

**PENERAPAN ALGORITMA GENETIKA UNTUK OPTIMASI RUTE
TERPENDEK SUPLIER BARANG INDOMARET
DI WILAYAH TEGAL TIMUR**

Tesis S -2

untuk memenuhi sebagian persyaratan mencapai derajat sarjana S- 2
Program Magister Teknik Elektro



Diajukan oleh

IRAWAN PUDJA HARDJANA

20602000006

**PROGRAM MAGISTER TEKNIK ELEKTRO
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2024

HALAMAN PENGESAHAN PEMBIMBING

Tesis S-2

**PENERAPAN ALGORITMA GENETIKA UNTUK OPTIMASI RUTE
TERPENDEK SUPLIER BARANG INDOMARET
DI WILAYAH TEGAL TIMUR**

yang diajukan oleh:

Irawan Pudja Hardjana

20602000006

Telah diperiksa dan disetujui,
Pada tanggal 04 September 2024

Pembimbing 1

Pembimbing 2

Imam Muh. Ibnu Subroto, S.T., M.Sc., Ph. D.

NIDN: 061303730001

Ir. Suryani Alifah, M.T., Ph.D.

NIDN: 0625036901

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Teknik

Tanggal 06 Desember 2024

Rektor Program Studi Magister Teknik Elektro

Prof. Dr. H. Arttini Dwi Prasetyowati, M.Si.

NIDN: 0620026501

LEMBAR PENGESAHAN

**PENERAPAN ALGORITMA GENETIKA UNTUK OPTIMASI
RUTE TERPENDEK SUPLIER BARANG INDOMARET
DI WILAYAH TEGAL TIMUR**

yang dipersiapkan dan disusun oleh
Irawan Pudja Hardjana
20602000006

telah dipertahankan didepan Dewan Penguji
Pada tanggal: September 2024

Susunan Dewan Penguji

Ketua penguji


Prof. Dr. Ir. Muhammad Haddin, MT
NIDN. 0618066301

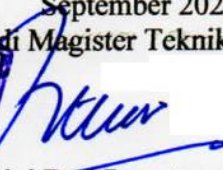
Penguji 1


Ir. Suryani Alifah, M.T., Ph.D.
NIDN. 0625036901

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Teknik

Tanggal: September 2024

Ketua Program Studi Magister Teknik Elektro


Prof. Dr. Ir. Sri Artini Dwi Prasetyowati, MSi
NIDN : 0620026501

PERNYATAAN KEASLIAN TESIS

Saya yang bertanda tangan dibawah ini:

Nama : Irawan Pudja Hardjana
NIM : 20602000006
Program Studi : Magister Teknik Elektro
Fakultas : Teknologi Industri

Dengan ini saya menyatakan bahwa Tesis yang diajukan kepada Program Studi Magister Teknik Elektro dengan Judul:

“Penerapan Algoritma Genetika untuk Optimasi Rute Terpendek Suplier Barang Indomaret Di Wilayah Tegal Timur”

Adalah hasil karya sendiri, judul tersebut belum pernah diajukan untuk memperoleh gelar Magister Teknik pada Program Studi Magister Teknik Elektro Universitas Islam Sultan Agung (UNISSULA) ataupun pada universitas lain serta belum pernah ditulis maupun diterbitkan oleh orang lain kecuali secara tertulis diacu, disitasi dan ditunjuk dalam daftar pustaka. Tesis ini adalah milik saya, segala bentuk kesalahan dan kekeliruan dalam Tesis ini adalah tanggung jawab saya.

Semarang, September 2024

Penulis



Irawan Pudja Hardjana
20602000006

KATA PENGANTAR

Puji syukur kehadirat Allah Subhanahu wa ta'ala. Pada hari ini telah tuntas sudah tugas menyusun laporan tesis. Tesis ini adalah untuk memenuhi sebagian persyaratan mencapai derajat sarjana S- 2 pada Program Magister Teknik Elektro Universitas Islam Sultan Agung Semarang.

Ucapan terimakasih penulis sampaikan kepada:

1. Bapak Prof. Dr. Gunarto, SH, MH selaku Rektor Universitas Islam Sultan Agung Semarang.
2. Ibu Dr. Novi Marlyana, ST, MT selaku Dekan Fakultas Teknologi Industri Universitas Islam Sultan Agung Semarang.
3. Ibu Prof. Dr. Ir. Sri Artini Dwi Prasetyowati selaku Ketua Jurusan Program Magister Teknik Elektro Universitas Islam Sultan Agung Semarang.
4. Bapak Dr. Eka Nuryanto BS, ST., MT selaku Sekretaris Program Studi Magister Teknik Elektro Fakultas Teknologi Industri Universitas Sultan Agung Semarang
5. Bapak Imam Muh. Ibnu Subroto, S.T., M.Sc., Ph. D. selaku Dosen Pembimbing I.
6. Ibu Ir. Suryani Alifah, M.T., Ph. D. selaku Dosen Pembimbing II.
7. Dan pihak-pihak lain yang tidak dapat penulis sebutkan satu persatu yang ikut membantu hingga terselesaikannya penulisan ini.

Kekurangan sudah tentu ada. Karena itu kritik dan saran yang bersifat membangun sangat penulis harapkan karena setidaknya penulis ingin berkarya lebih baik lagi di masa-masa yang akan datang.

Semoga karya yang jauh dari sempurna ini dapat bermanfaat bagi pembaca pada umumnya dan penulis pada khususnya.

Semarang, September 2024

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN PENGUJI.....	iii
PERNYATAAN KEASLIAN TESIS	iv
KATA PENGANTAR	v
DAFTAR ISI.....	viii
DAFTAR GAMBAR	viii
DAFTAR LAMPIRAN.....	ix
ABSTRAK	x
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	5
1.5 Manfaat Penelitian	5
1.6 Sistematika penulisan.....	5
BAB II KAJIAN PUSTAKA DAN LANDASAN TEORI	7
2.1 Penelaahan Studi / Kajian Pustaka.....	7
2.2 Landasan Teori.....	9
2.2.1 <i>Travelling</i> Salesman Problem (TSP).....	9
2.2.2 Algoritma Genetika.....	10
BAB III METODE PENELITIAN.....	12
3.1 Tahap-tahap penelitian.....	12
3.2 Pembuatan model jalur TSP.....	13
3.3 Perancangan algoritma genetika	13
3.3.1 Pembentukan populasi awal.....	13

3.3.2	Penentuan nilai <i>fitness</i>	15
3.3.3	Seleksi individu (penentuan parents)	16
3.3.4	<i>Crossover</i> / pindah silang.....	19
3.3.5	Mutasi.....	21
3.4	Simulasi.....	25
3.4.1	Algoritma	25
3.4.2	<i>Flowchart</i>	27
3.4.3	Coding.....	31
3.5	Kinerja.....	37
BAB IV HASIL DAN PEMBAHASAN		39
4.1	Hasil	39
4.2	Analisis	40
BAB V KESIMPULAN DAN SARAN.....		41
5.1	Kesimpulan.....	41
5.2	Saran	41
DAFTAR PUSTAKA.....		42
LAMPIRAN.....		43



DAFTAR GAMBAR

Gambar 3.1. Membuat populasi awal	27
Gambar 3.2. Menentukan nilai fitness	28
Gambar 3.3. Membuat probabilitas seleksi	28
Gambar 3.4. Membuat komulatif probabilitas	28
Gambar 3.5. Membuat <i>Roulette well</i>	29
Gambar 3.6. <i>Crossover</i>	30
Gambar 3.7. Mencari panjang total gen	30
Gambar 3.8. Menentukan jumlah gen yang akan dimutasi	30
Gambar 3.9. Proses mutasi	31



DAFTAR LAMPIRAN

Lampiran 1.	Peta Lokasi Indomaret Wilayah Tegal Timur	431
Lampiran 2.	Rute awal titik kota	442
Lampiran 3.	Rute terpendek yang didapatkan	442
Lampiran 4.	Siklus Algoritma Genetika	442
Lampiran 5.	Foto Citra Satelit Peta Indomaret Tegal Timur	453
Lampiran 6.	Proses Algoritma Genetika	453
Lampiran 7.	Representasi <i>Roulette wheel</i> pada garis bilangan.....	464
Lampiran 8.	Jarak 1	464
Lampiran 9.	Jarak 2	464
Lampiran 10.	Jarak 3	475
Lampiran 11.	Jarak 4	475
Lampiran 12.	Jarak 5	486
Lampiran 13.	Jarak 6	486
Lampiran 14.	Jarak 7	49
Lampiran 15.	Jarak 8	497
Lampiran 16.	Jarak 9	508
Lampiran 17.	Jarak 10	508
Lampiran 18.	Jarak 11	519
Lampiran 19.	Jarak 12	519
Lampiran 20.	Jarak 13	50
Lampiran 21.	Jarak 14	51
Lampiran 22.	Jarak 15	531
Lampiran 23.	Jarak 16	531
Lampiran 24.	Jarak 17	542
Lampiran 25.	Jarak 18	542
Lampiran 26.	Jarak 19	553
Lampiran 27.	Jarak 20	553
Lampiran 28.	Jarak 21	564
Lampiran 29.	Jarak 22	564
Lampiran 30.	Jarak 23	575
Lampiran 31.	Jarak 24	575
Lampiran 32.	Jarak 25	586
Lampiran 33.	Jarak 26	586
Lampiran 34.	Jarak 27	597
Lampiran 35.	Jarak 28	597
Lampiran 36.	Jarak 29	608
Lampiran 37.	Jarak 30	608
Lampiran 38.	Jarak 31	619
Lampiran 39.	Kode tempat	60
Lampiran 40.	Jarak antar node.....	60

ABSTRAK

Indomaret merupakan jaringan minimarket yang menyediakan berbagai kebutuhan harian, dan konsistensi dalam ketersediaan barang sangat penting untuk menjaga kepuasan pelanggan. Efisiensi distribusi barang dari supplier ke gerai-gerai Indomaret di wilayah Kecamatan Tegal Timur menjadi tantangan tersendiri karena berbagai faktor, seperti jarak antar gerai, kondisi jalan, serta biaya pengiriman. Penelitian ini bertujuan untuk mengoptimalkan rute pengiriman barang dari supplier ke 8 gerai Indomaret di wilayah Tegal Timur menggunakan Algoritma Genetika dalam pemodelan Travelling Salesman Problem (TSP). Algoritma Genetika dipilih karena kemampuannya dalam menyelesaikan masalah optimasi yang kompleks secara efisien.

Penelitian ini menggunakan data lokasi gerai Indomaret yang diperoleh dari peta satelit, dengan pengukuran jarak antar lokasi melalui Google Maps. Proses optimasi melibatkan seleksi, crossover, mutasi, dan evaluasi fitness untuk menemukan rute pengiriman barang dengan jarak tempuh terpendek. Hasil penelitian menunjukkan bahwa penerapan Algoritma Genetika berhasil mengoptimalkan rute pengiriman dengan jarak terbaik sebesar 9.700 meter dan nilai fitness 0,0001031. Simulasi yang dilakukan dengan MATLAB menunjukkan bahwa algoritma ini dapat mengurangi biaya operasional dan waktu tempuh secara signifikan dibandingkan metode distribusi konvensional.

Penelitian ini memberikan kontribusi dalam memperkaya literatur mengenai penerapan Algoritma Genetika untuk optimasi rute distribusi, serta memberikan rekomendasi praktis bagi supplier dalam meningkatkan efisiensi logistik. Dengan demikian, penerapan teknologi optimasi berbasis evolusi seperti Algoritma Genetika dapat membantu meningkatkan kinerja distribusi dan kepuasan pelanggan..

Kata kunci: Algoritma Genetika, Travelling Salesman Problem, Optimasi Rute, Distribusi Barang, Indomaret

BAB I

PENDAHULUAN

1.1 Latar Belakang

Indomaret merupakan organisasi minimarket yang berjalan di tanah air. Aspek bisnis primernya mencakup pengadaan beraneka macam komoditas keperluan sehari-hari seperti makanan, minuman, produk kebersihan, perlengkapan rumah tangga, dan barang konsumsi lainnya disamping layanan pembayaran seperti: pembayaran tagihan listrik, air, telepon, internet, dan berbagai tagihan lainnya. Indomaret juga menawarkan pembelian tiket, penyetoran angsuran, dan *top-up* pulsa telepon. Maka organisasi minimarket ini memberikan kerjasama bersama produsen domestik untuk penyediaan komoditas tersebut. Kerjasama ini bisa membuahkan keuntungan bagi segenap pihak yang berperan, termasuk produsen yang memperoleh akses penjualan yang lebih mendalam, ritel yang memegang variasi produk yang lebih beraneka ragam, dan pembeli yang memperoleh akses ke produk bermutu dengan harga bersaing. [1]

Keteraturan dalam ketersediaan produk mendukung penilaian dan nama baik Indomaret sebagai toko yang handal. Maka pengiriman produk yang tepat waktu oleh para produsen sebagai pemasok ke Indomaret sangat berarti untuk meyakinkan bahwa ketersediaan barang di Indomaret selalu siap. Jika terjadi keterlambatan pasokan, stok bisa kosong, yang berakibat ketidakpuasan konsumen dan berpeluang putus penjualan. Jadwal pendistribusian yang stabil membantu dalam penjadwalan operasional yang semakin ekonomis, termasuk penjadwalan tenaga kerja dan manajemen inventaris. [1]

Para produsen bisa mengalami berbagai macam hambatan dan tantangan dalam mekanisme distribusi produk [2]. Tantangan ini disebabkan oleh minimnya *planning* yang tepat dalam penetapan rute distribusi, yang mengimbas pada operasional yang tidak efisien. Selain itu, banyak produsen sebagai suplier yang belum seutuhnya mengambil teknologi modern yang menopang proses distribusi produk, seperti sistem pengelolaan stok dan penelusuran pengiriman secara real-time.

Pengiriman yang tidak efisien dapat berakibat kenaikan anggaran logistik, keterlambatan produk sampai di toko, sehingga dapat menurunkan tingkat kepuasan konsumen. Ketersediaan barang yang tidak *aksetabel* di gerai Indomaret juga dapat mengurangi kemampuan bersaing ritel tersebut.

Untuk menanggulangi kasus ini, dibutuhkan solusi optimasi rute pengiriman yang lebih efektif. Salah satu pendekatan yang mungkin adalah dengan menerapkan Travelling Salesman Problem (TSP), yaitu model matematis yang bertujuan mendapatkan rute terpendek yang mengunjungi setiap titik hanya sekali dan kembali ke titik awal. Dalam kerangka ini, rute pengiriman ke titik-titik lokasi Indomaret di Tegal Timur perlu dioptimalkan untuk memotong waktu tempuh dan anggaran logistik. Pemanfaatan teknologi-teknologi ini dapat membantu supplier dalam manajemen proses pengiriman barang ke Indomaret yang lebih efisien, menurunkan biaya operasional, dan menambah kepuasan pelanggan [3].

Ada 8 Indomaret di wilayah kecamatan Tegal Timur, yaitu :

1. Indomaret Veteran 31
2. Indomaret Serayu
3. Indomaret Perintis Kemerdekaan 2
4. Indomaret Pancasila
5. Indomaret Alun-alun Tegal
6. Indomaret Perintis Kemerdekaan
7. Indomaret um Arjuna
8. Indomaret Sultan Agung

Terletak strategis di pusat kota. Dengan dukungan produsen lokal, toko ini juga memasok produk-produk khas daerah yang menjadi unggulan masyarakat setempat. Sasaran utama supplier di Indomaret wilayah Tegal Timur ialah meyakinkan ketersediaan produk-produk yang diminati bagi masyarakat setempat, dengan melihat kualitas dan stabilitas pasokan. Melalui menjaga ikatan yang baik dengan petani dan produsen lokal, supplier berusaha untuk memastikan bahwa Indomaret di Tegal Timur ini selalu memiliki ketersediaan yang memadai, dan

dapat menjamin pelayanan terbaik kepada pelanggannya dan mendukung terjaganya ekonomi lokal.

Peta lokasi Indomaret di wilayah Tegal Timur memperlihatkan sebaran toko-toko ritel yang strategis ini di banyak titik penting di wilayah tersebut dan memberikan potret visual utama untuk memahami distribusi jaringan ritel dalam kerangka penkajian ini, seperti nampak pada gambar 1.1. Peta tersebut juga menampakkan jarak yang relatif dekat antar Indomaret, memastikan kenyamanan dan ketersediaan barang untuk masyarakat sekitar. Langkah suplier menyertakan pengadaan langsung dari sumber lokal untuk memotong biaya logistik [4]. Dalam hal ini, dibutuhkan solusi optimasi rute pengiriman yang lebih efisien. Salah satu pendekatan yang mungkin adalah menerapkan Travelling Salesman Problem (TSP), yaitu model matematika yang bertujuan mendapatkan rute terpendek yang mengunjungi setiap titik sekali dan kembali ke titik awal. Dalam hal ini, rute-rute pengiriman ke setiap gerai Indomaret di Tegal Timur perlu dioptimalkan untuk memotong waktu tempuh dan biaya logistik.

Problema Travelling Salesman Problem (TSP) menjadi penting dalam kajian ini, di mana sasaran utamanya adalah mendapatkan rute terpendek yang mengunjungi setiap gerai Indomaret sekali dan kembali ke titik awal. Algoritma genetika, merupakan salah satu metode optimasi yang berbasis evolusi, merekomendasikan solusi potensial untuk TSP dengan efisien. Sekarang ini, pemakaian Algoritma Genetika (AG) menjadi salah satu tren solusi yang banyak digunakan untuk penyelesaian TSP. Gabungan dari TSP dan algoritma genetika diharapkan dapat menghasilkan solusi terbaik bagi suplier dalam pendistribusian barang ke Indomaret-Indomaret di wilayah Tegal Timur.

Pada penelitian ini akan dibahas penerapan algoritma genetika dengan model Travelling Salesman Problem (TSP) untuk optimasi sistem pengiriman barang dari suplier ke Indomaret di wilayah Kecamatan Tegal Timur. Mengenal rute-rute pengiriman barang yang paling praktis ke gerai-gerai Indomaret di Tegal Timur. Mengevaluasi kinerja algoritma genetika untuk memecahkan masalah TSP dibandingkan dengan metode konvensional. Merekomendasikan langkah pendistribusian barang yang lebih efektif bagi suplier. Penelitian ini juga akan

menanggapi pertanyaan bagaimana penerapan algoritma genetika dalam menyelesaikan TSP untuk masalah distribusi barang ke Indomaret di Tegal Timur. Seberapa efektif algoritma genetika dalam menghemat waktu dan biaya pengiriman dibandingkan metode distribusi yang digunakan saat ini. Faktor-faktor apa sajakah yang mempengaruhi keberhasilan algoritma genetika pada optimasi rute pengiriman.

Hasil penelitian ini juga diharapkan bisa memberi manfaat teoritis dan praktis, diantaranya:

1. penelitian ini secara teoritis memperkaya literatur mengenai penerapan algoritma genetika untuk optimasi rute distribusi.
2. Penelitian ini secara praktis diharapkan dapat menolong supplier untuk menaikkan efisiensi distribusi barang, sehingga dapat mengurangi biaya operasional dan meningkatkan kepuasan pelanggan.

1.2 Perumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan dapat yang dirumuskan sebagai berikut :

- 1) Bagaimana menerapkan Algoritma Genetika untuk optimasi rute terpendek pada pencarian rute terpendek supplier menuju beberapa lokasi Indomaret yang berada di wilayah kecamatan Tegal Timur
- 2) Bagaimana mensimulasikan Algoritma Genetika untuk optimasi rute terpendek pada pencarian rute terpendek supplier menuju beberapa lokasi Indomaret yang berada di wilayah kecamatan tegal timur
- 3) Bagaimana kinerja optimasi pencarian jalur terpendek pada pencarian rute terpendek supplier menuju beberapa lokasi Indomaret yang berada di wilayah kecamatan Tegal Timur.

1.3 Batasan Masalah

1. Lokasi target merupakan lokasi indomaret di wilayah kecamatan tegal timur
2. Kriteria jalur terpendeknya adalah menggunakan ukuran jarak tempuh

3. Implementasinya masih berupa simulasi menggunakan bahasa pemrograman Matlab

1.4 Tujuan Penelitian

1. Menerapkan Algoritma Genetika untuk optimasi rute terpendek pada pencarian rute terpendek suplier menuju beberapa lokasi Indomaret yang berada di wilayah kecamatan Tegal Timur
2. Mensimulasikan Algoritma Genetika untuk optimasi rute terpendek pada pencarian rute terpendek suplier menuju beberapa lokasi Indomaret yang berada di wilayah kecamatan Tegal Timur
3. Mengetahui kinerja optimasi pencarian jalur terpendek pada pencarian rute terpendek suplier menuju beberapa lokasi Indomaret yang berada di wilayah kecamatan Tegal Timur menggunakan algoritma genetika.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini:

1. Mempermudah penyelesaian masalah optimasi
2. Mempercepat tercapainya solusi optimum.

1.6 Sistematika penulisan

BAB I Pendahuluan

Menjelaskan tentang latar belakang, perumusan masalah, batasan masalah, keaslian penelitian, tujuan dan manfaat penelitian, dan metode penelitian.

BAB II tinjauan Pustaka

Menjelaskan tentang penelaahan studi / kajian pustaka serta dasar teori yang berkaitan dengan tesis ini.

BAB III Metode Penelitian

Menjelaskan tentang metode dan tahapan penelitian dari penerapan algoritma genetika dengan merepresentasikan pada jarak tempuh antar node.

BAB IV Pembahasan

Menjelaskan tentang percobaan / eksperimen simulasi penerapan algoritma genetika dengan model TSP untuk sebuah lintasan, pencatatan hasil pengujian.

BAB V Kesimpulan dan Saran

Menjelaskan tentang kesimpulan yang bisa diambil dari pembahasan tesis serta saran-saran pengembangan.



BAB II

KAJIAN PUSTAKA DAN LANDASAN TEORI

2.1 Penelaahan Studi / Kajian Pustaka

Penelitian ini mengacu pada beberapa penelitian yang pernah dilakukan sebelumnya.

Peneliti	Metode/Model	Tujuan Penelitian	Parameter yang Digunakan	Hasil Penelitian
Rizki Rino Pratama, Rintho Rante Rerung, Aditya Erfina (2020)	Algoritma Genetika	Mengoptimalkan ukuran generasi dan populasi untuk TSP	Ukuran populasi: 50, ukuran generasi: kelipatan 500 (500–3000), Probabilitas crossover dan mutasi: 0,2	Generasi ke-2000 optimal, fitness stabil pada generasi 2000–3000
Rachmad Sanuri, Herdiesel Santoso (2019)	Algoritma Genetika, TSP with Time Windows (TSP-TW)	Penjadwalan rute kunjungan multi-destinasi	Ukuran populasi: 150, Probabilitas crossover: 0,3, Probabilitas mutasi: 0,7	Fitness tertinggi dicapai pada generasi ke-127, kombinasi probabilitas optimal
Fitriana Yuli Saptaningtyas (2012)	Algoritma Genetika, Multi Travelling Salesman Problem (MTSP)	Optimasi rute distribusi koran	Order crossover, swapping mutation, 8 looper, 160 pelanggan	Jarak terpendek 144,16 km, iterasi ke-98
Ari Yulianto Nugroho, Amin Suyitno, Riza Arifudin (2016)	Algoritma Genetika vs. Branch and Bound	Perbandingan algoritma untuk TSP	Rute distribusi PT. JNE Semarang	Algoritma Genetika menghasilkan panjang sirkuit lebih pendek

				daripada Branch and Bound
Kus Sri Wahyuni (2011)	Algoritma Genetika, TSP	Menentukan rute distribusi optimal	Rute distribusi, sekali melewati agen, kembali ke gudang	Jarak tempuh berkurang dari 588,2 km menjadi 437,25 km, pengurangan 25,66%
Syarifah Elza Ramadhania (2020)	Algoritma Genetika dan Tabu Search (Hibrid)	Penyelesaian TSP dengan hibridasi algoritma	TSP simetris, populasi berbeda-beda	Hibridasi memberikan hasil lebih optimal dibandingkan Algoritma Genetika, meski membutuhkan waktu lebih lam

Tabel di atas menganalisis berbagai penelitian terdahulu. Penelitian ini membedakan diri dengan penerapan teknik mutasi dan crossover yang lebih spesifik dan penyesuaian parameter algoritma yang lebih mendalam. Adapun parameter yang digunakan: crossover probability (p_c) sebesar 15%, penghindaran Gen Duplikat dengan menggunakan file `fix_duplicate_genes` untuk memperbaiki duplikasi gen. Penelitian ini juga mengaplikasikan algoritma genetika dalam kerangka distribusi barang dari suplier ke lokasi Indomaret di Kecamatan Tegal Timur, yang merekomendasikan sudut pandang baru dalam optimasi rute distribusi. Temuan ini memperlihatkan peningkatan efisiensi komputasi dan tingkat solusi dibandingkan dengan hasil penelitian sebelumnya.

2.2 Landasan Teori

2.2.1 *Travelling Salesman Problem (TSP)*

Pada tahun 1800 William Rowan Hamilton dan Thomas Pennington, matematikawan Irlandia dan Inggris mengangkat permasalahan matematika yang menjadi pendahulu dari Traveling Salesman Problem (TSP). TSP semula muncul dalam bentuk game dengan nama Icosian. Di dalam permainan ini, para pemain harus mengerjakan permainan yang menghubungkan 20 titik jalan cukup dengan beberapa rute saja. Maka pada saat itu teori tersebut dinamakan dengan teori lingkaran Hamilton pada matematika diskrit [5].

Bentuk umum TSP lahir di tahun 1930 yaitu pada saat dipelajari pertama kali oleh Karl Menger di Vienna, seorang matematikawan dari Harvard University. Permasalahan kemudian ini dipublikasikan oleh Hassler Whitney dan Merrill Flood di Princeton. Studi lebih lanjut mengenai interaksi antara Menger dan Whitney dan perkembangannya berdasarkan TSP sebagai topik pembahasan terdapat pada makalah milik Alexander Schrijver "*On the history of combinatorial optimization (till 1960)*" [6].

Menurut Manggolo *et al.* permasalahan TSP adalah masalah optimasi untuk mendapatkan rute terpendek bagi seorang *salesman* yang ingin mengunjungi banyak kota. Persoalan pada TSP merupakan salah satu dari masalah optimasi yang sekian banyak, dan menarik perhatian para matematikawan, khususnya ilmuwan komputer karena mendefinisikan sangat mudah, namun untuk dipecahkannya bagi TSP sangat sulit [7].

Yang perlu diperhatikan pada TSP, seorang *salesman* memulai kunjungannya dari kota keberangkatan, hingga berlanjut ke kota n , dan pada akhirnya akan kembali ke kota keberangkatan semula. Tetapi aturan yang harus dipatuhi adalah bahwa semua kota selain kota awal hanya dapat sekali saja dikunjungi. Masalah yang harus dipecahkan adalah cara membangun rute optimal untuk mencapai total jarak tempuh tersingkat itu bagaimana? Sehingga dapat menekan biaya transportasi.

Masalah pada kasus ini tidak mudah untuk dihadapi, karena banyak kombinasi rute yang diberikan oleh jumlah titik kota, disamping itu harus diperhatikan pula aturan yang berlaku. Representasi TSP ke dalam bentuk graf bentuk sederhananya dapat dilihat dari pada gambar 2.1. Berdasarkan gambar tersebut salesman harus mengunjungi tujuh titik kota, dimana satu titik persegi yang didefinisikan sebagai rute awal dan sekaligus merupakan rute akhir dalam perjalanan. Untuk suatu proses pencarian rute, jarak yang menghubungkan setiap kota kita harus ketahui terlebih dahulu. Jika nilai jarak belum diketahui maka jarak-jarak tersebut dapat pula dihitung berdasarkan koordinat tiap titik kota. Maka dengan menjumlah jaraknya akan diketahui rute terpendeknya setelah mencoba semua kombinasi rute, seperti gambar 2.2 [8].

2.2.2 Algoritma Genetika

Algoritma genetika menurut Carwoto [9], adalah algoritma pencarian yang mengikuti mekanisme seleksi alam darwin dan juga prinsip-prinsip genetik penentu struktur berkualitas tinggi pada sebuah domain atau populasi, yang kemudian disebut sebagai individu. Menurut Carwoto [9], algoritma genetika termasuk ke dalam studi kecerdasan buatan (*Artificial Intelligence*) dan komputasi lunak (*Soft Computing*) pada ilmu komputer. Algoritma Genetika dapat pula menyelesaikan masalah-masalah pada optimasi dan pencarian (*searching*) seperti *Knapsack Problem* dan TSP, seperti banyak terdapat pada dynamic programming dimana kompleksitasnya cukup tinggi.

Algoritma genetika dapat digambarkan mekanismenya sebagai individu-individu di dalam suatu populasi, dimana individu-individu tersebut merepresentasikan kumpulan solusi-solusi dari permasalahan. Masing-masing individu tersebut selanjutnya akan dievaluasi untuk dicari nilai *fitness*-nya. Individu-individu akan selanjutnya diseleksi, dan yang berpeluang untuk diproses selanjutnya adalah yang memiliki *fitness* tertinggi. Proses berikutnya yaitu reproduksi, yang memiliki tujuan menambah keanekaragaman atau variasi individu-individu yang terseleksi.

Cara melakukan reproduksi adalah dengan crossover, yaitu kawin silang antara individu-individu yang terpilih. Yang diharapkan dari crossover ini adalah munculnya individu-individu baru yang lebih baik dari individu sebelumnya. Proses selanjutnya setelah mengalami crossover, individu-individu baru yang terbentuk akan dimutasi. Proses ini sebagai representasi perubahan nilai gen di dalam kromosom, satu atau lebih. Proses ini merupakan akhir proses reproduksi dan akan dihasilkan populasi yang baru. Ilustrasi siklus algoritma genetika adalah seperti nampak pada gambar 2.3.

Proses ini secara lengkap juga akan dilakukan terhadap populasi-populasi baru yang terbentuk setelahnya, dan berlanjut hingga beberapa generasi yang dikehendaki. Harapannya adalah bahwa individu dengan fitness tertinggi akan muncul sebagai solusi masalah dari yang ada [10].



BAB III METODE PENELITIAN

3.1 Tahap-tahap penelitian

Dengan mengacu pada tujuan penelitian dan rumusan masalah, penelitian ini diawali melakukan studi literatur dari sumber-sumber terkait dan penelitian-penelitian yang sebelumnya. Tujuannya adalah menghimpun bahan-bahan yang menunjang topik yang sedang diangkat dalam penelitian ini. Setelah itu dilanjutkan dengan pembuatan peta titik-titik lokasi Indomaret yang berada di wilayah kecamatan Tegal Timur yang akan dikunjungi oleh seorang suplier dari sebuah titik lokasi dan membuat pemodelan jalurnya sesuai dengan kondisi real dari pandangan citra satelit. Pada pemodelan ini digunakan 2 aplikasi yaitu google my maps dan google maps . Pemodelan jalur dibuat dengan google my maps dengan membagi titik-titik singgahan tersebut menjadi sekumpulan individu yang terdiri dari 10 gen pada tiap kromosomnya yang menunjukkan rute-rute tempuh yang memungkinkan. Jarak antara setiap 2 titik lokasi dicari menggunakan google maps, sehingga didapati ukuran jarak yang real dan dapat dikenali pula rute-rute jalur searah dari rute-rute yang lainnya. Peta pemodelan jalur adalah seperti tampak pada gambar 3.1.

Algoritma yang digunakan dalam penelitian ini adalah Algoritma Genetika dengan model TSP. Maka setiap titik hanya boleh disinggahi satu kali saja kecuali titik pemberangkatan yang juga merupakan titik kembali. Dan dalam pemodelan jalur ini dipilih titik A.

Tahap perancangannya dimulai membuat populasi awal yang direpresentasikan dengan sekumpulan titik-titik persinggahan atau *node* yang memperlihatkan rute-rute tempuh yang memungkinkan. Sekumpulan rute-rute tersebut kemudian disebut dengan individu. Individu-individu ini selanjutnya akan mengalami proses seleksi sebagai *parents* berdasarkan *fitness function* berdasarkan probabilitas seleksi yang diberikan. Selanjutnya *parents* ini akan mengalami *crossover* berdasarkan probabilitas *crossover*. Proses terakhirnya adalah mutasi yang juga ditentukan oleh *mutationrate*. Dimana nilai *mutationrate*

ini ditentukan dan dilakukan mulai dari *gen* 1 sampai panjang *gen* seluruh kromosom. Selanjutnya akan muncul kromosom generasi 1, yaitu setelah seluruh proses dari algoritma genetika dilalui secara sempurna. Kromosom ini kemudian akan menjalani semua proses yang sama dari awal sampai akhir sehingga kemudian diperoleh kromosom generasi 2, dan seterusnya. Proses kemudian berulang hingga jumlah generasi yang dikehendaki. Setelah itu dievaluasi pada perulangan keberapakah generasi terbaik akan dihasilkan. Hasil akhir memperlihatkan generasi terbaik sudah ditunjukkan rata-rata pada perulangan ke 1-100. Serangkaian proses ini diilustrasikan dengan flowchart seperti nampak pada gambar 3.2.

3.2 Pembuatan model jalur TSP

Langkah awal dari penelitian ini yaitu membuat model jalur TSP. Jalur tersebut dibuat dengan menghubungkan 9 titik dengan jaraknya dan masing-masing hanya 1 kali saja dikunjungi kecuali titik pemberangkatan yang juga merupakan titik kembali, sebagaimana gambar 3.1. dimana A merupakan titik pemberangkatan yang sekaligus merupakan titik kembali. Rute yang terbentuk merupakan jalur yang menghubungkan titik-titik lokasi Indomaret di wilayah kecamatan Tegal Timur yang dihasilkan dari citra pandangan satelit. Rute ditempuh dengan menggunakan mobil yang di dalamnya juga terdapat pula jalur searah.

3.3 Perancangan algoritma genetika

3.3.1 Pembentukan populasi awal

Untuk memudahkan pemodelan maka titik-titik lokasi kunjungan tersebut diwakili oleh sebuah karakter huruf seperti pada tabel 3.1.

Aturan dalam pengambilan rute jalur:

1. Dikarenakan terdapat rute ganda yang menghubungkan 2 titik lokasi Indomaret dengan ukuran panjang yang berbeda maka di sini pengambilan rute jalur selalu memilih rute terpendek.

2. Kaidah di dalam TSP, setiap titik lokasi hanya akan dikunjungi tepat satu kali saja. Dalam pengambilan rute yang ada di pemodelan ini Indomaret Serayu (diwakili oleh karakter C) memiliki posisi hampir di pusat simpang 3 memungkinkan untuk dilewati kembali meski dari ruas jalur yang berbeda, ini terdapat pada kromosom k_6 .
3. Rute dari Indomaret Alun-alun Tegal (F) ke Indomaret Pancasila (E) lebih panjang dari arah sebaliknya dikarenakan terdapat jalur searah.

Di sini diambil sebanyak 8 kromosom (k) yang mewakili 8 urutan rute perjalanan. Titik A dituliskan sebanyak 2 kali karena merupakan titik pemberangkatan sekaligus titik kembali sehingga populasi awal adalah:

$$k_1 = [A B I H G F E C D A]$$

$$k_2 = [A B E F I H G D C A]$$

$$k_3 = [A B C E F I H G D A]$$

$$k_4 = [A B F I H G E C D A]$$

$$k_5 = [A C D G H I F E B A]$$

$$k_6 = [A C B I H G F E D A]$$

$$k_7 = [A D G H I F E C B A]$$

$$k_8 = [A D C E F G H I B A]$$

Kedelapan kromosom tersebut selanjutnya disebut dengan individu, dimana individu ini menyatakan salah satu solusi (penyelesaian) yang mungkin dari permasalahan tersebut. Tahap berikutnya adalah menentukan *cost* dari masing-masing kromosom atau individu tersebut. *Cost* dari masing-masing kromosom merupakan nilai jarak lintasan dari urutan masing-masing node yang membentuk kromosom tersebut. Panjang ruas jalan setiap rute nampak pada tabel 3.2 diukur secara reel dari citra satelit menggunakan *Google Maps*.

Maka *cost* dari masing-masing kromosom tersebut adalah:

1. $K_1 = [A B I H G F E C D A]$
 $= AB+BI+IH+HG+GF+FE+EC+CD+DA$
 $= 1.600+3.300+2.000+900+1.400+650+1.100+1.100+800$
 $= 12.850$
2. $K_2 = [A B E F I H G D C A]$

$$\begin{aligned}
 &= AB+BE+EF+FI+IH+HG+GD+DC+CA \\
 &= 1.600+1.400+300+1.700+2.000+900+1.000+1.100+950 \\
 &= 10.950
 \end{aligned}$$

3. $K_3 = [A B C E F I H G D A]$

$$\begin{aligned}
 &= AB+BC+CE+EF+FI+IH+HG+GD+DA \\
 &= 1.600+1.700+1.100+300+1.700+2.000+900+1.000+800 \\
 &= 11.100
 \end{aligned}$$
4. $K_4 = [A B F I H G E C D A]$

$$\begin{aligned}
 &= AB+BF+FI+IH+HG+GE+EC+CD+DA \\
 &= 1.600+1.700+1.700+2.000+900+1.400+1.100+1.100+800 \\
 &= 12.300
 \end{aligned}$$
5. $K_5 = [A C D G H I F E B A]$

$$\begin{aligned}
 &= AC+CD+DG+GH+HI+IF+FE+EB+BA \\
 &= 950+1.100+1.000+900+2.000+1.700+650+1.400+1.600 \\
 &= 11.300
 \end{aligned}$$
6. $K_6 = [A C B I H G F E D A]$

$$\begin{aligned}
 &= AC+CB+BI+IH+HG+GF+FE+ED+DA \\
 &= 950+650+3.300+2.000+900+1.400+650+2.500+800 \\
 &= 13.150
 \end{aligned}$$
7. $K_7 = [A D G H I F E C B A]$

$$\begin{aligned}
 &= AD+DG+GH+HI+IF+FE+EC+CB+BA \\
 &= 800+1.000+900+2.000+1.700+650+1.100+650+1.600 \\
 &= 10.400
 \end{aligned}$$
8. $K_8 = [A D C E F G H I B A]$

$$\begin{aligned}
 &= AD+DC+CE+EF+FG+GH+HI+IB+BA \\
 &= 800+1.100+1.100+300+1.400+900+2.000+ 3.300+1.600 \\
 &= 12.500
 \end{aligned}$$

3.3.2 Penentuan nilai *fitness*

Selanjutnya kita ambil nilai *fitness* individu yang akan kita jadikan calon kromosom *parents*. Perlu diketahui bahwa ukuran baik tidaknya sebuah individu

atau baik tidaknya suatu solusi yang didapatkan di sini ditentukan oleh nilai *fitness* ini. Karena yang dicari adalah jarak terpendek maka nilai *fitness*nya merupakan fungsi *fitness* $fk=1/\text{jarak}$ atau $1/\text{cost}$.

Maka nilai-nilai *fitness* (fk) dari masing-masing individu tersebut adalah:

1. k_1 dengan cost **12.850**
 $fk_1 : 1/12.850 = 0,0000778$
2. k_2 dengan cost **10.950**
 $fk_2 : 1/10.950 = 0,0000913$
3. k_3 dengan cost **11.100**
 $fk_3 : 1/11.100 = 0,0000901$
4. k_4 dengan cost **12.300**
 $fk_4 : 1/12.300 = 0,0000813$
5. k_5 dengan cost **11.300**
 $fk_5 : 1/11.300 = 0,0000885$
6. k_6 dengan cost **13.150**
 $fk_6 : 1/13.150 = 0,0000760$
7. k_7 dengan cost **10.400**
 $fk_7 : 1/10.400 = 0,0000962$
8. k_8 dengan cost **12.500**
 $fk_8 : 1/12.500 = 0,0000800$

Total *fitness* dari ke delapan kromosom tersebut adalah

$$\begin{aligned}\sum fk &= 0,0000778 + 0,0000913 + 0,0000901 + 0,0000813 + 0,0000885 + \\ &0,0000760 + 0,0000962 + 0,0000800 \\ &= 0,0006812\end{aligned}$$

3.3.3 Seleksi individu (penentuan parents)

Adalah seleksi terhadap individu-individu untuk dijadikan kromosom *parents* menggunakan metode *roulette wheel*. Pada seleksi dengan *roulette wheel* ini penentuan *parents* dilakukan dengan melihat probabilitas yang diberikan, yaitu *fitness* dibagi dengan total *fitness* populasi. Dan *Parents* dipilih menurut *fitness* mereka. Nilai *fitness* terbaik lebih berpeluang untuk terpilih dari yang lain.

Adapun tahapannya adalah :

1. Menghitung nilai fitness (fK) setiap individu
2. Menghitung probabilitas seleksi (ρ_k) di mana $\rho_k = \frac{fK}{\sum fK}$
3. Mengacak suatu bilangan [0,1]
4. Memilih individu dimana bilangan random tersebut berada sebagai parent
5. Lakukan langkah 4 dan 5 sebanyak individu dalam populasi

Maka:

$$\rho_{k_1} = \frac{0,0000778}{0,0006812}$$

$$= 0,1142102$$

$$\rho_{k_2} = \frac{0,0000913}{0,0006812}$$

$$= 0,1340282$$

$$\rho_{k_3} = \frac{0,0000901}{0,0006812}$$

$$= 0,1322666$$

$$\rho_{k_4} = \frac{0,0000813}{0,0006812}$$

$$= 0,1193482$$

$$\rho_{k_5} = \frac{0,0000885}{0,0006812}$$

$$= 0,1299178$$

$$\rho_{k_6} = \frac{0,0000760}{0,0006812}$$

$$= 0,1115678$$

$$\rho_{k_7} = \frac{0,0000962}{0,0006812}$$

$$= 0,1412214$$

$$\rho_{k_8} = \frac{0,0000800}{0,0006812}$$

$$= 0,1174398$$

Karena proses seleksi adalah menggunakan *rouletewheel*, maka yang kita cari terlebih dahulu nilai komulatif (C) dari probabilitas.

$$\begin{aligned}
 C_1 &= 0,1142102 \\
 C_2 &= 0,1142102 + 0,1340282 \\
 &= 0,2482384 \\
 C_3 &= 0,2482384 + 0,1322666 \\
 &= 0,3805050 \\
 C_4 &= 0,3805050 + 0,1193482 \\
 &= 0,4998532 \\
 C_5 &= 0,4998532 + 0,1299178 \\
 &= 0,6297710 \\
 C_6 &= 0,6297710 + 0,1115678 \\
 &= 0,7413388 \\
 C_7 &= 0,7413388 + 0,1412214 \\
 &= 0,8825602 \\
 C_8 &= 0,8825602 + 0,1174398 \\
 &= 1,0000000
 \end{aligned}$$

Lalu kita bangkitkan bilangan *random* (R) sebanyak jumlah individu, yaitu antara 0-1.

Misal sebagai berikut:

$$\begin{aligned}
 R_1 &: 0.92 \\
 R_2 &: 0.53 \\
 R_3 &: 0.81 \\
 R_4 &: 0.33 \\
 R_5 &: 0.04 \\
 R_6 &: 0.42 \\
 R_7 &: 0.18 \\
 R_8 &: 0.71
 \end{aligned}$$

Apabila kita gambarkan di atas sebagai sebuah garis bilangan *roulette wheel* dari hasil hitungan kumulatif adalah seperti pada gambar 3.3.

Jika putaran *rouletewheel* / *random* ke- x atau (R_x) lebih kecil dari kumulatif x atau (k_x) maka dia sebagai induk, namun harus lebih besar dari (k_{x-1}). Dengan kata lain $(k_{x-1}) < (R_x) < (k_x)$

Maka terbentuklah populasi baru dengan:

1. kromosom ke 1 adalah k_8 [A D C E F G H I B A]
2. kromosom ke 2 adalah k_5 [A C D G H I F E B A]
3. kromosom ke 3 adalah k_7 [A D G H I F E C B A]
4. kromosom ke 4 adalah k_3 [A B C E F I H G D A]
5. kromosom ke 5 adalah k_1 [A B I H G F E C D A]
6. kromosom ke 6 adalah k_4 [A B F I H G E C D A]
7. kromosom ke 7 adalah k_2 [A B E F I H G D C A]
8. kromosom ke 8 adalah k_6 [A C B I H G F E D A]

3.3.4 *Crossover* / pindah silang

Yaitu pertukaran gen-gen antar sesama kromosom *parents*. Gen-gen saling dipertukarkan dengan tetap menjaga urutan node yang bukan bagian dari kromosom tersebut namun dipengaruhi oleh parameter *crossover probability* (p_c). Misal kita tentukan $p_c = 15\%$, kemudian kita bangkitkan bilangan *random* R lagi sebanyak jumlah populasi yaitu 8 kali, dimana masing-masing mewakili urutan kromosom terseleksi berdasarkan probabilitas seleksi.

$$R_1 = 0,136$$

$$R_2 = 0,134$$

$$R_3 = 0,308$$

$$R_4 = 0,772$$

$$R_5 = 0,214$$

$$R_6 = 0,133$$

$$R_7 = 0,457$$

$$R_8 = 0,878$$

Maka dalam 1 generasi ini diharapkan ada 3 kromosom dari populasi mengalami *crossover* dengan induk, yaitu:

1. kromosom ke 1 yaitu k_8 [A D C E F G H I B A]
2. kromosom ke 2 yaitu k_5 [A C D G H I F E B A]

3. dan kromosom ke 7 yaitu k_4 [A B F I H G E C D A].

Untuk menentukan posisi *crossover* dibangkitkanlah bilangan *random* antara 1 sampai dengan panjang kromosom-1 yang dalam kasus ini adalah antara 1-3. Misal bilangan acaknya adalah 1, maka gen yang ke-1 pada kromosom induk pertama diambil lalu ditukar dengan gen pertama pada kromosom induk kedua yang belum ada pada induk pertama dengan tetap memperhatikan urutannya. Jika bilangan acaknya 2, maka gen yang ke-2 pada kromosom induk tersebut diambil lalu ditukar dengan gen ke dua pada kromosom induk pasangannya yang belum ada pada induk tersebut dengan tetap memperhatikan urutannya pula.

Misalkan bilangan *random* untuk 3 kromosom induk yang akan di-*crossover* juga diberikan 2, maka gen yang akan dicrossover adalah 1-2 juga:

Kromosom ke 1 (k_8) = 2

Kromosom ke 2 (k_5) = 2

Kromosom ke 6 (k_4) = 2

Proses *crossover* :

Proses ke 1 = kromosom ke 8 >> kromosom ke 5

= [ADCEFGHIBA] >> [ACDGHIFEB A]

Memotong pada titik ke-2:

Segmen 1 : [AD] dan [AC]

Segmen 2 : [CEFGHIBA] dan [DGHIFEB A]

Hasil *crossover*:

Anak 1 : [AD DGHIFEB A] = [AD C DGHIFEB A]

Anak 2 : [AC C EFGHIBA] = [AC D EFGHIBA]

Proses ke 2 = kromosom ke 5 >> kromosom ke 4

= [ACDGHIFEB A] >> [ABFIHGE CDA]

Memotong pada titik ke-1:

Segmen 1: [AC] dan [AB]

Segmen 2: [DGHIFEB A] dan [FIHGE CDA]

Hasil *crossover*:

Anak 1 : [AC FIHGE C D A]

: [A C F I H G E B D A] → (perhatikan urutan gen selanjutnya)

Anak 2 dan [A B D G H I F E B A] → (perhatikan urutan gen selanjutnya)

: [A B D G H I F E C A]

Proses ke 3 = kromosom ke 1 (k_3) >> kromosom ke 6 (k_4)

= [A D C E F G H I B A] >> [A B F I H G E C D A]

Memotong pada titik ke-4:

Segmen 1 : [A D] dan [A B]

Segmen 2 : [C E F G H I B A] dan [F I H G E C D A]

Hasil crossover:

Anak 1 : [A D F I H G E C D A] → (perhatikan urutan gen selanjutnya)
: [A D F I H G E C B A]

Anak 2 : [A B C E F G H I B A] → (perhatikan urutan gen selanjutnya)
: [A B C E F G H I D A]

Maka Populasi setelah di-crossover :

kromosom ke 1 = [A D C G H I F E B A]

kromosom ke 2 = [A C F I H G E B D A]

kromosom ke 3 = [A D G H I F E C B A]

kromosom ke 4 = [A B C E F I H G D A]

kromosom ke 5 = [A B I H G F E C D A]

kromosom ke 6 = [A D F I H G E C B A]

kromosom ke 7 = [A B E F I H G D C A]

kromosom ke 8 = [A C B I H G F E D A]

3.3.5 Mutasi

Skema mutasi yang digunakan adalah *swapping mutation*. Yaitu kromosom-kromosom yang akan dimutasi dalam satu populasi jumlahnya ditentukan oleh parameter *mutation rate* (ρ_m). Prosesnya adalah dengan cara menukar gen yang dipilih secara acak dengan gen sesudahnya. Apabila posisi gen

berada di akhir kromosom, maka penukarannya dengan gen yang pertama. Langkah awalnya, panjang total gen dalam satu populasi, adalah:

$$\begin{aligned} \text{Panjang total gen} &= \text{jumlah gen dalam 1 kromosom} * \text{jumlah Kromosom} \\ &= 10 * 8 \\ &= 80 \end{aligned}$$

Pemilihan posisi gen yang akan dimutasi dilakukan dengan membangkitkan bilangan acak antara 1 – Panjang total gen, dan dalam kasus ini adalah 1- 80. Dan misal kita tentukan pula $\rho_m = 15\%$.

$$\text{Maka jumlah semua gen yang akan dimutasi adalah} = 0,15 * 80 = 12$$

Sejumlah 12 posisi gen yang akan dimutasi, maka posisi setelah diacak adalah 2,13,18,25, 32, 38, 44, 47, 52, 58, 62, 76.

Proses mutasi :

kromosom ke 1 = [A D C G H I F E B A] menjadi mutated child_1 [A C D G H I F E B A]

kromosom ke 2 = [A C F I H G E B D A] menjadi mutated child_2 [A C I F H G E D B A]

kromosom ke 3 = [A D G H I F E C B A] menjadi mutated child_3 [A D G H F I E C B A]

kromosom ke 4 = [A B C E F I H G D A] menjadi mutated child_4 [A C B E F I H G D A]

kromosom ke 5 = [A B I H G F E C D A] menjadi mutated child_5 [A B I G H F C E D A]

kromosom ke 6 = [A D F I H G E C B A] menjadi mutated child_6 [A F D I H G E B C A]

kromosom ke 7 = [A B E F I H G D C A] menjadi mutated child_7 [A E B F I H G D C A]

kromosom ke 8 = [A C B I H G F E D A] menjadi mutated child_8 [A C B I H F G E D A]

Maka proses algoritma genetik untuk 1 generasi telah selesai, dengan nilai *cost*:

Kromosom ke 1 generasi ke 1

$$= [A C D G H I F E B A]$$

$$\begin{aligned}
&= AC + CD + DG + GH + HI + IF + FE + EB + BA \\
&= 950 + 1.100 + 1.000 + 900 + 2.000 + 1.700 + 650 + 1.400 + 1.600 \\
&= 11.400
\end{aligned}$$

$$\begin{aligned}
fk_{1 \text{ generasi } 1} &= 1/13.600 \\
&= 0,0000877
\end{aligned}$$

Kromosom ke 2 generasi ke 1

$$\begin{aligned}
&= [A C I F H G E D B A] \\
&= AC + CI + IF + FH + HG + GE + ED + DB + BA \\
&= 950 + 3.300 + 1.700 + 2.300 + 900 + 1.400 + 2.500 + 1.750 + 1.600 \\
&= 16.400
\end{aligned}$$

$$\begin{aligned}
fk_{2 \text{ generasi } 1} &= 1/16.400 \\
&= 0,0000610
\end{aligned}$$

Kromosom ke 3 generasi ke 1

$$\begin{aligned}
&= [A D G H F I E C B A] \\
&= AD + DG + GH + HF + FI + IE + EC + CB + BA \\
&= 800 + 1.000 + 900 + 2.300 + 1.700 + 2.000 + 1.100 + 1.100 + 650 \\
&= 11.550
\end{aligned}$$

$$\begin{aligned}
fk_{3 \text{ generasi } 1} &= 1/11.550 \\
&= 0,0000866
\end{aligned}$$

Kromosom ke 4 generasi ke 1

$$\begin{aligned}
&= [A C B E F I H G D A] \\
&= AC + CB + BE + EF + FI + IH + HG + GD + DA \\
&= 950 + 650 + 1.400 + 300 + 1.700 + 2.000 + 900 + 1.000 + 800 \\
&= 9.700
\end{aligned}$$

$$\begin{aligned}
fk_{4 \text{ generasi } 1} &= 1/9.700 \\
&= 0,0001031
\end{aligned}$$

Kromosom ke 5 generasi ke 1

$$\begin{aligned}
&= [A B I G H F C E D A] \\
&= AB + BI + IG + GH + HF + FC + CE + ED + DA \\
&= 1.600 + 3.300 + 2.000 + 900 + 2.300 + 1.700 + 1.100 + 2.500 + 800 \\
&= 16.200
\end{aligned}$$

$$\begin{aligned}fk_{5 \text{ generasi } 1} &= 1/16.200 \\ &= 0,0000617\end{aligned}$$

Kromosom ke 6 generasi ke 1

$$\begin{aligned}&= [A F D I H G E B C A] \\ &= AF+FD+DI+IH+HG+GE+EB+BC+CA \\ &= 2.350 + 2.400 + 3.000 + 2.000 + 900 + 1.400 + 1.400 + 650 + 950 \\ &= 15.050\end{aligned}$$

$$\begin{aligned}fk_{6 \text{ generasi } 1} &= 1/15.050 \\ &= 0,0000664\end{aligned}$$

Kromosom ke 7 generasi ke 1

$$\begin{aligned}&= [A E B F I H G D C A] \\ &= AE+EB+BF+FI+IH+HG+GD+DC+CA \\ &= 2.050 + 1.400 + 1.700 + 1.700 + 2.000 + 900 + 1.000 + 1.100 + 950 \\ &= 13.100\end{aligned}$$

$$\begin{aligned}fk_{7 \text{ generasi } 1} &= 1/13.100 \\ &= 0,0000763\end{aligned}$$

Kromosom ke 8 generasi ke 1

$$\begin{aligned}&= [A C B I H F G E D A] \\ &= AC+CB+BI+IH+HF+FG+GE+ED+DA \\ &= 950 + 650 + 3.300 + 2.000 + 2.300 + 1.400 + 1.400 + 2.500 + 800 \\ &= 15.300\end{aligned}$$

$$\begin{aligned}fk_{8 \text{ generasi } 1} &= 1/15.300 \\ &= 0,0000654\end{aligned}$$

Sehingga nilai optimum solusi pada generasi 1 ini diperoleh pada:

Kromosom ke 4 generasi ke 1

$$\begin{aligned}&= [A C B E F I H G D A] \\ &= AC+CB+BE+EF+FI+IH+HG+GD+DA\end{aligned}$$

Dengan jarak tempuhnya 9.700 meter dan $fk_{\text{generasi } 1}$ nya: 0,0001031.

Jika dalam beberapa generasi berturut-turut diperoleh nilai *fitness* yang tertinggi tidak berubah maka didapat kriteria berhenti. Pada 1 generasi ini nilai *fitness* tertingginya telah terlihat tidak berubah dan jika perhitungan dilanjutkan sampai generasi ke-N maka diyakinkan bahwa nilai *fitness* yang tertinggi akan tetap pula. Maka pada generasi ke-1 ini perhitungan cukup mendekati dan mewakili solusi optimum yang didapat. Sehingga algoritma genetika untuk penyelesaian masalah TSP ini dapat dibuktikan.

3.4 Simulasi

Nilai-nilai *fitness* dan ukuran jarak akan ditampilkan selama berlangsungnya proses optimasi. Pada akhir proses akan ditampilkan nilai terbaik yang diperolehnya. Dalam proses ini akan dilakukan perulangan dilakukan sebanyak 100 kali, lampiran 1 menunjukkan sebaran nilai *fitness* yang diperoleh dari hasil optimasi.

3.4.1 Algoritma

Pada pembentukan populasi awal, di sini akan dihasilkan sebanyak 8 kromosom (k) yang mewakili 8 urutan rute perjalanan, yaitu:

$$k_1 = [A B I H G F E C D A]$$

$$k_2 = [A B E F I H G D C A]$$

$$k_3 = [A B C E F I H G D A]$$

$$k_4 = [A B F I H G E C D A]$$

$$k_5 = [A C D G H I F E B A]$$

$$k_6 = [A C B I H G F E D A]$$

$$k_7 = [A D G H I F E C B A]$$

$$k_8 = [A D C E F G H I B A]$$

Dengan masing-masing nilai fitnesnya adalah:

$$fk_1 = 0,0000778$$

$$fk_2 = 0,0000913$$

$$fk_3 = 0,0000901$$

$$fk_4 = 0,0000813$$

$$fk_5 = 0,0000885$$

$$fk_6 = 0,000088$$

$$fk_7 = 0,0000760$$

$$fk_8 = 0,0000800$$

Setelah melalui seleksi parents yang dipengaruhi oleh probabilitas seleksi akan diperoleh kromosom yang terpilih untuk mengalami crossover dan mutasi adalah

kromosom ke 1 adalah k_8 [D C E F G H I B]

kromosom ke 2 adalah k_5 [C D G H I F E B]

kromosom ke 3 adalah k_7 [D G H I F E C B]

kromosom ke 4 adalah k_3 [B C E F I H G D]

kromosom ke 5 adalah k_1 [B I H G F E C D]

kromosom ke 6 adalah k_4 [B F I H G E C D]

kromosom ke 7 adalah k_2 [B E F I H G D C]

kromosom ke 8 adalah k_6 [C B I H G F E D]

Selanjutnya proses crossover dan mutasi akan menghasilkan keturunan generasi 1 adalah:

A C D G H I F E B A dengan fitness 0,0000877

A C I F H G E D B A dengan fitness 0,0000610

A D G H F I E C B A dengan fitness 0,0000866

A C B E F I H G D A dengan fitness 0,0001031

A B I G H F C E D A dengan fitness 0,0000617

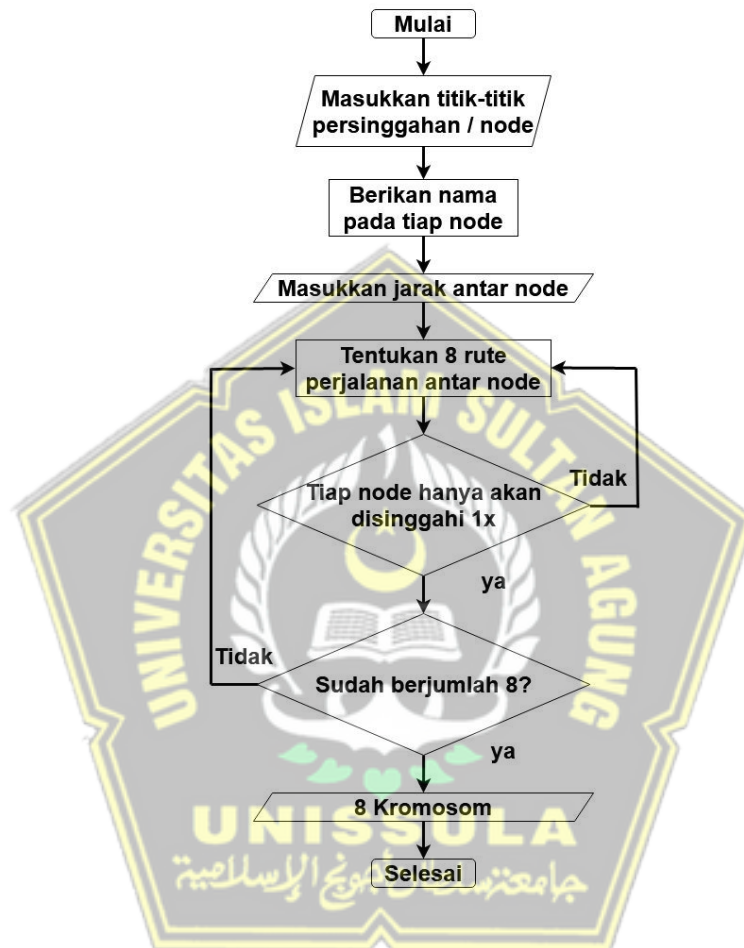
A F D I H G E B C A dengan fitness 0,0000664

A E B F I H G D C A dengan fitness 0,0000763

A C B I H F G E D A dengan fitness 0,0000654

Maka pada kromosom generasi 1 yang merupakan perulangan ke-1 tercatat nilai fitness dan jarak terbaik adalah 0,0001031 dan 9.700m. Dan jika dilakukan proses iterasi selanjutnya tidak ada nilai yang lebih baik dari itu.

3.4.2 Flowchart



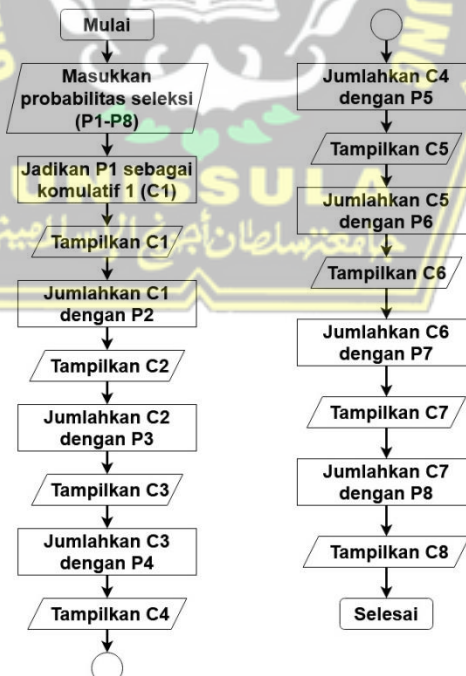
Gambar 3.1. Membuat populasi awal



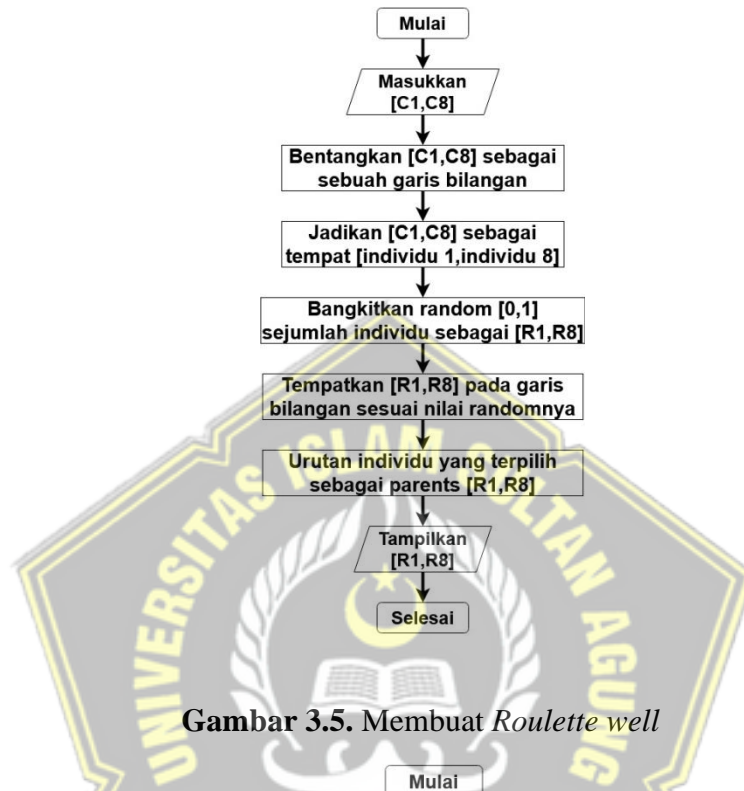
Gambar 3.2. Menentukan nilai fitness



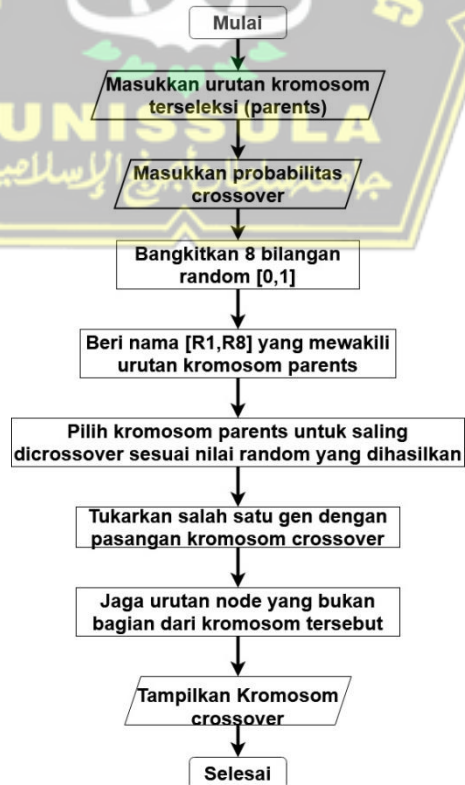
Gambar 3.3. Membuat probabilitas seleksi

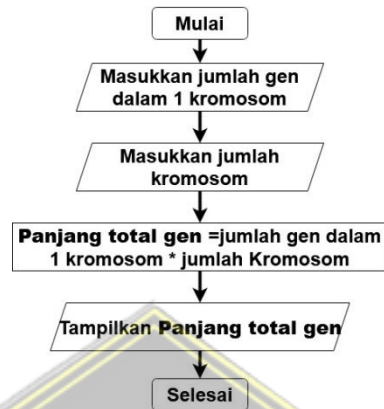


Gambar 3.4. Membuat komulatif probabilitas

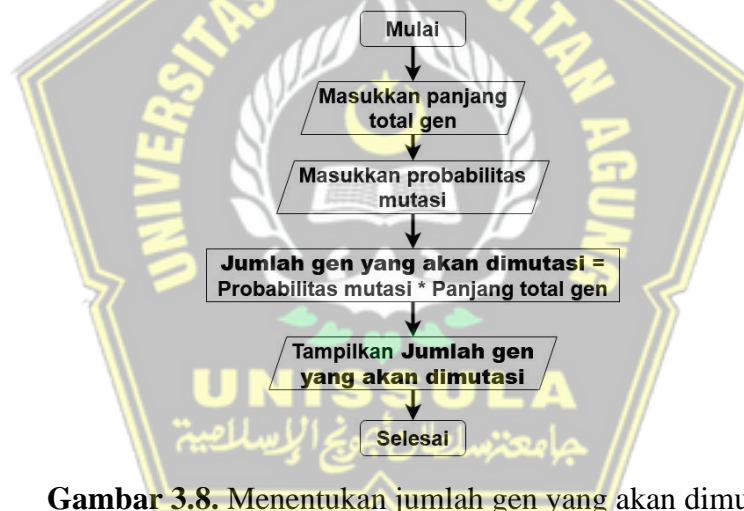


Gambar 3.5. Membuat *Roulette well*

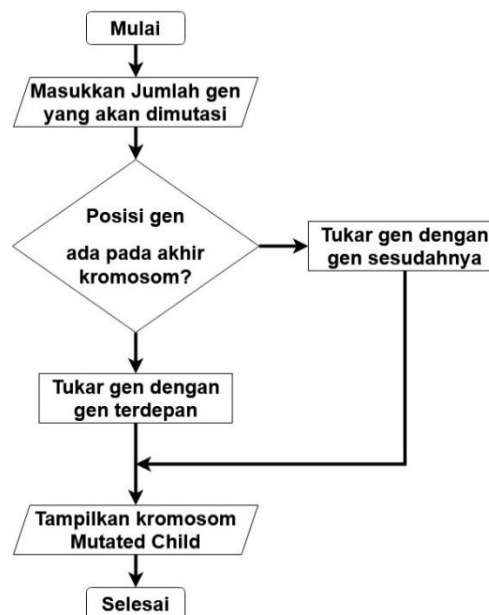


Gambar 3.6. *Crossover*

Gambar 3.7. Mencari panjang total gen



Gambar 3.8. Menentukan jumlah gen yang akan dimutasi



Gambar 3.9. Proses mutasi

3.4.3 Coding

genetic_algorithm_TSP.m:

```
% Definisi Populasi Awal
populasi = {
    {'A', 'B', 'I', 'H', 'G', 'F', 'E', 'C', 'D', 'A'},
    {'A', 'B', 'E', 'F', 'I', 'H', 'G', 'D', 'C', 'A'},
    {'A', 'B', 'C', 'E', 'F', 'I', 'H', 'G', 'D', 'A'},
    {'A', 'B', 'F', 'I', 'H', 'G', 'E', 'C', 'D', 'A'},
    {'A', 'C', 'D', 'G', 'H', 'I', 'F', 'E', 'B', 'A'},
    {'A', 'C', 'B', 'I', 'H', 'G', 'F', 'E', 'D', 'A'},
    {'A', 'D', 'G', 'H', 'I', 'F', 'E', 'C', 'B', 'A'},
    {'A', 'D', 'C', 'E', 'F', 'G', 'H', 'I', 'B', 'A'}
};

num_populasi = length(populasi);
costs = zeros(num_populasi, 1);
fitness = zeros(num_populasi, 1);

% Hitung Cost dan Fitness untuk Masing-Masing Kromosom
for i = 1:num_populasi
    rute = populasi{i};
    total_cost = 0;
    for j = 1:(length(rute) - 1)
        % Cari indeks dari rute saat ini dan selanjutnya
        rute_name = [rute{j}, rute{j+1}];
        idx = find(strcmp(routes, rute_name));
        if ~isempty(idx)
            total_cost = total_cost + distances(idx);
        end
    end
    costs(i) = total_cost;
    fitness(i) = 1 / total_cost; % Fitness adalah 1 / cost
end

% Buat Tabel Individu dengan Cost dan Fitness
rute_str = cellfun(@(x) strjoin(x, '-'), populasi, 'UniformOutput', false);
```

```

% Pastikan semua variabel memiliki ukuran yang sama
T = table(rute_str(:), costs, fitness, 'VariableNames', {'Rute', 'Cost', 'Fitness'});

% Tampilkan Tabel Populasi Awal dengan Cost dan Fitness
disp('Tabel Populasi Awal dengan Cost dan Fitness:');
disp(T);

% Hitung Total Fitness
total_fitness = sum(fitness);

% Hitung Probabilitas Seleksi
probabilitas_seleksi = fitness / total_fitness;

% Hitung Probabilitas Kumulatif
probabilitas_kumulatif = cumsum(probabilitas_seleksi);

% Tampilkan Total Fitness, Probabilitas Seleksi, dan Probabilitas Kumulatif
fprintf('\nTotal Fitness: %.7f\n', total_fitness);
disp('Probabilitas Seleksi:');
disp(probabilitas_seleksi);
disp('Probabilitas Kumulatif:');
disp(probabilitas_kumulatif);

% Definisi R nilai random untuk seleksi parent
R = [0.92, 0.53, 0.81, 0.33, 0.04, 0.42, 0.18, 0.71];

% Inisialisasi array untuk menyimpan parent yang terpilih
selected_parents = cell(num_populasi, 1);
selected_costs = zeros(num_populasi, 1);
selected_fitness = zeros(num_populasi, 1);

% Pembentukan Parent dengan Roulette Wheel Selection
for i = 1:num_populasi
    for j = 1:num_populasi
        if R(i) <= probabilitas_kumulatif(j)
            selected_parents{i} = populasi{j};
            selected_costs(i) = costs(j);
            selected_fitness(i) = fitness(j);
            break;
        end
    end
end

% Buat Tabel Parent yang Terpilih dengan Cost dan Fitness

```

```

selected_rute_str = cellfun(@(x) strjoin(x, '-'), selected_parents,
'UniformOutput', false);

T_parents = table(selected_rute_str(:), selected_costs, selected_fitness, ...
'VariableNames', {'Rute', 'Cost', 'Fitness'});

% Tampilkan Tabel Parent yang Terpilih
disp('Tabel Parent yang Terpilih:');
disp(T_parents);

% Definisi nilai random untuk crossover
R_c = [0.136, 0.134, 0.308, 0.772, 0.214, 0.133, 0.457, 0.878];
crossover_prob = 0.15;

% Inialisasi array untuk menyimpan hasil crossover
crossover_population = cell(num_populasi, 1);
crossover_costs = zeros(num_populasi, 1);
crossover_fitness = zeros(num_populasi, 1);

% Pembentukan Kromosom Hasil Crossover
for i = 1:num_populasi
    if R_c(i) <= crossover_prob
        % Pilih dua parent secara acak
        parent1 = selected_parents{randi(num_populasi)};
        parent2 = selected_parents{randi(num_populasi)};

        % Terapkan Crossover
        crossover_point = randi([1, length(parent1) - 1]);
        offspring1 = [parent1(1:crossover_point), setdiff(parent2,
parent1(1:crossover_point))];
        offspring2 = [parent2(1:crossover_point), setdiff(parent1,
parent2(1:crossover_point))];

        % Perbaiki kromosom dengan gen ganda
        offspring1 = fix_duplicate_genes(offspring1);
        offspring2 = fix_duplicate_genes(offspring2);

        % Hitung cost dan fitness untuk offspring jika tidak kosong
        if ~isempty(offspring1) && length(offspring1) == numel(unique(offspring1))
            rute = offspring1;
            total_cost = 0;
            for k = 1:(length(rute) - 1)
                rute_name = [rute{k}, rute{k+1}];
                idx = find(strcmp(routes, rute_name));

```

```

        if ~isempty(idx)
            total_cost = total_cost + distances(idx);
        end
    end
    crossover_population{i} = offspring1;
    crossover_costs(i) = total_cost;
    crossover_fitness(i) = 1 / total_cost;
end

if ~isempty(offspring2) && length(offspring2) == numel(unique(offspring2))
    rute = offspring2;
    total_cost = 0;
    for k = 1:(length(rute) - 1)
        rute_name = [rute{k}, rute{k+1}];
        idx = find(strcmp(routes, rute_name));
        if ~isempty(idx)
            total_cost = total_cost + distances(idx);
        end
    end
    crossover_population{i} = offspring2;
    crossover_costs(i) = total_cost;
    crossover_fitness(i) = 1 / total_cost;
end
end
end

% Hapus kromosom yang kosong dari crossover_population
valid_indices = ~cellfun('isempty', crossover_population);
crossover_population = crossover_population(valid_indices);
crossover_costs = crossover_costs(valid_indices);
crossover_fitness = crossover_fitness(valid_indices);

% Pilih 3 individu terbaik dari hasil crossover
[~, sort_idx] = sort(crossover_costs);
top_3_indices = sort_idx(1:3);

% Gabungkan 3 individu crossover terbaik ke populasi parent
final_population = [selected_parents; crossover_population(top_3_indices)];
final_costs = [selected_costs; crossover_costs(top_3_indices)];
final_fitness = [selected_fitness; crossover_fitness(top_3_indices)];

% Buat Tabel Individu Akhir dengan Cost dan Fitness
final_rute_str = cellfun(@(x) strjoin(x, '-'), final_population, 'UniformOutput',
false);

```

```
T_final = table(final_rute_str(:), final_costs, final_fitness, 'VariableNames',
{'Rute', 'Cost', 'Fitness'});
```

```
% Tampilkan Tabel Individu Akhir
disp('Tabel Individu Akhir:');
disp(T_final);
```

fix_duplicate_genes.m

```
function new_chromosome = fix_duplicate_genes(chromosome)
    all_genes = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I'};
    unique_genes = unique(chromosome);
    missing_genes = setdiff(all_genes, unique_genes);
    if length(missing_genes) > 0
        duplicates = setdiff(chromosome, unique_genes);
        for i = 1:length(duplicates)
            index = find(strcmp(chromosome, duplicates{i}));
            chromosome{index} = missing_genes{mod(i-1, length(missing_genes)) + 1};
        end
    end
    new_chromosome = chromosome;
end

% Terapkan Mutasi pada Populasi Akhir
mutated_population = cell(size(final_population));
for i = 1:length(final_population)
    mutated_population{i} = mutate_chromosome(final_population{i}, routes);
end

% Hitung Cost dan Fitness untuk Kromosom yang Terkena Mutasi
mutated_costs = zeros(length(mutated_population), 1);
mutated_fitness = zeros(length(mutated_population), 1);

for i = 1:length(mutated_population)
    rute = mutated_population{i};
    total_cost = 0;
    for j = 1:(length(rute) - 1)
        % Cari indeks dari rute saat ini dan selanjutnya
        rute_name = [rute{j}, rute{j+1}];
        idx = find(strcmp(routes, rute_name));
        if ~isempty(idx)
            total_cost = total_cost + distances(idx);
        end
    end
    mutated_costs(i) = total_cost;
```

```

    mutated_fitness(i) = 1 / total_cost; % Fitness adalah 1 / cost
end

% Buat Tabel Individu Setelah Mutasi dengan Cost dan Fitness
mutated_rute_str = cellfun(@(x) strjoin(x, '-'), mutated_population,
'UniformOutput', false);
T_mutated = table(mutated_rute_str(:), mutated_costs, mutated_fitness,
'VariableNames', {'Rute', 'Cost', 'Fitness'});

% Tampilkan Tabel Individu Setelah Mutasi
disp('Tabel Individu Setelah Mutasi:');
disp(T_mutated);

mutate_chromosome.m
% Terapkan Mutasi pada Populasi Akhir
mutated_population = cell(size(final_population));
for i = 1:length(final_population)
    mutated_population{i} = mutate_chromosome(final_population{i}, routes);
end

% Hitung Cost dan Fitness untuk Kromosom yang Terkena Mutasi
mutated_costs = zeros(length(mutated_population), 1);
mutated_fitness = zeros(length(mutated_population), 1);

for i = 1:length(mutated_population)
    rute = mutated_population{i};
    total_cost = 0;
    for j = 1:(length(rute) - 1)
        % Cari indeks dari rute saat ini dan selanjutnya
        rute_name = [rute{j}, rute{j+1}];
        idx = find(strcmp(routes, rute_name));
        if ~isempty(idx)
            total_cost = total_cost + distances(idx);
        end
    end
    mutated_costs(i) = total_cost;
    mutated_fitness(i) = 1 / total_cost; % Fitness adalah 1 / cost
end

% Buat Tabel Individu Setelah Mutasi dengan Cost dan Fitness
mutated_rute_str = cellfun(@(x) strjoin(x, '-'), mutated_population,
'UniformOutput', false);
T_mutated = table(mutated_rute_str(:), mutated_costs, mutated_fitness,
'VariableNames', {'Rute', 'Cost', 'Fitness'});

```

```
% Tampilkan Tabel Individu Setelah Mutasi
disp('Tabel Individu Setelah Mutasi:');
disp(T_mutated);
```

3.5 Kinerja

Selama berjalannya proses optimasi lintasan akan dipilih secara acak dengan pencatatan masing-masing nilai *fitness*nya. Rute terbaik diketahui dengan *fitness* tertinggi pada masing-masing perulangannya yang kemudian menghasilkan rute terpendek.

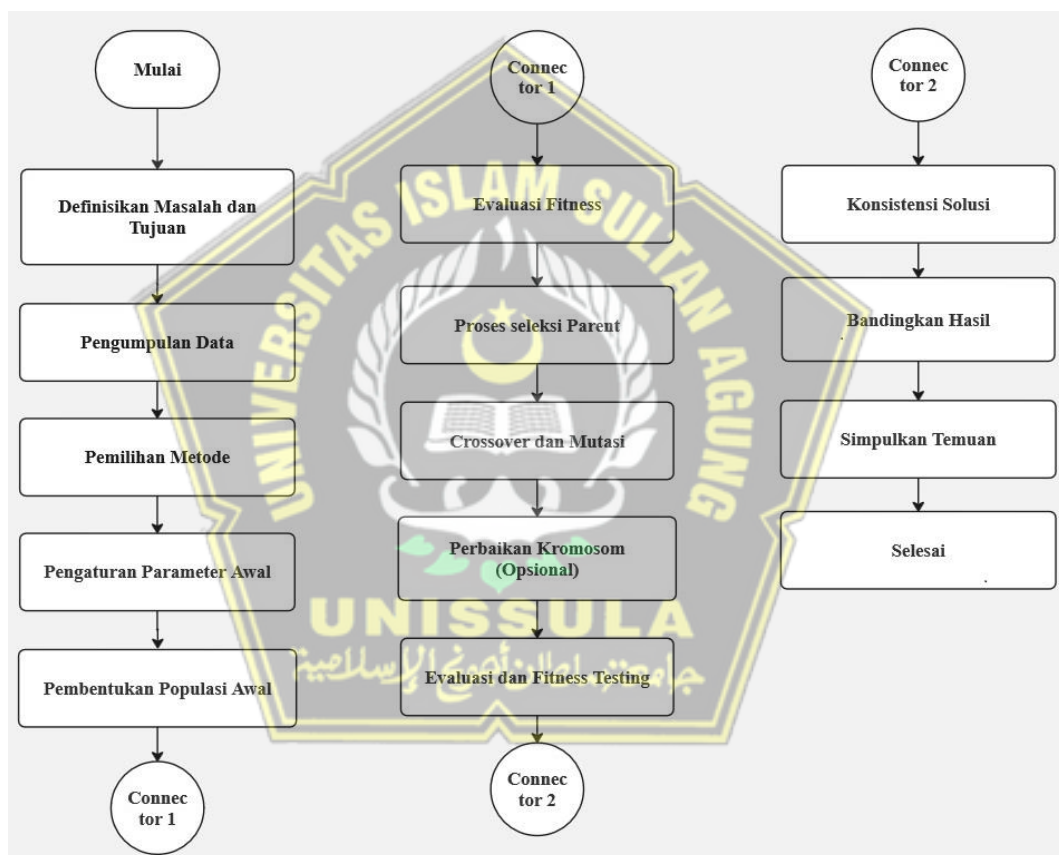
Panjang jalur tidak selalu berubah, tapi ada kemungkinan 1 atau 2 kali berubah. Berdasarkan pengujian hampir selalu panjang terbaiknya adalah 9.700m. Perbedaan jalur terbaik (yang kadang sangat kecil kemungkinan muncul) itu menandakan bahwa program dan algoritma ini memang mengolah peta yang diberikan.

Program optimasi ini untuk iterasi diset di awal sebanyak 100 kali. Dan di sini dapat dikatakan 100 itu adalah 100 generasi atau 100 perulangan. Jadi dalam menerapkan dan mensimulasi algoritma genetika untuk optimasi pada kasus TSP ini, kinerja yang diukur dalam penelitian adalah nilai jarak, sedangkan untuk waktu sifatnya relatif karena ini menyesuaikan kondisi medan yang real. Cepat tidaknya tergantung kondisi lalu lintas / traffic kendaraan. Karena itu waktu tidak secara eksplisit dinyatakan dalam simulasi, tetapi di sini untuk semua lintasan dan generasi / perulangan berlaku semakin tinggi nilai *fitness* akan semakin pendek jarak tempuh dan berpeluang mempersingkat pula waktu.

Iterasi terakhir bukan berarti yang terbaik. Adalah keliru jika mengatakan bahwa algoritma genetika akan berhenti ketika mencapai titik optimum. Pertanyaannya berapa nilai optimum yang dimaksud? Kalau untuk kasus ini karena petanya tidak begitu sulit, tentu bisa dikatakan nilai optimum sudah ada. Tapi *real-world*nya tidak begitu. Maka dibutuhkan iterasi sebanyak 100 tadi, lalu dibandingkan mana yang terbaik. Rute yang paling optimum disimpan dalam variabel pada nilai jarak 9.700m. Mereka akan selalu menyimpan *length* terbaik

setiap kali iterasi/*population update*. Jadi, yang paling optimum adalah bisa di iterasi beberapa saja.

Demikianlah kinerja sesungguhnya dari algoritma genetika. Untuk memberikan gambaran yang lebih jelas mengenai alur penelitian yang telah dilakukan, berikut disajikan flowchart penelitian. Flowchart ini menggambarkan tahapan-tahapan utama dalam penelitian, mulai dari identifikasi masalah hingga analisis hasil, serta bagaimana setiap tahapan saling berhubungan dalam upaya mencapai tujuan penelitian.



Gambar 3.10. *Flowchart Penelitian*

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil

Sebagaimana yang diketahui bahwa terdapat 3 tujuan penelitian yang dikemukakan pada bab 1, diantara 3 tujuan penelitian ini secara umum adalah :

1. Menerapkan Algoritma Genetika untuk optimasi rute terpendek pada kasus TSP
2. Mensimulasikan Algoritma Genetika untuk optimasi rute terpendek pada kasus TSP
3. Mengetahui kinerja optimasi pencarian jalur terpendek pada kasus TSP menggunakan algoritma genetika

Adapun untuk mencapai tujuan penelitian tersebut maka peneliti mendeskripsikan menjadi beberapa uraian sebagai berikut:

1. Penerapan Algoritma Genetika untuk optimasi rute terpendek pada kasus TSP dapat dilakukan dengan mengikuti semua prosesnya. Pada kasus ini datanya adalah jalur nyata penghubung titik-titik lokasi Indomaret dari suatu titik lokasi pemberangkatan yang juga merupakan titik kembali, yang diambil dari foto citra satelit melalui Google My Map dan pengukuran jarak antara setiap 2 titik lokasi menggunakan Google Map seperti pada gambar 3.1. Setelah melalui proses seleksi melalui fitness function, probabilitas seleksi, crossover dan mutasi didapat hasil akhir yang tidak berubah yaitu dengan nilai fitness 0,0001031 dengan jarak terbaiknya adalah 9700. Jarak ini dimiliki oleh jalur-jalur yang jika dilakukan perhitungan manual dengan algoritma genetika, pada generasi 1 jarak tersebut dimiliki oleh:

Kromosom ke 4 generasi ke 1

= AC+CB+BE+EF+FI+IH+HG+GD+DA

Dengan jarak tempuh 9.700m dan f_k *generasi 1* nya: 0,0001031

2. Untuk mengetahui kinerja optimasi pencarian jalur terpendek pada kasus TSP menggunakan algoritma genetika ini kita harus melihat pada kondisi medan. Karena kondisi medan real maka kinerja yang diukur adalah jarak. Rute dan fitness terbaik diketahui pada setiap perulangannya yang kemudian dapat disimpulkan rute terpendeknya. Di sini dihasilkan nilai fitness terbaiknya adalah 0,0001031 dan jarak terpendeknya adalah 9.700m.

4.2 Analisis

Kinerja dan simulasi pada penelitian ini pada dasarnya berangkat dari unsur yang sama yaitu seperti yang tertuang dalam semua proses yang terjadi pada algoritma genetika. Pada hitungan manual untuk pembentukan populasi awal, semua rute yang mungkin dipilih dengan hanya menyinggahi setiap titik tepat satu kali saja untuk memenuhi terbentuknya 8 kromosom yang akan dijadikan sebagai individu yang akan diseleksi. Namun variasi rute yang dihasilkan pada langkah simulasi dengan Matlab jumlahnya lebih banyak dibanding hitungan manual meskipun pada akhirnya hanya akan diambil sebanyak 8 rute saja pada setiap perulangannya.

Sebaran bilangan random yang dihasilkan pada langkah seleksi menggunakan roulette well lebih bersifat merata pada simulasi Matlab, sedangkan pada perhitungan secara manual akan cenderung dipengaruhi keinginan peneliti sehingga hasil dari simulasi Matlab lebih bersifat obyektif. Namun dari semua langkah yang ditempuh baik dari hitungan manual maupun simulasi dengan Matlab dihasilkan nilai fitness optimum yang sama.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil uraian di atas maka dapat disimpulkan bahwa:

- 1) Algoritma Genetika dapat diterapkan untuk optimasi rute terpendek pada kasus TSP
- 2) Simulasikan Algoritma Genetika untuk optimasi rute terpendek pada kasus TSP dapat dilakukan dengan menggunakan program Matlab
- 3) Kinerja optimasi pencarian jalur terpendek pada kasus TSP menggunakan algoritma genetika yang diukur adalah jarak karena kondisi medan adalah real tergantung dari traffic yang ada saat itu. Rute terbaik dan fitness tertinggi akan diketahui pada masing-masing perulangannya

5.2 Saran

1. Sebaran bilangan random baik pada saat seleksi individu maupun crossover pada perhitungan manual agar lebih bervariasi karena pada bagian ini sangat berpengaruh kepada nilai obyektivitas hasil.
2. Perlu adanya peningkatan variasi dari rute tempuh untuk menguji kemampuan algoritma genetika.

DAFTAR PUSTAKA

- [1] Susilo dan Y. Suryo, "Pengaruh supply chain management dan total quality management terhadap kinerja distribution center Indomaret Cluster Sidoarjo yang dimediasi oleh keunggulan bersaing," *Repository*, 22 Februari 2019.
- [2] A. Syamil, R. D. M. Danial, S. Saori, E. Waty, M. A. Fahmi, V. Hartati, R. P. Ishak, C. k. Dewi, H. Padilah, M. Fauzi dan R. M. Haryadi, "MANAJEMEN RANTAI PASOK," dalam *BUKU AJAR MANAJEMEN RANTAI PASOK*, Kota Jambi, PT Sonpedia Publishing Indonesia, 2023, pp. 2-4.
- [3] D. F. Rizki dan N. Supriadi, "Comparison of Alfamart and Indomaret Supply Chain Efficiency in Handling Consumer Demand in Indonesia," *American Journal of Economic and Management Business*, no. 3, p. 6, 2024.
- [4] D. Hertina, L. Afiati, M. Munizu, S. Riyadi, J. R. Thamrin dan D. A. Irawan, "MANAJEMEN RANTAI PASOK : Efektifitas MRP dalam mencapai kesuksesan bisnis," dalam *MANAJEMEN RANTAI PASOK : Efektifitas MRP dalam mencapai kesuksesan bisnis*, Kota Jambi, PT. Sonpedia Publishing Indonesia, 2023, pp. 1-45.
- [5] M. Z. Usman dan T. Oktiarmo, "Implementasi Algoritma Greedy Untuk Menyelesaikan Travelling Salesman Problem di Distributor PT. Z," *Journal of Integrated System*, vol. 1 , no. Vol. 1 No. 2 (2018) , p. 216–229, 2019.
- [6] R. Dimastara, H. Manurung dan H. Sembiring, "THE DESIGN OF GAS STATION TERMINAL SEARCH APPLICATION BY APPLYING THE METHOD OF TRAVELING SALESMAN PROBLEM," *Journal of Mathematics and Technology (MATECH)*, vol. 3, no. Vol. 3 No. 1 (2024): Journal MATECH (May 2024) , p. 76–89, 2024.
- [7] I. W. Supriana, "Implementasi Algoritma Genetika dalam Penentuan Rute Terbaik Pendistribusian BBM pada PT Burung Laut," *Jurnal Teknologi Informasi dan Komputer*, vol. 1, no. Vol.3 No.1 (2017), pp. 285-294, 2017.
- [8] T. A. o. G. A. i. S. R. Wulan dan N. P. Apriani, "The Application of Genetic Algorithm in Solving," *Iconistech*, 2019.
- [9] M. Rayungsari, N. Imamah, A. Imaniyah dan V. B. Kusuma, "ESTIMASI PARAMETER MODEL PREDATOR-PREY," *Jurnal Gammath*, vol. 4 , no. Vol. 4 No. 2 (2019), pp. 61-67 , 2019.
- [10] j. jasmani dan A. Mahmudi, "Optimization Of The Shortest Path Using Genetic Algorithm," *J-Intech*, vol. 1, no. Vol. 11 No. 1 (2023), pp. 128 - 139, 2023.