
Lampiran G	<i>Source Code</i> Menghitung Jarak ..... 76
Lampiran H	<i>Source Code</i> Menghitung Pinalti ..... 78
Lampiran I	<i>Source Code</i> Menghitung Pinalti <i>Time</i> <i>Windows</i> ..... 80



# **BAB I**

## **PENDAHULUAN**

Pada bab ini dijelaskan hal – hal yang menjadi latar belakang permasalahan dalam tugas akhir ini. Kemudian permasalahan tersebut disusun kedalam rumusan masalah. Selanjutnya dijabarkan batasan masalah untuk memperoleh tujuan serta manfaat.

### **1.1 Latar Belakang**

Distribusi adalah suatu kegiatan untuk memindahkan produk dari pihak *supplier* ke pihak konsumen dalam suatu *supply chain* [1]. Kegiatan distribusi adalah kegiatan yang hampir tidak bisa lepas dalam dunia industri, terutama yang bergerak dalam bidang produksi. Dalam dunia industri, kegiatan distribusi merupakan salah satu faktor yang mempengaruhi pendapatan. Dalam pendistribusian suatu produk harus mempertimbangkan beberapa faktor antara lain waktu tempuh dari satu tempat ke tempat yang lain, jarak tempuh dari satu tempat ke tempat yang lain, biaya transportasi, dan rute yang akan dilalui.

Penentuan rute terbaik dengan mengoptimalkan sumber daya yang tersedia dapat menekan biaya operasional industri dan juga dapat mengoptimalkan pendistribusian produk dengan tepat waktu. Pendistribusian produk dari sumber ke beberapa tempat tujuan tentunya merupakan suatu permasalahan yang cukup kompleks, karena dengan adanya beberapa tempat tujuan pengiriman produk akan menimbulkan beberapa jalur distribusi yang jarak dan waktu tempuh yang semakin panjang dan lama, hal tersebut tentunya akan berimbas pada biaya pengiriman (transportasi) yang cukup besar [2]. Kurang baiknya perencanaan sistem distribusi akan mempengaruhi pemborosan biaya distribusi

dan memakan waktu yang lama yang selanjutnya dapat menyebabkan hilangnya kepercayaan konsumen. Oleh karena itu, pendistribusian yang baik dalam menentukan rute, waktu, dan biaya distribusi yang minimum merupakan salah satu hal yang penting pada suatu industri.

Masalah pendistribusian ini dapat digolongkan sebagai *Vehicle Routing Problem* (VRP). VRP mendefinisikan permasalahan penentuan rute distribusi dimana seseorang mendatangi beberapa titik dan titik tersebut hanya dapat dikunjungi oleh seseorang tersebut yang berasal dari titik awal dan harus kembali ke titik awal tersebut pada akhir perjalanannya. Pada VRP juga mendefinisikan permasalahan rute dimana setiap titik digambarkan sebagai konsumen, dan tiap kendaraan yang dipakai untuk proses pendistribusian dianggap memiliki kapasitas tertentu.

Algoritma genetika adalah solusi yang sangat efektif untuk permasalahan optimasi. Algoritma ini didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Algoritma genetika menawarkan suatu solusi pemecahan masalah yang terbaik, dengan memanfaatkan metode seleksi, crossover, dan mutasi [3]. Algoritma genetika memungkinkan daerah solusi didapatkan pada ruang pencarian yang sangat luas dan kompleks sehingga memungkinkan daerah solusi didapatkan melampaui daerah optimum lokal [4].

Dalam Tugas Akhir ini, penulis menggunakan algoritma genetika untuk menyelesaikan permasalahan VRPTW (*Vehicle Routing Problem With Time Window*) dalam kasus pendistribusian kue karena algoritma genetika memiliki kelebihan dalam menghasilkan output yang optimal. Dengan demikian

diharapkan dapat diperoleh output berupa jalur distribusi kue yang optimal.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang tersebut, dapat dirumuskan permasalahan dalam Tugas Akhir ini sebagai berikut :

1. Bagaimana pengaruh nilai parameter algoritma genetika terhadap hasil dari optimasi VRPTW dalam pendistribusian kue.
2. Bagaimana nilai parameter terbaik yang dihasilkan algoritma genetika untuk menyelesaikan masalah VRPTW dalam pendistribusian kue.
3. Bagaimana rute optimal yang didapatkan dari optimasi VRPTW dalam pendistribusian kue dengan menggunakan algoritma genetika.

## **1.3 Batasan Masalah**

Batasan permasalahan yang digunakan untuk menghindari melebarnya masalah yang akan dibahas dalam tugas akhir ini yaitu :

1. Data yang digunakan adalah data distribusi dari industri rumah tangga “Matoh Tenan” yang merupakan industri rumah tangga yang bergerak di bidang produksi kue tradisional.
2. Penelitian diterapkan pada kasus dimana setiap sumber diasumsikan selalu memiliki persediaan dan setiap tujuan diasumsikan memiliki permintaan sesuai dengan jumlah persediaan serta tidak nol.
3. Jarak dan waktu tempuh dari toko kue ke toko kue lainnya didapatkan dengan menggunakan bantuan aplikasi Google Maps. Pada tugas akhir ini menggunakan satuan jarak kilometer (km) dan satuan waktu menit.

4. Biaya transportasi adalah biaya bahan bakar yang didapatkan dari total biaya yang dihabiskan selama proses pendistribusian dibagi dengan total jarak tempuh kemudian dikalikan dengan jarak masing-masing toko.
5. Parameter algoritma genetika yaitu *popsi* (30, 60, 90, 120, 150), *probability crossover* (0,2; 0,3; 0,4; 0,5; 0,6), dan *probability mutation* ( 0,3; 0,4; 0,5; 0,6; 0,7).
6. Variabel yang diperhatikan dalam Tugas Akhir ini meliputi jarak, waktu tempuh, *time windows*, kapasitas kendaraan, permintaan tiap toko, dan jumlah kendaraan.
7. Pada tugas akhir ini hanya memperhatikan jumlah box dan tidak memperhatikan isi dari masing – masing box yang didistribusikan.

#### **1.4 Tujuan Penelitian**

Berdasarkan permasalahan yang telah dirumuskan sebelumnya, tujuan penelitian Tugas Akhir ini adalah :

1. Mengetahui pengaruh parameter algoritma genetika terhadap hasil optimasi VRPTW dalam pendistribusian kue.
2. Mendapatkan parameter terbaik yang dihasilkan algoritma genetika dalam menyelesaikan masalah VRPTW pendistribusian kue.
3. Mendapatkan rute distribusi yang optimal pada masalah optimasi VRPTW dalam pendistribusian kue.

#### **1.5 Manfaat Penelitian**

Manfaat yang bisa diperoleh dari penelitian tugas akhir ini yaitu :

1. Memberikan bahan pertimbangan dan sumbangan pemikiran mengenai penentuan rute distribusi untuk meminimalkan waktu dan biaya distribusi.



2. Sebagai informasi tambahan yang dapat digunakan pihak lain dalam optimasi yang berhubungan dengan penentuan rute dalam kasus VRPTW.

## **1.6 Sistematika Penulisan Tugas Akhir**

Penulisan Tugas Akhir ini disusun dalam lima bab, yaitu :

1. **BAB I PENDAHULUAN**  
Bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.
2. **BAB II TINJAUAN PUSTAKA**  
Bab ini berisi penelitian terdahulu dan teori-teori yang terkait dengan permasalahan dalam Tugas Akhir ini seperti VRP, VRPTW, algoritma genetika beserta bagian-bagiannya.
3. **BAB III METODE PENELITIAN**  
Pada bab ini dibahas mengenai langkah-langkah pengerjaan Tugas Akhir ini mulai dari studi literatur, pengumpulan data, analisa dan perancangan sistem, implementasi perangkat lunak, evaluasi hasil uji coba, pengambilan keputusan sampai penulisan laporan.
4. **BAB IV ANALISIS DAN PEMBAHASAN**  
Bab ini menjelaskan mengenai analisa data dan pembahasan penerapan metode algoritma genetika dalam kasus optimasi VRPTW pada pendistribusian kue sehingga dihasilkan solusi yang terbaik dan rute yang optimal.
5. **BAB V PENUTUP**  
Bab ini berisi kesimpulan akhir yang diperoleh dari pembahasan masalah sebelumnya serta saran untuk pengembangan penelitian selanjutnya.



## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini dibahas mengenai penelitian terdahulu dan dasar teori yang digunakan dalam penyusunan Tugas Akhir ini.

#### **2.1 Penelitian Terdahulu**

Penulisan tugas akhir ini merujuk pada beberapa penelitian sebelumnya yang sesuai dengan topik yang diambil. Rujukan pertama yang digunakan adalah penelitian yang dilakukan oleh Alim Setiawan Slamet, dkk (2014) yang berjudul “*Vehicle Routing Problem (VRP) Dengan Algoritma Genetika Pada Pendistribusian Sayuran Dataran Tinggi*” [5]. Dalam penelitian tersebut diperoleh solusi berupa pembagian saluran distribusi dengan memenuhi kendala antara lain kapasitas tiap-tiap kendaraan, jumlah kendaraan, dan permintaan. Juga diperoleh penyelesaian *VRP* berupa pengurangan waktu pengiriman dan pengurangan penggunaan jumlah armada.

Rujukan penelitian selanjutnya adalah penelitian yang dilakukan oleh Mahmudy (2014) yang berjudul “*Penerapan Metode Simulated Annealing untuk Optimasi Vehicle Routing Problem With Time Window (VRPTW)*” [6]. Dalam penelitian tersebut Mahmudy menggunakan metode *simulated annealing* untuk menyelesaikan permasalahan *VRPTW*. Solusi yang didapatkan pada penelitian ini adalah dengan menggunakan metode *simulated annealing* menghasilkan rata-rata waktu komputasi yang baik yaitu 82,29 detik dari 25 percobaan yang dilakukan. Akan tetapi kekurangan metode ini adalah tidak bisa menangani data yang banyak secara efisien.

## 2.2 *Vehicle Routing Problem (VRP)*

VRP mendefinisikan permasalahan penentuan rute distribusi dimana seorang salesmen yang mendatangi beberapa kota dan tiap kota tersebut hanya dapat dikunjungi oleh satu salesmen saja dimana tiap salesman berasal dari tempat produksi dan harus kembali ke tempat produksi tersebut pada akhir perjalanannya. VRP juga mendefinisikan permasalahan rute dimana sebuah kota diasosiasikan sebagai sebuah demand atau konsumen, dan tiap kendaraan yang dipakai untuk perjalanan dianggap memiliki kapasitas tertentu. Total jumlah demand dalam suatu rute tidak boleh melebihi kapasitas dari kendaraan yang ditugasi melewati rute tersebut [7]. VRP juga dapat dipakai sebagai kombinasi dari dua permasalahan optimasi lain, yaitu *Bin Packing Problem (BPP)* dan *Travelling Salesman Problem (TSP)* [8]. Salah satu contoh kasus VRP adalah pendistribusian bahan baku makanan.

## 2.3 *Vehicle Routing Problem with Time Window*

VRP adalah suatu masalah pencarian jalur yang akan dilalui dengan tujuan mencari rute yang paling cepat atau pendek. VRPTW yang merupakan sebutan bagi VRP dengan kendala tambahan berupa adanya *time window* pada masing-masing pelanggan. Waktu ketersediaan pada masing-masing pelanggan dapat berbeda dan dinyatakan dalam selang waktu berupa batas waktu awal sampai akhir pelayanan pada pelanggan tersebut [9]. Salah satu contoh kasus VRPTW adalah distribusi catering. Bentuk model matematis VRPTW secara umum didefinisikan sebagai berikut [7] :

$$Z_{vrptw} = \text{minimum} \sum_{i,j \in N} c_{ij} + \sum_{i \in N} P_i \quad (2.1)$$

dengan:

$C_{ij}$  : jarak tempuh dari titik  $i$  ke titik  $j$ .

$P_i$  : Penalti apabila kendaraan melebihi kapasitas dan *time window*

$N$  : kumpulan titik yang terdiri dari pelanggan

Fungsi tujuan digambarkan pada persamaan (2.1) menyatakan bahwa tujuan dari model yaitu untuk meminimalkan jarak dengan batasan sebagai berikut:

1. Setiap pelanggan dikunjungi tepat satu kali. Kendaraan yang beroperasi dari pelanggan pertama menuju pelanggan ketiga lalu ke pelanggan kelima tidak boleh kembali lagi ke pelanggan yang sudah dilalui sebelumnya yaitu pelanggan pertama dan ketiga. Seperti yang ditunjukkan oleh Gambar 2.1.
2. Setiap kendaraan bermula dari tempat produksi. Kendaraan yang akan menuju ke setiap pelanggan bermula dari tempat produksi dilanjutkan ke setiap pelanggan sampai kapasitas yang dibawanya habis terdistribusikan ke masing – masing pelanggan. Seperti pada Gambar 2.1 rute kendaraan pertama hanya sampai pelanggan kelima, maka untuk rute selanjutnya yaitu ke pelanggan ketujuh dilanjutkan oleh kendaraan kedua dan kendaraan kedua bermula dari rumah produksi langsung menuju pelanggan ketujuh tidak dari pelanggan kelima.
3. Kendaraan tidak boleh mengangkut melebihi kapasitas yang dimiliki kendaraan tersebut. Masing-masing kendaraan memiliki kapasitas yang sama. Jika kendaraan mengangkut melebihi kapasitas maka akan dikenakan penalti. Seperti yang ditunjukkan oleh Gambar 2.1 adalah gambaran rute yang tidak terkena penalti. Jika kapasitas setiap kendaraan adalah 8 maka mulai dari tempat produksi ke pelanggan pertama menuju pelanggan ketiga dan kemudian pelanggan kelima adalah rute

dari kendaraan pertama agar tidak terkena pinalti. Tetapi apabila melebihi kapasitas kendaraan maka akan terkena pinalti seperti pada Gambar 2.2. Pada kendaraan kedua rute dimulai dari tempat produksi menuju pelanggan ketujuh setelah itu ke pelanggan kedelapan selanjutnya ke pelanggan kesembilan dilanjutkan ke pelanggan kesepuluh dan kemudian ke pelanggan keenam, total yang diangkut kendaraan adalah 9 sedangkan kapasitas kendaraan adalah 8 maka kendaraan kedua terkena pinalti karena mengangkut melebihi kapasitas yang dimiliki kendaraan.

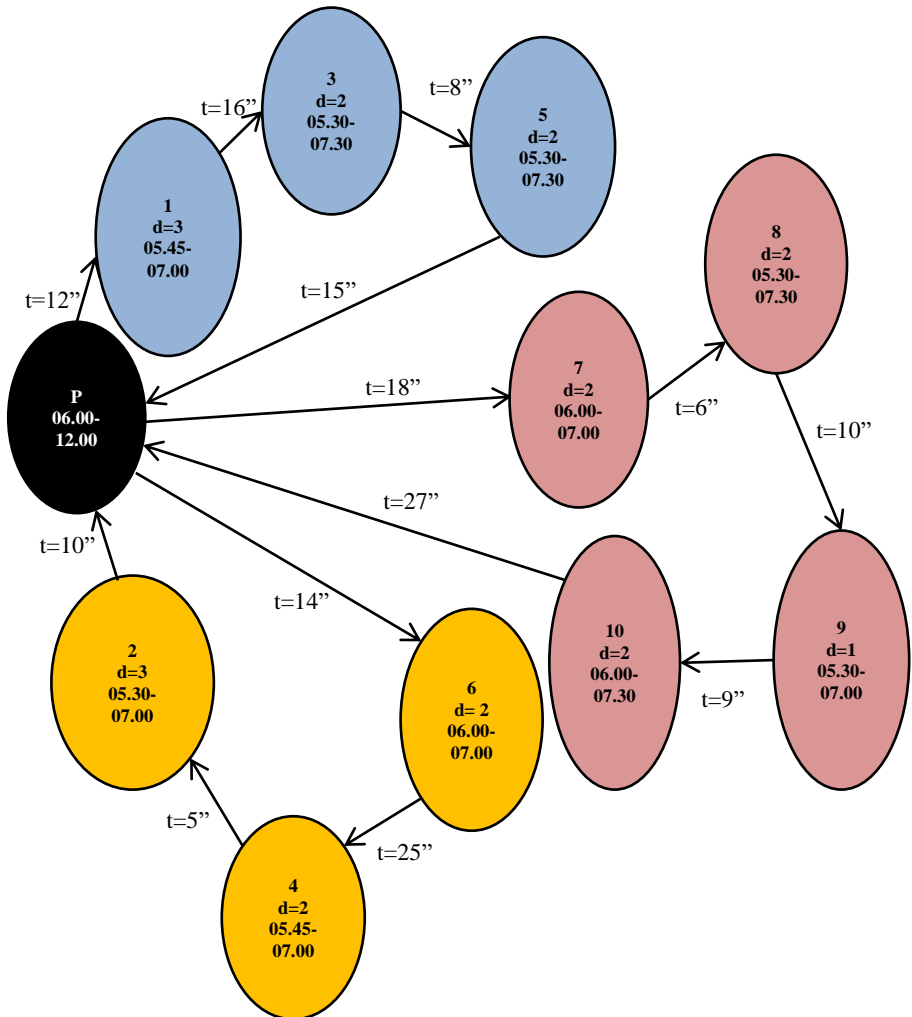
4. Setiap kendaraan berakhir di tempat produksi. Pada setiap rute kendaraan selalu berawal dari tempat produksi menuju ke setiap pelanggan hingga pelanggan terakhir pada rute tersebut. Berakhirnya rute tersebut tidak pada pelanggan terakhir melainkan pada tempat produksi, jadi setiap kendaraan yang sudah berada pada pelanggan terakhir selanjutnya akan menuju ke tempat produksi pada akhirnya seperti pada Gambar 2.1.
5. Kendaraan tidak boleh sampai di pelanggan sebelum waktu pelayanan oleh pelanggan dimulai. Dari Gambar 2.1 diketahui kendaraan pertama mulai beroperasi pukul 06.00 berawal dari tempat produksi akan menuju tempat pelanggan pertama yang waktu pelayanannya mulai pada pukul 05.45 dengan waktu yang ditempuh dari tempat produksi ke tempat pelanggan pertama adalah 12 menit. Jika kendaraan berangkat pukul 06.00 maka akan tiba di tempat pelanggan pertama pukul 06.12 yang artinya kendaraan sampai di tempat pelanggan pertama setelah waktu dimulai pelayanan. Selanjutnya ada waktu proses (waktu untuk bertransaksi dengan pelanggan) selama 5 menit. Jadi kendaraan berangkat dari pelanggan pertama menuju ke pelanggan kedua pada pukul 06.17 dengan waktu tempuh dari pelanggan pertama menuju ke pelanggan kedua adalah 16

menit. Kendaraan tiba di tempat pelanggan kedua pada pukul 06.33 yang artinya kendaraan tiba di tempat pelanggan kedua setelah waktu dimulai pelayanan, begitupun seterusnya sampai pada pelanggan terakhir. Jika melihat rute dan jam dimulai pelayanan pada Gambar 2.1 maka dapat disimpulkan bahwa kendaraan akan tiba di tempat masing – masing pelanggan setelah waktu dimulai pelayanan pada masing – masing pelanggan. Batasan *time window* pada masing-masing pelanggan terpenuhi, apabila tidak terpenuhi maka terkena denda. Batasan *time window* pada masing – masing pelanggan tidak hanya waktu dimulai pelayanan saja tetapi juga ada waktu berakhirnya pelayanan. Dimana setiap kendaraan tidak hanya harus datang setelah dimulainya waktu pelayanan tetapi kendaraan juga harus datang sebelum berakhirnya waktu pelayanan pada masing-masing pelanggan. Pada Gambar 2.1 kendaraan beroperasi mulai pukul 06.00 dari tempat produksi menuju tempat pelanggan pertama dengan waktu tempuh dari tempat produksi ke tempat pelanggan pertama adalah 12 menit. Kendaraan tiba di tempat pelanggan pertama pada pukul 06.12 dan melakukan proses transaksi selama 5 menit sampai pukul 06.17, artinya kendaraan tiba dan selesai melakukan transaksi sebelum waktu pelayanan di pelanggan pertama berakhir. Kemudian kendaraan berangkat dari tempat pelanggan pertama menuju tempat pelanggan ketiga pada pukul 06.17 dengan waktu tempuh 16 menit. Kendaraan tiba di tempat pelanggan ketiga pada pukul 06.33 dan melakukan transaksi selama 5 menit sampai pukul 06.38 yang artinya kendaraan tiba dan selesai melakukan transaksi sebelum waktu pelayanan di pelanggan ketiga berakhir. Selanjutnya menuju ke tempat pelanggan kelima pada pukul 06.38 dengan waktu tempuh 8 menit. Kendaraan tiba di tempat pelanggan kelima pada pukul

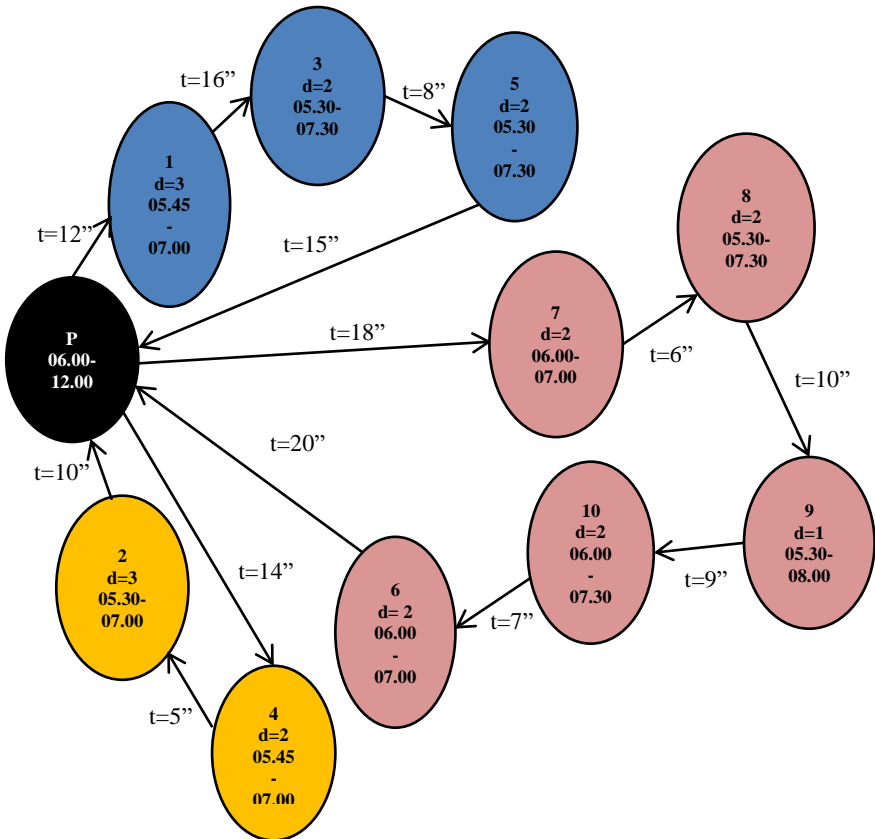
06.46 dan melakukan transaksi selama 5 menit sampai pukul 06.51, artinya kendaraan tiba dan selesai melakukan transaksi sebelum waktu pelayanan di pelanggan kelima berakhir. Dan yang terakhir kendaraan akan menuju atau kembali ke tempat produksi pada pukul 06.51 dengan waktu tempuh 15 menit. Kendaraan akan tiba di tempat produksi pada pukul 07.06. Dengan demikian dapat dikatakan bahwa rute tersebut memenuhi *time window* dari masing-masing pelanggan karena kendaraan tiba di tiap lokasi pelanggan sebelum waktu pelayanan berakhir sehingga dalam hal ini kendaraan pertama tidak terkena pinalti. Sedangkan pada Gambar 2.2 ditunjukkan bahwa kendaraan kedua memulai perjalanan dari tempat produksi pada pukul 06.00 menuju pelanggan ketujuh dengan waktu tempuh 18 menit. Kendaraan tiba di tempat pelanggan ketujuh pada pukul 06.18 dan melakukan proses transaksi selama 5 menit sampai pukul 06.23, artinya kendaraan tiba dan selesai melakukan transaksi sebelum waktu pelayanan di pelanggan ketujuh berakhir. Kemudian kendaraan berangkat dari tempat pelanggan ketujuh menuju pelanggan kedelapan pada pukul 06.23 dengan waktu tempuh 6 menit. Kendaraan tiba di pelanggan kedelapan pada pukul 06.29 dan melakukan proses transaksi selama 5 menit sampai pukul 06.34, artinya kendaraan tiba dan selesai melakukan proses transaksi sebelum waktu pelayanan pada pelanggan kedelapan berakhir. Setelah itu kendaraan berangkat menuju pelanggan kesembilan pada pukul 06.34 dengan waktu tempuh 10 menit. Kendaraan tiba di pelanggan ke sembilan pada pukul 06.44 dan melakukan transaksi selama 5 menit sampai pukul 06.49, artinya kendaraan tiba dan selesai melakukan transaksi sebelum waktu pelayanan pada pelanggan kesembilan berakhir. Selanjutnya kendaraan berangkat menuju pelanggan kesepuluh dari tempat



pelanggan kesembilan pada pukul 06.49 dengan waktu tempuh 9 menit.



**Gambar 2.1** Rute kendaraan dengan *demand* dan *time windows* masing-masing pelanggan



**Gambar 2.2** Rute kendaraan yang terkena pinalti

Kendaraan tiba di tempat pelanggan kesepuluh pada pukul 06.58 dan melakukan transaksi selama 5 menit sampai pukul 07.03, artinya kendaraan tiba dan selesai melakukan transaksi sebelum waktu pelayanan pada pelanggan kesepuluh berakhir. Kemudian kendaraan berangkat dari pelanggan kesepuluh menuju ke pelanggan keenam pada pukul 07.03 dengan waktu tempuh 7 menit. Kendaraan tiba di tempat pelanggan keenam pada pukul

07.10 dan melakukan transaksi selama 5 menit sampai dengan pukul 07.15, artinya kendaraan tiba dan selesai melakukan transaksi setelah waktu pelayanan pada pelanggan keenam berakhir. Ketika waktu pelayanan sudah berakhir maka tidak akan bisa untuk dilayani. Dan yang terakhir kendaraan akan kembali ke tempat produksi pada pukul 07.15 dengan waktu tempuh selama 20 menit. Kendaraan tiba di tempat produksi pada pukul 07.35. Dengan demikian kendaraan kedua tidak memenuhi *time windows* dari salah satu pelanggan yaitu pelanggan keenam. Oleh karena itu kendaraan kedua dalam hal ini terkena pinalti karena tiba di salah satu lokasi pelanggan setelah waktu pelayanan dari salah satu pelanggan tersebut berakhir.

## 2.4 Algoritma Genetika

Algoritma genetika merupakan algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Dalam proses evolusi, gen secara terus menerus mengalami penyesuaian dengan lingkungan hidupnya sehingga hanya individu-individu yang kuat yang mampu bertahan. Algoritma genetika merepresentasikan suatu solusi permasalahan dalam bentuk kromosom. Kromosom adalah individu yang terdapat di dalam populasi yang didapatkan dari himpunan solusi yang diperoleh secara acak. Jumlah populasi solusi yang besar merupakan keunggulan algoritma genetika. Terdapat beberapa aspek penting dalam algoritma genetika antara lain definisi fungsi fitness, definisi dan implementasi representasi genetika, serta definisi dan implementasi operasi genetika merupakan tiga aspek pendukung kinerja algoritma genetika [10].

Dasar dari algoritma genetika adalah individu dalam populasi bersaing untuk sumber daya dan pasangannya dengan

melakukan reproduksi untuk memperoleh individu baru yang mempunyai gen lebih baik dari orang tuanya yang kemudian akan melalui proses genetik yaitu seleksi alam individu yang lebih kuat akan bertahan untuk mendapatkan generasi yang lebih baik [11]. Algoritma genetika sering kali memecahkan masalah dengan cukup baik. Secara umum siklus dari algoritma genetika adalah :

1. Representasi kromosom
2. Inisialisasi populasi awal
3. Menghitung nilai *fitness*
4. Seleksi untuk mendapatkan individu baru untuk generasi selanjutnya
5. Proses reproduksi dengan *crossover* dan mutasi
6. Evaluasi

#### 2.4.1 Representasi Kromosom

Representasi kromosom merupakan suatu proses untuk menyelesaikan masalah, dimana suatu permasalahan dapat dikodekan ke dalam kromosom [4]. Algoritma genetika memiliki beberapa jenis representasi kromosom untuk permasalahan yang berbeda, seperti representasi biner, integer, real dan permutasi.

Node	1	2	3	4	5	6	7	8	9	10
<b>Kromosom</b>	2	5	7	4	1	9	6	10	3	8

**Gambar 2.3 Bentuk kromosom pada representasi permutasi**

Representasi permutasi telah sukses diterapkan pada berbagai masalah kombinatorial seperti Travelling Sales Person

Problem, perencanaan dan penjadwalan produksi industri manufaktur [12]. Contoh penggunaan representasi permutasi dijelaskan pada Gambar 2.3.

### 2.4.2 Inisialisasi

Inisialisasi yaitu membangkitkan individu secara acak yang memiliki susunan *chromosome* yang mewakili solusi dari permasalahan yang ingin dipecahkan. Tempat penampungan individu tersebut dikatakan sebagai populasi [12].

**Tabel 2.1 Contoh individu**

Individu ke-	Susunan Kromosom
Individu 1	2 3 4 5 6 1 8 9 7 10
Individu 2	6 1 2 3 4 5 8 7 9 10
Individu 3	4 5 6 1 2 3 7 8 9 10
Individu 4	3 6 1 2 7 8 4 5 9 10
Individu 5	5 6 1 2 3 4 10 7 9 8

Representasi yang digunakan adalah representasi permutasi yang merupakan bentuk pengkodean yang direpresentasikan dengan angka sebagai gen. Dalam satu individu terdapat gen dimana gen tersebut mewakili lokasi distribusi dan susunan gen dalam satu individu tersebut mewakili sebuah urutan lokasi distribusi sebagai jalur distribusi yang digunakan untuk mendapatkan hasil yang optimal. Misalkan ada sebanyak 10 lokasi maka contoh satu individu dengan susunan kromosom adalah sebagai berikut = [ 3 4 2 5 1 6 10 8 9 7], berarti jalur distribusi bergerak dari kota 3 ke kota 4 dan seterusnya hingga ke kota 10. Tabel 2.1 menunjukkan contoh individu yang dibangkitkan dengan representasi permutasi.

### 2.4.3 Fitness

*Fitness* adalah nilai yang dimiliki oleh masing-masing individu untuk menentukan tingkat kesesuaian individu tersebut dengan kriteria atau tujuan (obyektif) permasalahan yang ingin dicapai [13]. Nilai *fitness* suatu kromosom menggambarkan kualitas kromosom dalam populasi tersebut. Fungsi tujuan untuk sistem optimasi distribusi kue menggunakan Algoritma genetika adalah [12]:

$$\text{Nilai } fitness = \frac{1}{f_x} \quad (2.2)$$

dengan :

$$f_x = Z_{vrptw}$$

$C_{ij}$  : jarak tempuh dari titik  $i$  ke titik  $j$ .

$P_i$  : Penalti apabila kendaraan melebihi kapasitas dan *time window*

$N$  : kumpulan titik yang terdiri dari pelanggan

### 2.4.4 Seleksi

Seleksi merupakan pemilihan individu terbaik yang akan dijadikan *parent* untuk generasi selanjutnya [14]. Seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang dipilih untuk melakukan proses genetik pada generasi berikutnya [12]. Ada beberapa metode seleksi diantaranya yaitu:

#### 1. Seleksi *Elitism*

Seleksi *elitism* merupakan seleksi yang dilakukan dengan mengambil nilai *fitness* tertinggi suatu individu [15]. Metode seleksi ini bekerja dengan mengumpulkan semua *offspring* dan

*parent* dalam satu penampungan, populasi dengan *fitness* individu terbaik akan lolos menjadi generasi selanjutnya [16].

## 2. Seleksi *Roulette Wheel*

Metode seleksi *roulette wheel* merupakan metode yang paling sederhana dan sering juga dikenal dengan nama *stochastic sampling with replacement*. Pada metode ini, orang tua dipilih berdasarkan nilai *fitness*nya, semakin baik nilai *fitness*nya maka semakin besar kemungkinannya untuk terpilih. Diandaikan semua kromosom diletakkan pada sebuah roda roulet, besarnya kemungkinan bagi setiap kromosom adalah tergantung dari nilai *fitness*nya [15].

## 3. *Stocastic Universal Sampling*

Pada metode ini, individu – individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran *fitness*nya seperti halnya pada seleksi roda *roulette* seperti pada Gambar 2.4. Kemudian diberikan sejumlah pointer sebanyak individu yang ingin diseleksi pada garis tersebut [17]. Pada seleksi ini diperlukan perhitungan total *fitness*, peluang *fitness* tiap individu, dan peluang kumulatif tiap individu. Selain itu diperlukan juga  $P_c$  (Probabilitas *Crossover*) yang sudah ditentukan sebelumnya untuk menentukan jumlah individu yang akan menjadi induk. Langkah-langkah untuk mencari total *fitness*, peluang *fitness* tiap individu, dan peluang kumulatif tiap individu adalah sebagai berikut [18] :

1. Mencari total *fitness* dari seluruh individu

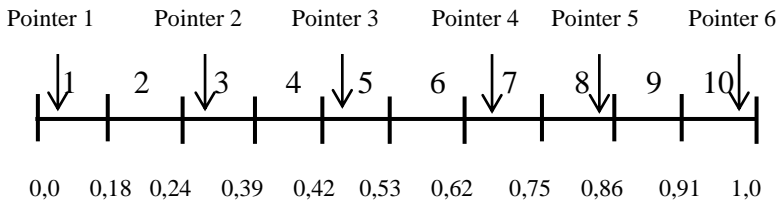
$$Total\ fitness = \sum \frac{1}{f(x)} \quad (2.3)$$

2. Mencari peluang *fitness* tiap individu ( $p_i$ )

$$p_i = \frac{1}{\text{Total fitness}} \quad (2.4)$$

3. Mencari peluang kumulatif tiap individu ( $q_i$ )

$$q_i = \sum_{i=1}^N p_i \quad (2.5)$$



**Gambar 2.4 Seleksi *Stochastic Universal Sampling***

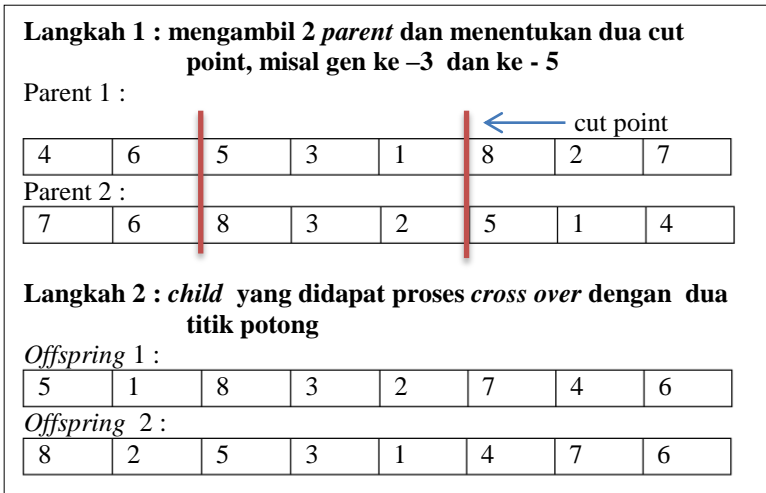
### 2.4.5 Crossover

*Crossover* adalah mekanisme yang dimiliki algoritma genetika dengan menggabungkan dua kromosom sehingga menghasilkan anak kromosom yang mewarisi ciri-ciri dasar dari *parent*. *Crossover* bekerja membangkitkan *offspring* baru dengan mengganti sebagian informasi dari *parent* [12]. Sebuah individu baru yang mengarah pada solusi optimal bisa diperoleh melalui proses *crossover*. Proses *crossover* adalah memilih dua buah kromosom sebagai *parent*, sehingga masing-masing kromosom *parent* terpisah menjadi dua segmen.

Setelah itu, dilakukan penukaran segmen kromosom *parent* untuk menghasilkan *offspring* atau individu baru. Pada representasi permutasi untuk proses *crossover* dapat digunakan metode *order crossover*. Misalkan terdapat dua *parent* terpilih yang memiliki panjang kromosom delapan satuan. Dalam



kromosom ditentukan dua titik potong atau cut point secara random dengan *cut point* pada posisi gen ke-3 dan ke - 5 [12]. *Offspring* yang dihasilkan dari dua titik potong digambarkan pada Gambar 2.8.



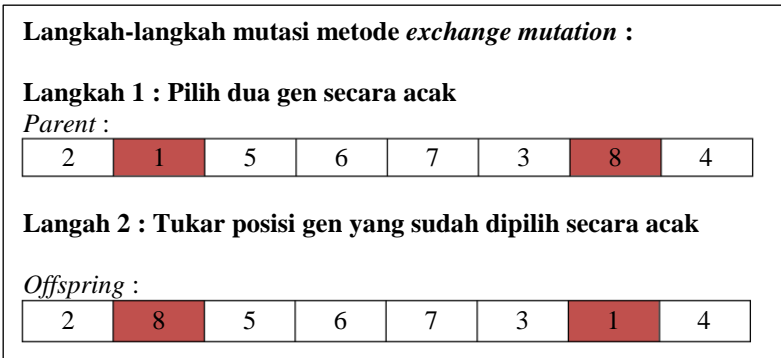
**Gambar 2.5 Proses *Crossover* dengan Metode Order *Crossover***

Pada Gambar 2.8 dijelaskan bahwa didapatkan cut point secara acak pada posisi gen ke-3 dan ke-5. Maka mulai dari gen ke-3 sampai gen ke-5 akan menjadi *substring* yaitu gen-gen yang akan ditukarkan antara *parent* 1 dan *parent* 2. Setelah dilakukan proses *cross over* maka akan didapatkan *offspring* yang gennya berasal dari *parent* kemudian disusun kembali agar tidak ada gen-gen yang sama dalam satu *offspring*.

#### 2.4.6 Mutasi

Mutasi merupakan proses suatu operator genetika untuk menghasilkan perubahan acak pada satu kromosom [19]. Metode

mutasi yang biasa digunakan dalam representasi permutasi adalah *exchange mutation* dan *insertion mutation*.



**Gambar 2.6 Proses Mutasi dengan Metode *Exchange Mutation***

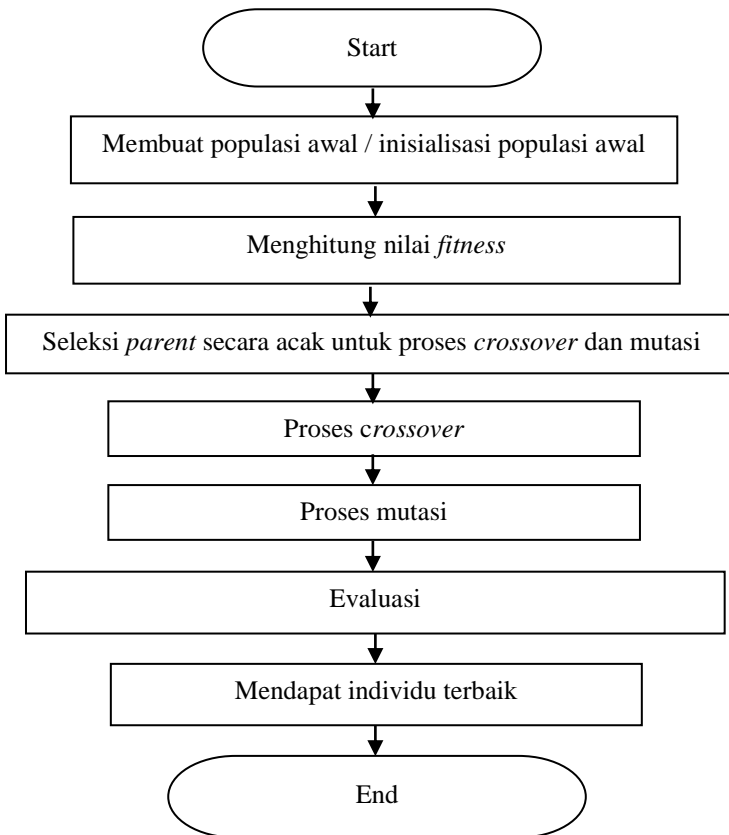
Sebelum dilakukan mutasi, sudah ditentukan parameter  $P_m$  (Probabilitas Mutasi). Probabilitas mutasi dipandang sebagai presentase gen yang diharapkan mengalami mutasi dari jumlah total gen pada populasi. Metode mutasi dengan *insertion mutation* bekerja dengan memilih dua posisi gen secara random kemudian memotong posisi yang pertama kemudian menyisipkannya setelah posisi yang kedua. Sedangkan Metode *exchange mutation* bekerja dengan memilih dua posisi gen secara acak kemudian menukarkan gen terpilih pertama ke posisi gen terpilih kedua sehingga gen terpilih pertama berada pada posisi gen terpilih kedua dan gen terpilih kedua berada pada posisi gen terpilih pertama seperti yang dijelaskan pada Gambar 2.6.

### 2.4.7 Evaluasi

Pada tahap evaluasi, ukuran populasi akan dipertahankan agar sama dengan ukuran populasi awal. Secara umum terdapat

dua tipe penggantian generasi. Tipe pertama, generasi berikutnya adalah semua individu baru atau anak semua (hasil dari proses *crossover* dan mutasi). Tipe kedua, generasi berikutnya adalah individu baru ditambah individu sebelumnya dengan nilai *fitness* yang terbesar. Individu lama akan digantikan dengan individu baru yang memiliki nilai *fitness* lebih baik.

Diagram alir algoritma genetika dapat dilihat pada Gambar 2.7.



**Gambar 2.7 Diagram Alir Algoritma Genetika**



## **BAB III**

### **METODE PENELITIAN**

Pada bab ini dijelaskan langkah-langkah pengerjaan tugas akhir. Metode penelitian yang digunakan berguna sebagai acuan agar tugas akhir ini dapat berjalan secara sistematis.

#### **3.1 Studi Literatur**

Pada tahap ini dilakukan adalah identifikasi masalah dan pemahaman teori melalui pencarian dan pengumpulan materi atau sumber pendukung dari permasalahan yang diambil pada penulisan Tugas akhir. Referensi yang digunakan adalah buku-buku literatur, jurnal ilmiah, tugas akhir yang berkaitan dengan permasalahan yang dibahas yaitu tentang penerapan algoritma genetika dalam optimasi distribusi dengan menggunakan kendala *time window*.

#### **3.2 Pengumpulan Data**

Pada tahap ini dilakukan pengumpulan data distribusi yang meliputi lokasi tujuan distribusi, jarak dari satu lokasi ke lokasi yang lain, waktu yang dibutuhkan dari satu lokasi ke lokasi lain, kapasitas angkut kendaraan, jumlah kendaran, *time window* tiap lokasi dan data permintaan dari setiap lokasi distribusi yang diperoleh dari industri rumah tangga “Matoh Tenan”.

#### **3.3 Analisa dan Perancangan Sistem**

Pada tahap ini dilakukan analisa dan perancangan perangkat lunak pada setiap langkah dengan metode algoritma genetika untuk optimasi VRPTW pada distribusi kue. Seperti yang digambarkan pada Gambar 2.7.

### **3.4 Implementasi Perangkat Lunak**

Setelah melakukan analisa dan perancangan perangkat lunak, tahap selanjutnya adalah mengimplementasikan hasil analisis dan perancangan yang telah dilakukan dalam bentuk perangkat lunak. Pada tugas akhir ini digunakan perangkat lunak MATLAB. Implementasi yang dilakukan adalah untuk mendapatkan solusi terbaik dari proses algoritma genetika.

### **3.5 Evaluasi Hasil Uji Coba**

Pada tahap ini akan dilakukan evaluasi hasil dari sistem yang sudah dibangun dengan bantuan perangkat lunak. Dan akan terus dilakukan evaluasi hasil uji coba sampai menemukan solusi yang terbaik.

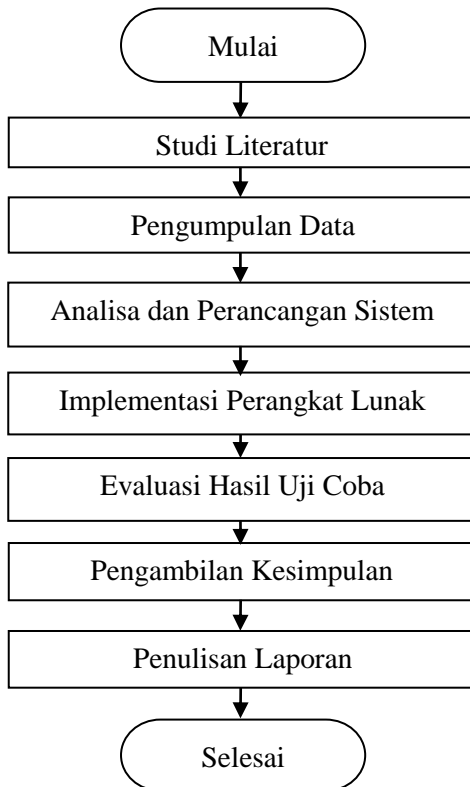
### **3.6 Pengambilan Kesimpulan**

Setelah melakukan evaluasi hasil uji coba dan didapatkan solusi yang terbaik, maka tahap selanjutnya dilakukan penarikan kesimpulan dari hasil pembahasan sebelumnya. Pemberian saran guna perbaikan dan pengembangan atas Tugas Akhir ini.

### **3.7 Penulisan Laporan**

Pada tahap terakhir adalah penulisan Tugas Akhir yang meliputi hasil kajian mengenai permasalahan yang dibahas dalam bentuk laporan Tugas Akhir.

Diagram alir pengerjaan Tugas Akhir ditunjukkan pada Gambar 3.1.



**Gambar 3.1 Diagram Alir Pengerjaan Tugas Akhir**





## BAB IV ANALISA DAN PERANCANGAN SISTEM

### 4.1 Proses Pengambilan Data

Pada tahap ini dilakukan pengumpulan data yang akan digunakan untuk menyelesaikan permasalahan rute distribusi kue. Data yang dibutuhkan adalah jarak antar toko kue, waktu yang dibutuhkan dalam proses pengiriman barang antar toko kue, waktu pelayanan (*time windows*), serta data-data pendukung lainnya seperti jumlah kendaraan, kapasitas kendaraan dan biaya distribusi (biaya bahan bakar).

**Tabel 4.1 Biaya Distribusi**

Kendaraan	Biaya	Jarak yang ditempuh/hari	Biaya per kilometer
Kendaraan 1	9.000/ hari	21,9 km	410,95
Kendaraan 2	9.000/ hari	22,3 km	403,58
Kendaraan 3	10.000/ hari	25,4 km	393,70
Kendaraan 4	7.500/ hari	17,3 km	433,52
Jumlah			1641,75
Rata-rata biaya per kilometer			410,43

Jumlah kendaraan yang dipakai dalam proses distribusi sebanyak 4 motor dengan kapasitas angkut setiap motor adalah 14 box kue. Dan waktu mulai pendistribusian adalah pukul 06.00 WIB. Data yang digunakan dapat dilihat pada Lampiran A.

### 4.2 Pengolahan Data

Pada tahap pengolahan data, data yang diperoleh akan diolah sesuai dengan metode yang digunakan, adapun tahapan pengolahan data adalah sebagai berikut:

1. Menentukan jarak dan waktu perjalanan dari tempat produksi kue ke toko kue dan dari toko kue satu ke toko kue lainnya. Jarak dan waktu perjalanan didapatkan dengan menggunakan bantuan *google maps*.
2. Membentuk populasi awal kemudian melakukan seleksi individu, selanjutnya reproduksi dengan melakukan *crossover* dan mutasi sehingga dihasilkan populasi yang baru setelah itu dihitung nilai fitness dari masing-masing individu dan yang terakhir dilakukan evaluasi (penggantian populasi).
3. Membuat implementasi program algoritma genetika.
4. Kemudian melakukan input parameter terhadap program algoritma genetika yang telah dibuat dengan menggunakan nilai parameter-parameter algoritma genetika yang paling baik agar mendapatkan hasil yang terbaik.

#### 4.2.1 Representasi Kromosom

Untuk menyelesaikan permasalahan VRPTW menggunakan algoritma genetika, langkah pertama adalah representasi kromosom. Dalam kasus VRPTW ini, kromosom direpresentasikan dalam bentuk permutasi dari toko kue – toko kue yang akan dikunjungi. Setiap kromosom harus memenuhi batasan atau syarat yaitu adanya batasan kapasitas pada kendaraan. Representasi kromosom dapat dilihat pada Gambar 4.1.

26	4	7	1	18	...	...	...	16	5	21	3	8	25
----	---	---	---	----	-----	-----	-----	----	---	----	---	---	----

**Gambar 4.1 Representasi Bentuk kromosom**

Dari Gambar 4.1 dapat dilihat bahwa nomor yang tertulis pada representasi kromosom merupakan urutan toko kue yang

akan dikunjungi sebanyak 26 toko kue, sehingga gen yang ada pada kasus ini mewakili 1 toko kue.

#### 4.2.2 Inisialisasi Populasi

Sebelum melakukan inisialisasi populasi, terlebih dahulu ditentukan parameter-parameter yang akan digunakan. Berdasarkan rekomendasi dari Grefensette bila rata-rata *fitness* setiap generasi digunakan sebagai indikator, maka ukuran populasi yang baik adalah tidak kurang dari 30 dan untuk  $P_c$  dan  $P_m$  adalah kombinasi yang berjumlah 0,9. Ukuran populasi untuk kasus ini adalah 150,  $P_c=0,2$  dan  $P_m=0,7$ . Kriteria penghentian yang digunakan adalah iterasi maksimum (banyaknya generasi) yaitu 50.

Proses inisialisasi adalah proses pembentukan populasi awal yang diperoleh dengan rekombinasi kromosom sebanyak ukuran populasi (*popsiz*e). Proses inisialisasi populasi dilakukan secara acak/*random*. Kemudian setiap individu dalam populasi awal dibagi kedalam empat blok.

**Tabel 4.2 Individu Hasil Inisialisasi Populasi Awal**

$K_i$	Blok 1	Blok 2	Blok 3	Blok 4
1	9 16 8 22 3 12 2	20 10 14 18 17 24	13 26 17 25 6 4	5 19 23 11 15 21
2	3 1 21 25 24 14 15	7 10 6 22 20 9 11	19 2 17 23 16 18	13 4 8 26 5 12
3	7 26 3 14 23 8 13	4 5 1 2 18 10 12	6 17 16 21 11 25	22 9 20 15 19 24
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
150	4 15 16 12 5 6 10	8 9 20 3 14 7 17	13 26 11 19 23 25	22 24 21 2 18 1

Setiap blok merepresentasikan kendaraan dan setiap kendaraan memiliki kapasitas yang sama yaitu 14. Selanjutnya dari bilangan acak yang dibangkitkan didapatkan 150 individu atau kromosom pada populasi awal seperti pada Tabel 4.2.

#### 4.2.3 Nilai *Fitness*

Setelah melakukan pembangkitan populasi awal dan mendapat kan populasi yang berjumlah sesuai dengan ukuran populasi yaitu 150 selanjutnya akan dilakukan perhitungan nilai *fitness*. Perhitungan nilai *fitness* dilakukan sesuai dengan fungsi *fitness* pada persamaan (2.2). Nilai *fitness* individu 1 yang memiliki urutan rute 9-16-8-22-3-12-2-20-10-14-18-1-7-24-13-26-17-25-6-4-5-19-23-11-15-21 dijelaskan pada Lampiran D. Dengan waktu keberangkatan kendaraan pada pukul 06.00 dan lama pelayanan di setiap toko adalah 10 menit.

#### 4.2.4 Seleksi

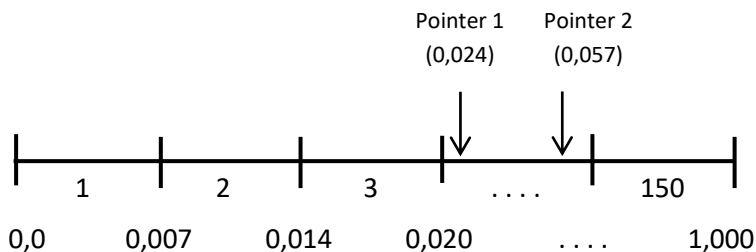
Sebelum dilakukan *crossover* akan dilakukan seleksi terlebih dahulu untuk memilih *parent* (induk). Pada penelitian ini digunakan seleksi metode *stochastic universal sampling*, di dalam metode ini diperlukan nilai *fitness*, total *fitness*, peluang *fitness* tiap individu dan peluang kumulatif tiap individu. Tabel 4.3 menunjukkan nilai *fitness* yang dihitung berdasarkan persamaan (2.2), total *fitness* yang dihitung berdasarkan persamaan (2.3), peluang *fitness* yang dihitung berdasarkan persamaan (2.4) dan peluang kumulatif yang dihitung berdasarkan persamaan (2.5).

Pada seleksi metode ini setiap individu dipetakan dalam suatu segmen garis secara berurutan. Kemudian diberikan sejumlah pointer sebanyak individu yang ingin diseleksi pada garis tersebut menggunakan rumus  $P_c$  (Probabilitas *Crossover*)  $\times$  ukuran populasi, sehingga dalam penelitian ini dipilih  $0,2 \times$

150 = 30 individu untuk menjadi *parent*. Jarak antar pointer ditentukan dengan rumus  $1/t$ , dengan  $t$  adalah banyaknya individu yang akan dipilih. Jarak antar pointer dalam penelitian ini adalah  $1/30 = 0,033$  Kemudian letak pointer pertama ditentukan dengan membangkitkan bilangan acak antara 0 dan 0,033.

**Tabel 4.3 Nilai *fitness*,  $p_i$  dan  $q_i$  tiap individu**

Kromosom $i$	Nilai <i>fitness</i>	$p_i$	$q_i$
Kromosom 1	0,0096	0,0066	0.007
Kromosom 2	0,0098	0,0067	0,014
Kromosom 3	0,0090	0,0062	0,020
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
Kromosom 150	0,0127	0,0087	1,00
Jumlah	1,4505		



**Gambar 4.2 Pemetaan Individu pada Seleksi *Stochastic Universal Sampling***

Pembangkitan bilangan acak antara 0 dan 0,033 diperoleh bilangan 0,024 sehingga pointer ke-1 terletak pada titik 0,024, pointer ke-2 terletak pada titik  $0,024+0,033=0,057$  dan seterusnya seperti yang digambarkan pada Gambar 4.2. Sehingga hasil yang diperoleh setelah seleksi seperti pada Tabel 4.4.

**Tabel 4.4 Hasil Seleksi Untuk Mendapatkan *Parent***

Parent $i$	Urutan Rute										
Parent 1	4	16	19	23	9	...	22	12	10	5	14
Parent 2	2	25	24	1	17	...	10	4	14	5	15
Parent 3	1	3	18	2	10	...	26	4	11	12	8
⋮	⋮										
⋮	⋮										
Parent 30	12	25	16	3	10	...	20	22	5	18	13

#### 4.2.5 *Crossover*

Setelah dilakukan seleksi untuk mendapatkan *parent* selanjutnya dilakukan proses *crossover*. Penentuan dilakukan *crossover* atau tidak dari individu yang menjadi *parent* adalah dengan membangkitkan bilangan acak  $r$  antara 0 dan 1. Jika  $r < P_c$  maka *crossover* dilakukan. Parameter  $P_c$  telah ditentukan sebelumnya yaitu sebesar 0,2. Pembangkitan bilangan acak  $r$  diperoleh nilai  $r = 0,1435$ , sehingga *crossover* dilakukan. Pada penelitian ini menggunakan metode *order crossover* dengan dua titik potong. Dua *parent* dari total banyaknya *parent* akan dipilih untuk dilakukan *crossover*. Langkah- langkah untuk melakukan *order crossover* dengan dua titik potong adalah sebagai berikut:

1. Membangkitkan bilangan acak antara 1 dan panjang kromosom sebagai dua titik potong untuk memilih *substring*

dari kedua induk. Hasil pengacakan bilangan didapatkan bilangan 2 dan 8 yang artinya titik 2 dan 8 sebagai titik potong. Gen-gen dari gen ke -2 sampai gen ke-8 pada kromosom *parent* menjadi *substring* yang akan ditukar.

K 1	4	16 19 23 9 3 15 13	18 21 ... 12 10 15 4
K 2	2	25 24 1 17 7 12 19	20 6 ... 4 14 5 15

2. Untuk mendapatkan kromosom anak yang pertama yaitu dengan menukar *substring* kromosom *parent* yang pertama dengan kromosom *parent* yang kedua. Kosongkan terlebih dahulu untuk gen – gen selain *substring*.

A 1	-	16 19 23 9 3 15 13	- - - - - ... - - -
A 2	-	25 24 1 17 7 12 19	- - - - - ... - - -

3. Isikan gen – gen *parent* yang pertama ke gen-gen anak yang pertama dengan dimulai dari gen setelah titik potong yang kedua hingga akhir gen pada kromosom *parent* yang pertama, kemudian dilanjutkan dari gen pertama hingga gen pada titik potong kedua pada kromosom *parent* yang pertama.

A 1	12	16 19 23 9 3 15 13	20 6 ... 24 1 17 7
-----	----	--------------------	--------------------

4. Isikan gen-gen *parent* yang kedua ke gen-gen anak yang kedua seperti pada langkah ketiga.

A 2	13	25 24 1 17 7 12 19	18 21 ... 2 9 3 15
-----	----	--------------------	--------------------

Dengan langkah-langkah tersebut didapat anak 1 dan 2, dan dengan cara yang sama dilakukan proses *crossover* secara berulang hingga mendapat anak seperti pada Tabel 4.5.

**Tabel 4.5 Kromosom Hasil *Crossover***

Kromosom	Urutan Rute
Kromosom 1	12 16 19 23 9 3 ... 25 24 1 17 7
Kromosom 2	13 25 24 1 17 7 ... 16 23 9 3 15
Kromosom 3	8 18 2 21 24 6 ... 1 14 9 11 17
⋮	⋮
Kromosom 30	26 22 17 2 19 11 ... 16 24 23 20 25

#### 4.2.6 Mutasi

Setelah proses *crossover* selanjutnya adalah proses mutasi. Proses mutasi yang dilakukan pada kromosom adalah untuk mendapatkan urutan rute yang berbeda dari sebelumnya. Nilai  $P_m = 0,7$  menunjukkan probabilitas mutasi, artinya pada setiap iterasi atau generasi kemungkinan terjadinya mutasi adalah 0,7. Proses mutasi dilakukan pada kromosom hasil dari poses *crossover*. Metode mutasi yang digunakan pada masalah VRPTW ini adalah mutasi dengan metode *exchange*. Dari total kromosom hasil proses *crossover*, dibangkitkan bilangan  $r$  antara 0 dan 1. Pembangkitan bilangan random ini adalah untuk menentukan kromosom tersebut dikenakan proses mutasi atau tidak. Apabila nilai  $r < P_m$  maka akan dilakukan proses mutasi. Pembangkitan bilangan acak dapat dilihat pada Tabel 4.6.

Pada Tabel 4.6 dapat kita lihat bahwa kromosom 1 memiliki nilai  $r < P_m$  yaitu 0,0992 maka akan dilakukan proses mutasi pada kromosom 1 dengan metode *exchange* seperti pada Gambar 4.3.

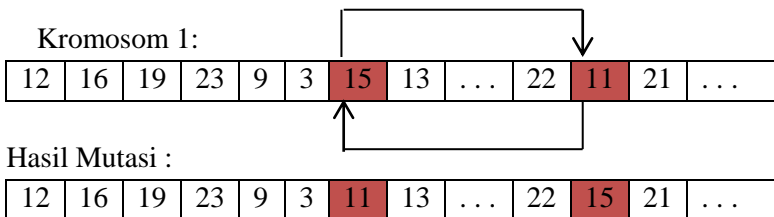


Langkah – langkah untuk melakukan mutasi *exchange* adalah:

1. Memilih secara acak dua posisi gen yaitu sebagai posisi gen yang akan ditukar posisi gennya, untuk kromosom 1 didapatkan bilangan acak 7 dan 13.
2. Menukar dua posisi gen yang sudah dipilih secara acak sebelumnya.

**Tabel 4.6 Pembangkitan Bilangan Acak ( $r$ )**

K	Urutan Rute	$r$
K 1	12 16 19 23 9 3 ... 25 24 1 17 7	0.0992
K 2	13 25 24 1 17 7 ... 16 23 9 3 15	0.5074
K 3	8 18 2 21 24 6 ... 1 14 9 11 17	0.3645
⋮	⋮	⋮
K 30	26 22 17 2 19 11 ... 16 24 23 20 25	0.6089



**Gambar 4.3 Proses Mutasi Menggunakan Metode *Exchange Mutation***

#### 4.2.7 Penggantian Populasi (Evaluasi)

Setelah melalui proses seleksi, *crossover* dan mutasi, maka langkah selanjutnya adalah penggantian populasi. Penggantian populasi ini dilakukan untuk menjamin bahwa individu pada generasi selanjutnya adalah individu terbaik. Pada penelitian ini

dilakukan proses penggantian dengan menggabungkan populasi awal yang sudah dibangkitkan sebelumnya pada Tabel 4.2 dengan populasi baru hasil dari proses *crossover* dan mutasi pada Tabel 4.7.

**Tabel 4.7 Populasi Hasil Proses *Crossover* Dan Mutasi**

<b>Kromosom</b>	<b>Urutan Rute</b>
Kromosom 1	12 16 19 23 9 3 ... 25 24 1 17 7
Kromosom 2	13 25 24 1 17 7 ... 11 23 9 3 15
Kromosom 3	8 18 2 21 24 6 ... 1 14 9 11 17
⋮	⋮
Kromosom 30	26 22 17 2 7 11 ... 16 24 23 20 25

Hasil dari penggabungan populasi seperti pada Tabel 4.8. Dari gabungan populasi tersebut, kromosom - kromosom akan dievaluasi berdasarkan nilai *fitness*. Nilai *fitness* pada kasus ini adalah jumlah jarak tempuh ditambah dengan jumlah pinalti. Semakin besar nilai *fitness*nya maka semakin berkemungkinan menjadi generasi selanjutnya.

**Tabel 4.8 Populasi Gabungan**

<b>Kromosom</b>	<b>Urutan Rute</b>
Kromosom 1	9 16 8 22 3 12 ... 19 23 11 15 21
Kromosom 2	3 1 21 25 24 14 ... 11 23 9 3 15
Kromosom 3	7 26 3 14 23 8 ... 1 14 9 11 17
⋮	⋮
⋮	⋮
Kromosom 180	26 22 17 2 7 11 ... 16 24 23 20 25

Populasi baru yang sesungguhnya didapatkan dari populasi gabungan yang nilai *fitness*nya terbaik dan akan dipertahankan jumlahnya sesuai dengan jumlah populasi awal. Kromosom yang memiliki nilai *fitness* terendah akan dihapus dari populasi. Sehingga populasi baru tersebut memiliki jumlah kromosom sama dengan populasi awal dengan kromosom yang memiliki nilai *fitness* terbaik dari populasi gabungan.

### 4.3 Implementasi Program dan Analisa Data

Implementasi program untuk permasalahan distribusi kue dengan mempertimbangkan jarak antar toko, kapasitas kendaraan dan *time windows* menggunakan algoritma genetika didasarkan pada perumusan algoritma genetika yang telah dibahas sebelumnya.

Dalam implementasi program ini, dibuat beberapa *script* sebagai berikut:

#### 1. VRPTW

*Script* VRPTW ini adalah *script* utama yang digunakan untuk menjalankan program. Isi *script* ini adalah mendefinisikan jarak masing - masing toko dalam bentuk matriks, waktu tempuh masing - masing toko dalam bentuk matriks, *demand* (permintaan) masing – masing toko, *time widows* dari masing – masing toko, panjang dari satu kromosom (individu) serta banyaknya iterasi (generasi). Di dalam *script* ini juga dibangkitkan bilangan random untuk mendapatkan populasi awal sebagai berikut:

```
for i=1: UkuranPopulasi
    Populasi(i,:)=randperm(PanjangKromosom);
end
```

Selain itu juga ada proses iterasi yang didalamnya terjadi *crossover* dan mutasi serta penggantian populasi. Untuk menjalankan proses iterasi, *source code* yang digunakan digunakan adalah sebagai berikut:

```

Iterasi=1;
while Iterasi <= Term
Parent=MemilihInduk(Populasi,UkuranPopulasi,Pc,Q);
    disp('Parent =');
    disp(Parent);

randc = rand;
disp(randc);
if(randc<Pc)
    t=floor(UkuranPopulasi*Pc);
    for i=1:t-1;
        j = i+1;
        if (mod(j,2) == 0)
            Child(i,j,:)=Crossover(Parent(i,:),Parent(j,:));
        end
    end

    disp('Child hasil crossover = ');
    disp(Child);

for i=1:t
    randm=rand;
    disp(randm);
    if (randm<Pm)
        MutasiPoint=ceil(rand(1,2).*PanjangKromosom);
        trans(i)=Child(i,MutasiPoint(2));
        Child(i,MutasiPoint(2))= Child(i,MutasiPoint(1));
        Child(i,MutasiPoint(1))=trans(i);
    end
end
Populasi=[Populasi;Child]

```

dalam *script* utama ini juga akan menampilkan grafik ketika dilakukan running. Untuk melihat *source code* lebih lengkap ada pada Lampiran E.

## 2. Memilih Induk

Pada *script* ini tertulis *source code* fungsi untuk menyeleksi beberapa kromosom yang akan dijadikan *parent* (induk) dari populasi yang sudah dibangkitkan (sudah ada). *Source code* untuk memilih *parent* (induk) adalah sebagai berikut:

```
function
Parent=MemilihInduk(Populasi,UkuranPopulasi,Pc,Q)
t=floor(length(Populasi)*Pc);
Pointer(1)=0+((1/t)*rand);
for i=2:t
    Pointer(i)=(1/t)+Pointer(i-1);
end

for i=1:t
    for m=2:UkuranPopulasi
        if ((Pointer(i)>=Q(m-1)) && (Pointer(i)<Q(m)))
            Parent(i,:)=Populasi(m,:);
        elseif (Pointer(i)<=Q(1))
            Parent(i,:)=Populasi(1,:);
        end
    end
end
end
```

## 3. Crossover

Di dalam *script* ini terdapat fungsi untuk melakukan proses crossover. Fungsi tersebut menjelaskan bagaimana proses *crossover* dengan menggunakan metode *order crossover*. *Source*

*code* proses *crossover* menggunakan metode *order crossover* adalah sebagai berikut :

```
function Child=Crossover(Parent1,Parent2)

XoverPoint=[ceil(rand*(length(Parent1)-1))
ceil(rand*(length(Parent1)-1))];
a=min(XoverPoint);
b=max(XoverPoint);
Child=zeros(2,length(Parent1));

for i=a:b
    Child(1,i)=Parent1(i);
    Child(2,i)=Parent2(i);
end

for i=1:length(Parent1)
    if(i>b)
        Reorder(1,(i-b))=Parent1(i);

        Reorder(2,(i-b))=Parent2(i);
    end
    if (i<=b)
        Reorder(1,(length(Parent1)-b+i))=Parent1(i);
```

*Source code* secara keseluruhan dapat dilihat pada Lampiran F. Sedangkan untuk proses mutasi menggunakan metode *Exchange mutation* dapat dijalankan dengan menggunakan *source code* seperti berikut:

```
for i=1:t
    randm=rand;
    disp(randm);
    if (randm<Pm)
```

```

MutasiPoint=ceil(rand(1,2).*PanjangKromosom);
trans(i)=Child(i,MutasiPoint(2));
Child(i,MutasiPoint(2))=Child(i,MutasiPoint(1));
Child(i,MutasiPoint(1))=trans(i);
end
end

```

#### 4. Hitung Jarak

Pada *script* ini terdapat fungsi untuk menghitung jarak dari satu toko ke toko lainnya yang telah dikunjungi oleh beberapa kendaraan dalam satu kromosom (individu) pada sebuah populasi. Untuk menjalankan fungsi menghitung jarak digunakan *source code* sebagai berikut:

```

function TotalJarak = HitungJarak(Populasi,Jarak)
for i=1:length(Populasi)
    for k=1:4
        if (k==1)
            Distance(i,k)=Jarak(1,(Populasi(1,1)+1));
            for j=1:(B1-1)
                Distance(i,k)=Distance(i,k)+Jarak((Populasi(i,j)+1),(Populasi
                (i,j+1)+1));
            end
            Distance(i,k)=
            Distance(i,k)+Jarak((Populasi(i,B1)+1),1);
        elseif (k==2)
            Distance(i,k)=Jarak(1,(Populasi(1,B1+1)+1));
            for j=(B1+1):(B1+B2-1)
                Distance(i,k)=Distance(i,k)+Jarak((Populasi(i,j)+1),(Populasi
                (i,j+1)+1));
            end
        end
    end
end

```

*Source code* secara keseluruhan dapat dilihat pada Lampiran G.

## 5. Hitung Pinalti

Pada *script* ini terdapat fungsi untuk menghitung pinalti kapasitas yang diangkut oleh setiap kendaraan yang berdistribusi dari satu toko menuju ke toko yang lain. Untuk menjalankan fungsi tersebut menggunakan *source code* seperti berikut:

```
function PinaltiKapasitas = HitungPinalti(Populasi,Demand)

for i=1:length(Populasi)
    for k=1:4
        if (k==1)

load(i,k)=0;
        for d=1:B1
            load(i,k)=load(i,k)+Demand(Populasi(i,d)+1);
        end
        elseif (k==2)
            load(i,k)=0;
            for d=(B1+1):(B1+B2)
                load(i,k)=load(i,k)+Demand(Populasi(i,d)+1);
            end
            elseif (k==3)
                load(i,k)=0;
                for d=(B1+B2+1):(B1+B2+B3)
                    load(i,k)=load(i,k)+Demand(Populasi(i,d)+1);
                end
            else
                load(i,k)=0;
                for d=(B1+B2+B3+1):(B1+B2+B3+B4)
                    load(i,k)=load(i,k)+Demand(Populasi(i,d)+1);
                end
            end
            if (load(i,k)>14)
                Pinalti(i k)=load(i k)-14;
            end
        end
    end
end
```

Untuk *source code* secara lengkap dapat dilihat pada Lampiran H.



## 6. Hitung Pinalti TW

Di dalam *script* ini terdapat fungsi untuk menghitung pinalti *time windows* yang dilakukan oleh kendaraan yang berdistribusi apabila dalam proses pendistribusian melampaui *time windows* yang dimiliki oleh masing – masing toko. Fungsi tersebut dapat berjalan dengan menggunakan *source code* sebagai berikut:

```
function PinaltiWaktu = HitungPinaltiTW(Populasi,Waktu)
akhir = [ 720 510 540 510 540 540 570 540 540 570 600
510 540 570 540 570 540 570 570 540 540 540 540 540
540 570 ];

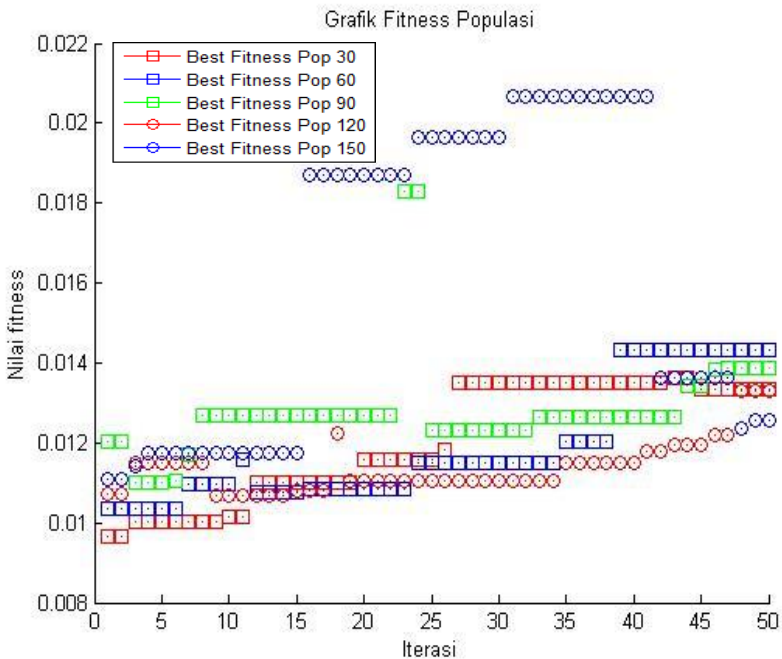
for i=1:length(Populasi)
    for k=1:4
        if (k==1)
            TW(i,k)=360+Waktu(1,(Populasi(1,1)+1));
            for j=1:(B1-1)
                TW(i,k)=TW(i,k)+Waktu((Populasi(i,j)+1),(Populasi(i,j+1)+1)
                ))+10;
                if (TW(i,k)>akhir(1,(Populasi(i,j+1)+1)))
                    PinaltiTW(i,k)=TW(i,k)-akhir(1,(Populasi(i,j)+1));
                else
                    PinaltiTW(i,k)=0;
                end
            end
        end
        elseif (k==2)
            TW(i,k)=360+Waktu(1,(Populasi(1,B1+1)+1));
            for j=(B1+1):(B1+B2-1)
                TW(i,k)=TW(i,k)+Waktu((Populasi(i,j)+1),(Populasi(i,j+1)+1)
                ))+10;

                if TW(i,k)>akhir(1,(Populasi(i,j+1)+1)))
                    PinaltiTW(i,k)=TW(i,k)-akhir(1,(Populasi(i,j)+1));
                else
                    PinaltiTW(i,k)=0;
                end
            end
        end
    end
end
```

Untuk *source code* secara lengkap dapat dilihat pada Lampiran I.

### 4.3.1 Analisa Uji Coba Ukuran Populasi

Uji coba populasi ini menggunakan ukuran populasi kelipatan 30, mulai dari 30 sampai 150. Pada percobaan ini banyaknya generasi sebesar 50. Kombinasi nilai  $P_c$  dan  $P_m$  adalah 0,4 dan 0,5. Didapatkan hasil uji coba seperti pada Gambar 4.4.



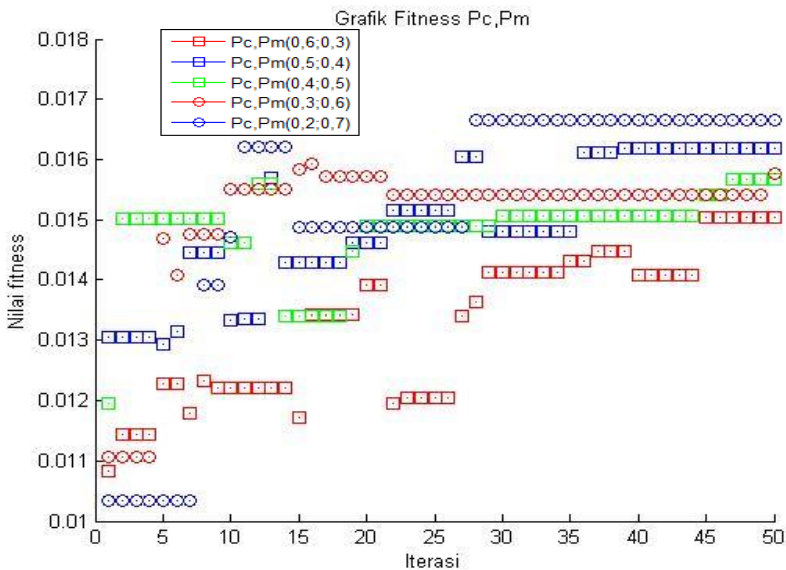
**Gambar 4.4 Grafik Uji Coba Populasi**

Pada Gambar 4.4 rata-rata nilai *fitness* terbesar terjadi pada ukuran populasi 150. Pada ukuran populasi 150 merupakan

ukuran populasi yang paling optimal untuk permasalahan VRPTW pada distribusi kue ini. Semakin tinggi ukuran populasi dapat mempengaruhi nilai *fitness* yang dihasilkan, tetapi waktu yang dibutuhkan untuk proses algoritma genetika juga semakin besar apabila semakin banyak ukuran populasinya.

### 4.3.2 Analisa Uji Coba Kombinasi Probabilitas Crossover dan Probabilitas Mutasi

Uji coba populasi ini menggunakan kombinasi  $P_c$  dan  $P_m$  mulai 0,6 : 0,3 sampai 0,2 : 0,7. Pada percobaan ini banyaknya generasi sebesar 50. Ukuran populasi sebesar 150. Didapatkan hasil uji coba seperti pada Gambar 4.5.



**Gambar 4.5 Grafik Uji Coba Kombinasi Probabilitas Crossover dan Probabilitas Mutasi**

Pada Gambar 4.5 rata-rata nilai *fitness* terbesar terjadi pada kombinasi  $P_c$  dan  $P_m$  sebesar 0,2 dan 0,7. Semakin besar nilai  $P_c$  dan  $P_m$  maka kemungkinan individu mengalami proses *crossover* dan mutasi semakin besar dan akan banyak juga individu baru yang dihasilkan. Dengan banyaknya individu baru yang dihasilkan maka nilai *fitness* yang dihasilkan juga bermacam-macam. Sehingga kemungkinan untuk mendapatkan nilai *fitness* yang lebih besar juga akan semakin besar.

#### 4.4 Beberapa Contoh Rute

Ada beberapa rute yang didapat dari parameter berbeda yang mungkin digunakan dalam masalah pendistribusian kue pada tugas akhir ini yaitu :

1. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,2 dan 0,7. Dimulai dari lokasi/titik awal  
 Rute kendaraan 1 : 9 → 3 → 5 → 6 → 10 → 4 → 12  
 Rute kendaraan 2 : 1 → 18 → 24 → 11 → 20 → 23 → 13  
 Rute kendaraan 3 : 26 → 7 → 14 → 25 → 21 → 19  
 Rute kendaraan 4 : 2 → 8 → 22 → 17 → 16 → 15  
 Dengan total jarak yang ditempuh sejauh 62,31 km dan nilai *fitness* sebesar 0,0160 dan biaya yang dibutuhkan adalah sebesar  $62,31 \times 410,43 = 25.573$  untuk setiap harinya.
2. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,3 dan 0,6. Dimulai dari lokasi/titik awal  
 Rute kendaraan 1 : 12 → 20 → 11 → 23 → 14 → 17 → 16  
 Rute kendaraan 2 : 1 → 15 → 13 → 8 → 2 → 21 → 22  
 Rute kendaraan 3 : 6 → 5 → 4 → 3 → 9 → 10  
 Rute kendaraan 4 : 25 → 19 → 7 → 24 → 18 → 26  
 Dengan total jarak yang ditempuh sejauh 84,3 km dan nilai *fitness* sebesar 0,0118 dan biaya yang dibutuhkan adalah sebesar  $84,3 \times 410,43 = 34.599$  untuk setiap harinya.

3. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,4 dan 0,5. Dimulai dari lokasi/titik awal  
 Rute kendaraan 1 : 11 → 20 → 23 → 8 → 12 → 21 → 25  
 Rute kendaraan 2 : 18 → 22 → 13 → 17 → 15 → 1 → 24  
 Rute kendaraan 3 : 10 → 6 → 9 → 2 → 7 → 3  
 Rute kendaraan 4 : 5 → 14 → 4 → 16 → 26 → 19  
 Dengan total jarak yang ditempuh sejauh 85,76 dan nilai *fitness* sebesar 0,0116 dan biaya yang dibutuhkan adalah sebesar  $85,76 \times 410,43 = 35.198$  untuk setiap harinya.
4. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,5 dan 0,4. Dimulai dari lokasi/titik awal  
 Rute kendaraan 1 : 19 → 11 → 15 → 26 → 17 → 18 → 20  
 Rute kendaraan 2 : 13 → 16 → 1 → 12 → 23 → 24 → 25  
 Rute kendaraan 3 : 3 → 14 → 21 → 22 → 2 → 7  
 Rute kendaraan 4 : 10 → 6 → 5 → 8 → 4 → 9  
 Dengan total jarak yang ditempuh sejauh 82,46 km dan nilai *fitness* sebesar 0,0121 dan biaya yang dibutuhkan adalah sebesar  $82,46 \times 410,43 = 33.844$  untuk setiap harinya.
5. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,6 dan 0,3. Dimulai dari lokasi/titik awal  
 Rute kendaraan 1 : 1 → 18 → 17 → 22 → 19 → 14 → 25  
 Rute kendaraan 2 : 6 → 5 → 10 → 4 → 7 → 9 → 3  
 Rute kendaraan 3 : 11 → 20 → 2 → 6 → 15 → 21  
 Rute kendaraan 4 : 23 → 26 → 8 → 13 → 12 → 24  
 Dengan total jarak yang ditempuh sejauh 80,85 km dan nilai *fitness* sebesar 0,0123 dan biaya yang dibutuhkan adalah sebesar  $80,85 \times 410,43 = 33.183$  untuk setiap harinya.
6. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,2 dan 0,7 dengan kombinasi banyaknya toko

yang dikunjungi tiap kendaraan adalah 7 7 6 6 . Dimulai dari lokasi/titik awal

Rute kendaraan 1 : 12 → 2 → 24 → 11 → 9 → 22 → 13

Rute kendaraan 2 : 15 → 8 → 7 → 16 → 26 → 20 → 17

Rute kendaraan 3 : 10 → 6 → 14 → 4 → 5 → 3

Rute kendaraan 4 : 19 → 25 → 23 → 1 → 18 → 21

Dengan total jarak yang ditempuh sejauh 84,21 km dan nilai *fitness* sebesar 0,0119 dan biaya yang dibutuhkan adalah sebesar  $84,21 \times 410,43 = 34.562$  untuk setiap harinya.

7. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,2 dan 0,7 dengan kombinasi banyaknya toko yang dikunjungi tiap kendaraan adalah 6 6 7 7 . Dimulai dari lokasi/titik awal

Rute kendaraan 1 : 18 → 22 → 12 → 25 → 17 → 23

Rute kendaraan 2 : 1 → 2 → 6 → 9 → 4 → 3 → 15

Rute kendaraan 3 : 16 → 7 → 19 → 2 → 21 → 8 → 13

Rute kendaraan 4 : 20 → 14 → 11 → 24 → 5 → 10 → 6

Dengan total jarak yang ditempuh sejauh 87,81 km dan nilai *fitness* sebesar 0,0113 dan biaya yang dibutuhkan adalah sebesar  $87,81 \times 410,43 = 36.039$  untuk setiap harinya.

8. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,2 dan 0,7 dengan kombinasi banyaknya toko yang dikunjungi tiap kendaraan adalah 6 7 6 7 . Dimulai dari lokasi/titik awal

Rute kendaraan 1 : 12 → 21 → 24 → 1 → 18 → 17

Rute kendaraan 2 : 13 → 2 → 15 → 26 → 14 → 16 → 22

Rute kendaraan 3 : 6 → 20 → 23 → 8 → 25 → 19 → 5

Rute kendaraan 4 : 9 → 11 → 7 → 10 → 3 → 4

Dengan total jarak yang ditempuh sejauh 95,35 km dan nilai *fitness* sebesar 0,0104 dan biaya yang dibutuhkan adalah sebesar  $95,35 \times 410,43 = 39.134$  untuk setiap harinya.

9. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,2 dan 0,7 dengan kombinasi banyaknya toko yang dikunjungi tiap kendaraan adalah 7 6 7 6 . Dimulai dari lokasi/titik awal

Rute kendaraan 1 : 4 → 9 → 19 → 15 → 7 → 11 → 8

Rute kendaraan 2 : 1 → 22 → 20 → 21 → 23 → 24

Rute kendaraan 3 : 26 → 16 → 6 → 3 → 5 → 10 → 14

Rute kendaraan 4 : 2 → 12 → 25 → 18 → 17 → 13

Dengan total jarak yang ditempuh sejauh 84,75 km dan nilai *fitness* sebesar 0,0117 dan biaya yang dibutuhkan adalah sebesar  $84,75 \times 410,43 = 34.783$  untuk setiap harinya.

10. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,2 dan 0,7 dengan kombinasi banyaknya toko yang dikunjungi tiap kendaraan adalah 7 6 6 7 . Dimulai dari lokasi/titik awal

Rute kendaraan 1 : 9 → 19 → 17 → 18 → 13 → 7 → 21

Rute kendaraan 2 : 5 → 10 → 6 → 23 → 11 → 12

Rute kendaraan 3 : 16 → 4 → 22 → 15 → 3 → 25

Rute kendaraan 4 : 2 → 14 → 8 → 20 → 1 → 24 → 26

Dengan total jarak yang ditempuh sejauh 87,36 km dan nilai *fitness* sebesar 0,0114 dan biaya yang dibutuhkan adalah sebesar  $87,36 \times 410,43 = 35.855$  untuk setiap harinya.

11. Rute hasil dari parameter populasi sebesar 150 dan parameter  $P_c$  dan  $P_m$  0,2 dan 0,7 dengan kombinasi banyaknya toko yang dikunjungi tiap kendaraan adalah 6 7 7 6 . Dimulai dari lokasi/titik awal

Rute kendaraan 1 : 26 → 23 → 25 → 8 → 13 → 24

Rute kendaraan 2 : 20 → 10 → 6 → 11 → 15 → 16 → 22

Rute kendaraan 3 : 12 → 18 → 1 → 3 → 9 → 7 → 14

Rute kendaraan 4 : 4 → 17 → 19 → 2 → 21 → 5

Dengan total jarak yang ditempuh sejauh 87,16 km dan nilai *fitness* sebesar 0,0114 dan biaya yang dibutuhkan adalah sebesar  $87,16 \times 410,43 = 35.773$  untuk setiap harinya.

#### 4.5 Rute Optimal

Setelah mengetahui pengaruh dari masing – masing parameter dan mendapatkan parameter terbaik dari permasalahan VRPTW ini, akan didapatkan rute optimal dari permasalahan VRPTW dengan menggunakan parameter terbaik yang sudah didapatkan tersebut. Dengan menggunakan parameter populasi sebesar 150, parameter kombinasi antara probabilitas *crossover* dan probabilitas mutasi masing-masing sebesar 0,2 dan 0,7 maka didapatkan rute optimal sebagai berikut : 9 3 5 6 10 4 12 1 18 24 11 20 23 13 26 7 14 25 21 19 2 8 22 17 16 15  
Apabila rute digambarkan akan seperti pada Gambar 4.6. Dengan nilai *fitness* sebesar 0,0160 dan jarak tempuh dari total perjalanan oleh 4 kendaraan yaitu 62,31 km.

Rute optimal yang didapatkan adalah rute yang memiliki jarak terpendek, tidak terkena pinalti kapasitas artinya barang yang diangkut tidak melebihi kapasitas masing-masing kendaraan dan tidak terkena pinalti *time windows* artinya kendaraan melakukan pendistribusian kue sesuai dengan *time windows* yang dimiliki oleh masing-masing pelanggan.

Sehingga didapatkan rute optimal yang dapat menekan biaya pendistribusian kue. Dan berdasarkan Tabel 4.1 biaya yang dihabiskan dalam pendistribusian kue oleh 4 kendaraan menggunakan rute optimal yang telah didapatkan adalah sebesar  $62,31 \times 410,43 = Rp. 25.573$  untuk setiap harinya atau sebesar Rp. 51.146 untuk 2 hari. Apabila rute digambarkan akan seperti





Rute optimal yang didapatkan adalah rute yang memiliki jarak terpendek, tidak terkena pinalti kapasitas. Artinya barang yang diangkut tidak melebihi kapasitas masing-masing kendaraan dan tidak terkena pinalti *time windows* artinya kendaraan melakukan pendistribusian kue sesuai dengan *time windows* yang dimiliki oleh masing-masing pelanggan.

Sehingga didapatkan rute optimal yang dapat menekan biaya pendistribusian kue. Dan berdasarkan Tabel 4.1 biaya yang dihabiskan dalam pendistribusian kue oleh 4 kendaraan menggunakan rute optimal yang telah didapatkan adalah sebesar  $62,31 \times 410,43 = Rp. 25.573$  untuk setiap harinya atau sebesar *Rp.* 51.146 untuk 2 hari.

## BAB V PENUTUP

Bab ini berisi tentang kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan serta saran untuk penulisan selanjutnya.

### 5.1 Kesimpulan

Dari hasil pembahasan yang telah diperoleh pada bab sebelumnya, dapat disimpulkan bahwa:

1. Pengaruh parameter genetika untuk ukuran populasi dan kombinasi  $p_c$  dan  $p_m$  mempengaruhi hasil optimal penelitian ini. Pada Penelitian ini semakin besar ukuran populasi akan mempengaruhi nilai *fitness* yang dihasilkan akan semakin banyak. Dan untuk kombinasi  $p_c$  dan  $p_m$  pada penelitian ini semakin tinggi nilai  $p_c$  dan  $p_m$  maka individu semakin bervariasi yang menyebabkan nilai *fitness* juga akan semakin bervariasi akibat proses persilangan yang terjadi.
2. Dalam penelitian ini, parameter terbaik untuk ukuran populasi adalah 150 populasi dan untuk kombinasi  $p_c$  dan  $p_m$  adalah  $p_c$  sebesar 0,2 dan  $p_m$  sebesar 0,7.
3. Rute optimal yang didapatkan pada penelitian ini adalah

Dari lokasi/titik awal :

Rute kendaraan 1 : 9 → 3 → 5 → 6 → 10 → 4 → 12

Rute kendaraan 2 : 1 → 18 → 24 → 11 → 20 → 23 → 13

Rute kendaraan 3 : 26 → 7 → 14 → 25 → 21 → 19

Rute kendaraan 4 : 2 → 8 → 22 → 17 → 16 → 15

Dengan nilai *fitness* sebesar 0,0160, jarak tempuh dari total perjalanan oleh 4 kendaraan yaitu 62,31 km dan biaya yang harus dikeluarkan sebesar Rp. 25.573 untuk setiap harinya atau sebesar Rp. 51.146 untuk 2 hari.

## 5.2 Saran

Saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah :

1. Pada penelitian ini studi kasus yang digunakan hanya mempertimbangkan jarak, kapasitas, permintaan dan *time windows*. Untuk selanjutnya mungkin bisa ditambahkan kendala lain.
2. Untuk selanjutnya, diharapkan ada penelitian mengenai VRPTW menggunakan algoritma lain atau algoritma genetika dengan metode lain dalam seleksi, *crossover* maupun mutasinya agar diperoleh hasil yang lebih baik.

## DAFTAR PUSTAKA

- [1] Chopra, S., Meindl, Peter. 2010. *Supply Chain Management: Strategy, Planning and Operation*. New Jersey: Pearson Education.
- [2] Prana A, Raden. 2007. *Aplikasi Kombinatorial pada Vehicle Routing Problem*. Bandung: Jurusan Teknik Informatika Institut Teknologi Bandung.
- [3] Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Graha Ilmu.
- [4] Gen, M. and Cheng, R., 2000. *Genetic Algorithms and Engineering Optimization*. New York: John Wiley and Sons Inc.
- [5] Slamet, Alim S., dkk. 2014. *Vehicle Routing Problem (VRP) dengan Algoritma Genetika pada Pendistribusian Sayuran Dataran Tinggi*. Bogor: Institut Pertanian Bogor.
- [6] Mahmudy, Wayan F. 2014. *Improved Simulated Annealing for Optimization of Vehicle Routing Problem with Time Windows (VRPTW)*. Malang: Universitas Brawijaya.
- [7] Kallehauge, B., Larsen, J. and Madsen, O. 2006. *Lagrangian Duality Applied to The Vehicle Routing Problem with Time Windows*. Computers & Operations Research.
- [8] Áslaug SB. 2004. *Solving The Vehicle Routing Problem with Genetic Algorithm*. Odense: Informatics and Mathematical Modeling, Technical University of Denmark.
- [9] Gambardella, L. M., Taillard, E., & Agazzi, G. 1999. *A Multiple Ant Colony System for Vehicle Routing Problems With Time Windows*. New Ideas in Optimization: McGraw-Hill, London.
- [10] Nugraha, I. 2008. *Algoritma Genetik untuk Optimasi Penjadwalan Kegiatan Belajar mengajar*. Bandung: Institut Teknologi Bandung.
- [11] Sutojo, T., Mulyanto, E., dan Suhartanto. 2011. *Kecerdasan Buatan*. Yogyakarta: ANDI.

- [12] Mahmudy, F. W. 2013. *Algoritma Evolusi*. Malang: Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
- [13] Mahmudy, W.F., Marian, R.M., dan Luong, L.H.S., 2014. *Hybrid Genetic Algorithms for Part Type Selection and Machine Loading Problems with Alternative Production Plans in Flexible Manufacturing System*. *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*.
- [14] Setiawan, K. 2003. *Paradigma Sistem Cerdas*. Surabaya: Banyumedia.
- [15] Wati, A. W. 2011. *Penerapan Algoritma Genetika Daalam Optimasi Model Dan Simulasi Dari Suatu Sistem*. *Jurnal Keilmuan Teknik Industri*.
- [16] Prabowo, K. A. 2011. *Penerapan Algoritma Genetik Dua Populasi pada Kasus Transportasi Dua Tahap*. Malang: Fakultas Matematika dan Ilmu Pegetahuan Alam Universitas Brawijaya.
- [17] Kusumadewi, S., Purnomo, H. 2005. *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik*. Yogyakarta. Graha Ilmu.
- [18] Sarwadi, Anjar KSW. 2004. *Algoritma Genetika untuk Penyelesaian Masalah Vehicle Routing*. *Jurnal Matematika dan Komputer Universitas Diponegoro*.
- [19] Sulistiyorini, R., Mahmudi, W.F. 2015. *Penerapan Algoritma Genetika untuk Permasalahan Optimasi Distribusi Barang Dua Tahap*. Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.

## LAMPIRAN A

### Data Lokasi Toko Kue, *Time Windows*, dan *Demand*

No	Toko Kue	Alamat Toko Kue	<i>Time Window</i>	<i>Demand</i>
1	Melan Kue Basah	Jl. Raya Wiguna Timur	05.45 - 08.30	3
2	Kue Basah Tambak Rejo	Jl. Tambak Rejo (Samping Indomaret)	06.00 - 09.00	1
3	Toko Aneka Kue Manalagi	Jl. Tropodo	05.00 - 08.30	3
4	Chi Chik Kue Basah	Jl. Raya Sedati Gede	05.30 - 09.00	2
5	Kue Basah Depan Pasar Betro	Jl. Garuda	05.30 - 09.00	2
6	Kue Basah depan Koramil	Jl. Rajawali	06.00 - 09.30	1
7	Toko Kue Isti	Wisma Tropodo	06.00 - 08.30	2
8	Kue Basah Depan Indomaret	Jl. Raya Wadung Asri	06.00 - 08.30	1
9	Toko Kue Karunia	Jl. Raya Pabean	06.00 - 09.30	1
10	Kue Basah Depan Toko Karunia	Jl. Raya Pulungan	05.30 - 09.00	1
11	Warung Pagi Tropodo	Wisma Tropodo	05.30 - 08.30	2
12	Kue Basah depan Pasar Tambak Rejo	Jl. Tambak Sawah	05.30 - 09.00	2
13	Aneka Kue Basah Tikungan	Jl. Gununganyar Kidul	05.30 - 09.30	2
14	Kue Basah Depan Superindo	Jl. Raya Tropodo	05.30 - 09.00	1
15	Kue Basah Rungkut	Jl. Rungkut Menanggal Harapan	06.00 - 09.30	1
16	Kue Basah Bu Yayuk	Jl. Rungkut Menanggal Utara	06.00 - 09.00	2
17	Aneka kue Warna – Warni	Jl. Rungkut Mapan	06.00 - 09.30	2

### Lampiran A Lanjutan

No	Toko Kue	Alamat Toko Kue	Time Window	Demand
18	Kue Basah Medokan	Jl. Raya Medokan sawah No. 68	05.30 - 09.30	1
19	Kue Basah Depan Alfamidi	Jl. Brigjen Katamso	06.00 - 09.00	2
20	Kue Basah Toko Semangat	Jl. Brigjen Katamso	06.00 - 09.00	1
21	Toko Kue Basah 3 Putra	Jl. Raya Tambak Rejo	05.45 - 09.00	2
22	Toko Kue Pak Antok	Jl. Blimbing V	06.00 - 09.00	2
23	Lapak Sayur Pagi Bu Mus	Jl. Taman Lestari	05.30 - 09.00	1
24	Toko Kue Yoland	Jl. Taman Asri No. 31	05.30 - 09.00	2
25	Toko Kue Bu lidya	Jl. Jambu I	05.00 - 09.30	2
26	Glory Toko Kue Basah	Jl. Raya Taman Asri	05.30 - 09.30	3



Lampiran B

MATRIKS JARAK

	P	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
P	-	4,8	1,1	3	4,9	6,4	7,6	2,6	1,3	3,5	7,8	3	1,5	1,2	2,6	2,3	2,7	3	4,3	4,6	2,9	1,4	1,7	1,8	1,1	0,85	1,5
1	4,8	-	5,6	7,6	9,6	11,1	12,2	7,3	5,7	8,1	12,3	7,7	6,1	3,6	7,2	3,8	3,8	3,8	1	9,2	7,4	6,6	5,4	5,6	4,8	4,9	5,2
2	1,1	5,6	-	2,6	4	6	7,2	2,2	1,9	2,6	7,4	2,6	0,45	2	2,2	3,1	3,5	3,9	5,2	4,3	2,6	0,95	2,4	2	2,4	1,5	2,1
3	3	7,4	2,6	-	1,9	3,4	4,5	0,9	1,8	0,5	4,8	1,3	2,2	3,8	0,4	3,6	4	4,4	7	4,2	2,6	1,7	2,6	2,1	4,2	2,8	3,1
4	4,9	9,3	4	1,9	-	1,5	2,7	2,8	3,8	1,4	2,9	3,2	3,9	5,7	2,3	5,5	5,9	6,3	8,9	5,9	4,5	3,7	4,5	4	6,1	4,7	5
5	6,3	10,8	5,8	3,3	1,5	-	1,2	4,2	5,2	2,9	1,4	4,6	5,3	7,1	3,7	7	7,4	7,7	10,3	7,4	6	5,1	6	5,4	7,5	6,1	6,5
6	7,5	12,2	6,9	4,5	2,7	1,2	-	5,4	6,4	4	0,26	5,8	6,5	8,3	4,9	8,2	8,6	8,9	11,5	8,6	7,1	6,3	7,2	6,6	8,7	7,3	7,6
7	2,5	6,9	2,3	0,9	2,8	4,2	5,4	-	1,4	1,4	5,7	0,4	1,9	3,3	0,5	3,2	3,6	3,9	6,5	3,2	1,8	1,3	2,2	1,6	3,7	2,3	2,7
8	1,3	5,5	1,8	1,9	3,8	5,3	6,5	1,5	-	2,4	6,7	1,9	2,2	1,9	1,5	1,7	2,1	2,5	5	3,5	1,8	0,85	0,75	0,18	1,7	0,85	1,2
9	3,5	7,9	2,6	0,5	1,4	2,9	4	1,4	2,4	-	4,3	1,8	2,1	4,3	0,9	4,1	4,5	4,9	7,5	4,5	3,1	2,3	3,1	2,6	4,7	3,2	3,6
10	7,8	12,3	7,2	4,8	2,9	1,4	0,26	5,7	6,7	4,3	-	6,1	6,8	8,6	5,2	8,4	8,8	9,2	11,8	8,8	7,4	6,6	7,4	6,9	9	7,5	7,9
11	2,9	7,3	2,7	1,3	3,2	4,6	5,8	0,4	1,8	1,8	6,1	-	2,2	3,7	0,9	3,6	4	4,3	6,9	2,7	1,1	1,7	2,6	2	4,1	2,7	3,1
12	1,5	6,3	0,45	2,2	3,6	5,7	6,8	1,9	2,3	2,1	7,1	2,2	-	2,5	1,9	3,5	3,9	4,3	5,6	4,7	3	1,4	3	2,5	2,2	2	2,6
13	1,5	3,6	2,3	4,4	6,3	7,7	8,9	3,9	2,5	4,9	9,2	4,3	2,8	-	4	1,1	1,5	1,8	3,1	5,9	4,2	3,3	2,1	2,6	1,4	1,6	1,9
14	2,6	7	2,2	0,4	2,3	3,7	4,9	0,5	1,5	0,9	5,2	0,9	1,9	3,4	-	3,2	3,6	4	6,5	3,6	2,2	1,3	2,2	1,7	3,8	2,4	2,7
15	1,6	3,8	3,4	3,7	5,6	7	8,2	3,2	1,7	4,2	8,5	3,6	3,8	1,1	3,3	-	0,4	0,75	3,4	5,2	3,5	2,6	1,6	1,9	2,5	2,7	2,1
16	3	4,2	3,8	4,1	6	7,4	8,6	3,6	2,1	4,6	8,9	4	4,2	1,4	3,7	0,4	-	0,35	3,6	5,6	3,9	3	2	2,3	2,9	2,9	2,5
17	3,5	4,2	4,3	4,6	6,5	7,9	9,1	4,1	2,6	5,1	9,4	4,5	4,7	2	4,2	0,9	0,5	-	3,2	6,1	4,4	3,5	2,5	2,8	3,4	3,6	3
18	4,5	1,1	5,3	7,3	9,2	10,7	11,9	6,9	4,8	7,7	12,1	6,7	5,7	3,1	6,4	3,3	2,9	3	-	8,3	6,6	5,7	4,7	4,9	4,4	4,6	4,8
19	4,6	9	4,5	4,2	5,9	7,4	8,9	3,2	3,5	4,5	9,1	2,7	4,9	5,4	3,6	5,2	5,6	6	8,5	-	1,7	3,5	4,2	3,6	5,8	4,3	4,7
20	2,9	7,2	2,7	2,4	4,3	6	6,9	1,6	1,8	3,1	7,4	1,1	3,1	3,7	2	3,5	3,9	4,3	6,8	1,7	-	1,8	2,5	1,9	4,1	2,4	3
21	1,4	6,2	0,95	1,7	3,6	5	6,2	1,2	0,9	2,2	6,5	1,6	1,4	2,4	1,3	2,6	3	3,4	5,5	3,3	1,6	-	1,6	1,1	2,1	1,6	2,1
22	1,9	5,2	2,5	2,6	4,5	5,9	7,1	2,1	0,65	3,1	7,4	2,5	2,9	1,6	2,2	1,6	2	2,3	4,7	4,1	2,4	1,5	-	0,8	1,9	1,5	0,6
23	1,1	5,9	1,7	2,6	4,5	5,9	7,1	2,1	0,6	3	7,4	2,5	2,2	2,1	2,2	2,2	2,6	3	5,2	4,1	2,4	1,5	1,2	-	1,2	0,7	1,3
24	1	5,5	1,7	3	4,9	6,3	7,5	2,5	1	3,5	7,8	2,9	2,2	1,9	2,6	2	2,3	2,7	5,1	4,5	2,8	1,9	0,6	1,2	-	0,2	0,45
25	0,85	5,7	1,5	2,8	4,7	6,1	7,3	2,3	0,85	3,3	7,6	2,7	2	2,1	2,7	2,1	2,5	2,9	5,2	4,3	2,6	1,5	0,85	1,3	0,55	-	0,6
26	1,8	5,1	2,4	2,5	4,4	5,9	7	2,1	0,55	3	7,3	2,5	2,8	1,5	2,1	1,5	1,9	2,3	4,6	4	2,3	1,4	0,23	0,7	1,9	1,4	-



Lampiran C

MATRIKS WAKTU

	P	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
P	-	12	3	8	12	15	19	7	6	9	19	9	4	4	7	7	8	10	13	11	7	4	6	6	4	4	4
1	11	-	14	16	21	24	27	15	12	18	26	17	15	9	15	11	11	11	3	19	15	14	11	12	9	10	10
2	4	17	-	7	9	13	16	7	6	8	17	8	1	7	6	10	12	13	17	10	7	3	9	7	6	6	8
3	9	18	9	-	4	8	11	2	4	1	12	4	7	8	1	8	9	10	16	12	5	5	7	5	8	8	9
4	16	25	14	7	-	4	8	9	10	5	9	11	10	14	7	15	16	17	23	18	12	11	13	11	14	14	14
5	19	27	14	10	4	-	4	12	14	8	5	14	13	18	10	19	20	22	27	21	15	14	16	14	17	17	18
6	23	31	18	14	8	4	-	16	18	12	1	18	18	21	14	22	24	25	32	25	20	20	22	19	23	22	23
7	8	17	7	3	8	12	17	-	4	4	16	1	5	8	2	11	12	14	17	11	8	5	7	5	8	8	9
8	6	15	6	5	10	14	19	4	-	6	19	5	7	4	4	7	8	9	15	8	4	3	3	1	8	4	5
9	12	20	9	2	4	7	12	5	7	-	13	7	8	12	4	15	17	18	21	16	11	8	12	9	15	13	15
10	24	24	19	14	8	4	1	17	20	13	-	21	19	29	18	33	34	35	38	28	25	23	28	24	31	28	31
11	10	19	8	4	10	14	18	2	6	6	19	-	7	15	4	18	20	22	26	10	6	7	14	10	16	12	16
12	5	16	1	8	11	13	17	6	8	8	18	8	-	7	5	11	12	13	17	11	8	4	9	7	8	7	9
13	4	11	7	11	16	20	24	9	5	12	26	12	8	-	8	3	5	6	11	12	8	7	4	5	2	4	3
14	8	17	6	1	7	11	15	2	4	3	16	3	5	8	-	8	9	11	19	13	5	4	6	4	8	7	8
15	9	12	12	11	16	20	24	10	5	12	27	12	12	4	8	-	1	2	10	13	9	8	5	6	7	8	8
16	10	13	12	12	17	21	26	11	6	13	28	13	13	5	9	1	-	1	8	15	11	9	6	7	8	10	9
17	11	11	14	14	19	24	28	13	9	16	30	14	14	6	11	2	1	-	8	18	14	12	8	9	9	10	10
18	13	3	16	20	24	29	33	18	15	22	35	21	16	9	16	9	7	8	-	23	19	18	14	16	11	13	12
19	12	21	12	13	18	23	25	11	8	16	28	9	12	11	12	13	14	17	22	-	4	9	11	9	13	12	14
20	8	17	8	11	16	17	26	8	4	10	24	6	12	7	9	10	11	13	19	4	-	5	7	5	9	8	10
21	5	17	3	5	10	14	19	4	3	7	19	6	4	7	4	8	10	13	17	9	5	-	6	4	10	7	9
22	8	14	9	8	13	18	22	7	2	10	24	9	9	4	6	6	7	9	13	11	7	6	-	3	4	6	3
23	5	17	7	9	14	18	22	8	4	11	24	10	8	8	8	9	10	12	18	12	8	6	7	-	6	3	5
24	6	14	8	8	13	17	21	7	2	10	23	8	9	3	6	5	6	9	13	10	7	5	2	3	-	1	1
25	5	15	6	9	14	19	23	8	4	11	24	9	7	4	7	7	8	10	16	11	8	7	3	4	3	-	2
26	7	13	8	7	12	17	21	6	2	9	22	8	7	3	5	5	6	8	14	10	6	4	1	2	3	5	-



## Lampiran D

Tabel Perhitungan *Fitness*

	Individu [9-16-8-22-3-12-2-20-10-14-18-1-7-24-13-26-17-25-6-4-5-19-23-11-15-21]								Jumlah
	Blok 1 9-16-8-22-3-12-2								
Node	0 ke 9	9 ke 16	16 ke 8	8 ke 22	22 ke 3	3 ke 12	12 ke 2	2 ke 0	
Waktu Berangkat Kendaraan	06.00	06.19	06.46	07.02	07.15	07.33	07.50		
Waktu Perjalanan (menit)	9	17	6	3	8	7	1		
Kendaraan Tiba	06.09	06.36	06.52	07.05	07.23	07.40	07.51		
<i>Earliest Time</i>	06.00	06.00	06.00	06.00	05.00	05.30	06.00		
Mulai Dilayani	06.09	06.36	06.52	07.05	07.23	07.40	07.51		
Waktu Tunggu	0	0	0	0	0	0	0		
Selesai	06.19	06.46	07.02	07.15	07.33	07.50	08.01		
<i>Latest time</i>	09.30	9.00	08.30	09.00	08.30	09.00	09.00		
Penalti (menit)	0	0	0	0	0	0	0		0
Kapasitas Yang diangkut	1	2	1	2	3	2	1		
Kapasitas Maksimum	14								
Penalti (kapasitas)	0								0

Jarak yg ditempuh (km)	3,5	4,5	2,1	0,75	2,6	2,2	0,45	1,1	17,2
	Blok 2 20-10-14-18-1-7-24								
Node	0 ke 20	20 ke 10	10 ke 14	14 ke 18	18 ke 1	1 ke 7	7 ke 24	24 ke 0	
Waktu Berangkat Kendaraan	06.00	06.17	06.51	07.19	07.48	08.01	08.26		
Waktu Perjalanan (menit)	7	24	18	19	3	15	8		
Kendaraan Tiba	06.07	06.41	07.09	07.38	07.51	08.16	08.34		
<i>Earliest Time</i>	06.00	05.30	05.30	05.30	05.45	06.00	05.30		
Mulai Dilayani	06.07	06.41	07.09	07.38	07.51	08.16	08.34		
Waktu Tunggu	0	0	0	0	0	0	0		
Selesai	06.17	06.51	07.19	07.48	08.01	08.26	08.44		
<i>Latest time</i>	09.00	09.00	09.00	09.30	08.30	08.30	09.00		
Penalti (menit)	0	0	0	0	0	0	0		0
Kapasitas Yang diangkut	1	1	1	1	3	2	2		
Kapasitas Maksimum	14								
Penalti (kapasitas)	0								0
Jarak yg ditempuh (km)	2,9	7,4	5,2	6,5	1,1	7,3	3,7	1	35,1
	Blok 3 13-26-17-25-6-4								
Node	0 ke 13	13 ke	26 ke	17 ke	25 ke 6	6 ke 4	4 ke 0		

		26	17	25					
Waktu Berangkat Kendaraan	06.00	06.14	06.27	06.45	07.05	07.38			
Waktu Perjalanan (menit)	4	3	8	10	23	8			
Kendaraan Tiba	06.04	06.17	06.35	06.55	07.28	07.46			
<i>Earliest Time</i>	05.30	05.30	06.00	05.00	06.00	05.30			
Mulai Dilayani	06.04	06.17	06.35	06.55	07.28	07.46			
Waktu Tunggu	0	0	0	0	0	0			
Selesai	06.14	06.27	06.45	07.05	07.38	07.56			
<i>Latest time</i>	09.30	09.30	09.30	09.30	09.30	09.00			
Penalti (menit)	0	0	0	0	0	0			0
Kapasitas Yang diangkut	2	3	2	2	1	2			
Kapasitas Maksimum	14								
Penalti (kapasitas)	0								0
Jarak yg ditempuh (km)	1,2	1,9	2,3	3,6	7,3	2,7	4,9		23,9
Blok 4 5-19-23-11-15-21									
Node	0 ke 5	5 ke 19	19 ke 23	23 ke 11	11 ke 15	15 ke 21	21 ke 0		
Waktu Berangkat Kendaraan	06.00	06.25	06.56	07.15	07.35	08.03			
Waktu Perjalanan (menit)	15	21	9	10	18	8			
Kendaraan Tiba	06.15	06.46	07.05	07.25	07.53	08.11			

<i>Earliest Time</i>	05.30	06.00	05.30	05.30	06.00	05.45			
Mulai Dilayani	06.15	06.46	07.05	07.25	07.53	08.11			
Waktu Tunggu	0	0	0	0	0	0			
Selesai	06.25	06.56	07.15	07.35	08.03	08.21			
<i>Latest time</i>	09.00	09.00	09.00	08.30	09.30	09.00			
Penalti (menit)	0	0	0	0	0	0			0
Kapasitas Yang diangkut	2	2	1	2	1	2			
Kapasitas Maksimum	14								
Penalti (kapasitas)	0								0
Jarak yg ditempuh (km)	6,4	7,4	3,6	2,5	3,6	2,6	1,4		27,5
Jumlah									103,7
<i>Fitness</i>									$\frac{1}{103,7}$ = 0,00964



## Lampiran E

### *Source Code VRPTW*

```
%Populasi awal random permutasi 1-26
for i=1: UkuranPopulasi
    Populasi(i,:)=randperm(PanjangKromosom);
end
disp('Populasi Awal=');
disp(Populasi);

%Menghitung Fitness
%Hitung Jarak, Pinalti Kapasitas, Pinalti TW,
Fitness tiap kromosom di populasi awal
TotalJarak = HitungJarak (Populasi, Jarak);
PinaltiKapasitas =
HitungPinalti (Populasi, Demand);
PinaltiWaktu = HitungPinaltiTW (Populasi, Waktu);
TotalPinalti= PinaltiKapasitas + PinaltiWaktu;
Z=TotalJarak + TotalPinalti;
F=1./Z;

%Menghitung probabilitas P(i) dan Probabilitas
Kumulatif Q(i)
%kromosom i
P=F/sum(F);
Q=cumsum(P);

% Bagian Iterasi
Iterasi=1;
while Iterasi <= Term

%Memilih Induk
%memilih parent dg stochastic universal sampling
(SUS)
```

## Lampiran E lanjutan

```

Parent=MemilihInduk (Populasi,UkuranPopulasi,Pc,Q
);
    disp('Parent =');
    disp(Parent);

    randc = rand;
    if(randc<Pc) %crossover
        t=floor(UkuranPopulasi*Pc);
        for i=1:t-1;
            j = i+1;
            if (mod(j,2) == 0)

Child(i:j,:)=Crossover (Parent (i,:),Parent (j,:));
                end
            end
        %end

        disp('Child hasil crossover = ');
        disp(Child);

        for i=1:t
            randm=rand;
            disp(randm);

%Mutasi
if (randm<Pm)

MutasiPoint=ceil (rand(1,2) .*PanjangKromosom);
            trans (i)=Child(i,MutasiPoint (2));
            Child(i,MutasiPoint (2))=
Child(i,MutasiPoint (1));
            Child(i,MutasiPoint (1))=trans (i);
        end
    end
end

```

## Lampiran E Lanjutan

```
disp('hasil Mutasi =');
disp(Child);
```

```
Populasi=[Populasi;Child]
disp('Populasi Baru =');
disp(Populasi);
```

```
TotalJarak=HitungJarak(Populasi, Jarak);
PinaltiKapasitas =
HitungPinalti(Populasi, Demand)
PinaltiWaktu =
HitungPinaltiTW(Populasi, Waktu)
TotalPinalti= PinaltiKapasitas +
PinaltiWaktu;
Z=TotalJarak + TotalPinalti;
F=1./Z; %Nilai fitness
```

```
t=length(Populasi);
while t>UkuranPopulasi
    for i=1:t
        if(F(i)==min(F))
            Populasi(i, :)=[];
            TotalJarak(i)=[];
            TotalPinalti(i)=[];
            F(i)=[];
            break;
        end
    end
end
t=t-1;
```

## Lampiran E Lanjutan

```

end

end

P=F/sum(F);
Q=cumsum(P);

%Plot(Iterasi,max_fitness, mean_fitness)
hold on;
plot(Iterasi,max(F),'-rs');
xlabel('Iterasi');
ylabel('Nilai fitness');
title('Grafik Fitness Setiap Generasi');
legend('Best Fitness');
hold off

A(Iterasi) = max(F);
Iterasi=Iterasi+1;

end

disp('Populasi Akhir')
disp (Populasi)
TotalJarak = HitungJarak (Populasi, Jarak)
PinaltiKapasitas =
HitungPinalti (Populasi, Demand) %Pinalti Populasi
Akhir
disp(PinaltiKapasitas);
PinaltiWaktu = HitungPinaltiTW (Populasi, Waktu)
disp(PinaltiWaktu);

TotalPinalti= PinaltiKapasitas + PinaltiWaktu;
Z=TotalJarak + TotalPinalti;
F=1./Z;

```

## Lampiran E Lanjutan

```
disp('F = ');
disp(F);
disp('A = ');
disp(A);
MaksimumF = max(A);
disp('Fitness Maksimum dari seluruh Iterasi');
disp(MaksimumF);
disp('Waktu Eksekusi')
cputime
disp('Algoritme Genetika Berakhir')
```

**Lampiran F*****Source Code Proses Crossover***

```

%fungsi Crossover
function Child=Crossover(Parent1,Parent2)
XoverPoint=[ceil(rand*(length(Parent1)-1))
ceil(rand*(length(Parent1)-1))];
a=min(XoverPoint);
b=max(XoverPoint);

Child=zeros(2,length(Parent1)); %Nilai awal
kromosom child
    for i=a:b
        Child(1,i)=Parent1(i);
        Child(2,i)=Parent2(i);
    end

for i=1:length(Parent1)
    if(i>b)
        Reorder(1,(i-b))=Parent1(i);
        Reorder(2,(i-b))=Parent2(i);
    end
    if (i<=b)
        Reorder(1,(length(Parent1)-
b+i))=Parent1(i);
        Reorder(2,(length(Parent1)-
b+i))=Parent2(i);
    end
end

Inserted1=Reorder(1,:); %inserted1 akan
disisipkan ke child2
Inserted2=Reorder(2,:); %inserted2 akan
disisipkan ke child1

    for m=1:length(Reorder)

```

## Lampiran F Lanjutan

```

    for n=a:b
        if (Reorder(2,m)==Parent1(n))
            i=1;
            while i<=length(Inserted2)
                if
(Inserted2(i)==Reorder(2,m))
                    Inserted2(:,i)=[];
                end
                i=i+1;
            end
        end
        if (Reorder(1,m)==Parent2(n))
            j=1;
            while j<=length(Inserted1)
                if
(Inserted1(j)==Reorder(1,m))
                    Inserted1(:,j)=[];
                end
                j=j+1;
            end
        end
    end
end

for i=b+1:length(Child)
    Child(1,i)=Inserted2(i-b);
    Child(2,i)=Inserted1(i-b);
end
for i=1:(a-1)
    Child(1,i)=Inserted2(length(Child)-b+i);
    Child(2,i)=Inserted1(length(Child)-b+i);

end
end

```

## Lampiran G

### *Source Code Menghitung Jarak*

```

function TotalJarak =
HitungJarak (Populasi, Jarak, B1, B2, B3, B4)
for i=1:length (Populasi) %note:
length (Populasi)=UkuranPopulasi jika ukuran
populasi>=PanjangKromosom
    for k=1:4
        if (k==1)
            Distance (i, k)=Jarak (1, (Populasi (1, 1)+1));
            for j=1:(B1-1)

Distance (i, k)=Distance (i, k)+Jarak ((Populasi (i, j)
+1), (Populasi (i, j+1)+1));
                end
                Distance (i, k)=
Distance (i, k)+Jarak ((Populasi (i, B1)+1), 1);
            elseif (k==2)

Distance (i, k)=Jarak (1, (Populasi (1, B1+1)+1));
                for j=(B1+1):(B1+B2-1)

Distance (i, k)=Distance (i, k)+Jarak ((Populasi (i, j)
+1), (Populasi (i, j+1)+1));
                    end
                    Distance (i, k)=
Distance (i, k)+Jarak ((Populasi (i, B1+B2)+1), 1);
                elseif (k==3)

Distance (i, k)=Jarak (1, (Populasi (1, B1+B2+1)+1));
                    for j=(B1+B2+1):(B1+B2+B3-1)

Distance (i, k)=Distance (i, k)+Jarak ((Populasi (i, j)
+1), (Populasi (i, j+1)+1));
                        end

```



## Lampiran G Lanjutan

```

        Distance(i,k)=
Distance(i,k)+Jarak((Populasi(i,B1+B2+B3)+1),1);
        else

Distance(i,k)=Jarak(1,(Populasi(1,B1+B2+B3+1)+1)
);
        for j=(B1+B2+B3+1):(B1+B2+B3+B4-1)

Distance(i,k)=Distance(i,k)+Jarak((Populasi(i,j)
+1),(Populasi(i,j+1)+1));
        end
        Distance(i,k)=
Distance(i,k)+Jarak((Populasi(i,B1+B2+B3+B4)+1),
1);
        end
        end
        TotalJarak(i)=sum(Distance(i,:));
end
end

```

## Lampiran H

### *Source Code Menghitung Pinalti*

```

function PinaltiKapasitas =
HitungPinalti (Populasi, Demand, B1, B2, B3, B4)
for i=1:length(Populasi) %note:
length(Populasi)=UkuranPopulasi jika ukuran
populasi>=PanjangKromosom
    for k=1:4
        if (k==1)
            load(i, k)=0;
            for d=1:B1

load(i, k)=load(i, k)+Demand(Populasi(i, d)+1);
            end
            elseif (k==2)
                load(i, k)=0;
                for d=(B1+1):(B1+B2)

load(i, k)=load(i, k)+Demand(Populasi(i, d)+1);
            end
            elseif (k==3)
                load(i, k)=0;
                for d=(B1+B2+1):(B1+B2+B3)

load(i, k)=load(i, k)+Demand(Populasi(i, d)+1);
            end
            else
                load(i, k)=0;
                for d=(B1+B2+B3+1):(B1+B2+B3+B4)

load(i, k)=load(i, k)+Demand(Populasi(i, d)+1);
            end
            end
            if (load(i, k)>14)
                Pinalti(i, k)=load(i, k)-14;
            else
                Pinalti(i, k)=0;
            end
        end
    end
end

```

## Lampiran H Lanjutan

```
    end
    PinaltiKapasitas(i)=sum(Pinalti(i,:));
end
end
```

## Lampiran I

### *Source Code Menghitung Pinalti Time Windows*

```

%Fungsi Hitung Pinalti Time Window
function PinaltiWaktu =
HitungPinaltiTW(Populasi,Waktu,B1,B2,B3,B4)
akhir = [ 720 510 540 510 540 540 570 540 540
570 600 510 540 570 540 570 540 570 570 540 540
540 540 540 540 540 570 ];
for i=1:length(Populasi)
    for k=1:4
        if (k==1)
            TW(i,k)=360+Waktu(1,(Populasi(1,1)+1));
            for j=1:(B1-1)

TW(i,k)=TW(i,k)+Waktu((Populasi(i,j)+1),(Populas
i(i,j+1)+1))+10;
                if
(TW(i,k)>akhir(1,(Populasi(i,j+1)+1)))
                    PinaltiTW(i,k)=TW(i,k)-
akhir(1,(Populasi(i,j)+1));
                else
                    PinaltiTW(i,k)=0;
                end
            end
        elseif (k==2)

TW(i,k)=360+Waktu(1,(Populasi(1,B1+1)+1));
            for j=(B1+1):(B1+B2-1)

TW(i,k)=TW(i,k)+Waktu((Populasi(i,j)+1),(Populas
i(i,j+1)+1))+10;
                if
(TW(i,k)>akhir(1,(Populasi(i,j+1)+1)))
                    PinaltiTW(i,k)=TW(i,k)-
akhir(1,(Populasi(i,j)+1));
                else

```

## Lampiran I Lanjutan

```

        PinaltiTW(i,k)=0;
    end
    end
else if (k==3)

TW(i,k)=360+Waktu(1,(Populasi(1,B1+B2+1)+1));
    for j=(B1+B2+1):(B1+B2+B3-1)

TW(i,k)=TW(i,k)+Waktu((Populasi(i,j)+1),(Populas
i(i,j)+1))+10;
        if
            (TW(i,k)>akhir(1,(Populasi(i,j+1)+1)))
                PinaltiTW(i,k)=TW(i,k)-
akhir(1,(Populasi(i,j)+1));
            else
                PinaltiTW(i,k)=0;
            end
        end
    end
else if (k==4)

TW(i,k)=360+Waktu(1,(Populasi(1,B1+B2+B3+1)+1));
    for j=(B1+B2+B3+1):(B1+B2+B3+B4-1)

TW(i,k)=TW(i,k)+Waktu((Populasi(i,j)+1),(Populas
i(i,j)+1))+10;
        if
            (TW(i,k)>akhir(1,(Populasi(i,j+1)+1)))
                PinaltiTW(i,k)=TW(i,k)-
akhir(1,(Populasi(i,j)+1));
            else
                PinaltiTW(i,k)=0;
            end
        end
    end
    end
    end
    end
    PinaltiWaktu(i)=sum(PinaltiTW(1,:));
end
end

```



## BIODATA PENULIS



Penulis memiliki nama lengkap Dinah Razan Anshori dan dilahirkan di Surabaya, 15 Agustus 1997 dari pasangan Anshori dan Siti Mas'udah. Penulis merupakan anak pertama dari tiga bersaudara. Penulis bertempat tinggal di Desa Gununganyar Tambak, Kecamatan Gununganyar, Kota Surabaya. Penulis telah menempuh pendidikan formal mulai dari TK Dahlanuddin Gununganyar

Tambak, SD Dahlanuddin Gununganyar Tambak, SMP Al Islah Gununganyar, SMAN 16 Surabaya. Setelah lulus dari SMA, penulis melanjutkan studinya di S1 Departemen Matematika FMKSD ITS Surabaya tahun 2014. Selama perkuliahan penulis aktif mengikuti kegiatan organisasi di KM ITS, khususnya di Jurusan Matematika ITS dan Fakultas FMIPA. Penulis pernah menjadi Staff Departemen Community Service HIMATIKA ITS 2015/2016, Staff Departemen Sosial Masyarakat BEM FMIPA ITS 2015/2016, Sekretaris Departemen Sosial Masyarakat BEM FMIPA ITS 2016/2017 dan menjadi Sekretaris Departemen Social Development HIMATIKA ITS 2016/2017. Pada tahun 2017 penulis melakukan kerja praktek di Badan Pusat Statistik (BPS) Provinsi Jawa Timur. Segala saran dan kritik yang membangun untuk Tugas Akhir ini serta bagi yang ingin berdiskusi lebih lanjut dengan penulis dapat menghubungi via email dengan alamat [dinahrazan@gmail.com](mailto:dinahrazan@gmail.com).