

**PENCARIAN RUTE OPTIMUM DENGAN *EVOLUTION*
*STRATEGIES***

SKRIPSI

LABORATORIUM KOMPUTASI DAN SISTEM CERDAS

Untuk Memenuhi Sebagian Persyaratan Mencapai Gelar Sarjana Komputer



Disusun Oleh :

DIAH ARUM ENDARWATI

105060807111140

**PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA**

MALANG

2014

LEMBAR PERSETUJUAN

**PENCARIAN RUTE OPTIMUM DENGAN *EVOLUTION*
*STRATEGIES***

SKRIPSI

LABORATORIUM KOMPUTASI DAN SISTEM CERDAS

Untuk Memenuhi Sebagian Persyaratan Mencapai Gelar Sarjana Komputer



Disusun Oleh :

DIAH ARUM ENDARWATI

105060807111140

Skripsi ini telah disetujui oleh dosen pembimbing
pada tanggal 18 Juli 2014

Dosen Pembimbing I,

Dosen Pembimbing II,

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

NIP 19720919 199702 1 001

Dian Eka Ratnawati, S.Si, M.Kom.

NIP. 19730619 200212 2 001

LEMBAR PENGESAHAN

PENCARIAN RUTE OPTIMUM DENGAN *EVOLUTION STRATEGIES*

SKRIPSI

Untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer

Disusun oleh :

DIAH ARUM ENDARWATI

NIM. 1050607111140

Setelah dipertahankan di depan Majelis Penguji pada tanggal 18 Juli 2014 dan dinyatakan memenuhi syarat untuk memperoleh gelar Sarjana dalam bidang Ilmu Komputer

Penguji I

Edy Santoso, S.Si., M.Kom.
NIP. 19740414200312 1 004

Penguji II

Ir. Heru Nurwarsito, M.Kom.
NIP. 19650402 199002 1 001

Penguji III

Ahmad Afif Supianto, S.Si., M.Kom.
NIK.820623 16 1 1 0425

Mengetahui,
Ketua Program Studi Informatika / Ilmu Komputer

Drs. Marji, M.T.
NIP. 19670801 199203 1 001

LEMBAR PERNYATAAN ORISINALITAS SKRIPSI

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

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

Malang, 18 Juli 2014

Mahasiswa,

Diah Arum Enderwati
NIM. 105060807111140

KATA PENGANTAR

Bismillahirrohmanirrohim

Alhamdulillah, segala puji bagi Allah SWT, karena atas rahmat dan hidayahNya-lah penulis dapat menyelesaikan Skripsi yang berjudul “**Pencarian Rute Optimum dengan *Evolution Strategies***”. Sholawat dan salam semoga tetap tercurahkan atas junjungan besar kita Nabi Muhammad SAW. beserta keluarga dan para sahabat sekalian. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika Program Teknologi Informatika dan Ilmu Komputer Universitas Brawijaya Malang.

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

1. Wayan Firdaus Mahmudy, S.Si. MT., Ph.D., selaku dosen pembimbing utama atas kesediaan menjadi pembimbing, ilmu yang berharga, segala masukan, kesabaran dalam membimbing, nasehat dan motivasi yang telah diberikan.
2. Dian Eka Ratnawati, S.Si., M.Kom., selaku dosen pembimbing atas kesediaan menjadi pembimbing, ilmu yang berharga, segala masukan, kesabaran dalam membimbing, nasehat dan motivasi yang telah diberikan.
3. Eriq M. Adams J, ST, M.Kom., selaku dosen penasehat akademik yang telah memberikan pengarahan selama penulis menempuh pendidikan di Program Teknologi Informasi dan Ilmu Komputer Universitas Brawijaya.
4. Kedua Orang Tua penulis (H. Diansyah dan Hj. Sri Wahyuni) dan seluruh keluarga atas segala nasihat, perhatian, dan kesabarannya di dalam membesarkan dan mendidik penulis, serta yang senantiasa tiada henti hentinya memberikan doa demi terselesainya skripsi ini.

5. Drs. Marji, M.T., selaku Ketua Program Studi Informatika Universitas Brawijaya.
6. Issa Arwani, S.Kom.,M.Sc., selaku Sekretaris Program Studi Informatika Universitas Brawijaya.
7. Seluruh Dosen Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.
8. Eka Wahyu Diana dan Tri Wahyu Diningrum yang selalu memberi dukungan dan doa selama pengerjaan skripsi ini.
9. Yurika Caesarita, Shindy Ardita, Farhan, Patria Ludi, Gopi Yudoyo, Dzulkarnaen, Jihad Dzikri, Fawaz, Vicky, Sari, Dika Rizky, Rohman, Oky Prima, Nicky, Vika, Arik, Ana, dan Hanani terima kasih atas segala diskusi, motivasi, semangat, segala bantuan dan doa yang selalu mengiringi penulis.
10. Teman-teman angkatan 2010 terima kasih atas segala bantuan dan kebersamaannya selama menempuh studi di Teknik Informatika Universitas Brawijaya.
11. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya skripsi ini.

Hanya doa yang bisa penulis berikan semoga Allah SWT memberikan pahala serta balasan kebaikan yang berlipat. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, Juli 2014

Penulis

ABSTRAK

Diah Arum Endarwati. 2014. Pencarian Rute Optimum dengan *Evolution Strategies*. Program Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya. Malang. Dosen Pembimbing: Wayan Firdaus Mahmudy, S.Si. MT., Ph.D. dan Dian Eka Ratnawati, S.Si., M.Kom.

Delivery order merupakan salah satu layanan jasa pesan antar makanan siap saji yang cukup diminati masyarakat. Adanya layanan *delivery order* memberi kemudahan bagi konsumen dalam mendapatkan makanan. Salah satu perusahaan yang menggunakan layanan *delivery order* adalah Pizza Hut Delivery (PHD) Rungkut Yakaya Surabaya. Dalam melakukan pengantaran makanan, kurir perusahaan harus mengetahui rute yang harus dilewati. Adapun proses pencarian rute di PHD masih manual, sehingga tidak diketahui apakah rute yang dilewati merupakan rute yang optimal atau tidak. Untuk menyelesaikan masalah ini maka dibutuhkan suatu aplikasi yang dapat membantu kurir untuk mengetahui rute yang optimal dalam proses pengantaran makanan.

Penelitian ini menggunakan *Evolution Strategies* dalam pemilihan rute menuju lokasi konsumen. *Evolution Strategies* dapat digunakan untuk memecahkan masalah pencarian rute optimum. Berdasarkan hasil pengujian yang dilakukan *Evolution Strategies* memperoleh hasil optimal pada jumlah generasi 50 dan 100 serta semakin besar jumlah populasi (*miu*) maka nilai *fitness* yang dihasilkan juga semakin besar.

Kata kunci: *Evolution Strategies*, Rute Optimum, *Delivery Order*

ABSTRACT

Diah Arum Enderwati. 2014. *Optimum Route Search with Evolution Strategies. Information Technology and Computer Science Program, Brawijaya University, Malang. Advisors: Wayan Firdaus Mahmudy, S.Si. MT., Ph.D. and Dian Eka Ratnawati, S.Si., M.Kom.*

Delivery order is one of the messaging services fast food who become the people interest. The existence of the service delivery order to provide convenience for consumers to acquire the food. One company that uses the service delivery order was Pizza Hut Delivery (PHD) Rungkut Yakaya Surabaya. Its conduct a food delivery, Pizza Hut Delivery (PHD) need to know the route should be skipped. The PHD route search process is still manual, so that it's not known whether a route that bypassed the optimal route or not. To complete this problem we need a application that can help the courier to find the optimal route in the delivery of food.

The study used the Evolution Strategies in the selection of the route to the location of the consumer. Evolution Strategies can be used to solve the problem of optimum route search. Based on the results of tests performed Evolution Strategies obtain optimal results on the number of generations 50 and 100 as well as greater number of the population (miu) so the result of the fitness value is also greater.

Keywords : *Evolution Strategies, Route Optimum, Delivery Order*

DAFTAR ISI

| | |
|---|-------------|
| LEMBAR PERSETUJUAN | ii |
| LEMBAR PENGESAHAN | iii |
| LEMBAR PERNYATAAN | iv |
| KATA PENGANTAR..... | v |
| ABSTRAK | vii |
| ABSTRACT | viii |
| DAFTAR ISI..... | ix |
| DAFTAR TABEL | xii |
| DAFTAR GAMBAR..... | xiii |
| DAFTAR SOURCE-CODE | xv |
| BAB I PENDAHULUAN | |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Batasan Masalah..... | 3 |
| 1.4 Tujuan..... | 4 |
| 1.5 Manfaat..... | 4 |
| 1.6 Sistematika Penulisan..... | 4 |
| BAB II TINJAUAN PUSTAKA | |
| 2.1 Kajian Pustaka | 6 |
| 2.2 <i>Evolution Strategies</i> | 7 |
| 2.3 Proses <i>Evolution Strategies</i> | 8 |
| 2.4 Istilah Pada <i>Evolution Strategies</i> | 9 |
| 2.5 Struktur Umum <i>Evolution Strategies</i> | 11 |
| 2.5.1 Representasi Kromosom | 11 |
| 2.5.2 Mutasi..... | 13 |
| 2.5.3 Evaluasi (Fungsi <i>Fitness</i>)..... | 14 |
| 2.5.4 Seleksi | 15 |

BAB III METODOLOGI DAN PERANCANGAN

3.1 Analisis Perangkat Lunak 18

 3.1.1 Deskripsi Masalah 18

 3.1.2 Deskripsi Umum 18

 3.1.3 Deskripsi Data 19

3.2 Perancangan Perangkat Lunak 19

 3.2.1 Proses Pencarian Rute Optimum dengan ES 20

 3.2.1.1 Inisialisasi 21

 3.2.1.2 Pembentukan populasi awal 22

 3.2.1.3 Mutasi 23

 3.2.1.4 Hitung *Fitness* 24

 3.2.1.5 Seleksi 24

 3.2.1.6 Populasi Baru 25

 3.2.2 Perancangan *Interface* 25

 3.2.2.1 Perancangan *Interface* Halaman Utama 26

 3.2.2.2 Perancangan *Interface* Proses Perhitungan 26

 3.2.2.3 Perancangan *Interface List* Data Wilayah 27

 3.2.2.4 Perancangan *Interface* Peta Wilayah 28

3.3 Perancangan Uji Coba 28

 3.3.1 Rancangan Uji Coba I 29

 3.3.2 Rancangan Uji Coba II 29

 3.3.3 Rancangan Uji Coba III 30

3.4 Perhitungan Manual 30

BAB IV IMPLEMENTASI

4.1 Lingkungan Implementasi 38

 4.1.1 Lingkungan Perangkat Keras 38

 4.1.2 Lingkungan Perangkat Lunak 38

4.2 Implementasi Kode Program 39

 4.2.1 Pembangkitan Populasi Awal 39

| | |
|--|----|
| 4.2.2 Hitung <i>Fitness</i> | 40 |
| 4.2.3 Proses Mutasi | 40 |
| 4.2.4 Proses Seleksi..... | 41 |
| 4.2.5 Menampilkan Peta Rute | 42 |
| 4.3 Implementasi Antarmuka | 45 |
| 4.3.1 Form Input Parameter <i>Evolution Strategies</i> | 46 |
| 4.3.2 Form Proses Perhitungan | 46 |
| 4.3.3 Form Peta | 50 |
| 4.3.4 Proses List Data | 51 |
| BAB V PENGUJIAN | |
| 5.1 Sistematika Pengujian | 52 |
| 5.1.1 Sistematika Uji Coba I | 52 |
| 5.1.2 Sistematika Uji Coba II..... | 53 |
| 5.1.2 Sistematika Uji Coba III..... | 53 |
| 5.2 Implementasi Pengujian..... | 53 |
| 5.2.1 Hasil dan Analisa Uji Coba I | 53 |
| 5.2.2 Hasil dan Analisa Uji Coba II..... | 54 |
| 5.2.3 Hasil dan Analisa Uji Coba III..... | 57 |
| BAB VI PENUTUP | |
| 6.1 Kesimpulan | 59 |
| 6.2 Saran..... | 60 |
| DAFTAR PUSTAKA | 61 |
| LAMPIRAN I | 62 |
| LAMPIRAN II | 65 |
| LAMPIRAN III | 68 |

DAFTAR TABEL

| Tabel | Halaman |
|--|---------|
| 2.1 Definisi Istilah <i>Evolution Strategies</i> | 9 |
| 3.1 Rancangan Uji Coba I | 29 |
| 3.2 Rancangan Uji Coba II..... | 30 |
| 3.3 Data Lokasi Jalan | 30 |
| 3.4 Jarak antar Jalan yang Terhubung..... | 31 |
| 3.5 Populasi Awal | 32 |
| 3.6 Node yang Terhubung..... | 33 |
| 3.7 Jarak Tempuh Node yang Terhubung..... | 33 |
| 3.8 Nilai <i>Fitness</i> Setiap Individu..... | 33 |
| 3.9 Proses Penukaran Gen Pada Kromosom | 34 |
| 3.10 Jarak Tempuh Node yang Terhubung Hasil Mutasi | 34 |
| 3.11 <i>Offspring</i> Hasil Mutasi..... | 35 |
| 3.12 Kumpulan Individu Populasi Awal dan Individu <i>Offspring</i> | 35 |
| 3.13 Hasil Sorting Kumpulan Individu..... | 36 |
| 3.14 Populasi Baru dan Individu Terbaik | 36 |
| 5.1 Hasil Uji Coba I | 54 |
| 5.2 Hasil Uji Coba II..... | 54 |
| 5.3 Hasil Uji Coba II..... | 57 |

DAFTAR GAMBAR

| Gambar | Halaman |
|--|---------|
| 2.1 Posisi ES di Optimasi Lain | 7 |
| 2.2 Pseudocode Prosedur ES | 9 |
| 2.3 Ilustrasi Istilah pada ES | 11 |
| 2.4 Contoh Representasi <i>Real Value/Floating</i> | 12 |
| 2.5 Contoh Representasi Permutasi..... | 12 |
| 2.6 Contoh <i>Insertion Mutation</i> | 14 |
| 2.7 Contoh <i>Reciprocal Exchange Mutation</i> | 14 |
| 2.8 Pseudocode <i>Elitism Selection</i> | 16 |
| 3.1 Diagram Alir Metodologi Penelitian..... | 17 |
| 3.2 Diagram Alir Sistem Perangkat Lunak | 19 |
| 3.3 Diagram Alir Penyelesaian Pencarian Rute Optimum dengan ES..... | 20 |
| 3.4 Diagram Alir Pembangkitan Populasi Awal..... | 22 |
| 3.5 Diagram Alir Mutasi | 23 |
| 3.6 Diagram Alir Seleksi <i>Elitism</i> | 24 |
| 3.7 Perancangan <i>Interface</i> Halaman Utama..... | 26 |
| 3.8 Perancangan <i>Imterface</i> Proses Perhitungan | 27 |
| 3.9 Perancangan <i>Iterface List</i> Data Wilayah..... | 27 |
| 3.10 Perancangan <i>Iterface</i> Peta Wilayah | 28 |
| 4.1 Form Input Parameter <i>Evolution Strategies</i> | 46 |
| 4.2 Hasil Input Parameter <i>Evolution Strategies</i> | 47 |
| 4.3 Populasi Awal | 47 |
| 4.4 <i>Offspring</i> Hasil Mutasi..... | 48 |
| 4.5 Hasil <i>Sorting</i> Kromosom Awal dan Kromosom Hasil Mutasi | 48 |
| 4.6 Populasi Baru Hasil <i>Elitism Selection</i> | 49 |

4.7 Hasil Solusi Setiap Generasi 49

4.8 Rute Optimal 50

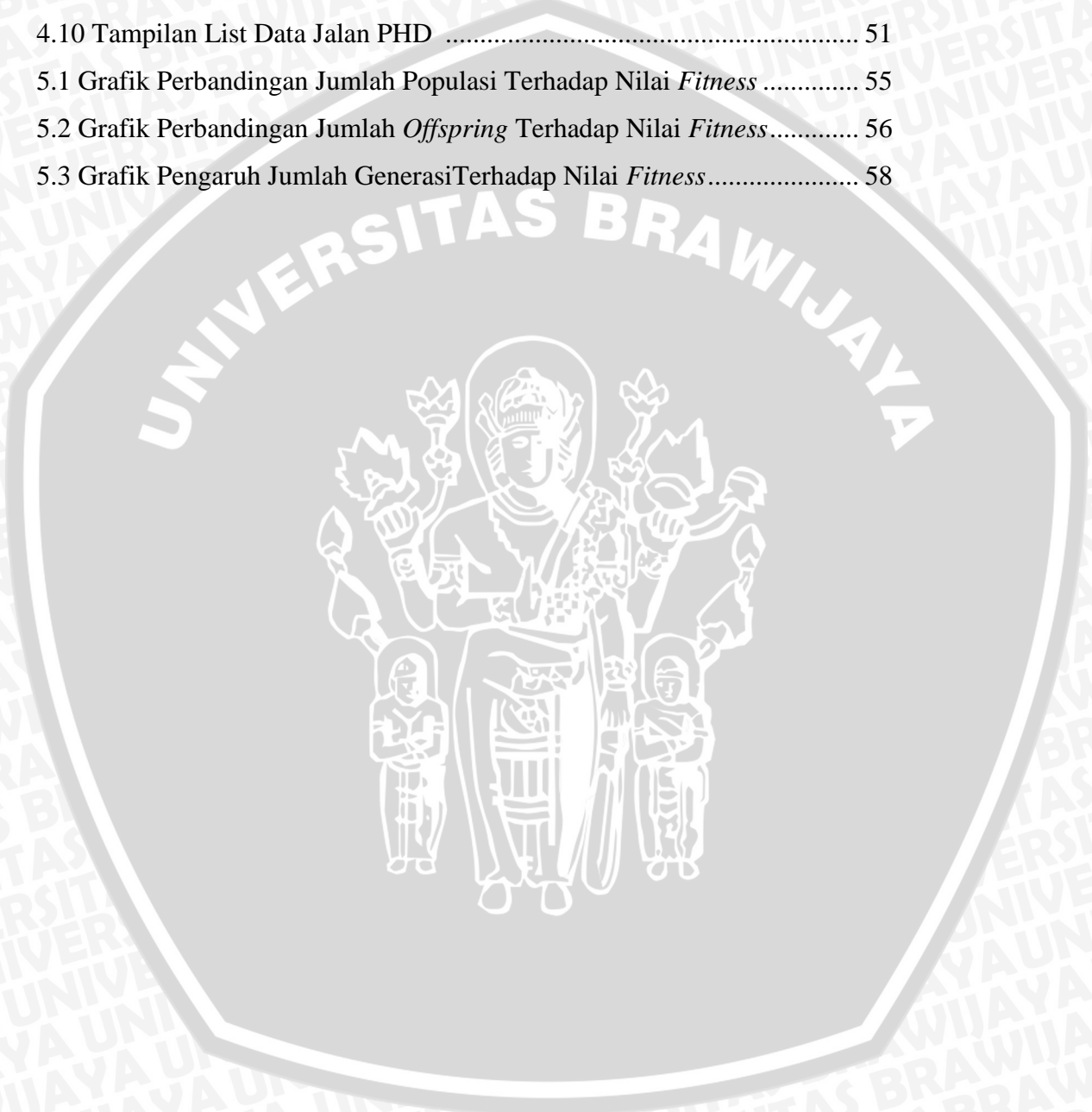
4.9 Peta Rute 50

4.10 Tampilan List Data Jalan PHD 51

5.1 Grafik Perbandingan Jumlah Populasi Terhadap Nilai *Fitness* 55

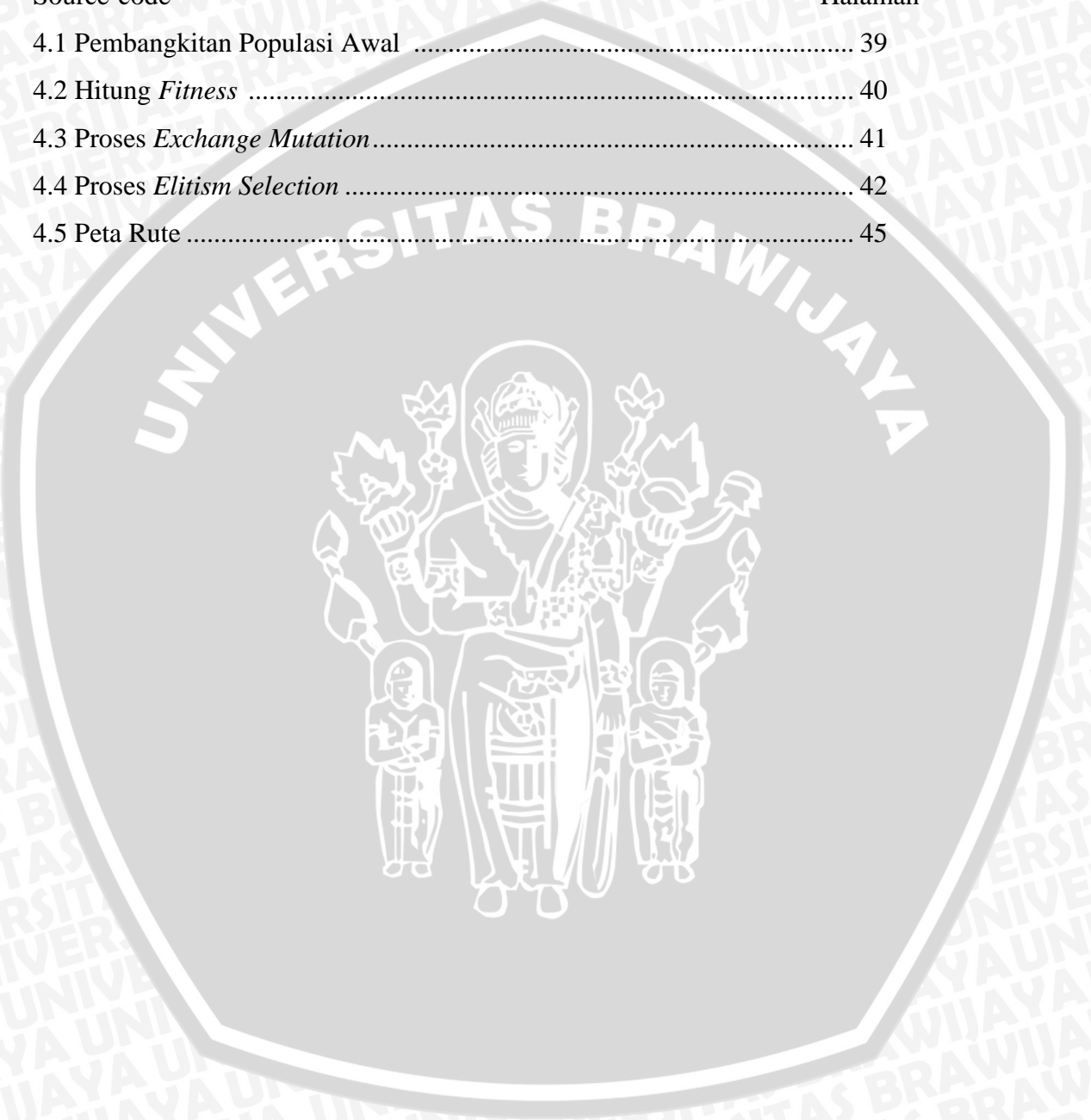
5.2 Grafik Perbandingan Jumlah *Offspring* Terhadap Nilai *Fitness* 56

5.3 Grafik Pengaruh Jumlah Generasi Terhadap Nilai *Fitness* 58



DAFTAR SOURCE-CODE

| Source-code | Halaman |
|---|---------|
| 4.1 Pembangkitan Populasi Awal | 39 |
| 4.2 Hitung <i>Fitness</i> | 40 |
| 4.3 Proses <i>Exchange Mutation</i> | 41 |
| 4.4 Proses <i>Elitism Selection</i> | 42 |
| 4.5 Peta Rute | 45 |



BAB I PENDAHULUAN

1.1 Latar Belakang

Delivery order merupakan salah satu layanan jasa pesan antar makanan siap saji yang cukup diminati masyarakat. Kebanyakan masyarakat cenderung memiliki kesibukan sehingga tidak sempat untuk mengunjungi restoran/rumah makan secara langsung. Adanya layanan *delivery order* memberi kemudahan bagi konsumen dalam mendapatkan makanan.

Pizza Hut Delivery (PHD) merupakan salah satu restoran yang menggunakan konsep pelayanan dengan pesan antar (*delivery order*) dan pesan bawa (*take away*). Dua jenis pelayanan yang diberikan oleh PHD ini, membuat outlet PHD hanya memerlukan tempat untuk membuat produk (*kitchen*) dan *counter* hanya untuk melayani pesan bawa [PHD]. PHD mempunyai beberapa cabang, salah satunya adalah PHD cabang Rungkut Surabaya. PHD Rungkut memiliki wilayah pengantaran yang luas dengan banyaknya rute jalan. Pencarian rute pengantaran masih dikerjakan secara manual. Pada pengantaran jarak pendek bukan menjadi masalah, tetapi untuk pengantaran jarak tempuh yang cukup jauh pencarian rute secara manual menjadi tidak optimum.

Tercapainya kepuasan konsumen akan dapat membantu perusahaan dalam meningkatkan penjualan. Salah satu strategi meningkatkan kepuasan konsumen adalah memaksimalkan pelayanan dengan pengantaran pesanan dalam waktu yang singkat. Untuk mengatasi hal tersebut maka ditekankan pada penentuan rute atau jalur yang akan dilewati secara optimal untuk menuju tempat konsumen. Perusahaan *delivery order* membutuhkan suatu aplikasi yang diharapkan dapat memberikan informasi rute jalan sehingga membantu kurir dalam proses pengantaran makanan ke lokasi tujuan.

Metode yang dapat digunakan untuk menyelesaikan permasalahan optimasi pencarian rute adalah *Evolutionary Algorithms* (EA). EA merupakan teknik optimasi

yang meniru proses evolusi biologi. Berbagai jenis EA yang digunakan dalam permasalahan optimasi diantaranya adalah *Genetic Algorithm* (GA), *Evolution Strategies* (ES), dan lain-lain. [MAH-13].

Anies Hannawati, Thiang, dan Eleazar dari Fakultas Teknologi Industri Universitas Kristen Petra pernah melakukan penelitian tentang pencarian rute optimum. Peneliti menggunakan GA dalam menyelesaikan masalah optimasi yang kompleks yaitu pencarian rute optimum dengan memperhatikan kondisi jalan. Penelitian ini menggunakan representasi kromosom string bit, mutasi bit, seleksi *roulette wheel* dan *elitism*, serta *crossover one-cut-point* dan *two-cut-point*. Dari hasil penelitian GA dapat berjalan dengan baik dalam menyelesaikan masalah dan efektif dalam mencari rute terpendek dengan waktu tersingkat berdasarkan kondisi rute [HAN-02].

Metode lain yang digunakan dalam optimasi yaitu *Evolution Strategies* (ES) yang juga merupakan salah satu jenis *Evolutionary Algorithms*. ES diperkenalkan tahun 1960 dan dikembangkan 1970 oleh Rechenberg, Hans Paul Schwefel, dan rekannya di *Technical University of Berlin* (TUB) [BEY-02]. *Evolution Strategies* (ES) adalah salah satu algoritma *meta-heuristic* yang menggunakan vektor bilangan pecahan (*real-vector*) sebagai representasi solusi. Pada perkembangannya, ES diadopsi untuk permasalahan kombinatorial yaitu representasi permutasi. ES hampir serupa dengan *Genetic Algorithm* (GA). Perbedaannya GA menggunakan *crossover* dalam operator reproduksi utama, sedangkan ES menggunakan operator mutasi. Notasi-notasi yang digunakan ES antara lain μ (*miu*) yang menyatakan ukuran populasi dan λ (*lamda*) yang menyatakan banyaknya *offspring* yang dihasilkan saat proses reproduksi berlangsung. ES menggunakan metode *elitism* untuk proses seleksi [MAH-13].

Dalam skripsi ini akan dilakukan penelitian dengan mengambil permasalahan pencarian rute optimum pada kasus *delivery order* dengan perhitungan menggunakan salah satu metode optimasi yaitu *Evolution Strategies* (ES). Penelitian ini diharapkan memperoleh solusi rute yang paling optimal melalui jarak yang paling minimal.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, rumusan masalah yang akan dibahas dalam skripsi ini adalah sebagai berikut :

1. Bagaimana menyelesaikan masalah pencarian rute optimum menggunakan *evolution strategies* dengan sebuah aplikasi perangkat lunak?
2. Bagaimana pengaruh ukuran populasi, ukuran *offspring*, dan banyaknya generasi terhadap nilai *fitness* dalam pencarian rute optimum?

1.3 Batasan Masalah

Untuk menghindari melebarnya pokok bahasan, maka batasan masalahnya adalah sebagai berikut :

1. Studi kasus pada penelitian ini adalah pada perusahaan *delivery order* Pizza Hut Delivery (PHD) Cabang Rungkut Yakaya Surabaya.
2. Pengambilan data dilakukan untuk sampel pengujian.
3. Untuk data jalan diambil jalan raya dan jalan utama.
4. Persoalan transportasi yang dibahas merupakan persoalan transportasi *delivery order* dari satu sumber ke satu tujuan.
5. Perhitungan yang dilakukan adalah untuk mencari rute optimal dalam pengantaran mulai dari outlet PHD menuju lokasi konsumen.
6. Perhitungan dilakukan berdasarkan jarak.
7. Algoritma yang digunakan dalam penelitian ini adalah *evolution strategies*
8. Pembuatan rute pada sistem tidak memperhatikan dan memperhitungkan faktor kemacetan serta faktor-faktor lain yang berada diluar kendali sistem seperti penutupan jalan dan sebagainya.

1.4 Tujuan

Tujuan pada penulisan skripsi ini adalah sebagai berikut :

1. Menyelesaikan masalah pencarian rute optimum menggunakan *Evolution Strategies* (ES) dengan sebuah aplikasi perangkat lunak
3. Mengetahui pengaruh ukuran populasi, ukuran *offspring*, dan banyaknya generasi terhadap nilai *fitness* dalam pencarian rute optimum?

1.5 Manfaat

Manfaat dari penyusunan skripsi ini adalah dijadikan usulan bagi perusahaan penyedia jasa pengiriman makanan (*delivery order*) dalam mempertimbangkan rute jalan yang akan dilalui untuk mengantar makanan kepada konsumen agar lebih optimum.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penyusunan skripsi ini terbagi menjadi enam bab. Adapun penjabaran dari pembahasan masing-masing bab sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini berisi Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, dan Sistematika Penulisan skripsi.

2. BAB II TINJAUAN PUSTAKA

Bab ini mendeskripsikan teori yang menjadi dasar dalam penelitian.

3. BAB III METODOLOGI DAN PERANCANGAN

Bab ini berisi penjelasan mengenai urutan tindakan yang dilakukan dalam penelitian, perhitungan manual, serta perancangan perangkat lunak yang akan dibuat.

4. BAB IV IMPLEMENTASI

Bab ini berisi tentang implementasi program untuk pencarian rute optimum dengan menggunakan bahasa pemrograman PHP.

5. BAB V PENGUJIAN

Bab ini akan dilakukan pengujian dan analisa hasil dari pengujian sistem yang digunakan untuk memaparkan hasil penelitian.

6. BAB VI PENUTUP

Bab ini berisi kesimpulan dari seluruh pembahasan penelitian serta saran pengembangan penelitian lebih lanjut.

UNIVERSITAS BRAWIJAYA



BAB II

TINJAUAN PUSTAKA

2.1 Kajian Pustaka

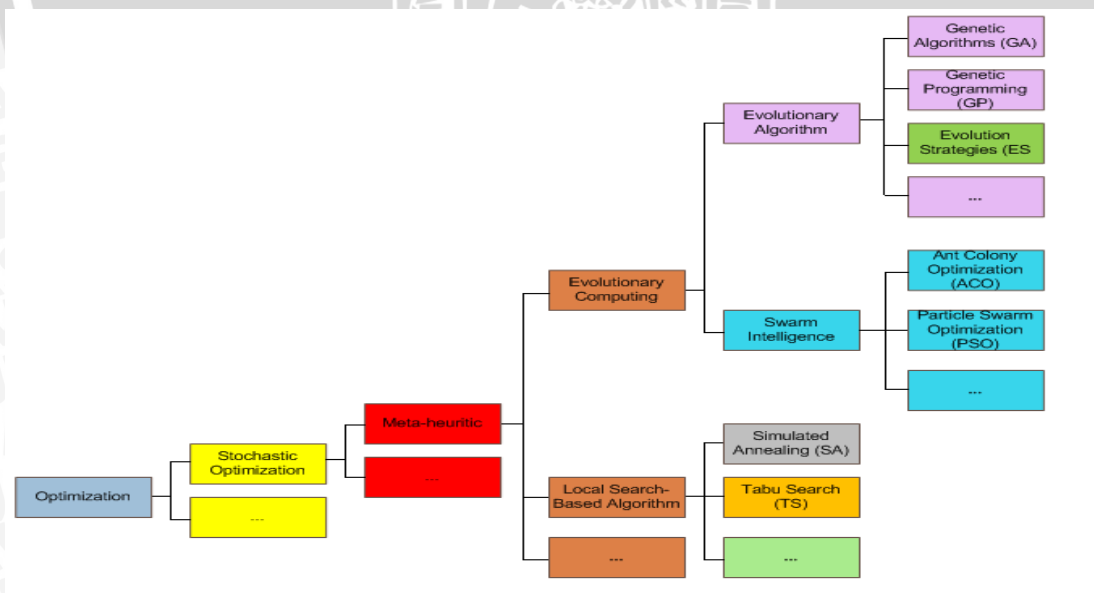
Anies Hannawati, Thiang, Eleazar dari Universitas Kristen Petra pernah melakukan penelitian mengenai pencarian rute optimum menggunakan metode algoritma genetika. Algoritma genetika digunakan untuk mencari rute yang paling optimum dari titik asal ke titik tujuan. Dalam penelitian ini masalah optimasi yang dipilih adalah masalah dalam bidang transportasi, dimana dicari optimasi dalam pencarian rute terpendek, waktu tercepat dan kondisi-kondisi yang timbul didalamnya untuk sebuah jalur perjalanan dari posisi awal menuju posisi tujuan pada suatu peta lokasi jalan/kota. Dari hasil pengujian dapat ditarik beberapa kesimpulan menarik mengenai metode ini. Algoritma genetika cukup efektif dan mudah digunakan khususnya dalam hal mencari rute terpendek dan waktu tersingkat berdasarkan kondisi rute. Algoritma ini menunjukkan keunggulannya pada saat dilakukan perhitungan dengan memakai bobot jarak terhadap waktu. Hal ini akan memakan waktu lebih lama untuk perhitungan matematika biasa. Semakin kompleks bentuk rutenya, maka makin sulit dilakukan perhitungan dengan metode matematika biasa. Secara keseluruhan, algoritma genetika yang telah didisain dapat berjalan dengan baik dan dapat menyelesaikan permasalahan. [HAN-02].

Algoritma genetika (*genetic algorithm*) merupakan salah satu cabang dari *Evolutionary Algorithm* (EA). Cabang EA selain algoritma genetika yaitu *Evolution Strategies* (ES) [MAH-13]. Naufar Rifqi, Warih Maharani, dan Shaufiyah dari Institut Teknologi Telkom Bandung pernah melakukan penelitian dengan menggunakan *Evolution Strategies* pada permasalahan analisis dan implementasi klasifikasi data mining menggunakan jaringan syaraf tiruan (JST). Pada penelitian ini didapatkan kesimpulan bahwa kombinasi nilai parameter pada ES mampu menghasilkan JST yang optimal untuk klasifikasi data Pima Indian Diabetes yaitu ukuran populasi struktur 5, ukuran populasi bobot 50 dan nilai *mutation step size* 0.01

– 0.2 dan untuk data Breast Cancer ukuran struktur 5, ukuran populasi bobot 50 dan nilai mutation step size 0.01 – 0.8 [RIF-11]. Pada hakikatnya AG dan ES memiliki proses perhitungan yang serupa, hanya perbedaannya terletak pada proses reproduksi, dimana ES menggunakan mutasi untuk reproduksi utama sedangkan AG menggunakan *crossover* (kawin silang). Pada penelitian ini penulis akan mencoba menyelesaikan permasalahan pencarian rute optimum dengan menggunakan *Evolution Strategies*.

2.2 Evolution strategies (ES)

Evolution Strategies (ES) pertama kali diperkenalkan tahun 1960-an dan dikembangkan tahun 1970-an oleh Ingo Rechenberg, Hans Paul Schwefel, dan teman-temannya. ES merupakan salah satu cabang dari *Evolutionary Algorithms* (EAs). EA merupakan bentuk generik dari algoritma optimasi *meta-heuristic* berbasis populasi [MAH-13]. ES adalah suatu algoritma pencarian yang berbasis pada mekanisme seleksi alam dan genetika. ES adalah salah satu algoritma yang sangat tepat digunakan dalam menyelesaikan masalah optimasi kompleks, yang sulit dilakukan oleh metode konvensional. Gambar 2.1 menunjukkan posisi ES pada EA.



Gambar 2.1 Posisi ES di Optimasi Lain
Sumber : [MAH-13]

Evolution Strategies (ES) memiliki kesamaan dengan *Genetic Algorithm* (GA). Perbedaan ES dan GA hanya terletak pada operator yang digunakan, jika ES menggunakan mutasi sebagai operator reproduksi utama, berbeda halnya dengan GA yang menggunakan *crossover* sebagai operator reproduksi utamanya dan mutasi hanya sebagai operator penunjang. Representasi yang digunakan oleh ES adalah vektor bilangan pecahan (*real-vector*), tetapi seiring dengan perkembangan ES dapat menggunakan representasi permutasi untuk permasalahan kombinatorial [MAH-13]. Notasi yang digunakan pada ES antara lain yaitu [MAH-13] :

1. μ (*miu*) menyatakan ukuran populasi (sama seperti *popSize* pada GA)
2. λ (*lambda*) menyatakan banyaknya *offspring* yang dihasilkan proses reproduksi (sama seperti *crossover rate* dan *mutation rate* pada GA).

2.3 Siklus *Evolution Strategies* (ES)

Terdapat empat (4) tipe siklus dari ES, yaitu [MAH-13] :

1. (μ, λ)
Proses ES (μ, λ) tidak menggunakan rekombinasi dalam proses reproduksi. Seleksi menggunakan *elitism selection* hanya melibatkan individu dalam *offspring*, individu induk dalam populasi tidak dilibatkan.
2. $(\mu/r, \lambda)$
Proses ES $(\mu/r, \lambda)$ serupa dengan ES (μ, λ) dengan tambahan melibatkan proses rekombinasi. Seleksi menggunakan *elitism selection* hanya melibatkan individu dalam *offspring*, individu induk dalam populasi tidak dilibatkan.
3. $(\mu + \lambda)$
Proses ES $(\mu + \lambda)$ tidak menggunakan rekombinasi dan proses seleksi menggunakan *elitism selection* melibatkan individu *offspring* dan individu induk.
4. $(\mu/r + \lambda)$
Proses ES $(\mu/r + \lambda)$ serupa dengan Proses ES $(\mu + \lambda)$ menggunakan rekombinasi dan proses seleksi menggunakan *elitism selection* melibatkan individu *offspring* dan induk

Pada penelitian ini akan menggunakan ES dengan proses $(\mu + \lambda)$ *pseudocode* dari proses $(\mu + \lambda)$ adalah sebagai berikut :

```

procedure EvolutionStrategies
begin
  t = 0
  inialisasi P(t): generate random  $\mu$ 
  individu
  while (bukan kondisi berhenti) do
    mutasi C(t) untuk
    menghasilkan  $\lambda$  dari P(t)
    seleksi P(t+1) dari P(t) dan
    C(t)
    t = t + 1
  end while
end

```

Gambar 2.2. Pseudocode prosedur ES

Sumber : [MAH-13]

Dari *pseudocode* tersebut terlihat bahwa proses pembentukan *offspring* hanya menggunakan proses mutasi saja. Proses seleksi melibatkan populasi awal P(t) dan juga anak C(t).

2.4 Istilah pada Evolution Strategies

Pada *evolution strategies* ini terdapat beberapa definisi yang harus dipahami sebelumnya, yaitu [SAT-09] :

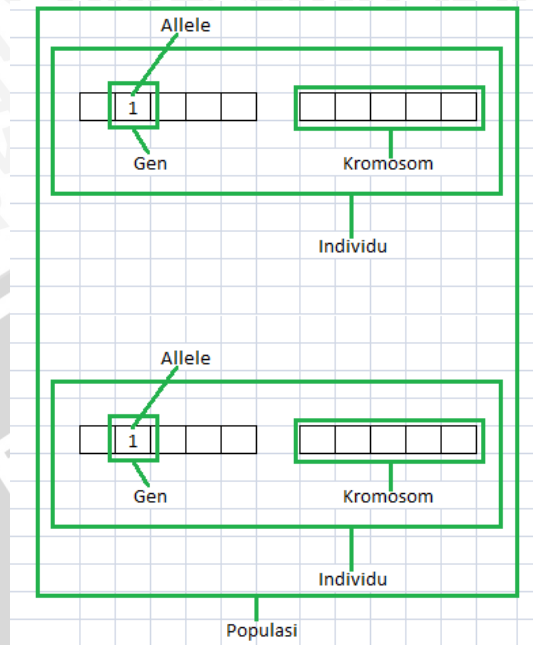
Tabel 2.1 : Definisi istilah yang terdapat pada ES

| Istilah | Definisi |
|---------|--|
| Gen | Nilai yang menyatakan satuan dasar yang membentuk suatu arti |

| | |
|------------------|--|
| | tertentu dalam satu kesatuan gen yang dinamakan kromosom. Gen dapat berupa nilai biner, <i>float</i> , integer/karakter, dan kombinatorial. |
| Allele | Nilai dari gen |
| Kromosom | Gabungan dari gen-gen yang membentuk nilai tertentu. |
| Individu | Suatu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat. Dalam beberapa masalah yang dapat dipecahkan dengan <i>evolution strategies</i> , individu ini dapat juga merupakan kromosom itu sendiri. |
| Populasi | Sekumpulan individu yang akan diproses bersama dalam satu satuan siklus evolusi |
| <i>Fitness</i> | Seberapa baik nilai dari suatu individu yang didapatkan |
| Seleksi | Proses untuk mendapatkan calon induk yang baik |
| Mutasi | Proses pergantian salah satu gen yang terpilih dengan nilai tertentu |
| Generasi | Urutan iterasi dimana beberapa kromosom bergabung |
| <i>Offspring</i> | Kromosom baru yang dihasilkan |

Sumber : [SAT-09]

Pada tabel 2.1 terdapat istilah-istilah yang terdapat pada *evolution strategies* (ES) yaitu antara lain : gen, allele, kromosom, individu, populasi, *fitness*, seleksi, mutasi, generasi, dan *offspring*. Berikut ilustrasi istilah-istilah yang terdapat pada *evolution strategies* ditunjukkan pada Gambar 2.3.



Gambar 2.3 Ilustrasi Istilah pada ES
Sumber : [SAT-09]

2.5 Struktur Umum *Evolution Strategies* (ES)

2.5.1 Pengkodean (*Encoding*)

Langkah pertama yang dilakukan dalam penerapan *evolution strategies* adalah melakukan penginisialisasian atau memrepresentasikan menjadi bentuk kromosom (*chromosome*). Hal ini dilakukan agar dapat diproses melalui *evolution strategies*. Langkah tersebut dinamakan pengkodean (*encoding*). Pada ES representasi menggunakan vektor bilangan pecahan (*real-vector*), tetapi seiring perkembangan saat ini ES dapat menggunakan representasi permutasi [MAH-13]. Menurut [OBI-98] yang dikutip dari [CIP-08], terdapat empat metode pengkodean atau representasi yang biasa digunakan, yaitu representasi biner, representasi permutasi, representasi nilai langsung (*Real-value/Floating*), dan representasi pohon (*tree*). Pada *evolution strategies* representasi yang digunakan hanya dua jenis yaitu representasi nilai langsung (*Real-value/Floating*) dan representasi permutasi.

Penjelasan lebih rinci representasi ES adalah sebagai berikut:

1. Representasi nilai Langsung (*real-value* atau *floating*)

Representasi nilai merupakan suatu metode pengkodean dimana representasi kromosom dikodekan dengan berupa suatu nilai tertentu. Nilai dapat berupa apa saja seperti angka biasa, angka real ataupun bentuk karakter dari suatu objek. Jenis metode ini dapat digunakan secara langsung dalam permasalahan dengan nilai yang rumit. Representasi nilai langsung akan dicontohkan pada Gambar 2.4.

| | |
|--------------|--|
| Chromosome A | [Blue], [Green], [Red],[Blue],[Brown], [Black] |
| Chromosome B | 0.077, 0.567, 0.563, 0.342, 1.432, 1,231 |
| Chromosome C | D,I,A,H,A,R,U,M,E,N,D,A,R,W,A,T,I |

Gambar 2.4 Contoh Representasi *Real-Value/Floating*

Sumber : [CIP-08]

2. Representasi permutasi

Representasi permutasi merupakan sebuah urutan permasalahan yang dikodekan dengan angka – angka. Metode representasi ini biasanya digunakan pada permasalahan kombinatorial, yang tidak dapat diselesaikan dengan pengkodean lain. Misalnya pada masalah penentuan rute dengan penyelesaian *Traveling Salesman Problem* (TSP). Pada permasalahan ini pengkodean digunakan kombinasi tempat dan urutan kunjungan. Contoh representasi permutasi ditunjukkan pada Gambar 2.5.

| | |
|------------|---------|
| Chromosome | 3 2 1 4 |
| Chromosome | 1 4 3 2 |

Gambar 2.5 Contoh Representasi Permutasi

Sumber : [CIP-08]

2.5.2 Mutasi

Mutasi merupakan suatu proses untuk menghasilkan perubahan acak pada suatu kromosom. Proses mutasi berfungsi untuk menghasilkan individu baru yang memiliki sifat baru dengan cara melakukan perubahan pada sebuah gen atau lebih dari suatu individu. Proses mutasi memiliki tujuan agar individu-individu yang ada dalam populasi semakin bervariasi. Dengan begitu dapat mempertahankan keanekaragaman individu dalam suatu populasi [SET-03].

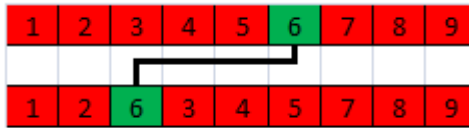
Mutasi merupakan operator utama *evolution strategies* (ES). Individu baru yang dihasilkan pada proses mutasi akan ditempatkan di penampungan *offspring*. Operasi ini dimulai dengan memilih suatu individu dari populasi yang ada, kemudian ditentukan titik secara random dan titik yang dipilih tersebut nilainya akan dirubah secara random juga atau dengan kata lain mengganti satu gen dari kromosom. Pada ES rekombinasi tidak digunakan maka hanya mutasi yang berperan menghasilkan *offspring*.

Pada penelitian ini digunakan representasi permutasi. Pengkodean permutasi melakukan proses mutasi dengan memilih dua posisi (*locus*) dari kromosom dan kemudian nilainya saling dipertukarkan. Beberapa operator mutasi telah diciptakan untuk representasi permutasi, seperti metode *insertio*, dan *reciprocal exchange mutation* [MAH-13].

Contoh mutasi adalah sebagai berikut :

a. Insertion Mutation

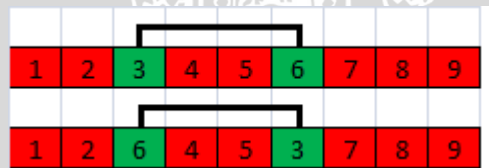
Metode *insertion mutation* bekerja dengan memilih satu posisi (*selected point / SP*) secara random kemudian mengambil dan menyisipkan nilainya pada posisi lain (*insertion point / IP*) secara random seperti yang ditunjukkan pada gambar 2.6.



Gambar 2.6 Contoh Insertion Mutation
Sumber : [MAH-13]

b. Reciprocal Exchange Mutation

Metode *reciprocal exchange mutation* bekerja dengan memilih dua posisi (*exchange point / XP*) secara random kemudian menukarkan nilai pada posisi yang telah dipilih. Contoh metode *reciprocal exchange mutation* ditunjukkan pada gambar 2.7.



Gambar 2.7 Reciprocal Exchange Mutation
Sumber : [MAH-13]

2.5.3 Evaluasi (Fungsi *Fitness*)

Fungsi *fitness* adalah fungsi yang dimiliki oleh masing-masing individu untuk menentukan tingkat kesesuaian individu tersebut dengan kriteria yang ingin dicapai. *Fitness* memiliki ukuran untuk kondisi dari kromosom yang mengekspresikan kemungkinan suatu kromosom akan tetap hidup dalam generasinya. Kromosom dipilih untuk diseleksi dan memperoleh generasi baru. Semakin besar nilai *fitness*-nya akan mempunyai kesempatan untuk menghasilkan generasi selanjutnya [SAR-11].

Dalam proses ini setiap kromosom dalam populasi yang telah dibangkitkan dihitung nilai *fitness*-nya. Perhitungan nilai *fitness* ini tergantung pada permasalahan yang diangkat. Semakin tinggi nilai *fitness*-nya berarti semakin baik juga individu atau solusi yang didapatkan. Nilai *fitness* ini biasanya dijadikan ukuran untuk melihat seberapa baiknya suatu individu atau kromosom yang dibangkitkan atau bisa disebut sebagai acuan untuk mendapatkan individu terbaik. Nilai *fitness* bisa terdiri dari

beberapa constraint atau nilai sesuai dengan parameter syarat yang digunakan dalam permasalahan, sehingga fungsi *fitness*-nya bisa bertambah atau berkurang sesuai dengan yang diinginkan asalkan tetap dalam jalur permasalahan yang diangkat.

Pada proses evaluasi individu dilakukan berdasarkan fungsi *fitness*-nya. Untuk permasalahan optimasi pencarian nilai maksimum nilai *fitness* (maksimasi) dapat digunakan secara langsung, persamaannya yaitu [MAH-13] :

$$fitness = f(x) \quad (2-1)$$

Sedangkan untuk permasalahan meminimalkan fungsi (minimasi) maka *fitness* tidak dapat digunakan secara langsung, yakni [MAH-13]:

$$Fitness = \frac{1}{f(x)} \quad (2-2)$$

2.5.4 Seleksi

Proses selanjutnya adalah seleksi dimana proses ini menentukan individu mana yang akan dimasukkan pada populasi baru dan digunakan untuk pembentukan generasi selanjutnya. Semakin tinggi nilai *fitness* suatu individu, maka semakin besar kemungkinan untuk dipilih [NIS-13]. Pada penelitian ini digunakan seleksi *elitism*. Metode seleksi *elitism* bekerja dengan mengumpulkan seluruh individu dalam populasi (*parent*) dan *offspring* dalam satu penampungan. *popSize* individu terbaik dalam penampungan ini akan lolos untuk masuk dalam generasi selanjutnya. Metode seleksi ini menjamin individu yang terbaik akan selalu lolos. Pseudocode *elitism selection* disajikan pada Gambar 2.8 sebagai berikut:

PROCEDURE ElitismSelection

Input:

POP: himpunan individu pada populasi

pop_size: ukuran populasi

OS: himpunan individu anak (*offspring*) hasil reproduksi menggunakan crossover and mutasi

Output:

POP: himpunan individu pada populasi setelah proses seleksi selesai

```
/* gabungkan individu pada POP dan OS ke dalam TEMP */
```

```
TEMP  $\leftarrow$  Merge (POP, OS)
```

```
/* urutkan individu berdasarkan fitness secara ascending */
```

```
OrderAscending (Temp)
```

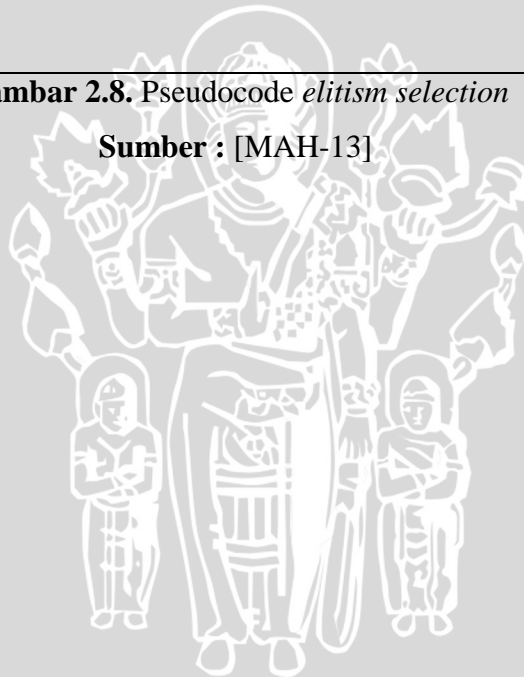
```
/* copy pop_size individu terbaik ke POP */
```

```
POP  $\leftarrow$  CopyBest (Temp, pop_size)
```

END PROCEDURE

Gambar 2.8. Pseudocode elitism selection

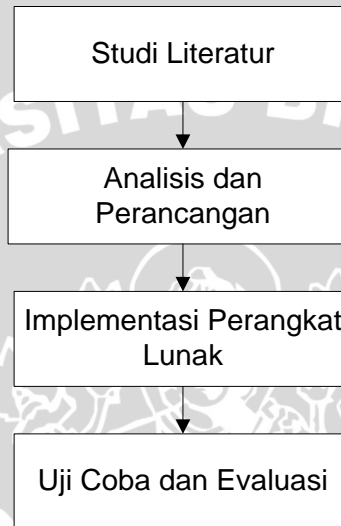
Sumber : [MAH-13]



BAB III

METODOLOGI DAN PERANCANGAN

Pada bab ini akan dibahas mengenai langkah-langkah yang akan digunakan dalam penelitian. Langkah-langkahnya ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian
Sumber : Perancangan

Berdasarkan gambar 3.1, penjelasan langkah-langkah yang dilakukan dalam penelitian skripsi ini adalah :

1. Mengumpulkan berbagai literatur yang terkait dengan penelitian dan mempelajari metode yang digunakan.
2. Melakukan analisa dan perancangan sistem pencarian rute optimum dengan menggunakan *Evolution Strategies*.
3. Membuat sistem perangkat lunak berdasarkan analisa dan perancangan yang telah dilakukan
4. Melakukan uji coba terhadap sistem perangkat lunak dan melakukan evaluasi (analisa hasil) yang diperoleh dari uji coba sistem perangkat lunak.

3.1 Analisis Perangkat Lunak

3.1.1 Deskripsi Masalah

Sistem transportasi pengantaran makanan pada Pizza Hut *Delivery* (PHD) masih menggunakan sistem transportasi manual, yang artinya rute transportasi ditentukan dengan menerka jalur yang akan dilalui untuk mencapai titik-titik tujuan yang telah ditentukan tanpa mengetahui apakah rute tersebut merupakan rute yang optimal. Sehingga waktu tempuh sulit untuk dikontrol. Alur perjalanan yang dilakukan oleh masing-masing kendaraan dalam satu kali perjalanan dimulai berangkat dari outlet dan menuju lokasi konsumen. Setiap kurir memiliki pengetahuan jalan yang berbeda sehingga rute yang dilalui untuk menuju lokasi yang sama kemungkinan berbeda. Dalam penelitian ini, sistem yang akan dibuat adalah pencarian rute optimum berdasarkan jarak untuk meminimalkan waktu tempuh. Sistem ini dibuat untuk memudahkan perusahaan dalam layanan *delivery order*. Diharapkan dengan adanya sistem ini akan sangat membantu perusahaan sehingga segala sesuatu yang akan dilakukan menjadi lebih efektif dan efisien

Algoritma yang akan digunakan pada sistem ini adalah *evolution strategies* (ES). ES salah satu metode *evolutionary algorithm* (EA) yang digunakan untuk menyelesaikan masalah optimasi yang kompleks dan pada penelitian ini akan diterapkan pada masalah pencarian rute optimum pada *delivery order*.

3.1.2 Deskripsi Umum

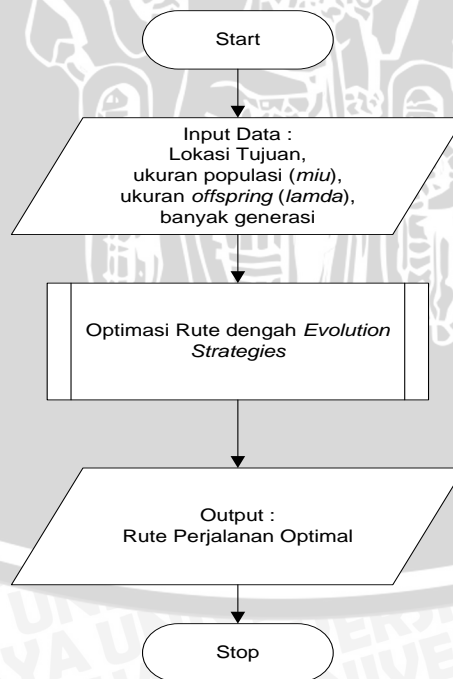
Perangkat lunak yang akan dibuat dalam penelitian ini adalah sebuah aplikasi yang dapat menyelesaikan permasalahan dalam pencarian rute optimum pada *delivery order*. Perangkat lunak ini bertujuan menentukan rute-rute yang akan dilewati untuk mencapai tempat tujuan secara optimal berdasarkan jarak tempuh yang minimal. Untuk mencapai tempat tujuan akan sangat efisien jika mengetahui jalur yang paling optimum sehingga bisa meminimalkan waktu tempuh dari outlet ke lokasi konsumen. Sistem ini akan dibuat dengan menggunakan bahasa pemrograman PHP dan database MySQL.

3.1.3 Deskripsi Data

Data yang digunakan sebagai objek penelitian ini adalah data jalan dan jarak jalan Kota Surabaya khusus untuk wilayah pengantaran makanan di Pizza Hut Delivery (PHD) Rungkut Yakaya Surabaya. Untuk data jalan untuk wilayah pengantaran makanan diperoleh dari PHD Rungkut Yakaya Surabaya dan untuk jarak diambil melalui *googlemaps*. Selanjutnya data yang sudah didapat akan dilakukan pengolahan sehingga data siap diproses untuk perhitungan pencarian rute dengan menggunakan *evolution strategies* (ES).

3.2 Perancangan Perangkat Lunak

Pada subbab ini akan dibahas mengenai proses pada perangkat lunak ini dengan menerapkan *evolution strategies* (ES). Perangkat lunak akan menerima *input*-lokasi jalan yang akan dituju, serta parameter ES yang terdiri dari ukuran populasi (*miu*), ukuran *offspring* (*lamda*), dan banyak generasi. Diagram alir perancangan perangkat lunak dapat dilihat pada Gambar 3.2.

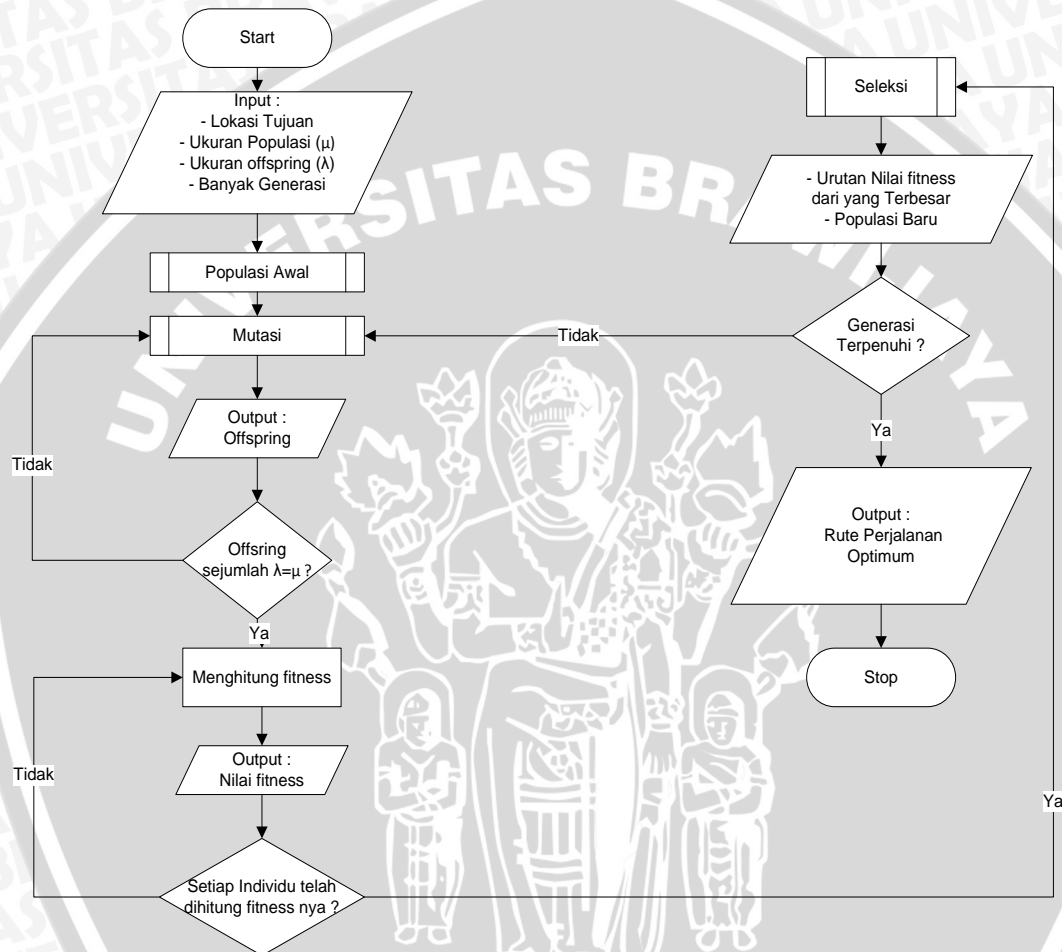


Gambar 3.2 Diagram Alir Sistem Perangkat Lunak

Sumber : Perancangan

3.2.1 Proses Pencarian Rute Optimum dengan *Evolution Strategies*

Diagram alir perangkat lunak dari langkah-langkah pencarian rute optimum dengan *evolution strategies* ditunjukkan pada gambar 3.3



Gambar 3.3 Diagram Alir Pencarian Rute Optimum dengan ES

Sumber : Perancangan

Dari diagram alir pada gambar 3.3 dapat dijelaskan langkah-langkah proses pencarian rute optimum dengan *evolution strategies* adalah sebagai berikut:

1. Inisialisasi nilai awal

Langkah ini merupakan peng-*input*-an nilai awal yang dilakukan oleh *user*.

Inputan yang akan dimasukkan dalam *evolution strategies*, antara lain : lokasi

tujuan, penentuan ukuran populasi (μ/miu), penentuan ukuran *offspring* yang akan digunakan ($\lambda/lamda$), dan banyaknya generasi.

2. Pembentukan populasi awal

Populasi awal dibentuk sebanyak jumlah populasi yang telah ditentukan berfungsi untuk membentuk generasi pertama. Pembentukan populasi awal ini dilakukan secara *random* untuk membentuk kromosom, dimana kromosom merupakan calon solusi rute. Pada kromosom terdiri dari beberapa gen atau *node* yang merupakan titik rute.

3. Proses Mutasi

Melakukan proses mutasi yang mana setiap kromosom akan dicari titik mutasi secara random sehingga memperoleh *offspring* sejumlah yang telah ditentukan pada awal penginputan ($\mu=\lambda$). Pada permasalahan ini digunakan metode mutasi *reciprocal exchange mutation*.

4. Proses hitung *fitness*

Setelah proses mutasi, maka akan dilakukan proses perhitungan *fitness* individu awal dan individu hasil mutasi. Hal ini dilakukan sampai semua individu telah terpenuhi.

5. Proses Seleksi

Melakukan penyeleksian dengan menggunakan metode *elitism selection*, seleksi ini dilakukan dengan melibatkan individu awal dan individu hasil mutasi yang telah ditampung di dalam *offspring*. Mengurutkan semua individu berdasarkan nilai *fitness* dari besar ke kecil. Jika generasi sudah sesuai yang diinput-kan maka proses pencarian rute optimum dengan *evolution strategies* akan berakhir dan mendapatkan output rute perjalanan.

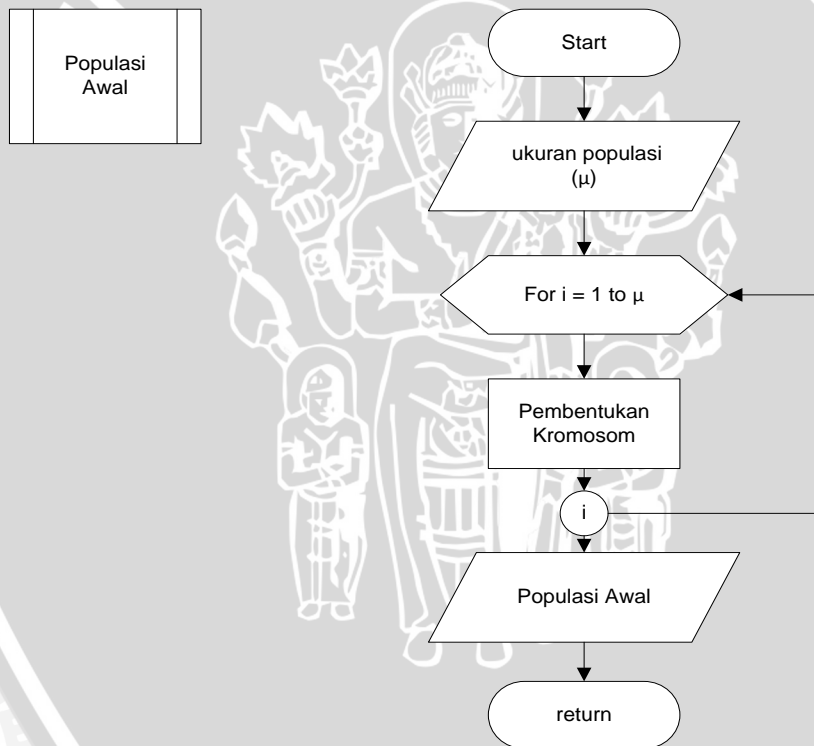
Langkah-langkah *evolution strategies* (ES) lebih rinci, sebagai berikut :

3.2.1.1 Inisialisasi

Inisialisasi adalah membentuk susunan gen yang mewakili satu variabel menjadi sebuah kromosom yang menghasilkan informasi atau solusi dari suatu

masalah. Kromosom dibentuk secara acak yang terdiri atas sejumlah string dan ditempatkan pada sebuah populasi. Pada penelitian ini digunakan representasi permutasi yang akan diimplementasikan untuk *evolution strategies*. Panjang kromosom setiap individu adalah dinamis berdasarkan banyak solusi yang terpilih. Setiap kromosom adalah kumpulan dari *node* yang merupakan calon solusi. Representasi permutasi adalah kumpulan angka yang mewakili posisi dalam sebuah jalur. Pada pencarian rute optimum ini, kromosom mewakili urutan jalan sebagai jalur pengantaran.

3.2.1.2 Pembentukan Populasi Awal



Gambar 3.4 Diagram Alir Pembangkitan Populasi Awal

Sumber : Perancangan

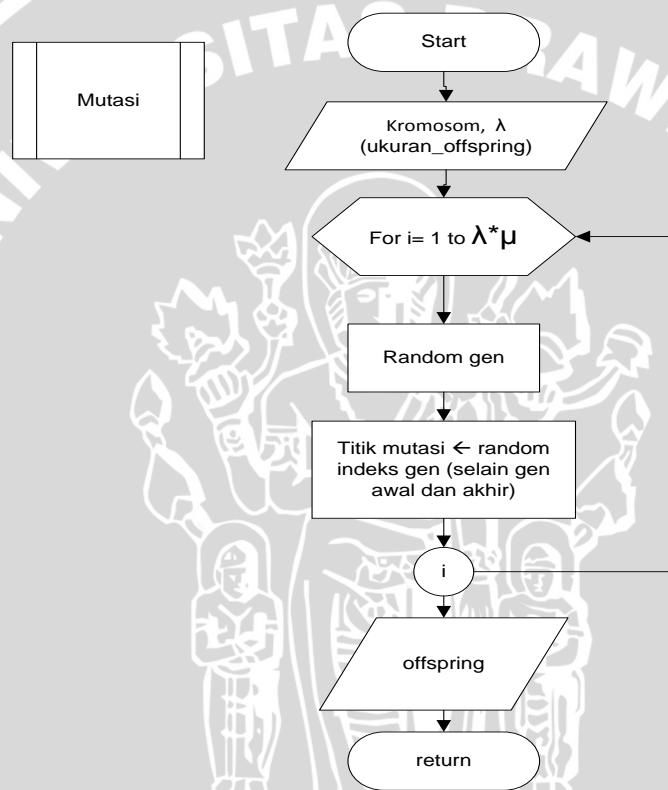
Diagram alir proses pembentukan populasi awal ditunjukkan pada gambar

3.4. Langkah-langkah pembentukan populasi awal adalah sebagai berikut :

1. Menentukan ukuran populasi yang akan dibentuk.

2. Melakukan proses pembentukan kromosom dengan mengecek inputan awal yaitu lokasi tujuan dan ukuran populasi.
3. Jika individu terpenuhi sesuai ukuran populasi yang telah ditentukan maka akan membentuk populasi awal, sedangkan jika tidak maka proses pembentukan kromosom akan terus berlanjut.

3.2.1.3 Mutasi



Gambar 3.5 Diagram Alir Mutasi

Sumber : Perancangan

Setelah proses pembentukan populasi, maka selanjutnya melakukan proses mutasi untuk membentuk *offspring*. Pembentukan *offspring* (λ) dilakukan sejumlah populasi (μ) atau ($\lambda=\mu$). Langkah-langkah Mutasi ditunjukkan pada Gambar 3.5, akan dijelaskan sebagai berikut :

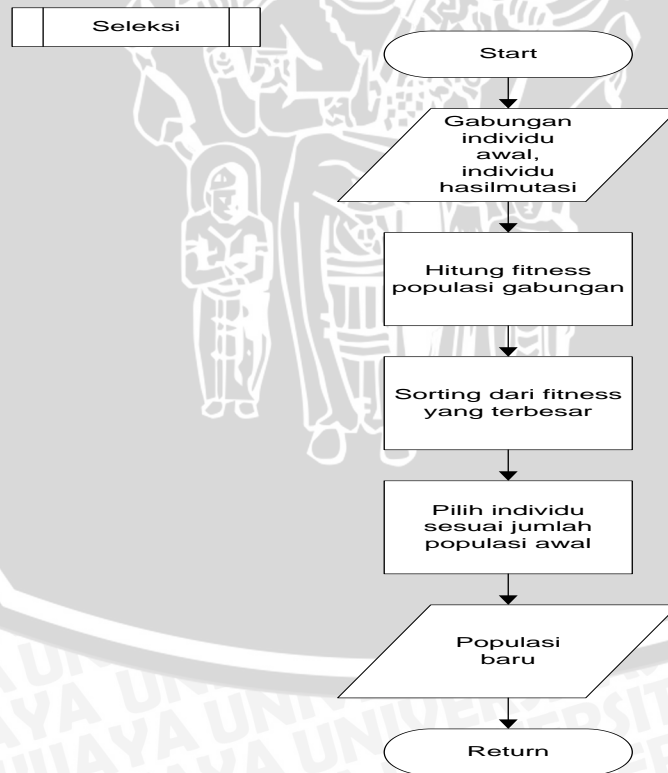
1. Input individu dan ukuran *offspring* (λ)
2. Pilih gen secara random untuk dijadikan titik mutasi selain gen awal dan akhir

3. Melakukan perulangan pembentukan hasil mutasi sampai sebanyak individu.
4. Tukar atau ubah gen-gen titik mutasi dengan gen yang menghubungkan antar gen pada kromosom
5. Setelah proses mutasi dilakukan maka menghasilkan individu-individu baru yang akan dimasukkan ke dalam *offspring*.

3.2.1.4 Hitung *Fitness*

Pada penelitian ini digunakan *fitness* untuk mencari rute optimum yang akan dilewati oleh seorang kurir dalam pengantaran makanan. Proses perhitungan *fitness* dimulai setelah mendapatkan kromosom, dimana kromosom merupakan kemungkinan solusi yang berisi informasi *node* rute. Rute terdiri dari beberapa *node* jalan yang memiliki panjang jalan. Perhitungan nilai $fitness = 1/total_jarak$.

3.2.1.5 Seleksi



Gambar 3.6 Diagram Alir Seleksi *Elitism*

Sumber : Perancangan

Metode seleksi yang digunakan pada penelitian ini adalah *elitism selection*. Langkah-langkah dari metode *elitism selection* yang ditunjukkan pada Gambar 3.6 adalah sebagai berikut :

1. Menggabungkan populasi awal dengan populasi hasil mutasi
2. Menghitung *fitness* tiap individu
3. Mengurutkan nilai *fitness* dari yang terbesar ke *fitness* yang memiliki nilai terkecil
4. Mengambil individu yang memiliki nilai *fitness* terbesar sebanyak ukuran populasi sebelumnya untuk dip roses menuju generasi selanjutnya.

3.2.1.6 Populasi Baru

Populasi baru merupakan hasil dari proses seleksi gabungan antara individu populasi awal dan *offspring* hasil mutasi yang diambil sebanyak jumlah populasi inputan awal (ukuran populasi / *miu*). Populasi baru yang di dapatkan dari beberapa proses tersebut akan digunakan untuk proses pembentukan generasi berikutnya. Populasi baru ini diharapkan mendapatkan solusi lebih baik dari generasi sebelumnya.

3.2.2 Perancangan Interface

3.2.2.1 Perancangan Interface Halaman Utama

Pada halaman utama ini, *user* dapat menginputkan lokasi tujuan yang akan dituju sebanyak dua inputan, dimana inputan ini terdapat beberapa pilihan (*options*). Halaman utama ini juga terdapat beberapa inputan parameter *evolutiom strategies* (ES) berupa ukuran populasi (ukuran populasi / *miu*), ukuran *offspring* (*lamda*), dan banyak generasi. Perancangan *interface* untuk halaman utama ditunjukkan pada Gambar 3.7.

| | |
|---|---|
| ① | ② |
| Home ③ Proses Perhitungan ④ Peta ⑤ List Data ⑥ | <p>Pencarian Rute PHD Rungkut Yakaya Surabaya dengan Evolution Strategies</p> <p>Lokasi Tujuan 1 <input type="text"/></p> <p>Lokasi Tujuan 2 ⑦ <input type="text"/></p> <p>Jumlah Populasi (miu) <input type="text"/></p> <p>Jumlah Offspring (lamda) <input type="text"/></p> <p>Jumlah Generasi <input type="text"/></p> <p> <input type="button" value="Subm ⑧"/> <input type="button" value="Rese ⑨"/> </p> |

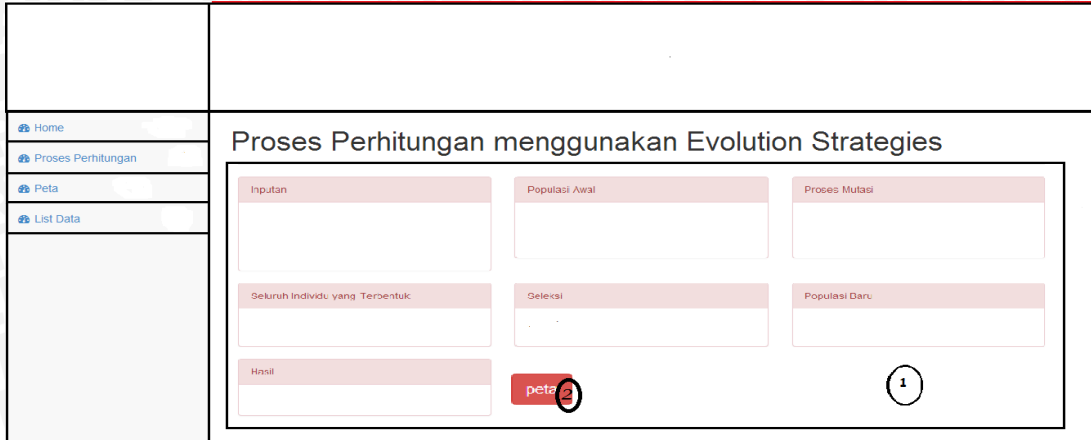
Gambar 3.7 Perancangan *Interface* Halaman Utama
Sumber: Perancangan

Keterangan gambar:

1. Logo yang merupakan identitas dari Pizza Hut Delivery
2. *Header* yang menampilkan gambar untuk menarik perhatian *user*
3. Tab *home* merupakan tab yang menampilkan panel *input*
4. Tab proses perhitungan merupakan tab yang menampilkan proses perhitungan menggunakan *evolution strategies* serta hasil rute yang terbaik
5. Tab peta menampilkan peta wilayah PHD
6. Tab Data menampilkan data jalan, node 1, node 2, dan jarak antar node.
7. Panel *input* parameter untuk perhitungan *evolution strategies*
8. Tombol *Submit* untuk memproses hasil *input*-an
9. Tombol *Reset*

3.2.2.2 Perancangan *Interface* Proses Perhitungan

Setelah melakukan proses peng-*input*-an, *user* akan mendapatkan informasi proses perhitungan dengan *evolution strategies* dan hasil rute yang paling optimum. Perancangan *interface* untuk halaman perhitungan ditunjukkan pada Gambar 3.8.



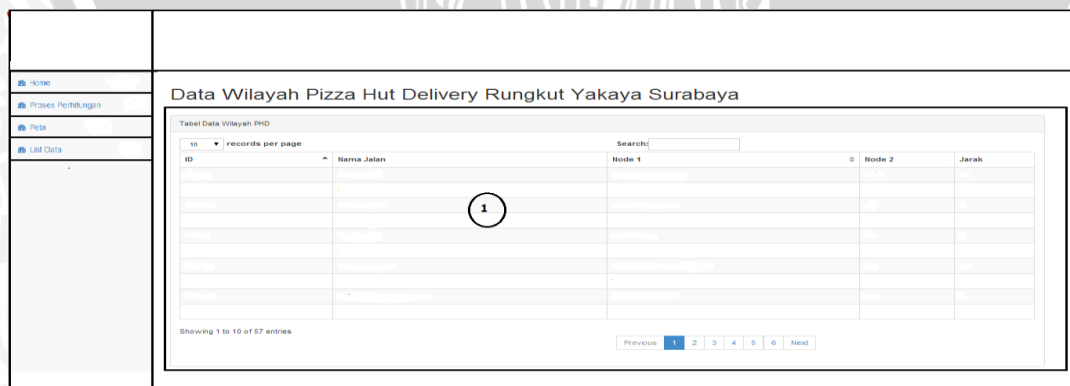
Gambar 3.8 Perancangan *Interface* Proses Perhitungan
Sumber: Perancangan

Keterangan gambar:

1. Menampilkan proses perhitungan pencarian rute optimum yang terdiri dari beberapa panel. Panelnya antara lain *input-an*, populasi awal, mutasi, seluruh individu yang terbentuk, Seleksi, populasi baru, dan hasil.
2. Tombol Peta digunakan untuk menunjukkan arah rute pada peta.

3.2.2.3 Perancangan *Interface* List Data Wilayah

Halaman ini merupakan halaman list data alamat untuk wilayah Pizza Hut Delivery Rungkut Yakaya Surabaya. Perancangan *interface* dari halaman list data alamat ditunjukkan pada gambar 3.9



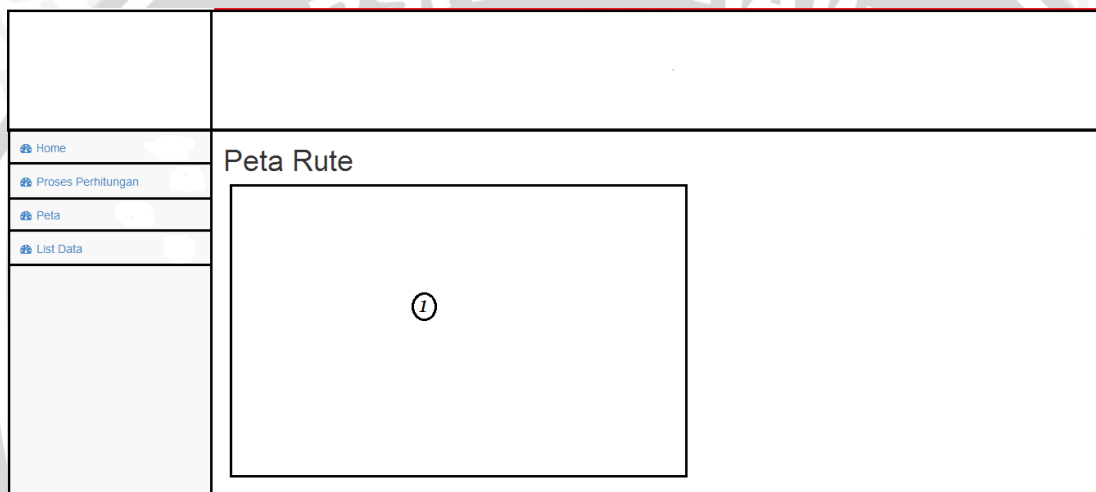
Gambar 3.9 Perancangan *Interface* List Alamat Wilayah PHD
Sumber: Perancangan

Keterangan gambar:

1. Tabel untuk menampilkan list wilayah PHD

3.2.2.4 Perancangan *Interface* Peta Wilayah

Halaman ini merupakan halaman peta untuk wilayah Pizza Hut Delivery Rungkut Yakaya Surabaya. Perancangan *interface* dari halaman peta ditunjukkan pada gambar 3.10



Gambar 3.10 Perancangan *Interface* Peta Wilayah PHD

Sumber: Perancangan

3.3 Perancangan Uji Coba

Proses pengujian pada sistem pencarian rute optimum dengan *evolution strategies* dilakukan dengan mencoba berbagai parameter. Parameternya antara lain ukuran populasi (miu / μ), ukuran *offspring* ($lamda / \lambda$), banyak generasi, dan banyak calon solusi yang dihasilkan. Pengujian dilakukan dengan menggunakan 3 rancangan uji coba, rancangan uji coba berdasarkan calon solusi yang dihasilkan, rancangan uji coba dengan kombinasi $miu (\mu)$ dan $lamda (\lambda)$ serta rancangan uji coba dengan banyak generasi berdasarkan miu dan $lamda$ terbaik. Pengujian ini bertujuan untuk mengetahui pengaruh parameter ES terhadap pencarian rute optimal *delivery order* berdasarkan nilai *fitness*-nya.

3.3.1 Rancangan Uji Coba I

Pada uji coba I adalah membandingkan alternative solusi atau individu-individu yang dihasilkan setelah proses perhitungan menggunakan *evolution strategies* yang merupakan calon solusi rute. Uji coba ini dilakukan untuk mengetahui beberapa solusi yang dihasil sehingga dapat mengambil rute yang paling optimal berdasarkan jarak minimal dan nilai *fitness* terbesar.

3.3.2 Rancangan Uji Coba II

Uji coba yang dilakukan pada uji coba II merupakan kombinasi antara ukuran populasi (μ) dan ukuran *offspring* (λ). Uji coba ini dilakukan untuk mengetahui pengaruh ukuran populasi, ukuran *offspring*, dan jumlah generasi terhadap nilai *fitness*. Ukuran populasi yang akan di uji coba sebesar 5 hingga 30 dan jumlah *offspring* sebesar 1μ hingga 10μ . Skenario uji coba II akan menghasilkan rute yang optimal dilihat dari nilai *fitness* terbesar.. Rancangan tabel hasil pengujian ukuran populasi (μ) dan ukuran *offspring* (λ) ditunjukkan pada Tabel 3.1.

Tabel 3.1 Rancangan Uji Coba II

| Ukuran <i>Offspring</i> (λ) | Ukuran Populasi | | | | | | Rata-Rata |
|--|-----------------|----|----|----|----|----|-----------|
| | (μ) | | | | | | |
| | 5 | 10 | 15 | 20 | 25 | 30 | |
| 1 μ | | | | | | | |
| 2 μ | | | | | | | |
| 3 μ | | | | | | | |
| 4 μ | | | | | | | |
| 5 μ | | | | | | | |
| 6 μ | | | | | | | |
| 7 μ | | | | | | | |
| 8 μ | | | | | | | |
| 9 μ | | | | | | | |
| 10 μ | | | | | | | |
| Rata-Rata | | | | | | | |

3.3.3 Rancangan Uji Coba III

Pengujian ini dilakukan dengan memberikan nilai ukuran populasi dan ukuran *offspring* yang diperoleh dari ukuran populasi dan ukuran *offspring* terbaik dalam uji coba II. Rancangan tabel hasil pengujian pengaruh banyak generasi terhadap rata-rata nilai *fitness* ditunjukkan pada Tabel 3.2.

Tabel 3.2 Rancangan Uji Coba III

| Banyak Generasi | $\mu = \dots, \lambda = \dots$ | | | | | Rata-Rata Nilai <i>Fitness</i> |
|-----------------|--------------------------------|---|---|---|---|--------------------------------|
| | Percobaan Ke - | | | | | |
| | 1 | 2 | 3 | 4 | 5 | |
| 10 | | | | | | |
| 20 | | | | | | |
| 30 | | | | | | |
| 40 | | | | | | |
| 50 | | | | | | |
| 100 | | | | | | |

3.4 Perhitungan Manual

Contoh perhitungan manual menggunakan algoritma *evolution strategies* (ES). Proses perhitungan manual ini menggunakan data sampel. Data sampel ini didapatkan dari sebagian data yang nantinya akan dijadikan sebagai data pada aplikasi yang dibuat. Untuk data lokasi tujuan setiap rute dapat dilihat pada tabel 3.3.

Tabel 3.3 Data Lokasi Tujuan

| NO | NAMA JALAN | NODE 1 | NODE 2 | PANJANG JALAN (KM) |
|----|---------------------|--------|--------|--------------------|
| 1 | RUNGKUT MAPAN UTARA | 26 | 46 | 0.260 |
| 2 | DELTA PERMAI | 1 | 5 | 0.300 |
| 3 | TENGGILIS MEJOYO | 6 | 19 | 0.230 |
| 4 | JL. KEDUNG BARUK | 1 | 2 | 1.000 |
| 5 | JL. RAYA RUNGKUT | 7 | 8 | 2.480 |
| 6 | WONOREJO RUNGKUT | 2 | 3 | 0.438 |

| | | | | |
|----|---------------------|----|----|-------|
| 7 | WONOREJO TAMBAK | 3 | 4 | 0.769 |
| 8 | JL. SOEKARNO HATTA | 2 | 10 | 1.110 |
| 9 | RUNGKUT PUSKESMAS | 8 | 9 | 1.190 |
| 10 | RUNGKUT ALANG-ALANG | 9 | 10 | 0.218 |
| 11 | PANDUGO | 10 | 11 | 3.330 |
| 12 | PANDUGO GG.I | 11 | 13 | 0.363 |
| 13 | MEDOKAN AYU | 28 | 29 | 0.523 |
| 14 | JL. RAYA TENGGILIS | 6 | 20 | 0.350 |

Keterangan :

Node 1 : Titik awal dari jalan

Node 2 : Titik akhir dari jalan

Tabel 3.4 merupakan jarak antar jalan yang saling terhubung, dimana ada beberapa jalan yang tidak terhubung.

Tabel 3.4 : Jarak antar Jalan yang Terhubung

| Jalan 1 | Jalan 2 | Total Jarak |
|---------|---------|-------------|
| 1 | 32 | 0.827 |
| 1 | 33 | 0.860 |
| 1 | 36 | 0.629 |
| 1 | 43 | 2.410 |
| 1 | 61 | 0.530 |
| 1 | 62 | 0.433 |
| 1 | 63 | 0.540 |
| 1 | 64 | 0.833 |
| 1 | 67 | 0.477 |
| 2 | 3 | 0.530 |
| 2 | 4 | 1.300 |
| 2 | 60 | 1.640 |
| 3 | 14 | 0.580 |
| 3 | 55 | 0.581 |
| 3 | 58 | 0.579 |
| 4 | 5 | 3.480 |

Dari data yang telah diperoleh PHD Rungkut Yakaya Surabaya memiliki 69 jalan dan 49 *node* penghubung dengan panjang jalan yang berbeda. *Node* 26 dan *node* 46 menandakan outlet PHD. Sebagian data jalan dapat dilihat pada tabel 3.3. Data selengkapnya dapat dilihat pada Lampiran.

Untuk perhitungan manual ini akan ditetapkan parameter awal adalah sebagai berikut:

| | |
|---------------------------------------|---------------------------|
| Lokasi awal | = 1 (Rungkut Mapan Utara) |
| Lokasi tujuan 1 | = 11 (Pandugo) |
| Lokasi tujuan 2 | = 12 (Pandugo Gg.1) |
| Ukuran populasi (μ) | = 4 |
| Ukuran <i>offspring</i> (λ) | = 2μ |
| Banyak generasi | = 1 |

Langkah 1 : Pembangkitan populasi awal (kromosom)

Pada kasus pencarian rute optimum ini digunakan representasi permutasi. Representasi permutasi digunakan untuk menyatakan sebuah solusi rute. Setiap gen pada kromosom berupa angka yang menyatakan nomor dari setiap *node*. Dimisalkan populasi awal terdiri dari 4 kromosom, dimana gen pertama ditentukan 1 pada semua kromosom dan gen tujuan diinputkan 11 dan 12. Populasi awal yang terbentuk terdapat pada tabel 3.5.

Tabel 3.5: Populasi awal

| No | Kromosom |
|----|----------------------------|
| 1 | [1-61-42-41-40-10-11-12] |
| 2 | [1-61-42-41-40-9-59-11-12] |
| 3 | [1-61-43-45-11-12] |
| 4 | [1-36-37-9-10-11-12] |

Langkah 2 : Hitung *Fitness*

Setelah mendapatkan kromosom maka akan dilakukan proses perhitungan *fitness*.

Untuk masalah pencarian nilai minimum maka digunakan persamaan (2.2) dengan

bobot total jarak. Semakin besar *fitness* maka semakin baik kromosom tersebut untuk dijadikan solusi. Untuk mempermudah melakukan perhitungan jarak tempuh, dibentuk terlebih dahulu *node-node* yang terhubung.

Tabel 3.6: *Node* yang terhubung

| Kromosom | Node yang Terhubung |
|----------------------------|--|
| [1-61-42-41-40-10-11-12] | 1-61; 61-42; 42-41 ; 41-40; 40-10; 10-11; 11-12 |
| [1-61-42-41-40-9-59-11-12] | 1-61; 61-42; 42-41; 41-40 ; 40-9; 9-59; 59-11; 11-12 |
| [1-61-43-45-11-12] | 1-61; 61-43; 43-45; 45-11; 11-12 |
| [1-36-37-9-10-11-12] | 1-36 ; 36-37; 37-9; 9-10; 10-11; 11-12 |

Selanjutnya menghitung jarak tempuh tiap *node* yang terhubung.

Tabel 3.7: Jarak tempuh *node* yang terhubung

| Jarak Node yang Terhubung | Total Jarak |
|--|--|
| $(0.26+0.27)+(0.27+0.428)+(0.428+0.948)+(0.948+0.932)+(0.932+0.218)+(0.218+3.33)+(3.33+0.363)$ | $0.53+0.698+1.376+1.88+1.15+3.548+3.693=12.875$ |
| $(0.26+0.27)+(0.27+0.428)+(0.428+0.948)+(0.948+0.932)+(0.932+1.19)+(1.19+0.609)+(0.609+3.33)+(3.33+0.363)$ | $0.53+0.698+1.376+1.88+2.122+1.799+3.939+3.693=16.037$ |
| $(0.26+0.27)+(0.27+2.15)+(2.15+2.29)+(2.29+3.33)+(3.33+0.363)$ | $0.53+2.42+4.44+5.62+3.693=16.703$ |
| $(0.26+0.359)+(0.369+0.491)+(0.491+1.19)+(1.19+0.218)+(0.218+3.33)+(3.33+0.363)$ | $0.619+0.86+1.681+1.408+3.548+3.693=11.809$ |

Perhitungan nilai *fitness* :

Menghitung nilai *fitness* dari tiap kromosom yang telah dibangkitkan dengan persamaan (2-2)

Tabel 3.8 : Nilai *Fitness* tiap Individu

| Kromosom | $fitness = 1 / total\ jarak$ |
|----------------------------|------------------------------|
| [1-61-42-41-40-10-11-12] | 0.077669902 |
| [1-61-42-41-40-9-59-11-12] | 0.062355802 |
| [1-61-43-45-11-12] | 0.059869484 |
| [1-36-37-9-10-11-12] | 0.084681175 |

Langkah 3 : mutasi

Proses selanjutnya adalah mutasi. Mutasi dilakukan terhadap induk populasi yang telah terbentuk. Karena rekombinasi tidak digunakan maka hanya mutasi yang berperan menghasilkan *offspring*. *Offspring* yang dihasilkan dari proses mutasi sebanyak $\lambda=2\mu=8$. Pembentukan *offspring* pada proses mutasi diawali dengan menentukan posisi gen secara random.

Tabel 3.9: Proses Penukaran Gen pada Kromosom

| Induk | Proses | Offspring |
|---------------------------------|----------------------|---------------------------------|
| P1 = [1-61-42-41-40-10-11-12] | tukar posisi 3 dan 4 | C1 = [1-61-41-42-40-10-11-12] |
| P1 = [1-61-42-41-40-10-11-12] | tukar posisi 4 dan 5 | C2 = [1-61-42-40-41-10-11-12] |
| P2 = [1-61-42-41-40-9-59-11-12] | tukar posisi 2 dan 3 | C3 = [1-42-61-41-40-9-59-11-12] |
| P2 = [1-61-42-41-40-9-59-11-12] | tukar posisi 4 dan 5 | C4 = [1-61-42-40-41-9-59-11-12] |
| P3 = [1-61-43-45-11-12] | tukar posisi 3 dan 4 | C5 = [1-61-45-43-11-12] |
| P3 = [1-61-43-45-11-12] | tukar posisi 2 dan 3 | C6 = [1-43-61-45-11-12] |
| P4 = [1-36-37-9-10-11-12] | tukar posisi 3 dan 4 | C7 = [1-36-9-37-10-11-12] |
| P4 = [1-36-37-9-10-11-12] | tukar posisi 4 dan 5 | C8 = [1-36-37-10-9-11-12] |

Tabel 3.10: Jarak Tempuh Node yang Terhubung Hasil Mutasi

| Node yang Terhubung | |
|--|----------|
| 1-61; 61-41; 41-42 ; 42-40; 40-10; 10-11; 11-12 | 1009.656 |
| 1-61; 61-42; 42-40; 40-41 ; 41-10; 10-11; 11-12; | 2009.897 |
| 1-42; 42-61; 61;41; 41-40; 40-9; 9-59; 59-11; 11-12; | 2010.183 |
| 1-61; 61-42; 42;40; 40-41; 41-9; 9-59; 59-11; 11-12; | 2011.609 |
| 1-61;61-45; 45-43; 43-11; 11-12; | 2007.254 |
| 1-43;43-61; 61-45; 45-11; 11-12; | 1008.640 |
| 1-36 ; 36-9; 9-37; 37-10; 10-11; 11-12 | 2003.618 |
| 1-36 ; 36-37; 37-10; 10-9; 9-11; 11-12 | 2003.580 |

Menghitung nilai *fitness* dari masing-masing *offspring*. Hasil perhitungan nilai *fitness* dapat dilihat pada tabel 3.6 hingga tabel 3.8

Tabel 3.11 : *Offspring* Hasil Mutasi

| Offspring | fitness =1 /total jarak |
|---------------------------------|-------------------------|
| C1 = [1-61-41-42-40-10-11-12] | 0.000989159 |
| C2 = [1-61-42-40-41-10-11-12] | 0.000497426 |
| C3 = [1-42-61-41-40-9-59-11-12] | 0.000496492 |
| C4 = [1-61-42-40-41-9-59-11-12] | 0.000496885 |
| C5 = [1-61-45-43-11-12] | 0.000497844 |
| C6 = [1-43-61-45-11-12] | 0.000986054 |
| C7 = [1-36-9-37-10-11-12] | 0.000497624 |
| C8 = [1-36-37-10-9-11-12] | 0.000498358 |

Langkah 4 : Seleksi

Seleksi dilakukan untuk memilih individu dari himpunan populasi dan *offspring* yang akan digunakan pada generasi selanjutnya. Metode seleksi yang digunakan adalah *elitism selection*. Metode ini memilih individu terbaik dari kumpulan individu populasi dan *offspring*.

Langkah 4.1 : Langkah awal yang dilakukan adalah menggabungkan individu populasi awal dan individu *offspring* terlebih dahulu. Setelah proses mutasi maka didapatkan individu baru (*children*). Berikut tabel kumpulan individu populasi awal dan individu *offspring*.

Tabel 3.12: Kumpulan individu populasi awal dan individu *offspring*

| Kromosom | fitness =1 /total jarak |
|-------------------------------|-------------------------|
| P1 [1-61-42-41-40-10-11-12] | 0.077669902 |
| P2 [1-61-42-41-40-9-59-11-12] | 0.062355802 |
| P3 [1-61-43-45-11-12] | 0.059869484 |
| P4 [1-36-37-9-10-11-12] | 0.084681175 |
| C1 [1-61-41-42-40-10-11-12] | 0.000989159 |
| C2 [1-61-42-40-41-10-11-12] | 0.000497426 |
| C3 [1-42-61-41-40-9-59-11-12] | 0.000496492 |
| C4 [1-61-42-40-41-9-59-11-12] | 0.000496885 |
| C5 [1-61-45-43-11-12] | 0.000497844 |
| C6 [1-43-61-45-11-12] | 0.000986054 |

| | | |
|----|----------------------|-------------|
| C7 | [1-36-9-37-10-11-12] | 0.000497624 |
| C8 | [1-36-37-10-9-11-12] | 0.000498358 |

Langkah 4.12 : Proses perangkaian (*sorting*)

Setelah proses menggabungkan individu populasi awal dan individu *offspring*, maka proses selanjutnya adalah proses *sorting*. *Sorting* dilakukan berdasarkan nilai *fitness* terbesar sampai terkecil.

Tabel 3.13: Hasil *sorting* kumpulan individu populasi awal dan individu *offspring*

| | Kromosom | fitness =1 /total jarak |
|----|----------------------------|-------------------------|
| P4 | [1-36-37-9-10-11-12] | 0.084681175 |
| P1 | [1-61-42-41-40-10-11-12] | 0.077669902 |
| P2 | [1-61-42-41-40-9-59-11-12] | 0.062355802 |
| P3 | [1-61-43-45-11-12] | 0.059869484 |
| C1 | [1-61-41-42-40-10-11-12] | 0.000989159 |
| C6 | [1-43-61-45-11-12] | 0.000986054 |
| C8 | [1-36-37-10-9-11-12] | 0.000498358 |
| C5 | [1-61-45-43-11-12] | 0.000497844 |
| C7 | [1-36-9-37-10-11-12] | 0.000497624 |
| C2 | [1-61-42-40-41-10-11-12] | 0.000497426 |
| C4 | [1-61-42-40-41-9-59-11-12] | 0.000496885 |
| C3 | [1-42-61-41-40-9-59-11-12] | 0.000496492 |

Langkah 4.3 : Setelah mengalami proses ranking (*sorting*), maka didapatkan populasi baru. Kromosom yang memiliki nilai *fitness* terkecil maka akan dibuang dan tidak akan dimasukkan pada populasi baru. Populasi baru diambil sesuai banyaknya populasi yang telah ditentukan sejak awal.

Tabel 3.14 : Populasi Baru dan Individu Terbaik

| | Kromosom | fitness =1 /total jarak |
|----|----------------------------|-------------------------|
| P4 | [1-36-37-9-10-11-12] | 0.084681175 |
| P1 | [1-61-42-41-40-10-11-12] | 0.077669902 |
| P2 | [1-61-42-41-40-9-59-11-12] | 0.062355802 |
| P3 | [1-61-43-45-11-12] | 0.059869484 |

Dari tabel 3.14. diketahui terdapat empat (4) individu yang memiliki nilai *fitness* terbesar dibandingkan dengan individu yang lain. Solusi optimal diperoleh berdasarkan *fitness* terbesar. Maka dari itu rute yang paling optimal yang harus dilewati berdasarkan hasil perhitungan manual adalah [1-36-37-9-10-11-12]

Setelah mendapatkan populasi baru makan langkah 3 sampai langkah 4 akan berlangsung terus-menerus sebanyak jumlah generasi yang telah ditentukan.

UNIVERSITAS BRAWIJAYA



BAB IV

IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai penerapan rancangan sistem yang telah dipaparkan dalam bab sebelumnya. Dalam bab ini juga menjelaskan hasil dari penelitian beserta analisisnya. Secara umum proses yang dijelaskan dalam bab ini adalah implementasi program.

4.1 Lingkungan Implementasi

Proses implementasi merupakan tahap perangkat lunak siap dioperasikan pada keadaan yang sebenarnya, sehingga akan diketahui apakah sistem yang dibuat telah menghasilkan tujuan yang diinginkan. Lingkungan implementasi yang akan dijelaskan pada bab ini adalah lingkungan implementasi perangkat keras dan perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan sistem pencarian rute optimum dengan *evolution strategies* adalah sebagai berikut :

1. Processor intel (R) Atom(TM) CPU N570 @1.66GHz 1.67 GHz
2. Memori 2GB
3. Monitor 12"
4. *Keyboard*
5. *Mouse*

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem pencarian rute optimum dengan *evolution strategies* adalah sebagai berikut :

1. Sistem operasi Microsoft Windows 7 Ultimate 32-bit sebagai lingkungan kerja dari sistem

2. Google Chrome yang berfungsi sebagai media untuk membuat dan *compile* program, dengan menggunakan bahasa PHP
3. *XAMPP 1.7.4* sebagai perangkat lunak yang digunakan untuk mengembangkan *database* dengan menggunakan *database* mySQL.
4. *Adobe Dreamweaver CS5* sebagai *software* untuk penulisan *code* program dalam PHP

4.2 Implementasi Kode Program

Berdasarkan perancangan perangkat lunak yang telah diuraikan dalam bab III, maka pada bagian ini akan dibahas mengenai implementasi kode program.

4.2.1 Pembangkitan Populasi Awal

Ukuran populasi ditentukan oleh *user* sesuai μ yang telah diinputkan. Dalam pembentukan populasi dilakukan proses pembentukan kromosom, dimana kromosom dibentuk dengan membuat kemungkinan jalur antar *node*. Sehingga pada saat proses pembentukan populasi awal kromosom dipanggil menggunakan *query* berdasarkan lokasi tujuan. Pemanggilan kromosom dilakukan secara random sesuai ukuran populasi. Proses pembentukan populasi awal ditunjukkan pada *source-code* 4.1.

```
1
2 $jumjalur_query = "select * from jalurjalan where nodeakhir = $idjalan";
3 $jumjalur_exe = mysql_query($jumjalur_query);
4 $jumjalur = mysql_num_rows($jumjalur_exe);
5 for($i=0; $i<$miu; $i++)
6 {
7     $rand = rand(1, $jumjalur);
8     $idrand[] = $rand;
9 }
10
11 function PopulasiAwal($idjalan,$idrand,$miu){
12 foreach($idrand as $row)
13 {
14     $query = "select * from jalurjalan where id=$row and nodeakhir= $idjalan";
15     $c = mysql_query($query);
```

```

16 $h=mysql_fetch_row($c);
17 for($j=2;$j<20;$j++) echo $h[$j]. " ";//populasi

```

Source-code 4.1 Pembangkitan Populasi Awal

4.2.2 Hitung *Fitness*

Proses perhitungan nilai *fitness*. Nilai *fitness* diperoleh dari persamaan $1/\text{total_jarak}$. Setiap kromosom yang dipanggil akan dihitung total jaraknya, sehingga mendapatkan nilai *fitness*. Proses perhitungan *fitness* ditunjukkan pada *source-code 4.2*.

```

1 //pencarian total jarak
2 $jarak = 0;
3 for ($j=2;$j<19;$j++)
4 {
5 $d = "select jarak from nodejalan where node1 = ".$h[$j]." and node2 =
6 ".$h[$j+1];
7 $e = mysql_query($d);
8 if($h[$j+1] != "")
9 {
10 $f=mysql_fetch_row($e);
11 $jarak += $f[0];
12 }
13 }
14 echo ($jarak);
15
16 //pencarian fitness
17 $fitnes = 0;
18 for ($j=2;$j<19;$j++)
19 {
20 $d = "select jarak from nodejalan where node1 = ".$h[$j]." and node2 =
21 ".$h[$j+1];
22 $e = mysql_query($d);
23 if($h[$j+1] != "")
24 {
25 $f=mysql_fetch_row($e);
26 $fitnes += $f[0];
27 }
28 }
29 echo (1/$fitnes);

```

Source-code 4.2 Hitung *Fitness*

4.2.3 Proses Mutasi

Metode mutasi yang digunakan pada implementasi program ini adalah *exchange mutation*. Proses mutasi ini diawali dengan menentukan kromosom yang akan di mutasi. Kemudian menentukan dua titik secara *random* untuk dimutasi, dimana gen awal dan gen akhir tidak boleh dilakukan proses penukaran. Proses mutasi ditunjukkan pada *source-code* 4.3.

```

1  $mut = array();
2  $query = "select * from jalurjalan where id=$row and nodeakhir= $idjalan";
3  $c = mysql_query($query);
4  $h=mysql_fetch_row($c);
5  for($j=2;$j<20;$j++) if($h[$j]!="") $mut[] = $h[$j];
6  if(count($mut) <=3) $h_mutasi[] = $mut;
7  else
8  {
9      $JumlahMutasi=0;
10     while ($JumlahMutasi < $lamda)
11     {
12         //proses mutasi
13         $temp = array();
14         for($i=1;$i<count($mut)-1; $i++) $temp[] = $mut[$i];
15         $rand_temp = array_rand($temp, rand(2, count($temp)-1));
16         $id1 = array_search($temp[$rand_temp[0]], $mut);
17         $id2 = array_search($temp[$rand_temp[1]], $mut);
18         $file = $mut[$id1];
19         $mut[$id1] = $mut[$id2];
20         $mut[$id2] = $file;
21         $h_mutasi[] = $mut;
22         $JumlahMutasi++;
23     }
24     foreach($h_mutasi as $row1)
25     {
26         if($it == $lamda){
27             foreach($row1 as $row)
28             {
29                 echo $row." ";

```

Source-code 4.3 Proses *Exchange Mutation*

4.2.4 Proses Seleksi

Proses seleksi dilakukan setelah dilakukan proses penggabungan kromosom populasi awal dan kromosom hasil mutasi dan proses sorting berdasarkan nilai fitness

terbesar. Kromosom yang mempunyai nilai fitness terkecil secara otomatis akan terbuang dan tidak digunakan pada proses pencarian rute pada generasi berikutnya. Proses ini akan menghasilkan rute yang optimal. Seleksi ini menggunakan metode *elitism selection*. Proses seleksi ditunjukkan pada *source-code* 4.4.

```

1 //Penyimpanan kromosom populasi awal pada database
2 $indv = "P$i";
3 $fitn = (1/$fitnes);
4 $jalurGabungan=" ";
5   for($j=2;$j<20;$j++)
6     {
7       if($h[$j]!="NULL")
8         $jalurGabungan.=$h[$j]." ";
9     }
10 $qinsert = mysql_query("insert into elitism (individu, kromosom,
11 total_jarak, fitnes) value('$indv', '$jalurGabungan', '$jarak', '$fitn')");
12 //Penyimpanan kromosom hasil mutasi pada database
13 $indve = "C$z";
14 $fitne = (1/$fitnes);
15 $jalurGabungane=" ";
16   for($je=2;$je<20;$je++)
17     {
18       if($row!="NULL")
19         $jalurGabungane.=$row." ";
20     }
21 $qinsert = mysql_query("insert into elitism (individu, kromosom,
22 total_jarak, fitnes) value('$indve', '$jalurGabungane', '$jarak',
23 '$fitne')");
24 //Penggabungan serta peyortingan kumpulan kromosom awal dan kromosom hasil
25 mutasi
26 $query = "SELECT * FROM elitism order by fitnes desc";
27 $exe = mysql_query($query);
28 while($row = mysql_fetch_assoc($exe))
29   {
30     $a = $row['individu'];
31     $b = $row['kromosom'];
32     $c = $row['total_jarak'];
33     $d = $row['fitnes'];
34     echo "<tr><td>$a</td><td>$b</td><td>$c</td><td>$d</td></tr>";
35   }

```

Source-code 4.4 Proses *Elitism Selection*

4.2.5 Menampilkan Peta Rute

Bagian ini menjelaskan kode program dari implementasi menampilkan hasil rute melalui *Google Maps*. Potongan kode program yaitu pada *source-code* 4.5.

```
1 <script
2 src="https://maps.googleapis.com/maps/api/js?v=3.exp?key=FGF2Gh8Ppjc2x1fwgC25
3 RSiU"></script>
4 <script>
5 // This example creates a 2-pixel-wide red polyline showing
6 // the path of William Kingsford Smith's first trans-Pacific flight between
7 // Oakland, CA, and Brisbane, Australia.
8
9 function initialize() {
10     var mapOptions = {
11         zoom: 25,
12         center: new google.maps.LatLng(-7.331459, 112.773779), //rungkut mapan
13         utara
14         mapTypeId: google.maps.MapTypeId.TERRAIN
15     };
16
17     var map = new google.maps.Map(document.getElementById('map-canvas'),
18         mapOptions);
19
20     var image = ['icons/marker2.png', 'icons/marker3.png', 'icons/marker4.png',
21 'icons/marker5.png', 'icons/marker6.png', 'icons/marker7.png',
22 'icons/marker8.png', 'icons/marker9.png', 'icons/marker10.png',
23 'icons/marker11.png', 'icons/marker12.png', 'icons/marker13.png',
24 'icons/marker14.png', 'icons/marker15.png', 'icons/marker16.png',
25 'icons/marker17.png', 'icons/marker18.png', 'icons/marker19.png',
26 'icons/marker20.png', 'icons/marker21.png', 'icons/marker22.png',
27 'icons/marker23.png', 'icons/marker24.png', 'icons/marker25.png',
28 'icons/marker26.png', 'icons/marker27.png', 'icons/marker28.png',
29 'icons/marker29.png', 'icons/marker30.png', 'icons/marker31.png',
30 'icons/marker32.png', 'icons/marker33.png', 'icons/marker34.png',
31 'icons/marker35.png', 'icons/marker36.png', 'icons/marker37.png',
32 'icons/marker38.png', 'icons/marker39.png', 'icons/marker40.png',
33 'icons/marker41.png', 'icons/marker42.png', 'icons/marker43.png',
34 'icons/marker44.png', 'icons/marker45.png', 'icons/marker46.png',
35 'icons/marker47.png', 'icons/marker48.png', 'icons/marker49.png',
36 'icons/marker50.png', 'icons/marker51.png', 'icons/marker52.png',
37 'icons/marker53.png', 'icons/marker54.png', 'icons/marker55.png',
38 'icons/marker56.png', 'icons/marker57.png', 'icons/marker58.png',
39 'icons/marker59.png', 'icons/marker60.png', 'icons/marker61.png',
40 'icons/marker62.png', 'icons/marker63.png', 'icons/marker64.png',
41 'icons/marker65.png', 'icons/marker66.png', 'icons/marker67.png',
42 'icons/marker68.png', 'icons/marker69.png'];
43
44     var flightPlanCoordinates = [
45
46         new google.maps.LatLng(-7.331459, 112.773779), //rungkut mapan utara
47         new google.maps.LatLng(-7.332500, 112.775350), //rungkut mapan raya
48         new google.maps.LatLng(-7.331655, 112.784178), //jl. rungkut madya
49         new google.maps.LatLng(-7.322927, 112.791551), //penjaringan asri ix
50         new google.maps.LatLng(-7.322417, 112.782558), //pandugo
51         new google.maps.LatLng(-7.325429, 112.781641) //pandugo gg.1
52     ];
53     var flightPath = new google.maps.Polyline({
54         path: flightPlanCoordinates,
55         geodesic: true,
56         strokeColor: '#FF0000',
57         strokeOpacity: 1.0,
```



```
58     strokeWeight: 2
59   });
60
61   function addMarker(lat,lng, position) {
62     var myLatLng = new google.maps.LatLng(lat,lng);
63     var marker = new google.maps.Marker({
64       map: map,
65       position: myLatLng,
66       icon: image[position],
67       name: "test"
68     })
69   }
70 }
71
72 function addMarker2(lat,lng) {
73   var myLatLng = new google.maps.LatLng(lat,lng);
74   var image2 = 'icons/marker1.png';
75   var marker = new google.maps.Marker({
76     map: map,
77     position: myLatLng,
78     icon: image2,
79     name: "test"
80   })
81 }
82 }
83
84
85
86
87   addMarker2(-7.331459, 112.773779); //rungkut mapan utara
88   addMarker(-7.313219, 112.763881,0); //delta permai
89   addMarker(-7.319141, 112.760993,1); //tenggilis mejoyo
90   addMarker(-7.309749, 112.773973,2); //jl. kedung baru
91   addMarker(-7.310595, 112.769619,3); //jl. raya rungkut
92   addMarker(-7.311803, 112.791766,4); //wonorejo rungkut
93   addMarker(-7.311683, 112.798378,5); //wonorejo tambak
94   addMarker(-7.318757, 112.78097,6); //jl. soekarno hatta
95   addMarker(-7.322829, 112.7735,7); //rungkut puskesmas
96   addMarker(-7.322725, 112.780099,8); //rungkut alang-alang
97   addMarker(-7.322417, 112.782558,9); //pandugo
98   addMarker(-7.325429, 112.781641,10); //pandugo gg.1
100  addMarker(-7.321663, 112.796361,11); //medokan ayu
101  addMarker(-7.320388, 112.759133,12); //jl. raya tenggilis
102  addMarker(-7.322982, 112.752989,13); //kendangsari
103  addMarker(-7.326856, 112.749481,14); //kendangsari (2)
104  addMarker(-7.326802, 112.755253,15); //rungkut industri 1
105  addMarker(-7.330186, 112.742056,16); //jl. kutisari
106  addMarker(-7.332389, 112.748493,17); //kutisari utara
107  addMarker(-7.336050, 112.745575,18); //kutisari selatan
108  addMarker(-7.339221, 112.758686,19); //berbek indudtri iii
109  addMarker(-7.332985, 112.751203,20); //rungkut industri iii
110  addMarker(-7.334549, 112.750972,21); //rungkut industri iii (2)
111  addMarker(-7.335608, 112.756390,22); //rungkut industri iii (3)
112  addMarker(-7.332187, 112.759335,23); //rungkut industri v
113  addMarker(-7.334570, 112.759120,24); //rungkut industri v (2)
114  addMarker(-7.333191, 112.754733,25); //rungkut industri iv
115  addMarker(-7.325496, 112.757121,26); //jl. raya rungkut industri
```



```

116 addMarker(-7.328210, 112.756896,27); //jl. raya rungkut industri (2)
117 addMarker(-7.325117, 112.761496,28); //rungkut indutri ii
118 addMarker(-7.324660, 112.754533,29); //rungkut indutri ii (2)
119 addMarker(-7.330902, 112.766267,30); //rungkut indutri kidul
120 addMarker(-7.334258, 112.769149,31); //rungkut tengah
121 addMarker(-7.339783, 112.767927,32); // rungkut menanggal
122 addMarker(-7.339318, 112.773526,33); //abdul karim
123 addMarker(-7.332544, 112.770409,34); //jl. rungkut kidul
124 addMarker(-7.325511, 112.76584,35); //jl. rungkut lor
125 addMarker(-7.344653, 112.768078,36); //wadungsari
126 addMarker(-7.340587, 112.780434,37); //amir machmud
127 addMarker(-7.327043, 112.775592,38); //rungkut asri utara xiii
128 addMarker(-7.326405, 112.773945,39); //rungkut asri xiii
129 addMarker(-7.328491, 112.77629,40); //rungkut asri tengah
130 addMarker(-7.331655, 112.784178,41); //jl. rungkut madya
131 addMarker(-7.320759, 112.796122,42); //jl. medokan ayu
132 addMarker(-7.325578, 112.781077,43); //jl. soekarno hatta (2)
133 addMarker(-7.333202, 112.795298,44); //medokan sawah
134 addMarker(-7.338065, 112.795137,45); //gunung anyar jaya
135 addMarker(-7.337416, 112.801832,46); //wiguna timur
136 addMarker(-7.333313, 112.805964,47); //gunung anyar timur
137 addMarker(-7.342097, 112.801756,48); //gunung anyar tambak
138 addMarker(-7.322927, 112.791551,49); //penjaringan asri ix
139 addMarker(-7.324908, 112.800117,50); //jl. medokan kampung
140 addMarker(-7.341023, 112.800393,51); //gunung anyar tambak i-iii
141 addMarker(-7.343744, 112.801186,52); //gunung anyar asri
142 addMarker(-7.322965, 112.757426,53); //tenggilis mulyo
143 addMarker(-7.325932, 112.757089,54); //rungkut industri vi
144 addMarker(-7.328401, 112.756875,55); //rungkut industri vi (2)
145 addMarker(-7.321816, 112.761889,56); //raya tenggilis
146 addMarker(-7.319180, 112.776077,57); //kedung asem
147 addMarker(-7.308222, 112.768332,58); //jl. panjang jiwo
148 addMarker(-7.331090, 112.770554,59); //jl. kh.zamhuri
149 addMarker(-7.332500, 112.775350,60); //rungkut mapan raya
150 addMarker(-7.332521, 112.771037,61); //rungkut permai
151 addMarker(-7.332739, 112.773338,62); //rungkut mapan barat
152 addMarker(-7.334687, 112.772893,63); //rungkut mapan selatan
153 addMarker(-7.336953, 112.772496,64); //rungkut barata
154 addMarker(-7.334234, 112.775275,65); //rungkut mapan raya (2)
155 addMarker(-7.333546, 112.780956,66); //gunung anayr lor
156 addMarker(-7.340486, 112.780585,67); //gunung anyar tengah
157 flightPath.setMap(map);
158 }
159
160 google.maps.event.addDomListener(window, 'load', initialize);
161
162 </script>

```

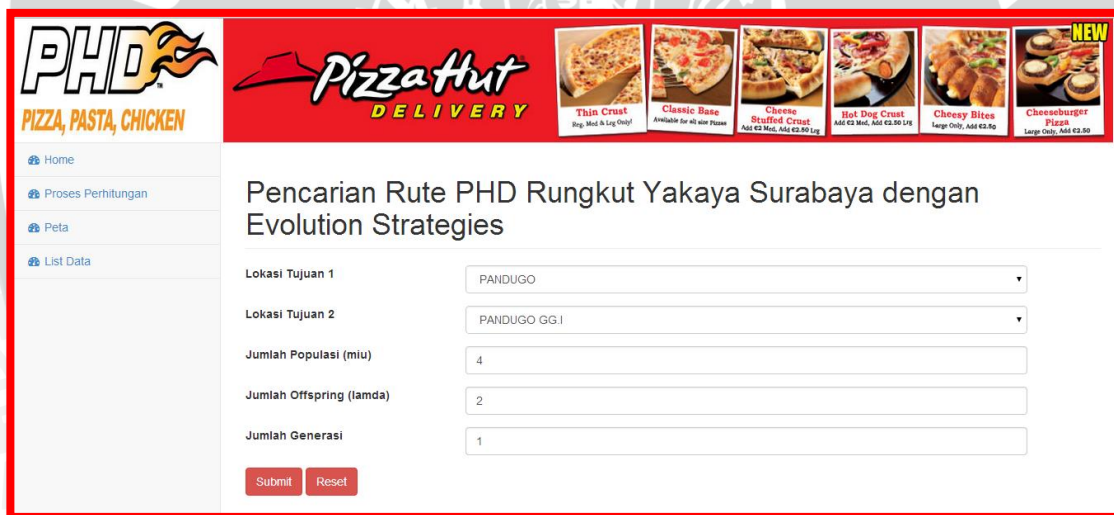
Source-code 4.5 Peta Rute

4.3 Implementasi Antarmuka

Berdasarkan rancangan antarmuka yang ada di Bab III, terdapat 3 form dalam tampilan sistem pencarian rute optimum delivery order dengan menggunakan *evolution strategies*.

4.3.1 Form Input Parameter *Evolution Strategies*

Pada form input rute *delivery order*, pegawai dapat melihat rute untuk pengantaran ke 2 (dua) lokasi tujuan. Outlet PHD merupakan *node* awal atau pusat dimulainya proses pencarian rute. Gambar 4.1 menunjukkan form input Input Parameter *Evolution Strategies*.



The screenshot shows the PHD Pizza Hut Delivery app interface. At the top, there is a navigation menu with options: Home, Proses Perhitungan, Peta, and List Data. The main content area is titled "Pencarian Rute PHD Rungkut Yakaya Surabaya dengan Evolution Strategies". Below the title, there are four input fields for parameters: "Lokasi Tujuan 1" (set to PANDUGO), "Lokasi Tujuan 2" (set to PANDUGO GG.1), "Jumlah Populasi (miu)" (set to 4), and "Jumlah Offspring (lamda)" (set to 2). There is also a field for "Jumlah Generasi" (set to 1). At the bottom of the form, there are two buttons: "Submit" and "Reset".

Gambar 4.1 Form Input Parameter *Evolution Strategies*

4.3.2 Form Proses Perhitungan

Form proses perhitungan merupakan tampilan proses hasil dari *evolution strategies*. Form ini terdapat proses input, populasi awal, *offspring* hasil mutasi *exchange*, sorting, proses *elitism selection*, hasil populasi baru, dan hasil rute yang optimal berdasarkan banyak generasi yang diinputkan. Gambar 4.2 merupakan hasil input berdasarkan parameter *evolution strategies*.

| Inputan | |
|------------------|----|
| Lokasi Tujuan 1 | 11 |
| Lokasi Tujuan 2 | 12 |
| Jumlah Populasi | 4 |
| Jumlah Offspring | 2 |
| Jumlah Generasi | 1 |

Gambar 4.2 Hasil Input Parameter *Evolution Strategies*

Berdasarkan gambar 4.2 ditunjukkan bahwa inputan lokasi tujuan 1 adalah jalan dengan *node* 11 (Pandugo) , lokasi tujuan 2 adalah 12 (Pandugo Gg. 11), jumlah populasi 4, jumlah *offspring* adalah 2, dan jumlah generasi 1. Setelah proses penginputan selesai, maka akan dilakukan proses perhitungan.

Pada gambar 4.3 sampai dengan gambar 4.7 merupakan proses perhitungan dengan menggunakan *evolution strategies* yang menghasilkan rute yang optimal.

| Populasi Awal | | | |
|---------------|--|-------------|-------------------|
| | Kromosom | Total Jarak | Fitness |
| P1 | 1 62 42 41 40 9 58 3 2 4 8 11 12 | 20.603 | 0.048536620880454 |
| P2 | 1 64 63 61 42 41 40 9 58 3 2 60 5 59 11 12 | 27.141 | 0.036844626211267 |
| P3 | 1 62 61 36 37 59 11 12 | 11.107 | 0.09003331232556 |
| P4 | 1 62 43 51 11 12 | 5.807 | 0.17220595832616 |

Gambar 4.3 Populasi Awal

Gambar 4.3 merupakan proses pembentukan populasi awal, dimana populasi awal sebanyak 4 kromosom . Setiap kromosom pembentukan dari beberapa gen atau *node* jalan yang merupakan calon solusi. Kromosom mempunyai nilai jarak dan nilai *fitness*.

| Offspring Hasil Mutasi | | | | |
|------------------------|-------|--|-------------|---------------------|
| Parent | Child | Kromosom | Total Jarak | Fitness |
| P1 | C1 | 1 62 42 2 40 9 58 3 41 4 8 11 12 | 2004.695 | 0.00049882899892502 |
| P1 | C2 | 1 42 62 2 40 9 58 3 41 4 8 11 12 | 3004.262 | 0.00033286044958795 |
| P2 | C3 | 1 63 64 61 42 41 40 9 58 3 2 60 5 59 11 12 | 1005.347 | 0.00099468143834915 |
| P2 | C4 | 1 63 11 61 42 41 40 9 58 3 2 60 5 59 64 12 | 2004.494 | 0.00049887901884466 |
| P3 | C5 | 1 62 61 36 59 37 11 12 | 2002.615 | 0.00049934710366196 |
| P3 | C6 | 1 62 61 36 37 59 11 12 | 7.414 | 0.13487995683841 |
| P4 | C7 | 1 11 43 62 51 12 | 2004.742 | 0.00049881730417181 |
| P4 | C8 | 1 62 43 51 11 12 | 5.807 | 0.17220595832616 |

Gambar 4.4 Offspring Hasil Mutasi

Gambar 4.4 menampilkan individu hasil mutasi sebanyak 8 *children*. Sebanyak 8 *children* dikarenakan inputan banyak *offspring* adalah 2, jadi setiap *parent* (populasi awal) menghasilkan 2 *children*.

| Hasil Sorting Kumpulan Individu Awal dan Offspring | | | |
|--|--|-------------|-------------|
| | Kromosom | Total Jarak | Fitness |
| C8 | 1 62 43 51 11 12 | 5.807 | 0.172206 |
| P4 | 1 62 43 51 11 12 | 5.807 | 0.172206 |
| C6 | 1 62 61 36 37 59 11 12 | 7.414 | 0.13488 |
| P3 | 1 62 61 36 37 59 11 12 | 11.107 | 0.0900333 |
| P1 | 1 62 42 41 40 9 58 3 2 4 8 11 12 | 20.603 | 0.0485366 |
| P2 | 1 64 63 61 42 41 40 9 58 3 2 60 5 59 11 12 | 27.141 | 0.0368446 |
| C3 | 1 63 64 61 42 41 40 9 58 3 2 60 5 59 11 12 | 1005.35 | 0.000994681 |
| C5 | 1 62 61 36 59 37 11 12 | 2002.61 | 0.000499347 |
| C4 | 1 63 11 61 42 41 40 9 58 3 2 60 5 59 64 12 | 2004.49 | 0.000498879 |
| C1 | 1 62 42 2 40 9 58 3 41 4 8 11 12 | 2004.69 | 0.000498829 |
| C7 | 1 11 43 62 51 12 | 2004.74 | 0.000498817 |
| C2 | 1 42 62 2 40 9 58 3 41 4 8 11 12 | 3004.26 | 0.00033286 |

Gambar 4.5 Hasil Sorting Kumpulan Individu Awal dan Individu Hasil Mutasi

Setelah proses mutasi maka individu awal dan individu hasil kromosom dikumpulkan, kemudian *disorting* atau diranking berdasarkan nilai *fitness* terbesar. Kumpulan individu awal dan hasil mutasi ditunjukkan pada Gambar 4.5.

| Populasi Baru Hasil Seleksi | | | |
|-----------------------------|------------------------|-------------|-----------|
| | Kromosom | Total Jarak | Fitness |
| C8 | 1 62 43 51 11 12 | 5.807 | 0.172206 |
| P4 | 1 62 43 51 11 12 | 5.807 | 0.172206 |
| C6 | 1 62 61 36 37 59 11 12 | 7.414 | 0.13488 |
| P3 | 1 62 61 36 37 59 11 12 | 11.107 | 0.0900333 |

Gambar 4.6 Populasi Baru Hasil *Elitism Selection*

Gambar 4.6 menampilkan populasi baru yang didapatkan setelah proses *sorting*. Populasi baru diambil sebanyak 4 individu, dikarenakan inputan populasi awal adalah 4. Populasi baru ini akan digunakan pada proses generasi selanjutnya sesuai inputan jumlah generasi. Kemudian dilakukan proses seleksi untuk menghasilkan individu yang paling baik dengan nilai *fitness* terbesar yang merupakan rute optimal

| Hasil Tiap Generasi |
|--|
| GENERASI 1 Kromosom : 1 62 43 51 11 12 Total Jarak : 5.807 fitness : 0.172206 |

Gambar 4.7 Hasil Solusi Setiap Generasi

Gambar 4.7 merupakan hasil solusi setiap generasi. Pada tampilan ini ditunjukkan hanya 1 generasi karena inputan jumlah generasi adalah 1.

| Rute Optimal | |
|--|-------------|
| Rute Jalan | Total Jarak |
| RUNGKUT MAPAN UTARA - RUNGKUT MAPAN RAYA- JL. RUNGKUT MADYA - PENJARINGAN ASRI IX - PANDUGO - PANDUGO GG.I | 5.807 |

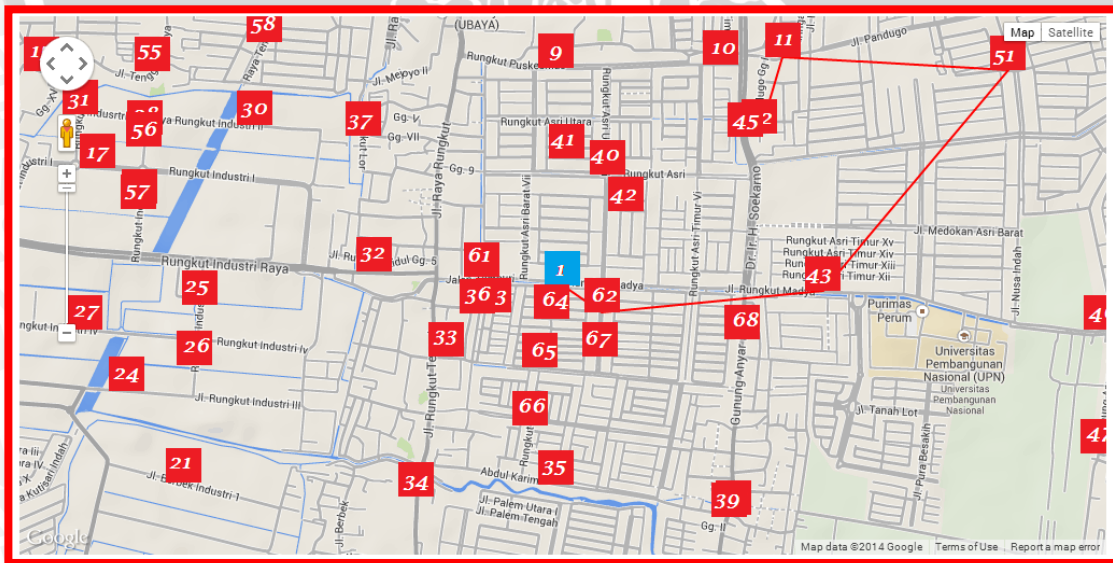
peta

Gambar 4.8 Rute Optimal

Setelah proses pembentukan populasi awal, mutasi, dan seleksi maka menghasilkan rute optimal berdasarkan jarak yang paling optimal. Rute optimal ditunjukkan pada Gambar 4.8. Untuk melihat peta rute yang dihasilkan maka terdapat tombol peta.

4.3.3 Form Peta

Form peta merupakan form yang menampilkan node jalan yang terhubung yang dilalui sesuai dengan lokasi tujuan yang diinputkan. Gambar 4.9 menunjukkan form peta.



Gambar 4.9 Peta Rute

Gambar 4.9 memperlihatkan tampilan form peta dari aplikasi ini. Form ini ditampilkan ketika proses perhitungan selesai yang menghasilkan rute jalan yang akan dilewati yang ditampilkan pada gambar 4.8 untuk menuju lokasi tujuan yang di inputkan di awal proses. Pada form ini terdapat 4 tombol panah dan tombol zoom in dan zoom out saja dalam bentuk yang kecil. Pada form ini memperlihatkan ada 69 marker yang masing-masing marker merupakan *node* jalan. Marker berwarna biru yang ditandai dengan *node* nomor 1 merupakan titik awal atau pusat lokasi pizza hut delivery (PHD) dengan nama jalan Rungkut Mapan Utara. Sedangkan marker berwarna merah yang ditandai dengan *node* 2 hingga *node* 3 merupakan marker untuk lokasi jalan tujuan. Pada peta tersebut dijelaskan bahwa rute berjalan dari *node* 1 menuju *node* 62, 43, 51, 11, dan 12. Dimana input lokasi tujuan adalah *node* 11 dan 12. Line pada form menghubungkan *node-node* rute yang dilewati.

4.3.4 Form List Data

Form list data merupakan form yang menampilkan list jalan untuk wilayah Pizza Hut *Delivery* Rungkut Yakaya Surabaya. Pada form ini juga terdapat data jarak antar *node*. Gambar 4.10 menunjukkan form list data.

Tabel Data Wilayah PHD

| ID | Nama Jalan | Node 1 | Node 2 | Panjang Jalan |
|----|---------------------|--------|--------|---------------|
| 1 | Rungkut Mapan Utara | 26 | 4 | 0.260 |
| 2 | Delta Permai | 1 | 5 | 0.300 |
| 3 | Tenggilis Mejoyo | 6 | 19 | 0.230 |
| 4 | Jl. Kedung Baru | 1 | 2 | 1.000 |
| 5 | Jl. Raya Rungkut | 7 | 8 | 2.480 |
| 6 | wonorejo Rungkut | 2 | 3 | 0.438 |
| 7 | Wonorejo Tambak | 3 | 4 | 0.769 |
| 8 | Jl. Soekarno Hatta | 2 | 10 | 1.110 |
| 9 | Rungkut Puskesmas | 8 | 9 | 1.190 |
| 10 | Rungkut Alang-Alang | 9 | 10 | 0.218 |

Showing 1 to 10 of 36 entries

Previous 1 2 3 4 Next

Gambar 4.10 Tampilan List Data Jalan PHD

BAB V PENGUJIAN

Bab ini akan menjelaskan mengenai pengujian sesuai dengan rancangan pengujian yang telah dipaparkan pada bab III.

5.1 Sistematika Pengujian

Pengujian *Evolution Strategies* dilakukan untuk menguji pengaruh banyak generasi serta pengujian pengaruh ukuran populasi (*miu*) dan ukuran *offspring* (*lamda*) untuk memecahkan permasalahan pencarian rute optimum.

Seperti yang telah dijelaskan pada bab III, pada penelitian ini terdapat 3 macam pengujian yaitu uji kombinasi ukuran populasi (*miu*) dan ukuran *offspring* (*lamda*) serta uji coba banyak generasi. Pengujian 1 dilakukan untuk mengetahui hasil dari banyaknya alternative solusi yang dihasilkan. Pengujian 2 dilakukan untuk mengetahui pengaruh ukuran populasi (*miu*) dan ukuran *offspring* (*lamda*). Uji coba dilakukan dengan mengkombinasikan nilai ukuran populasi (*miu*) dan ukuran *offspring* (*lamda*). Pengujian 3 dilakukan untuk mengetahui pengaruh banyak generasi terhadap nilai *fitness* dengan memberikan ukuran populasi (*miu*) dan ukuran *offspring* (*lamda*) hasil terbaik dari pengujian sebelumnya.

5.1.1 Sistematika Uji Coba I

Proses pengujian ini dilakukan dengan menginputkan populasi awal sebanyak 4, jumlah *offspring* sebanyak 2, jumlah generasi 1, serta lokasi tujuan adalah 1 (pandugo) dan pandugo Gg.1. Pengujian bertujuan untuk mengetahui pengaruh banyak alternative solusi yang diperoleh dengan perhitungan menggunakan *evolution strategies* berdasarkan jarak minimal dan nilai *fitness* terbesar.

5.1.2 Sistematika Uji Coba I I

Proses pengujian ini dilakukan dengan mengubah ukuran populasi (μ) dan ukuran *offspring* (λ). Ukuran ukuran populasi sebesar 5, 10, 15, 20, 25, dan 30, sedangkan untuk ukuran *offspring* adalah sebesar 1μ , 2μ , 3μ , 4μ , 5μ , 6μ , 7μ , 8μ , 9μ , dan 10μ . Pengujian ini bertujuan untuk mengetahui pengaruh perubahan ukuran populasi dan *offspring* terhadap nilai *fitness* rata-rata. Setiap nilai *fitness* rata-rata inilah yang akan dibandingkan, sehingga akan didapatkan analisa pengaruh dari perubahan nilai ukuran populasi (μ) dan ukuran *offspring* (λ). Analisa hasil pengujian dilakukan dengan tidak mempertimbangkan lama waktu pengujian.

5.1.3 Sistematika Uji Coba III

Proses pengujian ini dilakukan dengan mengubah banyaknya generasi. Banyaknya generasi sebesar 1 sampai 30. Pengujian ini bertujuan untuk mengetahui pengaruh perubahan banyaknya generasi terhadap nilai *fitness* rata-rata. Percobaan uji coba III ini dilakukan pengulangan sebanyak 5 kali. Hal ini dilakukan karena konsep dari *evolution strategies* yang random menyebabkan kromosom atau solusi yang beragam sehingga dilakukan beberapa kali uji coba untuk kemudian di rata-rata. Analisa hasil pengujian dilakukan dengan tidak mempertimbangkan lama waktu pengujian.

5.2 Implementasi Pengujian

Subbab ini akan dijelaskan tentang hasil pengujian uji coba I, uji coba II dan pengujian uji coba III berdasarkan sistematika pengujian yang telah dijabarkan.

5.2.1 Hasil dan Analisa Uji Coba I

Algoritma *Evolution Strategies* memiliki kelebihan yaitu dapat menghasilkan banyak solusi dari proses mutasi sesuai yang diinginkan, sehingga alternatif solusi yang ditawarkan muncul lebih banyak. Selain itu pada tahap seleksi, populasi yang dihasilkan dari tiap tahapan algoritma dikumpulkan kembali untuk diambil solusi

terbaik, sehingga tidak ada solusi terbaik yang terlewatkan. Dari hasil ujicoba algoritma yang dilakukan berulang kali tanpa batasan iterasi atau mencapai konvergen, sehingga mendapatkan solusi yang paling optimal, didapatkan hasil yang ditunjukkan pada tabel 5.1 di bawah ini:

Tabel 5.1 : Solusi Optimal oleh *Evolution Strategies*

| No | Rute | Jarak Tempuh (Km) | Fitness |
|----|--------------------------|-------------------|---------|
| 1 | [1-62-43-52-11-12] | 5.807 | 0.17221 |
| 2 | [1-62-43-51-11-12] | 5.807 | 0.17221 |
| 3 | [1-62-61-36-37-59-11-12] | 7.414 | 0.13488 |
| 4 | [1-62-61-36-38-59-11-12] | 11.107 | 0.09003 |

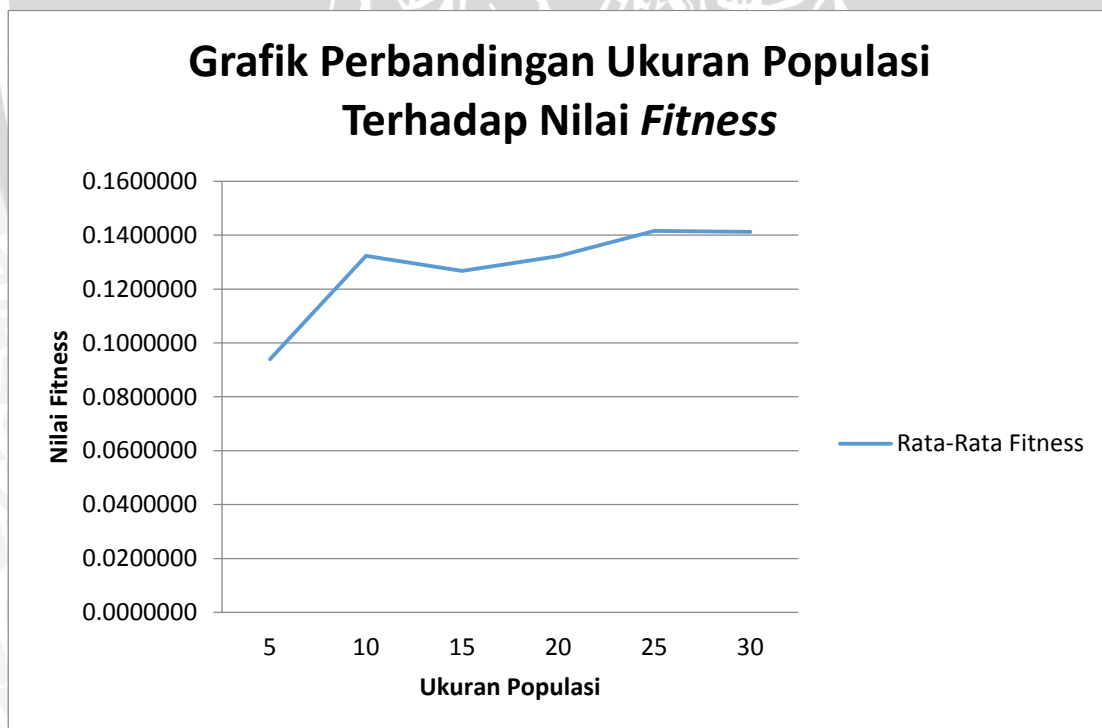
5.2.2 Hasil dan Analisa Uji Coba II

Berdasarkan sistematika pengujian yang telah dijabarkan sebelumnya, pengujian terhadap pengaruh kombinasi ukuran populasi (*miu*) dan ukuran *offspring* (*lamda*) dilakukan pada generasi 100. Ukuran jumlah populasi yang digunakan adalah 5, 10, 25, 30, 35, 40, 45, dan 50. Sedangkan untuk ukuran jumlah *offspring* adalah 1, 2, 3, 4, 5, 6, 7, 8, 9, dan 10. Hasil pengujian ditunjukkan pada tabel 5.1.

Tabel 5.2 Hasil Uji Coba II

| Ukuran <i>Offspring</i> (λ) | Ukuran Populasi (μ) | | | | | | Rata - Rata |
|---------------------------------------|---------------------------|-----------|------------|-----------|-----------|-----------|-------------|
| | 5 | 10 | 15 | 20 | 25 | 30 | |
| 1 μ | 0.0788022 | 0.1052630 | 0.141483 | 0.1414830 | 0.141483 | 0.141483 | 0.1249995 |
| 2 μ | 0.0858811 | 0.1093250 | 0.109325 | 0.1450750 | 0.1398010 | 0.14183 | 0.1218729 |
| 3 μ | 0.1153930 | 0.1153930 | 0.141483 | 0.1450750 | 0.141483 | 0.14183 | 0.1334428 |
| 4 μ | 0.1093250 | 0.1245430 | 0.141483 | 0.1031570 | 0.1398010 | 0.141483 | 0.1266320 |
| 5 μ | 0.1052630 | 0.1414830 | 0.115393 | 0.1414830 | 0.141483 | 0.141483 | 0.1310980 |
| 6 μ | 0.1153930 | 0.1539450 | 0.141483 | 0.1414830 | 0.141483 | 0.141483 | 0.1392117 |
| 7 μ | 0.1108160 | 0.0844737 | 0.141483 | 0.1398010 | 0.141483 | 0.141483 | 0.1265900 |
| 8 μ | 0.0564653 | 0.1653761 | 0.103157 | 0.1093250 | 0.141483 | 0.141483 | 0.1195482 |
| 9 μ | 0.0844737 | 0.1623109 | 0.0901551 | 0.1450750 | 0.141483 | 0.141483 | 0.1274968 |
| 10 μ | 0.0780275 | 0.1602286 | 0.141483 | 0.1093250 | 0.141483 | 0.141483 | 0.1286717 |
| Rata -Rata | 0.0939840 | 0.1322341 | 0.12669281 | 0.1321282 | 0.1411466 | 0.1415524 | ----- |

Pada tabel 5.2 terlihat bahwa nilai *fitness* terbesar dihasilkan oleh pengujian dengan ukuran populasi 30 dengan nilai *fitness* 0.1415524 dan yang terkecil adalah ukuran populasi 5 dengan nilai *fitness* 0.0939840. Pada tabel 5.1 juga ditunjukkan bahwa nilai *fitness* terbesar dihasilkan oleh pengujian dengan ukuran *offspring* 6 dengan nilai *fitness* 0.1392117 dan yang terkecil adalah ukuran *offspring* 8 dengan nilai *fitness* 0.1195482. Nilai *fitness* yang diperoleh terkadang tidak mengalami perubahan dan terkadang juga mengalami perubahan, baik perubahan kearah penurunan nilai *fitness* ataupun kenaikan nilai *fitness*. Dengan menggunakan populasi awal dan nilai parameter *evolution strategies* yang sama tidak dapat dipastikan akan memperoleh nilai *fitness* yang sama. Hal ini terjadi karena konsep *evolution strategies* adalah random. Grafik pengaruh ukuran populasi (*miu*) dan ukuran *offspring* (*lamda*) ditunjukkan pada gambar 5.1 dan gambar 5.2.



Gambar 5.1 Grafik Perbandingan ukuran Populasi (*miu*) Terhadap Nilai *Fitness*

Grafik perbandingan ukuran populasi dengan nilai *fitness* pada gambar 5.1 cenderung naik. Hal ini menunjukkan bahwa seiring bertambahnya ukuran populasi (*miu*), nilai *fitness* juga semakin besar. Pada grafik diatas menunjukkan bahwa ada penurunan nilai *fitness* saat ukuran populasi 15 dan naik kembali saat ukuran populasi 20. Secara teori ukuran populasi yang besar memungkinkan terdapat variasi individu yang beragam dan ukuran populasi yang kecil memungkinkan variasi individu sedikit tidak beragam. Tetapi pada grafik yang ditunjukkan pada gambar 5.1 tidak menjamin bahwa ukuran populasi yang besar akan menghasilkan nilai *fitness* yang besar. Semakin besar jumlah populasi yang digunakan maka kemungkinan untuk mendapatkan solusi semakin lambat.



Gambar 5.2 Grafik Perbandingan Ukuran *Offspring* (*lamda*) Terhadap Nilai *Fitness*

Grafik perbandingan ukuran *offspring* (*lamda*) dengan nilai *fitness* pada gambar 5.2 cenderung naik turun. Hal ini menunjukkan bahwa seiring bertambahnya ukuran *offspring* (*lamda*), nilai *fitness* juga semakin besar. Pada grafik diatas menunjukkan bahwa ada penurunan nilai *fitness* saat ukuran *offspring* 2 dan naik

kembali saat ukuran *offspring* 3. Kemudian turun lagi saat ukuran *offspring* 4 dan naik kembali saat ukuran *offspring* 5 dan 6. Secara teori ukuran *offspring* yang besar memungkinkan terdapat variasi individu yang beragam dan jumlah *offspring* yang kecil memungkinkan variasi individu sedikit tidak beragam. Tetapi pada grafik yang ditunjukkan pada gambar 5.2 tidak menjamin bahwa jumlah *offspring* yang besar akan menghasilkan nilai *fitness* yang besar. Hal ini disebabkan pada saat pemilihan titik / gen untuk digunakan pada proses mutasi, dilakukan secara random.

5.2.2 Hasil dan Analisa Uji Coba III

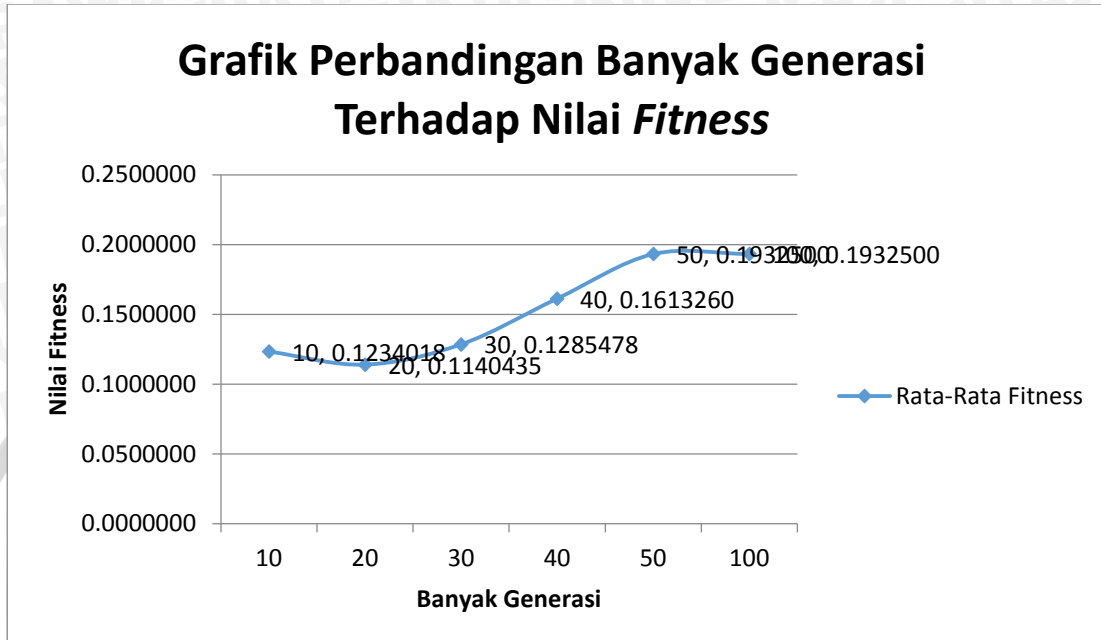
Berdasarkan sistematika pengujian yang telah dijabarkan sebelumnya, pengujian terhadap pengaruh ukuran generasi dilakukan dengan menggunakan ukuran populasi (μ) adalah 30 dan ukuran *offspring* (λ) adalah 6. Untuk ukuran generasi yang digunakan yaitu 10, 20, 30, 40, 50, dan 100. Untuk setiap pengujian dilakukan 5 kali uji coba untuk kemudian diambil rata-rata. Berikut adalah hasil uji ukuran generasi yang ditunjukkan pada tabel 5.3.

Tabel 5.3 Hasil Uji Coba III

| Banyak Generasi | $\mu = 30, \lambda = 6$ | | | | | Rata-Rata Nilai <i>Fitness</i> |
|-----------------|-------------------------|-----------|-----------|-----------|----------|--------------------------------|
| | Percobaan Ke - | | | | | |
| | 1 | 2 | 3 | 4 | 5 | |
| 10 | 0.1414830 | 0.1093250 | 0.1153930 | 0.1093250 | 0.141483 | 0.1234018 |
| 20 | 0.1153930 | 0.0564653 | 0.1414830 | 0.1414830 | 0.115393 | 0.1140435 |
| 30 | 0.1153930 | 0.1414830 | 0.1052630 | 0.1652070 | 0.115393 | 0.1285478 |
| 40 | 0.1414830 | 0.1414830 | 0.1652070 | 0.1652070 | 0.193250 | 0.1613260 |
| 50 | 0.1932500 | 0.1932500 | 0.1932500 | 0.1932500 | 0.193250 | 0.1932500 |
| 100 | 0.1932500 | 0.1932500 | 0.1932500 | 0.1932500 | 0.193250 | 0.1932500 |

Pada Tabel 5.3 dapat diketahui bahwa rata-rata nilai *fitness* tertinggi diperoleh ketika generasi berjumlah banyak yaitu 50 dan 100 generasi dengan rata-rata *fitness* sebesar 0.1932500. Sedangkan nilai *fitness* terkecil diperoleh pada saat jumlah generasi 10. Namun semakin banyak generasi, semakin lama pula waktu eksekusi

yang dibutuhkan. Grafik pengaruh pengaruh jumlah generasi ditunjukkan pada Gambar 5.3.



Gambar 5.3 Grafik Pengaruh Banyak Generasi Terhadap Rata-rata *Fitness*

Grafik perbandingan banyaknya banyak generasi dengan nilai *fitness* pada Gambar 5.3 secara keseluruhan cenderung naik. Hal ini menunjukkan bahwa seiring bertambahnya generasi nilai *fitness* juga semakin besar. Pada grafik diatas menunjukkan bahwa nilai *fitness* mengalami penurunan ketika banyak generasi 20 dan naik kembali ketika banyak populasi 30. Tetapi, semakin banyak banyak generasi dapat mempengaruhi waktu eksekusi dari *evolution strategies*.

BAB VI PENUTUP

6.1 Kesimpulan

Berdasarkan hasil penelitian dan pengujian serta pembahasan dapat diambil kesimpulan sebagai berikut:

1. *Evolution Strategies* (ES) dapat diimplementasikan untuk menyelesaikan permasalahan pencarian rute optimum pada permasalahan *delivery order* dengan menggunakan representasi permutasi, metode *elitism*, dan *exchange mutation*.
2. Pencarian rute optimum dengan menggunakan *evolution strategies* dipengaruhi oleh beberapa parameter. Pengaruh parameter ukuran populasi (μ / μ) dan ukuran *offspring* (λ / λ). Semakin besar ukuran populasi (μ / μ) maka hasil yang diperoleh atau nilai *fitness* akan semakin tinggi (besar). Sedangkan pengaruh parameter ukuran *offspring* (λ / λ) menunjukkan perubahan nilai *fitness* tidak stabil (naik dan turun).
3. Pengaruh parameter banyak generasi menunjukkan bahwa semakin banyak generasi yang digunakan, maka nilai *fitness* juga cenderung semakin tinggi. Namun, semakin banyak generasi waktu eksekusi yang diperlukan juga semakin lama. Berdasarkan hasil pengujian, rata-rata nilai *fitness* tertinggi diperoleh pada saat jumlah generasi 50 dan 100.

6.2 Saran

Berdasarkan penelitian yang telah dilakukan yang dapat diberikan untuk pengembangan lebih lanjut adalah sebagai berikut :

1. Pada penelitian selanjutnya dapat menggunakan metode mutasi yang berbeda, serta membandingkan metode mana yang lebih baik digunakan. Contoh metode mutasi yang dapat digunakan adalah *insertion mutation*.
2. Aplikasi perlu dikembangkan dengan menggunakan kondisi yang sebenarnya yaitu dengan menambahkan faktor lain yang mempengaruhi, misalnya kepadatan lalu lintas, lampu lalu lintas, waktu tempuh perjalanan, dan biaya perjalanan.



DAFTAR PUSTAKA

- [BEY-02] Beyer, H-G & Schwefel, H-P 2002, 'Evolution strategies – A comprehensive introduction', *Natural Computing*, vol. 1, no. 1, 2002/03/01, pp. 3-52.
- [CIP-09] Ciptayani, PI, Mahmudy, WF & Widodo, AW 2009, 'Penerapan algoritma genetika untuk kompresi citra fraktal', *Jurnal Ilmu Komputer*, vol. 2, no. 1, April, pp. 1-9.
- [HAN-02] Hannawati, A. Thiang, dan Eleazar. 2002. *Pencarian Rute Optimum Menggunakan Algoritma Genetika*.
- [MAH-13] Mahmudy, WF. 2013. *Algoritma Evolusi*. Modul Kuliah Semester Ganjil 2013-2014. PTIIK. Universitas Brawijaya. Malang.
- [NIS-13] Nissa, Choiron, 2013. *Optimasi Rute Angkutan Kota Malang dengan Penerapan Algoritma Genetika*. SKRIPSI. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya.
- [RIF-11] Rifqi, N., Warih, M., dan Shaufiah. 2011. *Analisis dan Implementasi Klasifikasi Data Mining Menggunakan Jaringan Syaraf Tiruan dan Evolution Strategies*. Bali.
- [SAR-11] Saranta, YM. 2011. *Pencarian Rute Perjalanan Tercepat Kendaraan bermotor Roda Empat Menggunakan Algoritma Genetika*. SKRIPSI. Fakultas MIPA. Universitas Brawijaya.
- [SAT-09] Satriyanto. 2009. *Algoritma Genetika*. <http://lecturer.eepis-its.edu/~kangedi/materi%20kuliah/Kecerdasan%20Buatan/Bab%207%20Algoritma%20Genetika> . Januari 2011
- [SET-03] Setiawan, Kuswara, 2003. *Paradigma Sistem Cerdas*. Bayumedia. Surabaya
- [WID-10] Widodo, AW & Mahmudy, WF 2010, 'Penerapan algoritma genetika pada sistem rekomendasi wisata kuliner', *Kursor*, vol. 5, no. 4, pp. 205-211

LAMPIRAN 1

Tabel Data Jalan Wilayah Pengantaran PHD Rungkut Yakaya Surabaya

| NO | NAMA JALAN | NODE 1 | NODE 2 | PANJANG JALAN |
|----|-------------------------------|--------|--------|---------------|
| 1 | RUNGKUT MAPAN UTARA | 26 | 46 | 0.260 |
| 2 | DELTA PERMAI | 1 | 5 | 0.300 |
| 3 | TENGGILIS MEJOYO | 6 | 19 | 0.230 |
| 4 | JL. KEDUNG BARUK | 1 | 2 | 1.000 |
| 5 | JL. RAYA RUNGKUT | 7 | 8 | 2.480 |
| 6 | WONOREJO RUNGKUT | 2 | 3 | 0.438 |
| 7 | WONOREJO TAMBAK | 3 | 4 | 0.769 |
| 8 | JL. SOEKARNO HATTA | 2 | 10 | 1.110 |
| 9 | RUNGKUT PUSKESMAS | 8 | 9 | 1.190 |
| 10 | RUNGKUT ALANG-ALANG | 9 | 10 | 0.218 |
| 11 | PANDUGO | 10 | 11 | 3.330 |
| 12 | PANDUGO GG.I | 11 | 13 | 0.363 |
| 13 | MEDOKAN AYU | 28 | 29 | 0.523 |
| 14 | JL. RAYA TENGGILIS | 6 | 20 | 0.350 |
| 15 | KENDANGSARI | 20 | 21 | 0.543 |
| 16 | KENDANGSARI (2) | 19 | 20 | 1.140 |
| 17 | RUNGKUT INDUSTRI I | 22 | 18 | 1.540 |
| 18 | JL. KUTISARI | 21 | 22 | 0.971 |
| 19 | KUTISARI UTARA | 21 | 42 | 0.630 |
| 20 | KUTISARI SELATAN | 42 | 43 | 0.452 |
| 21 | BERBEK INDUSTRI III | 44 | 45 | 0.879 |
| 22 | RUNGKUT INDUSTRI III | 22 | 41 | 0.363 |
| 23 | RUNGKUT INDUSTRI III (2) | 41 | 44 | 0.248 |
| 24 | RUNGKUT INDUSTRI III (3) | 44 | 25 | 1.370 |
| 25 | RUNGKUT INDUSTRI V | 24 | 40 | 0.568 |
| 26 | RUNGKUT INDUSTRI V (2) | 44 | 40 | 0.306 |
| 27 | RUNGKUT INDUSTRI IV | 41 | 40 | 0.235 |
| 28 | JL. RAYA RUNGKUT INDUSTRI | 22 | 24 | 0.887 |
| 29 | JL. RAYA RUNGKUT INDUSTRI (2) | 24 | 25 | 0.425 |
| 30 | RUNGKUT INDUSTRI II | 18 | 17 | 0.487 |
| 31 | RUNGKUT INDUSTRI II (2) | 17 | 25 | 1.010 |

| NO | NAMA JALAN | NODE 1 | NODE 2 | PANJANG JALAN |
|----|---------------------------|--------|--------|---------------|
| 32 | RUNGKUT INDUSTRI KIDUL | 25 | 26 | 0.567 |
| 33 | RUNGKUT TENGAH | 26 | 37 | 0.600 |
| 34 | RUNGKUT MENANGGAL | 37 | 38 | 0.430 |
| 35 | ABDUL KARIM | 38 | 36 | 0.769 |
| 36 | JL. RUNGKUT KIDUL | 16 | 26 | 0.369 |
| 37 | JL. RUNGKUT LOR | 8 | 16 | 0.491 |
| 38 | WADUNGSARI | 38 | 39 | 0.762 |
| 39 | AMIR MACHMUD | 36 | 35 | 0.292 |
| 40 | RUNGKUT ASRI UTARA XIII | 9 | 14 | 0.932 |
| 41 | RUNGKUT ASRI XIII | 14 | 15 | 0.948 |
| 42 | RUNGKUT ASRI TENGAH | 15 | 27 | 0.428 |
| 43 | JL. RUNGKUT MADYA | 27 | 28 | 2.150 |
| 44 | JL. MEDOKAN AYU | 28 | 29 | 1.020 |
| 45 | JL. SOEKARNO HATTA (2) | 10 | 28 | 2.290 |
| 46 | MEDOKAN SAWAH | 29 | 30 | 2.000 |
| 47 | GUNUNG ANYAR JAYA | 29 | 34 | 1.050 |
| 48 | WIGUNA TIMUR | 29 | 32 | 0.473 |
| 49 | GUNUNG ANYAR TIMUR | 35 | 34 | 0.765 |
| 50 | GUNUNG ANYAR TAMBAK | 34 | 33 | 0.543 |
| 51 | PENJARINGAN ASRI IX | 13 | 28 | 0.269 |
| 52 | JL. MEDOKAN KAMPUNG | 13 | 29 | 0.321 |
| 53 | GUNUNG ANYAR TAMBAK I-III | 32 | 31 | 0.745 |
| 54 | GUNUNG ANYAR ASRI | 32 | 33 | 0.765 |
| 55 | TENGGILIS MULYO | 6 | 19 | 0.621 |
| 56 | RUNGKUT INDUSTRI VI | 18 | 23 | 0.326 |
| 57 | RUNGKUT INDUSTRI VI (2) | 23 | 24 | 0.385 |
| 58 | RAYA TENGGILIS | 6 | 8 | 0.349 |
| 59 | KEDUNG ASEM | 8 | 10 | 0.609 |
| 60 | JL. PANJANG JIWO | 1 | 7 | 1.340 |
| 61 | JL. KH. ZAMHURI | 26 | 27 | 0.270 |
| 62 | RUNGKUT MAPAN RAYA | 27 | 46 | 0.173 |
| 63 | RUNGKUT PERMAI | 26 | 47 | 0.280 |
| 64 | RUNGKUT MAPAN BARAT | 47 | 46 | 0.573 |
| 65 | RUNGKUT MAPAN SELATAN | 47 | 48 | 0.510 |

| NO | NAMA JALAN | NODE 1 | NODE 2 | PANJANG JALAN |
|----|------------------------|--------|--------|---------------|
| 66 | RUNGKUT BARATA | 38 | 48 | 0.462 |
| 67 | RUNGKUT MAPAN RAYA (2) | 46 | 48 | 0.187 |
| 68 | GUNUNG ANYAR LOR | 28 | 49 | 1.340 |
| 69 | GUNUNG ANYAR TENGAH | 49 | 35 | 0.507 |



LAMPIRAN 2

Jarak Jalan yang Terhubung

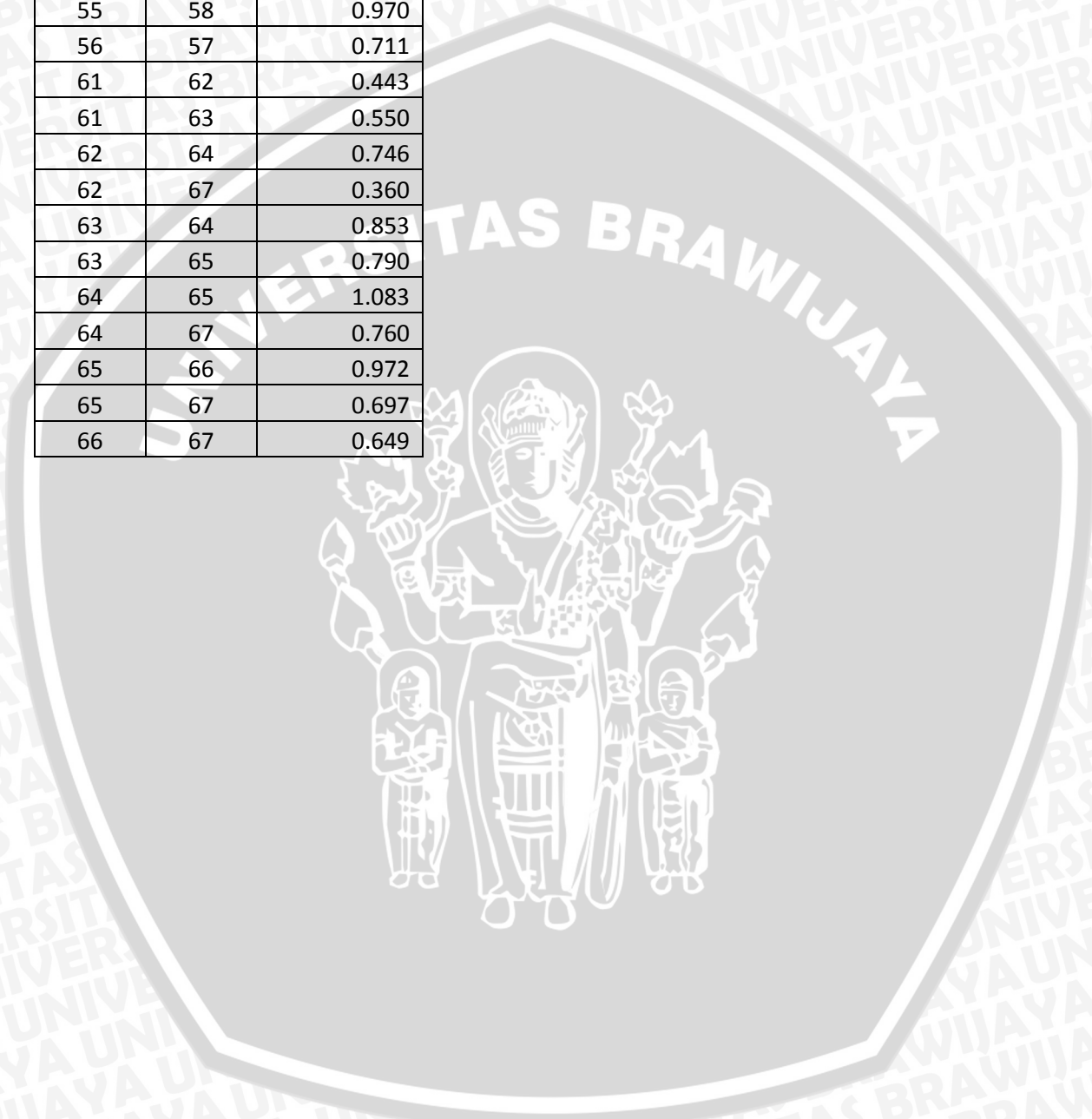
| Jalan 1 | Jalan 2 | Total Jarak |
|---------|---------|-------------|
| 1 | 32 | 0.827 |
| 1 | 33 | 0.860 |
| 1 | 36 | 0.629 |
| 1 | 43 | 2.410 |
| 1 | 61 | 0.530 |
| 1 | 62 | 0.433 |
| 1 | 63 | 0.540 |
| 1 | 64 | 0.833 |
| 1 | 67 | 0.477 |
| 2 | 3 | 0.530 |
| 2 | 4 | 1.300 |
| 2 | 60 | 1.640 |
| 3 | 14 | 0.580 |
| 3 | 55 | 0.581 |
| 3 | 58 | 0.579 |
| 4 | 5 | 3.480 |
| 4 | 8 | 2.110 |
| 4 | 60 | 2.340 |
| 5 | 9 | 3.670 |
| 5 | 37 | 2.971 |
| 5 | 58 | 2.829 |
| 5 | 59 | 3.089 |
| 5 | 60 | 3.820 |
| 6 | 7 | 1.207 |
| 6 | 8 | 1.548 |
| 8 | 10 | 1.328 |
| 8 | 11 | 4.440 |
| 8 | 45 | 3.400 |
| 8 | 59 | 1.719 |
| 9 | 10 | 1.408 |
| 9 | 37 | 1.681 |
| 9 | 40 | 2.122 |
| 9 | 58 | 1.539 |

| Jalan 1 | Jalan 2 | Total Jarak |
|---------|---------|-------------|
| 9 | 59 | 1.799 |
| 10 | 11 | 3.548 |
| 10 | 40 | 1.115 |
| 10 | 45 | 2.508 |
| 10 | 59 | 0.827 |
| 11 | 12 | 3.693 |
| 11 | 45 | 5.620 |
| 11 | 59 | 3.939 |
| 12 | 13 | 0.886 |
| 12 | 51 | 0.632 |
| 12 | 52 | 0.684 |
| 13 | 51 | 0.792 |
| 13 | 52 | 0.844 |
| 14 | 15 | 0.893 |
| 14 | 16 | 1.490 |
| 14 | 55 | 0.971 |
| 15 | 16 | 1.683 |
| 15 | 18 | 1.514 |
| 15 | 19 | 1.173 |
| 16 | 55 | 1.761 |
| 17 | 18 | 2.511 |
| 17 | 28 | 2.427 |
| 17 | 30 | 2.027 |
| 17 | 56 | 1.866 |
| 18 | 19 | 1.601 |
| 18 | 22 | 1.334 |
| 18 | 28 | 1.858 |
| 19 | 20 | 1.082 |
| 21 | 23 | 1.127 |
| 21 | 24 | 2.249 |
| 21 | 26 | 1.185 |
| 22 | 27 | 0.598 |
| 22 | 28 | 1.250 |

| Jalan 1 | Jalan 2 | Total Jarak |
|---------|---------|-------------|
| 23 | 24 | 1.618 |
| 23 | 26 | 0.554 |
| 23 | 27 | 0.438 |
| 24 | 26 | 1.676 |
| 24 | 29 | 1.795 |
| 24 | 31 | 2.380 |
| 24 | 32 | 1.937 |
| 25 | 26 | 0.874 |
| 25 | 27 | 0.803 |
| 25 | 28 | 1.455 |
| 25 | 29 | 0.993 |
| 25 | 57 | 0.953 |
| 26 | 27 | 0.541 |
| 28 | 29 | 1.312 |
| 28 | 57 | 1.272 |
| 29 | 31 | 1.435 |
| 29 | 32 | 0.992 |
| 29 | 57 | 0.810 |
| 30 | 31 | 1.497 |
| 30 | 56 | 0.813 |
| 31 | 32 | 1.577 |
| 32 | 33 | 1.167 |
| 32 | 36 | 0.936 |
| 32 | 61 | 1.577 |
| 32 | 63 | 0.857 |
| 33 | 34 | 1.030 |
| 33 | 36 | 0.969 |
| 33 | 61 | 0.870 |
| 33 | 63 | 0.880 |
| 34 | 35 | 1.199 |
| 34 | 38 | 1.192 |
| 34 | 66 | 0.892 |
| 35 | 38 | 1.531 |
| 35 | 39 | 1.061 |
| 35 | 66 | 1.231 |
| 36 | 37 | 0.860 |
| 36 | 61 | 0.639 |
| 36 | 63 | 0.649 |

| Jalan 1 | Jalan 2 | Total Jarak |
|---------|---------|-------------|
| 37 | 58 | 0.840 |
| 37 | 59 | 1.100 |
| 38 | 66 | 1.224 |
| 39 | 49 | 1.057 |
| 39 | 69 | 0.799 |
| 40 | 41 | 1.880 |
| 41 | 42 | 1.376 |
| 42 | 43 | 2.578 |
| 42 | 61 | 0.698 |
| 42 | 62 | 0.601 |
| 43 | 44 | 3.170 |
| 43 | 45 | 4.440 |
| 43 | 51 | 2.419 |
| 43 | 61 | 2.420 |
| 43 | 62 | 2.323 |
| 43 | 68 | 3.490 |
| 44 | 45 | 3.310 |
| 44 | 46 | 3.020 |
| 44 | 51 | 1.289 |
| 44 | 52 | 1.341 |
| 44 | 68 | 2.360 |
| 45 | 51 | 2.559 |
| 45 | 59 | 2.899 |
| 45 | 68 | 3.630 |
| 46 | 47 | 3.050 |
| 46 | 48 | 2.473 |
| 46 | 52 | 2.321 |
| 47 | 48 | 1.523 |
| 47 | 49 | 1.815 |
| 47 | 50 | 1.593 |
| 47 | 52 | 1.371 |
| 48 | 52 | 0.794 |
| 48 | 53 | 1.218 |
| 48 | 54 | 1.238 |
| 49 | 50 | 1.308 |
| 49 | 69 | 1.272 |
| 50 | 54 | 1.308 |
| 51 | 52 | 0.590 |

| Jalan 1 | Jalan 2 | Total Jarak |
|---------|---------|-------------|
| 51 | 68 | 1.609 |
| 53 | 54 | 1.510 |
| 55 | 58 | 0.970 |
| 56 | 57 | 0.711 |
| 61 | 62 | 0.443 |
| 61 | 63 | 0.550 |
| 62 | 64 | 0.746 |
| 62 | 67 | 0.360 |
| 63 | 64 | 0.853 |
| 63 | 65 | 0.790 |
| 64 | 65 | 1.083 |
| 64 | 67 | 0.760 |
| 65 | 66 | 0.972 |
| 65 | 67 | 0.697 |
| 66 | 67 | 0.649 |



LAMPIRAN 3

Peta Wilayah Pizza Hut Delivery Rungkut Yakaya Surabaya

